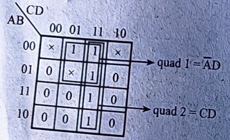
5. **Design a logic circuit to implement the Boolean function $F(A, B, C, D) = \Sigma(1, 3, 7, 11, 15)$ $D(A, B, C, D) = \Sigma(0, 2, 5)$ in term of i. Sum of product ii. Implement with NAND-NAND gate only.** [Fall 2013]

Solution:

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

$D(A, B, C, D) = \Sigma(0, 2, 5)$

Now, using k-map,



$\therefore$ SOP expression is, $F(A, B, C, D) = \bar{A}D + CD$

Now, implementation using NAND gates only,

$F = \bar{A}D + CD = (\bar{A}D + CD)''$

$= \{(\overline{\bar{A}D}) . (\overline{CD})\}'$

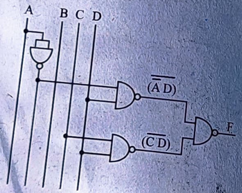


Fig.: Implementation of given function using NAND gates only

CHAPTER – 4

SIMPLIFICATION OF BOOLEAN FUNCTIONS

4.1 Venn Diagram and Test Vector

Venn Diagram

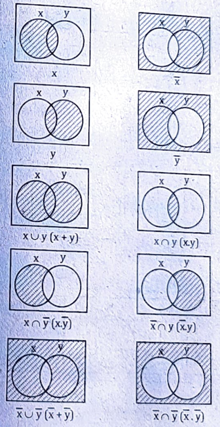
It is used to visualize the relationship among the variables of Boolean expression.

Note: NOT: Complement

AND: Intersection '∩'

OR: Union '∪'

Examples:

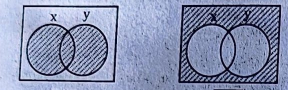


De-Morgan's Theorem

1. **Theorem 1:**

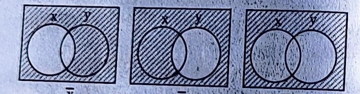
$\overline{(x+y)} = \bar{x} \cdot \bar{y}$

L.H.S. = $\overline{(x+y)}$



$x + y$ $\qquad$ $\overline{(x+y)}$

R.H.S. = $\bar{x} \cdot \bar{y}$



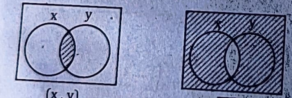
$\bar{x}$ $\qquad$ $\bar{y}$ $\qquad$ $\bar{x} \cdot \bar{y}$

Thus, $\overline{(x+y)} = \bar{x} \cdot \bar{y}$ Proved.

2. **Theorem 2**

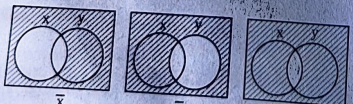
$\overline{(x \cdot y)} = \bar{x} + \bar{y}$

L.H.S. = $\overline{(x \cdot y)}$



$(x \cdot y)$ $\qquad$ $\overline{(x \cdot y)}$

R.H.S. = $\bar{x} + \bar{y}$



$\bar{x}$ $\qquad$ $\bar{y}$ $\qquad$ $\bar{x} + \bar{y}$

Thus, $\overline{(x \cdot y)} = \bar{x} + \bar{y}$ Proved.

Test Vector

Test vector is the set of inputs provided to a system in order to test that system.

4.2 Karnaugh Map (K-map)

K-map is a graphical chart which contains boxes and is a pictorial form of truth table. It is used to simplify the Boolean equations. The number of squares contained in the K–map depends upon the number of logical variables i.e., for n variables, there are 2^n squares arranged in specific order. It comprises a box for every line in truth table.

For

2 variables, there are $2^2 = 4$ cells in K-map

3 variables, there are $2^3 = 8$ cells in K-map

4 variables, there are $2^4 = 16$ cells in K-map

n variables, there are $2^n = 2^n$ cells in K-map.

Truth Table to K-Map

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

OR

A \ B	0	1
0	m_0	m_1
1	m_2	m_3

For two variables

AB \ C	0	1
00	m_0	m_1
01	m_2	m_3
11	m_6	m_7
10	m_4	m_5

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

For three variables

AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

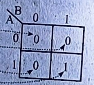
For four variables

Five-Variable K-Map

AB \ CDE	000	001	011	010	110	111	101	100
00	M_0	M_1	M_3	M_2	M_6	M_7	M_5	M_4
01	M_8	M_9	M_{11}	M_{10}	M_{14}	M_{15}	M_{13}	M_{12}
11	M_{24}	M_{25}	M_{27}	M_{26}	M_{30}	M_{31}	M_{29}	M_{28}
10	M_{16}	M_{17}	M_{19}	M_{18}	M_{22}	M_{23}	M_{21}	M_{20}

Relation between a Truth Table and K-Map

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



Truth Table of AND gate

Simplification Procedure

1. Plotting
2. Grouping
3. Simplification

1. **Plotting**
   - We have to plot 1, if the given expression is SOP
   - We have to plot 0, if the given expression is POS.
2. **Grouping techniques**

   We should group adjacent cells only; number of 1s or 0s should be in 2^n numbers.

   i.e., $2^0 = 1$ (isolated term)

   $2^1 = 2$ (pair)

   $2^2 = 4$ (quad)

   $2^3 = 8$ (octet)

   $2^4 = 16$ (hex)

A pair eliminates one variable and its complements, a quad eliminates two variables and their complements, an octet eliminates three variables and their complements, and a hex eliminates four variables and their complements. So, for greatest simplification results, we should encircle the hex first, octets second, the quads third, and the pairs last.

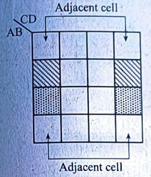
A \ BC	00	01	11	10
0	a	b	c	d
1	e	f	g	h

a and b are adjacent cells

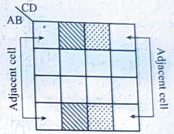
a and e are adjacent cells

a and f are not adjacent cells

a and d are adjacent cells



Leftmost and rightmost cells are adjacent cells.



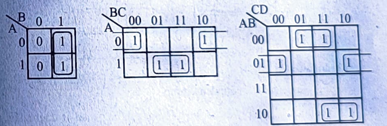
Top and corresponding bottom cells are adjacent cells.

Adjacent cell

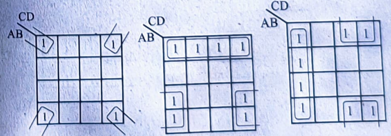
Adjacent cell

Fig.: Illustration of adjacent cells

Pairs:



Quads:



Five-Variable K-Map

AB \ CDE	000	001	011	010	110	111	101	100
00	M_0	M_1	M_3	M_2	M_6	M_7	M_5	M_4
01	M_8	M_9	M_{11}	M_{10}	M_{14}	M_{15}	M_{13}	M_{12}
11	M_{24}	M_{25}	M_{27}	M_{26}	M_{30}	M_{31}	M_{29}	M_{28}
10	M_{16}	M_{17}	M_{19}	M_{18}	M_{22}	M_{23}	M_{21}	M_{20}

Relation between a Truth Table and K-Map

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

Truth Table of AND gate

Simplification Procedure

1. Plotting
2. Grouping
3. Simplification

1. Plotting

- We have to plot 1, if the given expression is SOP
- We have to plot 0, if the given expression is POS.

2. Grouping techniques

We should group adjacent cells only; number of 1s or 0s should be in 2^n numbers.

i.e., $2^0 = 1$ (isolated term)

$2^1 = 2$ (pair)

$2^2 = 4$ (quad)

$2^3 = 8$ (octet)

$2^4 = 16$ (hex)

A pair eliminates one variable and its complements, a quad eliminates two variables and their complements, an octet eliminates three variables and their complements, and a hex eliminates four variables and their complements. So, for greatest simplification results, we should encircle the hex first, octets second, the quads third, and the pairs last.

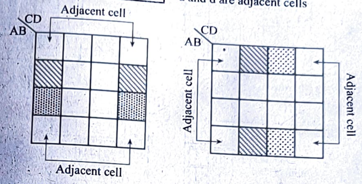
A \ BC	00	01	11	10
0	a	b	c	d
1	e	f	g	h

a and b are adjacent cells

a and e are adjacent cells

a and f are not adjacent cells

a and d are adjacent cells



Leftmost and rightmost cells are adjacent cells.

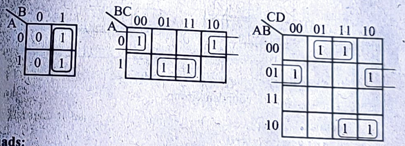
Top and corresponding bottom cells are adjacent cells.

Adjacent cell

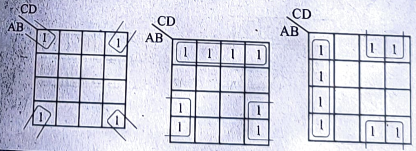
Adjacent cell

Fig.: Illustration of adjacent cells

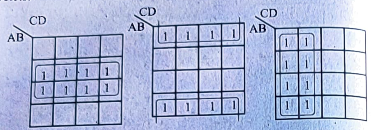
Pairs:



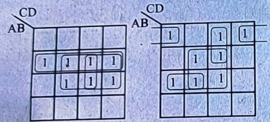
Quads:



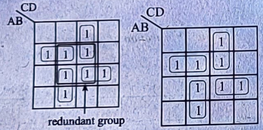
Octets:



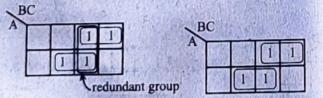
Overlapping groups:



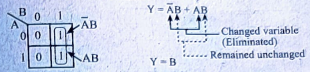
Eliminating redundant groups:

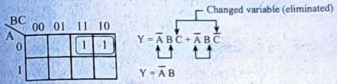


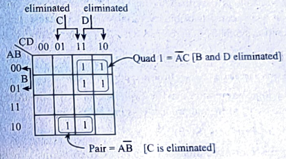
- If all the 1s in a group are already involved in some other group, then that group is called redundant group.
- A redundant group has to be eliminated, because it increases the number of gates required.



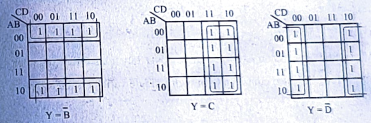
Simplification (SOP)

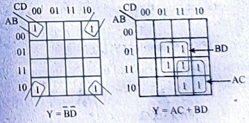


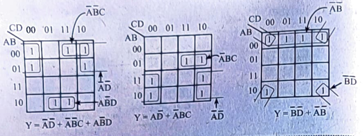




Some More Examples:







4.3 Minimum Realization

Boolean algebra deals with binary variables and logic operations. A Boolean function is described by an algebraic expression called Boolean expression which consists of binary variables, the constants 0 and 1, and the logic operation symbols. Let's consider the following illustrative examples.

$$\underbrace{F(A, B, C, D)}_{\text{Boolean function}} = \underbrace{A + B\bar{C} + ADC}_{\text{Boolean expression}}$$

Any logic expression may be expressed in two standard forms: Sum-of-products (SOP) form, and product-of-sums (POS) form.

1. **Sum-of-Products (SOP)**

When two or more product terms (or ANDed terms) are summed by Boolean algebra, the resulting expression is sum-of-products.

E.g., $F = AB + \bar{A}\,\bar{B}\,C$

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

SOP expression is two or more AND functions ORed together. SOP form can also contain a term with a single variable such as $A + \bar{A}\,\bar{B} + \bar{B}\,C$.

In SOP representation, 0 represents $\bar{A}$ and 1 represents A. Only 1s output is taken.

Standard SOP (or Canonical SOP)

Standard SOP is one in which all the variables in the domain appear in each product term in the expression.

$AB + B\,C + A\bar{D}$ is not standard SOP.

$A\bar{B}\,CD + ABCD + AB\bar{C}\,\bar{D}$ is standard SOP.

Each individual product term in standard SOP form is called as "**minterm**" and is represented by 'm'. Standardization makes the evaluation, simplification, implementation, realization of Boolean expression much more systematic and easier.

Minterms for three variables (A, B, C)

A	B	C	Term	Designation (m_i)
0	0	0	$\bar{A}\,\bar{B}\,\bar{C}$	m_0
0	0	1	$\bar{A}\,\bar{B}\,C$	m_1
0	1	0	$\bar{A}\,B\,\bar{C}$	m_2
0	1	1	$\bar{A}\,B\,C$	m_3
1	0	0	$A\,\bar{B}\,\bar{C}$	m_4
1	0	1	$A\,\bar{B}\,C$	m_5
1	1	0	$A\,B\,\bar{C}$	m_6
1	1	1	$A\,B\,C$	m_7

Consider the following example.

Input			Function output	
A	B	C	F_1	F_2
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

$$\begin{aligned} F_1 &= \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,BC + AB\bar{C} \\ &= m_0 + m_3 + m_6 \\ &= \Sigma m\,(0, 3, 6) \end{aligned}$$

$F_2 = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

$= m_3 + m_5 + m_6 + m_7$

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

Converting SOP to standard SOP:

1. Multiply each product term by $(A + \bar{A})$ where A is the missing term.
2. Expand the multiplication and repeat step 1 until standard SOP is obtained.

EXAMPLE:

Convert $A + A\bar{B} + \bar{B}C$ to standard SOP.

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

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

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

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

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

2. Product–Of–Sums (POS)

When two or more sum terms are multiplied, the resulting expression is product of sum. E.g., $(\bar{A} + B)(A + \bar{B} + \bar{C})(\bar{A} + B + C)$

In terms of logic function, it is the AND function of two or more ORed functions. In POS representation, 0 represents A and 1 represents $\bar{A}$ and only 0's output is taken. A POS expression can also contain a single variable terms. E.g., $A(\bar{B} + \bar{C})(B + C)$

Standard POS

Standard POS is one in which all the variables in the domain appear in each sum term in the expression.

$A(\bar{B} + \bar{C})(B + C)$ is not standard POS

$(A + B + \bar{C})(A + \bar{B} + \bar{C})$ is standard POS.

Each individual sum term in standard POS form is called "maxterm" and is represented by 'M'.

Maxterm of three variables

A	B	C	Term	Designation (M_j)
0	0	0	$A + B + C$	M_0
0	0	1	$A + B + \bar{C}$	M_1
0	1	0	$A + \bar{B} + C$	M_2
0	1	1	$A + \bar{B} + \bar{C}$	M_3
1	0	0	$\bar{A} + B + C$	M_4
1	0	1	$\bar{A} + B + \bar{C}$	M_5
1	1	0	$\bar{A} + \bar{B} + C$	M_6
1	1	1	$\bar{A} + \bar{B} + \bar{C}$	M_7

Consider the same example.

Input			Function output	
A	B	C	F_1	F_2
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

$F_1\,(A + B + \bar{C})(A + \bar{B} + C)(\bar{A} + B + C)(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + \bar{C})$

$= M_1 . M_2 . M_4 . M_5 . M_7$

$= \pi\,(1, 4, 5, 7)$

$F_2 = (A + B + C)(A + B + \bar{C})(A + \bar{B} + C)(\bar{A} + B + C)$

$= M_0, M_1, M_2, M_4,$

$= \pi\,(0, 1, 2, 4)$

Converting POS to Standard POS

1. Add $A.\bar{A}$ to each sum term, where A is missing term.
2. Apply rule $A + BC = (A + B)(A + C)$
3. $A.\bar{A} = A$
4. Repeat step 1 to 3 until standard POS is obtained.

EXAMPLE:

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

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

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

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

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

Relationship between Minterms and Maxterms

$m_i = \overline{M}_j$

E.g., $\bar{A}\,\bar{B}\,\bar{C} = \overline{A + B + C}$ [De-Morgan's law]

i.e., Each maxterm is the complement of its minterm & vice-versa.

Conversion of Standard SOP to POS

1. Evaluate each SOP terms and determine its binary representation.
2. Determine all the binary number not included in evaluation in step 1.
3. Write equivalent sum term for each binary number obtained in step 2.

Note: *The similar process is applied for POS to SOP.*

EXAMPLE:

i. **Convert SOP: $\bar{A}\,\bar{B}\,C + \bar{A}\,BC + A\,\bar{B}\,C + ABC$ to POS expression.**

⇒ **Step 1:** $\bar{A}\,\bar{B}\,C + \bar{A}\,BC + A\bar{B}\,C + ABC$

= 001 011 101 111

1 3 5 7

Step 2: Remaining binary numbers

= 000 010 100 110

0 2 4 6

Step 3: POS $= \pi(0, 2, 4, 6)$

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

ii. **Convert POS: $(A + B + \bar{C})(\bar{A} + B + C)(\bar{A} + \bar{B} + C)$ to SOP expression.**

⇒ Step 1: $(A + B + \bar{C})(\bar{A} + B + C)(\bar{A} + \bar{B} + C)$

= 001 100 110

1 4 6

Step 2: Remaining binary numbers

= 000 010 011 101 111

0 2 3 5 7

Step-3: SOP $= \Sigma m(0, 2, 3, 5, 7)$

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

Simplification (POS):

A \ B	B (0)	$\bar{B}$ (1)
A 0	$A + B$	$A + \bar{B}$
$\bar{A}$ 1	$\bar{A} + B$	$\bar{A} + \bar{B}$

Or,

A \ B	0	1
0	M_0	M_1
1	M_2	M_3

For 2 variables

A \ BC	00	01	11	10
0	$A + B + C$	$A + B + \bar{C}$	$A + \bar{B} + \bar{C}$	$A + \bar{B} + C$
1	$\bar{A} + B + C$	$\bar{A} + B + \bar{C}$	$\bar{A} + \bar{B} + \bar{C}$	$\bar{A} + \bar{B} + C$

Or,

A \ BC	00	01	11	10
0	M_0	M_1	M_3	M_2
1	M_4	M_5	M_7	M_6

For 3 variables

AB \ CD	00	01	11	10
00	$A+B+C+D$	$A+B+C+\bar{D}$	$A+B+\bar{C}+\bar{D}$	$A+B+\bar{C}+D$
01	$A+\bar{B}+C+D$	$A+\bar{B}+C+\bar{D}$	$A+\bar{B}+\bar{C}+\bar{D}$	$A+\bar{B}+\bar{C}+D$
11	$\bar{A}+\bar{B}+C+D$	$\bar{A}+\bar{B}+C+\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+D$
10	$\bar{A}+B+C+D$	$\bar{A}+B+C+\bar{D}$	$\bar{A}+B+\bar{C}+\bar{D}$	$\bar{A}+B+\bar{C}+D$

Or,

AB \ CD	00	01	11	10
00	M_0	M_1	M_3	M_2
01	M_4	M_5	M_7	M_6
11	M_{12}	M_{13}	M_{15}	M_{14}
10	M_8	M_9	M_{11}	M_{10}

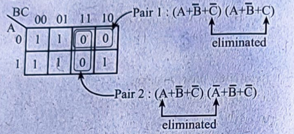
For 4 variables

Simplification procedure (POS):

1. The given POS expression consists of maxterms.

2. Corresponding to every maxterm, we enter a 0s in K-map and we enter 1s in the remaining cells of k-map.
3. We group 0s for carrying out simplification using grouping techniques of adjacent cells.

Example:



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

pair 2 = $(\overline{B}+\overline{C})$

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

4.4 Don't Care condition

"Don't care condition" is the condition of input which doesn't affect output. We plot 1s corresponding to the combination of input variables for SOP form which produces a high output and plot 0s for remaining cells of K-map. We plot 0s corresponding to the combination of input variables for POS form which produces a low output and plot 1s for remaining cells of K-map.

However, it is not always true that the cell not containing 1s (in SOP form) will contain 0s because some conditions of input variable do not occur. Also, for some function, the output corresponding to certain combinations of input variables do not occur. In such condition, we have a freedom to assume a 0 or 1 as output for each of these combinations. These conditions are known as the "don't care conditions". It is represented as 'x' (cross) mark in corresponding cells.

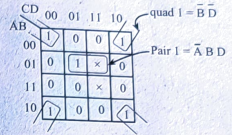
Note: *The don't care condition (X) may be assumed to be 0 or 1 as per the requirement for simplification.*

4.5 Logic Gates Implementation

Example:

Simplify F(A, B, C, D)=Σm(0, 2, 5, 8, 10)+d(7, 15). Write its standard SOP and implement the simplified circuit using NOR gates only.

⇒



$Y = \overline{B}\,\overline{D} + \overline{A}BD$

For standard SOP,

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

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

$= A\overline{B}C\overline{D} + \overline{A}\,\overline{B}C\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}BCD + \overline{A}B\overline{C}D$

For implementing $Y = \overline{B}\,\overline{D} + \overline{A}BD$ using NOR gates only, we write

$Y = \overline{B}\,\overline{D} + \overline{A}BD$

$= \overline{\overline{\overline{B}\,\overline{D}}} + \overline{\overline{\overline{A}BD}} = \overline{B+D} + \overline{A+\overline{B}+\overline{D}}$

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

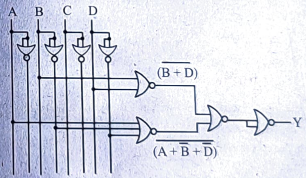


Fig.: Implementing using NOR gates only

SOLUTION TO IMPORTANT AND EXAM QUESTIONS

1. **Convert the following to the other canonical from.**
   i. **$F(x, y, z) = \Sigma(1, 5, 7)$**
   ii. **$F(A, B, C, D) = \Sigma(1, 2, 7, 11, 12, 14)$**
   iii. **$F(x, y, z) = \pi(0, 4, 6, 7)$**
   iv. **$F(A, B, C, D) = \pi(0, 1, 2, 3, 4, 6, 12)$** [2021 Fall]

Solution:

i. Given are minterms, we should convert them into maxterms,

We know, $m_i = \overline{M_j}$

Here, minterms = (1, 5, 7)

Thus, remaining terms are maxterms = (0, 2, 3, 4, 6)

$\therefore$ $F(x, y, z) = \pi(0, 2, 3, 4, 6)$

ii. Given minterms = (1, 2, 7, 11, 12, 14)

So, maxterms = (0, 3, 4, 5, 6, 8, 9, 10, 13, 15)

$\therefore$ $F(A, B, C, D) = \pi(0, 3, 4, 5, 6, 8, 9, 10, 13, 15)$

iii. $F(x, y, z) = \pi(0, 4, 6, 7)$

Given are maxterms. So, we should convert them into minterms.

Maxterms = (0, 4, 6, 7)

So, minterms = (1, 2, 3, 5)

$\therefore$ $F(x, y, z) = \Sigma(1, 2, 3, 5)$

iv. $F(A, B, C, D) = \pi(0, 1, 2, 3, 4, 6, 12)$

Given maxterms = (0, 1, 2, 3, 4, 6, 12)

So, minterms = (5, 7, 8, 9, 10, 11)

$\therefore$ $F(A, B, C, D) = \Sigma(5, 7, 8, 9, 10, 11)$

2. **Design a logic circuit to implement the Boolean function.**
   $F(A, B, C, D) = \Sigma(1, 3, 4, 5, 7, 9, 13, 14, 15)$
   $D(A, B, C, D) = \Sigma(0, 2, 8)$
   i. **Sum of product**
   ii. **Implement with NAND-NAND gates only.** [2021 Fall]

Solution:

Given function,

$F(A, B, C, D) = \Sigma(1, 3, 4, 5, 7, 9, 13, 14, 15)$

$D(A, B, C, D) = \Sigma(0, 2, 8)$

Now, using k-map, we get,

quad 2 = $\bar{A}\,\bar{C}$

AB \ CD	00	01	11	10
00	×	1	1	×
01	1	1	1	0
11	0	1	1	1
10	×	1	0	0

quad 1 = $\bar{A}\,\bar{B}$

quad 3 = BP

pair 1 = A B C

quad 4 = $\bar{B}\,\bar{C}$

Now, from k-map, we get,

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

which is required sum of product (SOP) from.

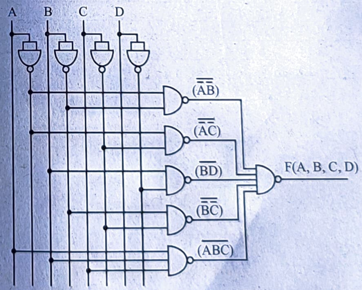
Implementing using NAND-NAND gates only,

$F = \bar{A}\,\bar{B} + \bar{A}\,\bar{C} + BD + \bar{B}\,\bar{C} + ABC$

$= \{\bar{A}\,\bar{B} + \bar{A}\,\bar{C} + BD + \bar{B}\,\bar{C} + ABC\}''$

$= \{(\overline{\bar{A}\bar{B}}) . (\overline{\bar{A}\bar{C}}) . (\overline{BD}). (\overline{\bar{B}\bar{C}}) . (\overline{ABC})\}'$

The circuit diagram is as follows:



3. For the given logic expression:

F(A, B, C, D) = A'B + BD + A'D' + B'D'

i. **Make a truth table**

ii. **Simplify it using k-map**

iii. **Realize the simplified expression using NOR gate (2 inputs) only.** [Fall 2020]

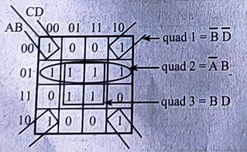
Solution:

Truth table is as follows:

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
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	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Here, $F = \Sigma m(0, 2, 4, 5, 6, 7, 8, 10, 13, 15)$

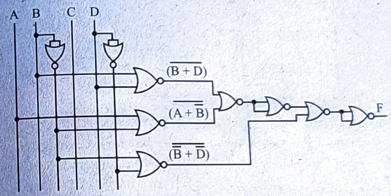
Now, using K-map,



$\therefore$ $F = \bar{B}\bar{D} + \bar{A}B + BD$ which is Boolean expression for given function.

Implementing the given function using 2-input NOR gate,

$$
\begin{aligned}
F &= \bar{B}\bar{D} + \bar{A}B + BD\\
&= \overline{\overline{(\bar{B}\,\bar{D})}} + \overline{\overline{(\bar{A}\,B)}} + \overline{\overline{(B\,D)}}\\
&= \overline{(B + D)} + \overline{(A + \bar{B})} + \overline{(\bar{B} + \bar{D})}\\
&= \overline{\overline{\{\overline{(B + D)} + \overline{(A + \bar{B})} + \overline{(\bar{B} + \bar{D})}\}}}
\end{aligned}
$$



4. **Use K-map scheme to obtain the minimized SOP expression for the given function.**

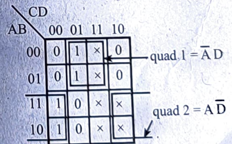
$F(A, B, C, D) = \Sigma_m(1, 5, 8, 12) + d(3, 7, 10, 11, 14, 15)$ and implement the result using NOR gates only. [Fall 2020]

Solution:

Given function,

$F(A, B, C, D) = \Sigma_m(1, 5, 8, 12) + d(3, 7, 10, 11, 14, 15)$

Now, using k-map,



$\therefore$ The required SOP expression is,

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

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

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

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

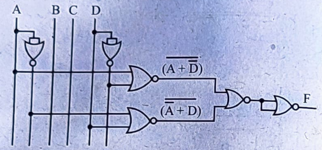


Fig.: Implementation of given function using NOR gates

5. **Simplify the Boolean function F and don't care condition d in i. SOP ii. POS and draw NAND-NAND equivalent logic. Given,**

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

$d=\bar{A}B\bar{C}D+ACD+A\bar{B}\,\bar{D}$ [Spring 2019]

Solution:

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

$$d=\bar{A}B\bar{C}D+ACD+A\bar{B}\,\bar{D}$$

Firstly converting into standard form,

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

$$=A\bar{B}C\bar{D}+\bar{A}\bar{B}C\bar{D}+\bar{A}BCD+\bar{A}\bar{B}CD+\bar{A}BCD+\bar{A}BC\bar{D}+\bar{A}\,\bar{B}\,\bar{C}\,\bar{D}$$

$$=A\bar{B}C\bar{D}+\bar{A}\bar{B}C\bar{D}+\bar{A}BCD+\bar{A}\bar{B}CD+\bar{A}BC\bar{D}+\bar{A}\,\bar{B}\,\bar{C}\,\bar{D}$$

$$=m_{10}+m_2+m_7+m_3+m_6+m_0$$

$$=\Sigma_m(0,2,3,6,7,10)$$

d. $$=A\bar{B}C\bar{D}+ACD+A\bar{B}\bar{D}$$

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

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

$$=m_5+m_{15}+m_{11}+m_{10}+m_8$$

$$=\Sigma_m(5,8,10,11,15)$$

i. **SOP form**

AB \ CD	00	01	11	10
00	1	0	1	1
01	0	×	1	1
11	0	0	×	0
10	×	0	×	×

quad 1 = $\bar{B}\,\bar{D}$

quad 2 = $\bar{A}\,C$

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

ii. **POS form**

AB \ CD	00	01	11	10
00	1	0	1	1
01	0	×	1	1
11	0	0	×	0
10	×	0	×	×

octet = $\bar{A}$

quad 1 = $(\bar{C}+D)$

quad 2 = $(\bar{B}+D)$

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

iii. **NAND-NAND equivalent logic**

For SOP expression,

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

$$=\overline{\overline{\bar{B}\,\bar{D}+\bar{A}\,C}}$$

$$=\overline{\{\overline{(\bar{B}\,\bar{D})}.\overline{(\bar{A}\,C)}\}}$$

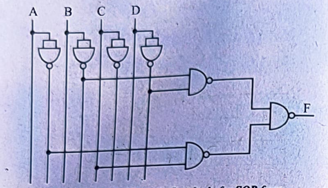


Fig.: NAND-NAND equivalent logic for SOP form.

For POS expression,

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

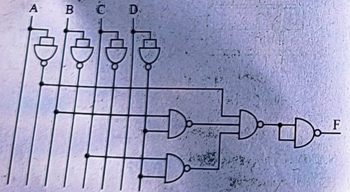


Fig.: NAND-NAND equivalent logic for POS form

6. **A Boolean function is given by, F(A, B, C, D) = Σ(0, 1, 2, 3, 10, 13, 14) and don't care condition d(A, B, C, D) = Σ(4, 7, 12). Simplify it using K-map and implement using NAND gate only.** [Spring 2019]

Solution:

$F(A, B, C, D) = \Sigma(0, 1, 2, 3, 10, 13, 14)$

$d(A, B, C, D) = \Sigma(4, 7, 12)$

Now, using K-map,

AB \ CD	00	01	11	10
00	1	1	1	1
01	×	0	×	0
11	×	1	0	1
10	0	0	0	1

quad = $\overline{A}\,\overline{B}$

pair = $AC\overline{D}$

pair = $AB\overline{C}$

$\therefore \quad F = \overline{A}\overline{B} + AC\overline{D} + AB\overline{C}$

Now, implementation using NAND gate only,

$$F = \{\overline{A}\overline{B} + AC\overline{D} + AB\overline{C}\}''$$
$$= \overline{\{\overline{(\overline{A}\,\overline{B})}.\overline{(AC\overline{D})}.\overline{(AB\overline{C})}\}}$$

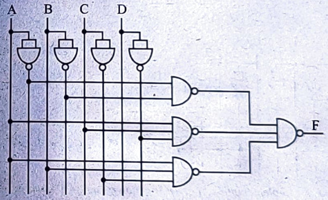


Fig.: Implementation of given function using NAND gates only.

7. **Simplify the following Boolean expression using K-map and implement using NOR gates only.**

F(A, B, C, D) = Σ(0, 2, 4, 6, 12, 15)

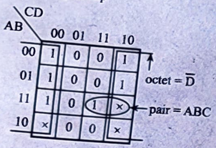
D(A, B, C, D) = Σ(8, 10, 14) [Fall 2019]

Solution:

$F(A, B, C, D) = \Sigma(0, 2, 4, 6, 12, 15)$

$D(A, B, C, D) = \Sigma(8, 10, 14)$

Now, using k-map



$\therefore \quad F(A, B, C, D) = \overline{D} + ABC$

Implementing using NOR gates only,

$$
\begin{aligned}
F(A, B, C, D) &= \overline{D} + ABC \\
&= \overline{D} + \overline{\overline{(ABC)}} \\
&= \overline{D} + \overline{(\overline{A} + \overline{B} + \overline{C})} \\
&= \{\overline{D} + \overline{(\overline{A} + \overline{B} + \overline{C})}\}''
\end{aligned}
$$

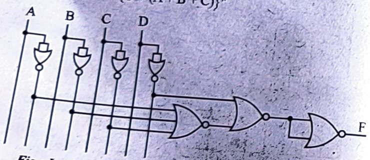


Fig.: Implementation of given function using NOR gates only

8. Simplify the expression mentioned below using K-map.

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

$d(A, B, C, D) = \Sigma(0, 2, 8)$

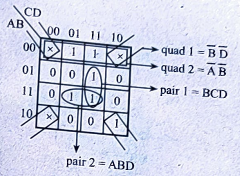
where d denotes don't care. Also implement the simplified function using NOR gates only. [Spring 2018]

Solution:

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

$d(A, B, C, D) = \Sigma(0, 2, 8)$

Using K-map,



$\therefore \quad F(A, B, C, D) = \overline{B}\overline{D} + \overline{A}\overline{B} + BCD + ABD$

Now, implementing using NOR gates only,

$$
\begin{aligned}
F &= \overline{B}\overline{D} + \overline{A}\overline{B} + BCD + ABD \\
&= \overline{\overline{(\overline{B}\,\overline{D})}} + \overline{\overline{(\overline{A}\,\overline{B})}} + \overline{\overline{(BCD)}} + \overline{\overline{(ABD)}} \\
&= \overline{(B + D)} + \overline{(A + B)} + \overline{(\overline{B} + \overline{C} + \overline{D})} + \overline{(\overline{A} + \overline{B} + \overline{D})} \\
&= \{\overline{(B + D)} + \overline{(A + B)} + \overline{(\overline{B} + \overline{C} + \overline{D})} + \overline{(\overline{A} + \overline{B} + \overline{D})}\}''
\end{aligned}
$$

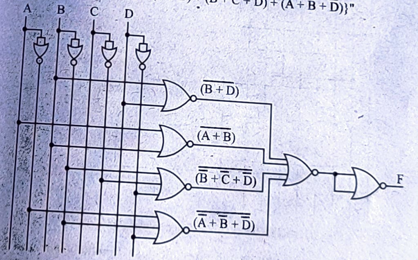


Fig.: Implementation of given function using NOR gates only

9. Use K-map to simplify the given Boolean function with don't care condition and realize it using basic gates only: $F = \Sigma(1, 4, 8, 12, 13, 15)$ and $d = \Sigma(3, 7, 11, 14)$. [Fall 2018]

Solution:

$F = \Sigma(1, 4, 8, 12, 13, 15)$

$d = \Sigma(3, 7, 11, 14)$

Using K-map,

AB \ CD	00	01	11	10
00	0	1	×	0
01	1	0	×	0
11	1	1	1	×
10	1	0	×	0

pair 1 = $\bar{A}\,\bar{B}\,D$

pair 2 = $B\,\bar{C}\,\bar{D}$

quad = AB

pair 3 = $A\,\bar{C}\,\bar{D}$

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

Implementing using basic gates,

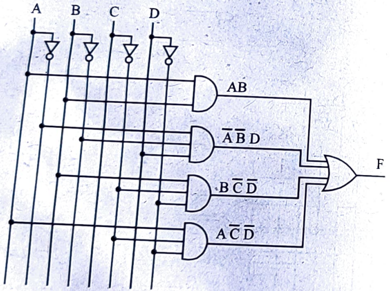


Fig.: Implementation of given function using basic gates

10. Use K-map to simplify the given Boolean function and once by considering the don't care condition and once by ignoring don't care condition and realize it using the basic gates.

F(A, B, C, D) = Σ(1, 4, 8, 12, 13, 15) and

don't care, d(A, B, C, D) = Σ(3, 14) [Spring 2017]

Solution:

Case I: Considering don't care condition,

$F(A, B, C, D) = \Sigma(1, 4, 8, 12, 13, 15)$

$d(A, B, C, D) = \Sigma(3, 14)$

Using K-map,

AB \ CD	00	01	11	10
00	0	1	×	0
01	1	0	0	0
11	1	1	1	×
10	1	0	0	0

pair 1 = $\bar{A}\,\bar{B}\,D$

pair 2 = $B\,\bar{C}\,\bar{D}$

quad = AB

pair 3 = $A\,\bar{C}\,\bar{D}$

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

Implementing using basic gates,

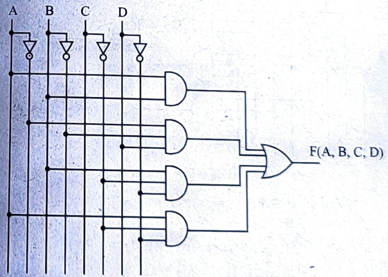


Fig.: Implementation using basic gates

Case II: Ignoring don't care condition,

$F(A, B, C, D) = \Sigma(1, 4, 8, 12, 13, 15)$

Using K-map,

AB \ CD	00	01	11	10
00	0	1	×	0
01	1	0	0	0
11	1	1	1	×
10	1	0	0	0

isolated = $\overline{A}\,\overline{B}\,\overline{C}\,D$

pair 2 = $B\overline{C}\,\overline{D}$

pair 1 = ABD

pair 3 = $A\overline{C}\,\overline{D}$

$\therefore \quad F = \overline{A}\overline{B}\overline{C}D + ABD + B\overline{C}\overline{D} + A\overline{C}\overline{D}$

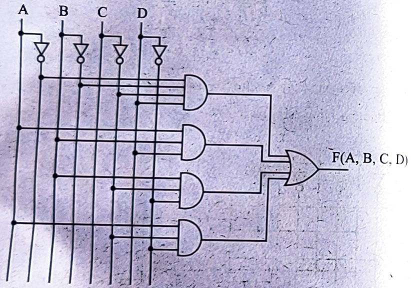


Fig.: Circuit diagram using basic gates

11. Find canonical SOP for this expression
F = ac + ab + bc [Fall 2016]

Solution:

$F = ac + ab + bc$

$= ac(b + \overline{b}) + ab(c + \overline{c}) + bc(a + \overline{a})$

$= abc + a\overline{b}c + abc + ab\overline{c} + abc + \overline{a}bc$

$= abc + a\overline{b}c + ab\overline{c} + \overline{a}bc$

$= m_7 + m_5 + m_6 + m_3$

$\therefore \quad F = \Sigma_m(3, 5, 6, 7)$

which is the required canonical SOP form.

12. Simplify given function using K-map and implement it in a circuit: F(W, X, Y, Z) = Σ(1, 4, 5, 6, 12, 14, 15) and don't care condition D(W, X, Y, Z) = Σ(11, 13). [Spring 2016]

Solution:

$F(W, X, Y, Z) = \Sigma(1, 4, 5, 6, 12, 14, 15)$

$D(W, X, Y, Z) = \Sigma(11, 13)$

Using K-map,

WX \ YZ	00	01	11	10
00	0	1	0	0
01	1	1	0	1
11	1	×	1	1
10	0	0	×	0

pair 1 = $W\overline{Y}Z$

quad 1 = $X\overline{Z}$

quad 2 = WX

$\therefore \quad F(W, X, Y, Z) = W\overline{Y}Z + X\overline{Z} + WX$

13. A Boolean function is given: F(w, x, y, z) = Σ(1, 4, 5, 6, 12, 14, 15) and don't care condition d(w, x, y, z) = Σ(1011). Simplify it using K-map with logic gate implementation. [Fall 2016]

Solution:

$F(w, x, y, z) = \Sigma(1, 4, 5, 6, 12, 14, 15)$

$d(w, x, y, z) = \Sigma(1011) = \Sigma(11)$

Using K-map,

wx \ yz	00	01	11	10
00	0	1	0	0
01	1	1	0	1
11	1	0	1	1
10	0	0	×	0

pair 1 = $\overline{w}\,\overline{y}\,z$

quad = $x\overline{z}$

quad 2 = wyz

$$\therefore \quad F(w, x, y, z) = x\bar{z} + \bar{w}\bar{y}z = wyz$$

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

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

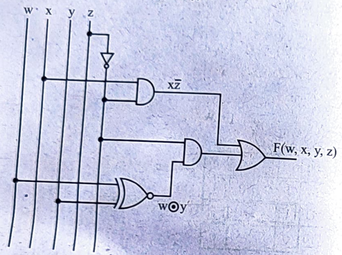


Fig.: Implementation of given function using logic gates

14. **Simplify the Boolean function F and dont care condition d in (1) SOP (2) POS and (3) draw NAND-NAND equivalent logic. Given**

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

$d = \bar{A}B\bar{C}D + ACD + A\bar{B}\bar{D}$

[Fall 2015]

Solution:

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

Converting to standard from,

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

$$= \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}BCD + \bar{A}\bar{B}CD + \bar{A}BCD + \bar{A}BC\bar{D}$$

$$= \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}BCD + \bar{A}\bar{B}CD + \bar{A}BC\bar{D}$$

$$= m_2 + m_0 + m_7 + m_3 + m_6$$

$$\therefore \quad F = \Sigma_m(0, 2, 3, 6, 7)$$

$$d = \bar{A}B\bar{C}D + ACD + A\bar{B}\bar{D}$$

Converting to standard form,

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

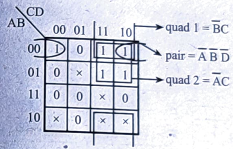
$$= \bar{A}B\bar{C}D + ABCD + A\bar{B}CD + A\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D}$$

$$= m_5 + m_{15} + m_{11} + m_{10} + m_8$$

$$\therefore \quad d = \Sigma_m(5, 8, 10, 11, 15)$$

i. **SOP form**

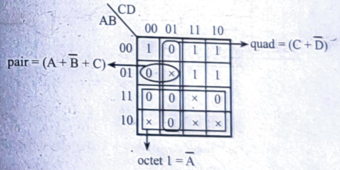
Using K-map,



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

ii. **POS form**

Using K-map,



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

iii. **NAND-NAND equivalent logic**

For SOP,

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

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

$$= \{\overline{(\bar{B} \cdot C)} \cdot \overline{(\bar{A} \cdot C)} \cdot \overline{(\bar{A} \cdot \bar{B} \cdot \bar{C})}\}'$$

For POS,

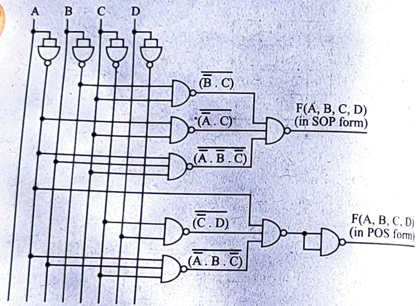
$$F = (\bar{A}) \cdot (C + \bar{D}) \cdot (A + \bar{B} + C)$$

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

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

$$= \{(\bar{A}) \cdot \overline{(\bar{C} \cdot D)} \cdot \overline{(\bar{A} \cdot B \cdot \bar{C})}\}''$$

Now, the circuit diagram is as follows:



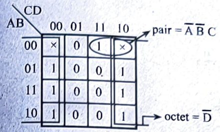
15. A Boolean function is given by F(A, B, C, D) = Σ(3, 4, 6, 8, 10, 12, 14) and don't care condition d(A, B, C, D) = Σ(0, 2, 8). Simplify it using K-map and implement using NAND gates only. [Fall 2015]

Solution:

$F(A, B, C, D) = \Sigma(3, 4, 6, 8, 10, 12, 14)$

$d(A, B, C, D) = \Sigma(0, 2, 8)$

Using K-map,



AB \ CD	00	01	11	10
00	×	0	1	×
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

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

Implementing using NAND gates only,

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

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

$$= \{\overline{(\bar{A} \cdot \bar{B} \cdot C)} \cdot \bar{\bar{D}}\}'$$

$$= \{\overline{(\bar{A} \cdot \bar{B} \cdot C)} \cdot D\}'$$

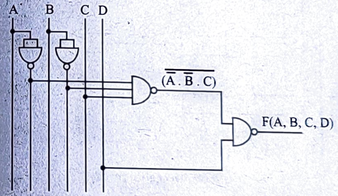


Fig.: Implementation of given function using NAND gates only

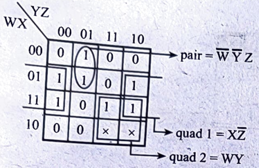
16. Simplify given function using k-map with circuit design.
F(W, X, Y, Z) = Σ(1, 4, 5, 6, 12, 14, 15) and
don't care condition D(W, X, Y, Z) = Σ(10, 11) [Spring 2014]

Solution:

$F(W, X, Y, Z) = \Sigma(1, 4, 5, 6, 12, 14, 15)$

$D(W, X, Y, Z) = \Sigma(10, 11)$

Using K-map,



$\therefore$ SOP expression is $F = \overline{W}\,\overline{Y}Z + X\overline{Z} + WY$

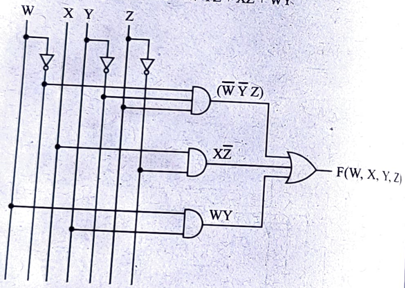


Fig.: Circuit design of given function

17. **Given the following Boolean function F = xy + x'y' + y'z**
   i. **Implement it with OR and NOT gate.**
   ii. **Implement it with only AND and NOT gate.** [Fall 2014]

Solution:

i. $F = xy + x'y' + y'z$

$F = \overline{\overline{xy}} + \overline{\overline{x'y'}} + \overline{\overline{y'z}}$

$= (x' + y)' + (x + y)' + (y + \overline{z})'$

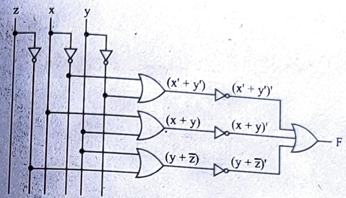


Fig.: Implementation of given function using NOT and OR gates only

ii. $F = (xy + x'y' + y'z)''$

$= \{(x.y)' . (x'y')' . (y'z)'\}'$

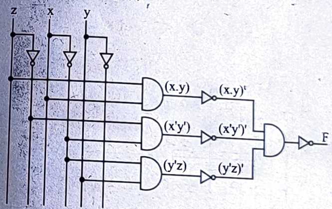


Fig.: Implementation of given function using AND gate and NOT gates only

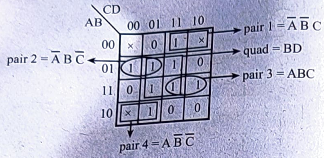
18. **A Boolean function is given by F(A, B, C, D) = Σ(3, 4, 5, 7, 9, 13, 14, 15) and don't care condition d(A, B, C, D) = Σ(0, 2, 8). Simplify it using K-map and implement using NAND gate only.** [Fall 2014]

Solution:

F(A, B, C, D) = Σ(3, 4, 5, 7, 9, 13, 14, 15)

d(A, B, C, D) = Σ(0, 2, 8)

Using K-map,



$\therefore\quad F(A, B, C, D) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C} + BD$

Implementing using NAND gates only,

$$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C} + BD$$
$$= (\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC + BD)''$$
$$= \{(\bar{A}\bar{B}C)'.(\bar{A}B\bar{C})'.(A\bar{B}\bar{C})'.(ABC)'.(BD)'\}'$$

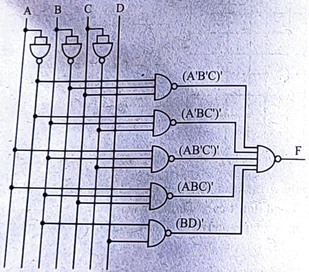


Fig.: Implementation of given function using NAND gates only

19. A logic circuit implements the following Boolean function F = A'C + AC'D'. It is found that the circuit input combination A = C = 1 can never occur. Using K-map with proper don't care conditions, find a simplified expressions and implement it using NAND gates only.
[Spring 2013]

Solution:

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

Dont care: when A = C = 1

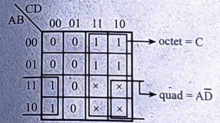
$\therefore\quad d = \Sigma_m(10, 11, 14, 15)$

Converting F to standard form,

$$F = \bar{A}C + A\bar{C}\bar{D}$$
$$= \bar{A}C\,(B + \bar{B})\,(D + \bar{D}) + A\bar{C}\bar{D}\,(B + \bar{B})$$
$$= (\bar{A}BC + \bar{A}\bar{B}C)\,(D + \bar{D}) + AB\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D}$$
$$= \bar{A}BCD + \bar{A}BC\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + AB\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D}$$
$$= m_7 + m_6 + m_3 + m_2 + m_{12} + m_8$$

$\therefore\quad F = \Sigma_m(2, 3, 6, 7, 8, 12)$

Using K-map,



$\therefore\quad F = C + A\bar{D}$

Implementing using NAND gates only,

$$F = C + AD'$$
$$= \{C + AD'\}''$$
$$= \{C\,.\,(AD')'\}'$$

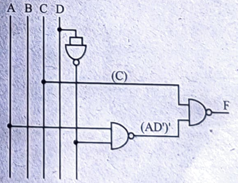


Fig.: Implementation of given function using NAND gates only.

20. **Design a combinational circuit with four input lines that represent a decimal digit in BCD and four output lines that generate the 9's complement of input digit.** [Fall 2013]

Solution:

Let, input variables (BCD) = B_3, B_2, B_1, B_0

Output variables (9's complement) = W X Y Z

Inputs (BCD)				Outputs (9's complement)			
B_3	B_2	B_1	B_n	W	X	Y	Z
0	0	0	0	1	1	1	1
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0

Note: 1010–1111 are dont care conditions.

Using K-map,

For W,

$B_3B_2 \backslash B_1B_0$	00	01	11	10
00	1	1	0	0
01	0	0	0	0
11	×	×	×	×
10	0	0	×	×

$\therefore \quad W = \bar{B}_3\bar{B}_2\bar{B}_1$

For X,

$B_3B_2 \backslash B_1B_0$	00	01	11	10
00	1	0	1	1
01	1	1	0	0
11	×	×	×	×
10	0	0	×	×

$\therefore \quad X = \bar{B}_3\bar{B}_1\bar{B}_0 + B_2\bar{B}_1 + \bar{B}_2B_1$

For Y,

$B_3B_2 \backslash B_1B_0$	00	01	11	10
00	1	0	1	1
01	0	0	1	1
11	×	×	×	×
10	0	0	×	×

$\therefore \quad Y = \bar{B}_3\bar{B}_2\bar{B}_0 + B_1$

For Z,

$B_3B_2 \backslash B_1B_0$	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	×	×	×	×
10	1	0	×	×

$\therefore \quad Z = \bar{B}_0$

Now, the required combinational circuit is as follows: