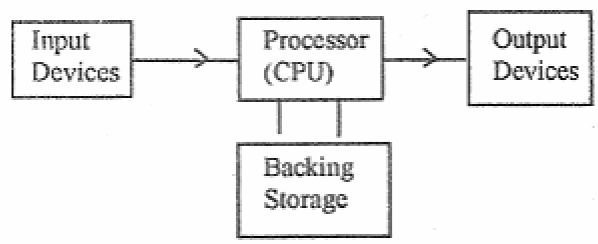
# UNIT II

## 2.1 DIGITAL LOGIC CIRCUIT: DIGITAL COMPUTERS

- Digital computer is a digital system that performs various computational tasks.
- Digital means it is information that represents the values
- A digital computer has two digits 0 and 1.
- A binary digit is called a bit
- Information is represented in digital computer in a group of bits.




**Block Diagram of a Digital Computer**

The input-process-output concepts of the computer are:

- **Input:** The computer accepts input data from the user via an input device like keyboard. The input data can be characters, word, text, sound, images, document, etc.
- **Process**: The computer processes the input data. For this, it performs some actions on the data by using the instructions or program given by the user of the data. The action could be an arithmetic or logic calculation, editing, modifying a document, etc. During processing, the data, instructions and the output are stored temporarily in the computer's main memory.
- **Output**: The output is the result generated after the processing of data. The output may be in the form of text, sound, image, document, etc. The computer may display the output on a monitor, send output to the printer for printing, play the output, etc.
- **Storage:** The input data, instructions and output are stored permanently in the secondary storage devices like disk or tape. The stored data can be retrieved later, whenever needed.

### 2.1.1 <u>LOGIC GATES :</u>

**Gates:**

- Gates are logic circuits that performs the manipulation of binary information with one or more inputs and a single output
- Gates are hardware that produce signals of binary 1 or 0
- It has distinct symbols and its operation can be described by means of logical expression

**Truth table:**

- A truth table is a table that describes the behavior of a logic gate
- It lists the value of output for every possible combination of the inputs

### 2.1.2 DEFINITION

**AND gate:**

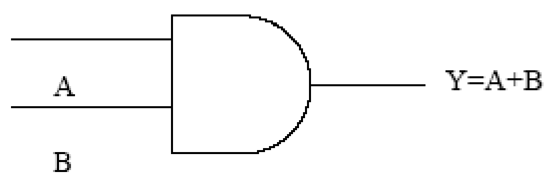
- AND gate has two or more inputs and only one output
- If both inputs are high the output is high
- If any one of the input is low the output is low

**Symbol:**

(a) Truth Table

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

(b) Graphic Symbol



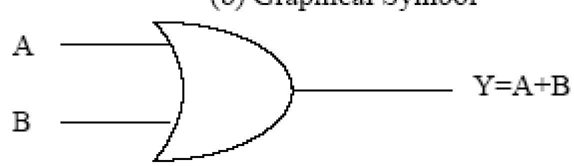

**OR gate:**

- OR gate has two or more inputs and only one output
- If one or more of its input is high the output is high
- If both the inputs are low the output is low

(a) Truth table

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

(b) Graphical Symbol



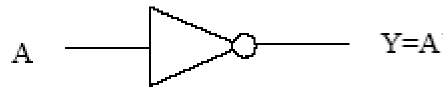

**NOT gate o**

- An inverter or not gate has only a single input and a single output signal
- It inverts or complements the given input

(a) Truth Table

| A | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

(b) Graphic Symbol



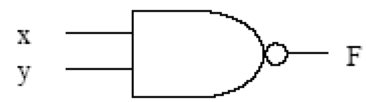

**NAND gate: (NOT- AND)**

- NAND gate has two or more inputs but only one output
- It produces the complement of AND output
- The output is high when any of the inputs are low and the output is low when all of its inputs are high

(a) Truth Table

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

(b) Graphical Symbol

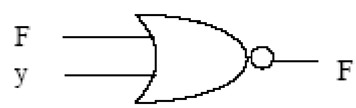



**NOR gate: (NOT- OR)**

- NOR gate has two or more inputs but only one output.
- It produces the complement of OR
- The output is high only when all inputs are low and the output is low if any of the input is high

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

(a) Truth Table




(b) Graphic Symbol

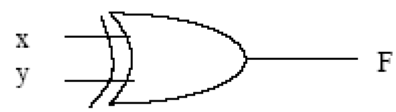
**EXCLUSIVE OR gate ( XOR) :**

- XOR gate has two or more inputs but only one output
- Output is high if one, and only one, of the inputs to the gate is high
- If both inputs are low or both inputs are high the output is low
- An encircled plus sign (⊕) is used to show the EOR operation.

**Symbol :**

| 2 Input EXOR gate | | |
|---|---|---|
| A | B | A⊕B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

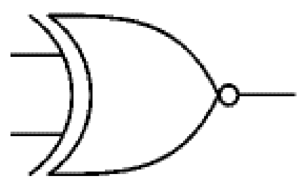
(b) Graphical Symbol




**EXCLUSIVE NOR gate (XNOR)*:***

- XOR  gate has two or more inputs but only one output
- It produces the complement or inverse of  XOR output
- Output is high  if both of the inputs are the same
- The XOR gate with inputs *A* and *B* implements the logical expression $A . \overline{B} + \overline{A} . B$

**Symbol**



| 2 Input EXNOR gate | | |
|---|---|---|
| A | B | $\overline{A \oplus B}$ |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Truth table**

## 2.2 BOOLEAN ALGEBRA :

- Boolean algebra is used to solve the logic problems by expressing the statements and functions as symbols and then manipulating them to arrive at a result
- It is a switching algebra that deals with binary variables and logical operations.

Consider

A+A' = 1
A.A' = 0
**ie (A+B)+(A+B)'=1 AND (A+B).(A+B)'=0**
**(A+B)+(A'.B')=1 AND (A+B).(A'.B')=0**

### 2.2.1 BOOLEAN EXPRESSIONS

**Consider the following Boolean expressions**

| Boolean Function | Operator Symbol | Name | Comments |
|---|---|---|---|
| $F_0$=0 | | Null | Binary constant 0 |
| $F_1$=xy | x.y | AND | x and y |
| $F_2$=xy' | x/y | Inhibition | x but not y |
| $F_3$=x | | Transfer | x |
| $F_4$=x'y | y/x | Inhibition | y but not x |
| $F_5$=y | | Transfer | Y |
| $F_6$=xy'+x'y | x ⊕ y | Exclusive – OR | x or y,but not both |
| $F_7$=x+y | x+y | OR | X or y |
| $F_8$=(x+y)' | x ˘ y | NOR | Not – OR |
| $F_9$=xy+x'y' | (x⊕ y)' | Equivalence | x equals y |
| $F_{10}$=y' | y' | Complement | Not y |
| $F_{11}$=x+y' | x ∨ y' | Implication | If y, then x |
| $F_{12}$=x' | x ∧ y' | Complement | Not x |
| $F_{13}$=x'+y | (x ∧ y)' | Implication | If x, then y |

1. **Simplify the Boolean expression**
   XY′Z′+XY′Z′W+XZ′
   The above expression can be written as
   XY′Z′ (1+W) +XZ′
   =XY′Z′+XZ′ as 1+W=1
   =XZ′ (Y′+1)
   =XZ′ as Y′+1=1
2. **Simplify the Boolean expression**
   X+X′Y+Y′+(X+Y′) X′Y
   The above expression can be written as
   X+X′Y+Y′+XX′Y+Y′X′Y
   =X+X′Y+Y′ as XX′=0, and YY′=0
   =X+Y+Y′ as X+X′Y=X+Y
   =X+1 as Y+Y′=1
   =1 as X + 1=1
3. **Simplify the Boolean expression**
   Z(Y+Z) (X+Y+Z)
   The above expression can be written as
   (ZY+ZZ)(X+Y+Z)
   = (ZY+Z) (X+Y+Z) as ZZ=Z
   =Z(X+Y+Z) as Z+ZY=Z
   =ZX+ZY+ZZ
   =ZX+ZY+Z as ZZ=Z,
   =ZX+Z as Z+ZY=Z
   =Z as Z+ZX=Z

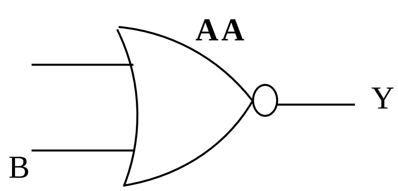
## DE MORGAN’S THEOREMS

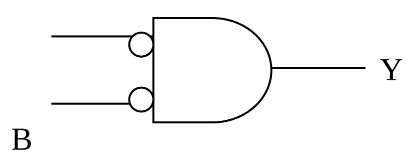
### De Morgan’s First Theorem:

**Statement:** It states that for any two elements A and B in Boolean Algebra, the complement of a sum is equal to the product of complements.

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

**Logic circuit:**




| A | B | Y=$\overline{A + B}$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |

| | | |
|---|---|---|
| **1** | **1** | **1** |

| **A** | **B** | **Y=$\overline{A}$ . $\overline{B}$** |
|---|---|---|
| **0** | **0** | **0** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |

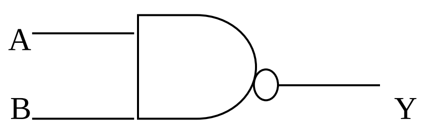
**<u>De Morgan’s Second Theorem:</u>**

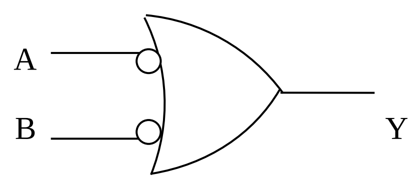
**Statement:**

It states that for any two elements A and B in a Boolean algebra, the complement of a product is equal to the sum of complements

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

**Logic circuit:**




| **A** | **B** | **Y=** $\overline{A . B}$ |
|---|---|---|
| **0** | **0** | **1** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

| **A** | **B** | **Y=** $\overline{A} + \overline{B}$ |
|---|---|---|
| **0** | **0** | **1** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

**2.2.2 BASIC IDENTITIES**

Boolean algebra equations can be manipulated by following a few basic rules.

| Manipulation Rules |
|---|
| A + B = B + A |
| A * B = B * A |
| (A + B) + C = A + (B + C) |
| (A * B) * C = A * (B * C) |
| A * (B + C) = (A * B) + (A * C) |

| |
|---|
| A + (B * C) = (A + B) * (A + C) |
| Equivalence Rules<br>$\overline{\overline{A}} = A$ (double negative)<br>$A + A = A$<br>$A * A = A$<br>$A * \overline{A} = 0$<br>$A + \overline{A} = 1$ |
| Rules with Logical Constants<br>$0 + A = A$<br>$1 + A = 1$<br>$0 * A = 0$<br>$1 * A = A$ |

### 2.2.3 DE MORGAN'S THEOREMS

**Example 1:**

Prove mathematically De Morgan's Theorems: for any two elements A and B in Boolean algebra

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

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

**Theorem 1:** $\overline{A+B} = \overline{A} . \overline{B}$

**Proof:**

We know that $a + \overline{a} = 1$ and $a . \overline{a} = 0$

i.e $(A+B) + \overline{A + B} = 1$ and $(A + B ). (\overline{A + B}) = 0$

**To Prove:**

$(A+B) + \overline{A} . \overline{B} = 1$ and $(A + B ). (\overline{A} . \overline{B}) = 0$

**Solution:**

$$
\begin{aligned}
(A+B) + \overline{A} . \overline{B} &= \{(A + B ) + \overline{A}\} . \{(A + B ) + \overline{B}\} \\
&= \{B+(A + \overline{A})\} . \{ A + (B + \overline{B})\} \\
&= (B+1) . ( A+1) \\
&= 1 . 1 \\
&= 1
\end{aligned}
$$

$$
\begin{aligned}
\text{And } (A + B ) . (\overline{A} . \overline{B}) &= \{A . ( \overline{A} . \overline{B} )\} + \{B . (\overline{A} . \overline{B})\} \\
&= \{ (A. \overline{A}) . \overline{B} \} + \{ (B . \overline{B}) . \overline{A} \} \\
&= ( 0 . B ) + ( 0 . A ) \\
&= 0 + 0 \\
&= 0
\end{aligned}
$$

**Theorem 2:** $\overline{A . B} = \overline{A} + \overline{B}$

**Proof:**

We know that $a + \overline{a} = 1$ and $a.\overline{a} = 0$

i.e., $A . B + \overline{A . B} = 1$ and $A . B . (\overline{AB}) = 0$

**To Prove:**

$A.B + (\overline{A} + \overline{B}) = 1$ and $A . B . (\overline{A} + \overline{B}) = 0$

**Solution:**

$$
\begin{aligned}
A . B + (\overline{A} + \overline{B}) &= (\overline{A} + \overline{B}) + A . B \\
&= \{ (\overline{A} + \overline{B}) + A\} . \{(\overline{A} + \overline{B}) + B \} \\
&= (A + \overline{A} + \overline{B}) . (B + \overline{B} + \overline{A}) \\
&= (1 + \overline{B}) . (1 + \overline{A}) \\
&= 1
\end{aligned}
$$

$$
\begin{aligned}
\text{And } A . B . (\overline{A} + \overline{B}) &= \{ (A . B) . \overline{A}\} + \{ (A . B) . \overline{B} \} \\
&= \{ (A . \overline{A}) . B ) \} + \{ (B . \overline{B}) . A ) \\
&= (0 . B) + (0 .+ A) \\
&= 0
\end{aligned}
$$

**Example*:*** prove the identities using Boolean algebra

(a) $A . B + C . D = (A + C)(A + D)(B + C)(B + D)$

**To Prove :** $AB + CD = (A + C)(A + D)(B + C)(B + D)$

**Solution :**

$$
\begin{aligned}
\text{L.H.S} &= AB + CD = (AB) + C . D \\
&= (AB+C) . (AB + D) \\
&= (C + A . B) . (D + A . B) \\
&= (C + A)(C + B)(D + A)(D + B) \\
&= (A + C)(A + D)(B + C)(B + D) \\
&= \text{R.H.S}
\end{aligned}
$$

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

**To Prove** :$(A + B\overline{C} + C)\overline{C} = AB\overline{C} + A\overline{B}\overline{C} + \overline{A}B\overline{C}$

**Solution :**

$$
\begin{aligned}
\text{L.H.S} &= (A + B\overline{C} + C)\overline{C} \\
&= A\overline{C} + B\overline{C}\overline{C} + C\overline{C} \\
&= A\overline{C} + B\overline{C} + 0 \\
\\
&= A(B+\overline{B}) . \overline{C} + (A+\overline{A}) B\overline{C} \\
&= AB\overline{C} + A\overline{B}\overline{C} + AB\overline{C} + \overline{A}B\overline{C} \\
&= AB\overline{C} + A\overline{B}\overline{C} + AB\overline{C} + \overline{A}B\overline{C} \\
&= AB\overline{C} + A\overline{B}\overline{C} + \overline{A}B\overline{C} \\
&= \text{R.H.S}
\end{aligned}
$$

(c) $A(\overline{A} + C)(AB + \overline{C}) = 0$

**To Prove** :$A(\overline{A} + C)(AB + \overline{C}) = 0$

**Solution :**

$$
\begin{aligned}
L.H.S &= A(\overline{A}+C)\ .\ (\overline{A}B + \overline{C}) \\
&= (A\overline{A} + AC)\ .\ (\overline{A}B + \overline{C}) \\
&= 0 + AC\ .\ (\overline{A}B + \overline{C}) \\
&= AC\overline{A}B + AC\overline{C} \\
&= A\overline{A}BC + A\ .\ C\overline{C} \\
&= 0\ .\ BC + A\ .\ 0 \\
&= 0 + 0 \\
&= 0 \\
&= R.H.S
\end{aligned}
$$

## 2.2.4 MAP SIMPLIFICATION:

### KARNAUGH’S MAP

- The map method was proposed by E.W.Veitch and later modified by M.Karnaugh.
- This provides a simple set procedure for minimizing the switching function.
- This map method /pictorial representation of the truth table is called as Veitch-Karnaugh (V-K) map or Karnaugh map.
- Made up of squares – each square represents one term.
- Each n variable map contains of $2^n$ cells. (If n=3 then map contains 8 cells)

A three- variable Karnaugh Map

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 2 | 6 | 4 |
| 1 | 1 | 3 | 7 | 5 |

A four – variable Karnaugh Map

| 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 |

Take 3-variable Karnaugh map : AB=01 C=0 so ABC = 010 whose decimal value = 2
AB – top values (00,01,11,10) C - down values (0,1)
Take 4-variable Karnaugh map : AB=01 CD=01 so ABCD=0101 whose decimal value= 5
AB – top values CD - down values (00,01,11,10)

### CANONICAL FORM 1

| Decimal Value | A | B | C | Minterm | Maxterm |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | A’ B’ C’ | A+B+C |
| 1 | 0 | 0 | 1 | A’ B’ C | A+B+C’ |
| 2 | 0 | 1 | 0 | A’ B C’ | A+B’+C |
| 3 | 0 | 1 | 1 | A’ B C | A+B’+C’ |
| 4 | 1 | 0 | 0 | A B’ C’ | A’+B+C |
| 5 | 1 | 0 | 1 | A B’ C | A’+B+C’ |
| 6 | 1 | 1 | 0 | A B C’ | A’+B’+C |
| 7 | 1 | 1 | 1 | A B C | A’+B’+C’ |

### 2.2.5 MINTERMS AND MAXTERMS FOR THREE VARIABLES

**Minterms :**

- A product term which has each of the variables as factors in either complemented or uncomplemented form is known as minterm.
- A function with n variables has $2^n$ minterms
- A three-variable function, such as f(x,y,z), has $2^3 = 8$ minterms.
- Any variable should be taken in uncomplemented form if it has the value ‘1’ and should be taken in complemented form if it has the value ‘0’.
- For example 001 should be written as $\overline{A}\ \overline{B}$ C

**Maxterms :**

- A sum term which has each of the variables in either complemented or uncomplemented form is known as maxterm.
- A function with n variables has $2^n$ maxterms
- A three-variable function, such as f(x,y,z), has $2^3 = 8$ maxterms.
- Any variable should be taken in uncomplemented form if it has the value ‘0’ and should be taken in complemented form if it has the value ‘1’.
- For example 001 should be written as A+B+$\overline{C}$ .

**SUM OF PRODUCTS (SOP)** :

- The switching function expressed as the sum of all the minterms for which the function attains the value ‘1’ is called the Canonical Sum of Products (SOP) or disjunctive normal expression.
- Σ denotes sum of product.
  Consider the following truth table. The function attains the value in ‘1’ in 4 cases.

| Decimal Value | A | B | C | *f* |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 |

Therefore, SOP = $\sum(1, 2, 4, 7)$ // 1,2,4,7 have the output 1

= 001+010+100+111

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

**Product of Sum (POS)**:

- The expression expressed as product of all the maxterms for which the function attains the value '0' is known as the canonical product of sums (POS) or conjunctive normal expression.
- π denotes product of sum.

POS = $\Pi(0, 3, 5, 6)$ // 0,3,5,6 have the output 0

= 0, 3, 5, 6

= (000) (011) (101) (110)

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

**FINDING THE CANONICAL SUM OF PRODUCTS (SOP) FORM FOR AN EXPRESSION:**

**Step 1** : When any switching expression is to be expressed in canonical SOP forms, then each term of the expression should be examined and if it is a minterm , then it should be kept as it is.

Eg : $AB\overline{C}$ - all the 3 variables are present – so it is a minterm and hence kept as it is.

**Step 2** : If any particular variable does not occur in any term, then for each variable A,B or C which does not occur, multiply that term by $(A+\overline{A})$,$(B+\overline{B})$ or $(C+\overline{C})$ as the case may be.

Eg : $A\overline{B}$ – here C is missing so multiply this term by $(C+\overline{C})$.

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

**Step 3 :** Eliminate the repeated terms.

**Example:** Find the canonical SOP and POS for the given expression.

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

$$\begin{aligned} f(A,B,C) &= AB + \bar{A}\bar{B} + AC + \bar{A}\bar{C} \\ &= AB(C+\bar{C}) + \bar{A}\bar{B}(C+\bar{C}) + AC(B+\bar{B}) + \bar{A}\bar{C}(B+\bar{B}) \quad \text{(step 2)} \\ &= ABC + AB\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + ACB + AC\bar{B} + \bar{A}\bar{C}B + \bar{A}\bar{C}\bar{B} \end{aligned}$$

Eliminating the repeated terms, we get (step 3)

$$\begin{aligned} f(A,B,C) &= ABC + AB\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + +A\bar{B}C + \bar{A}B\bar{C} \\ &= 111 + 110 + 001 + 000 + 101 + 010 \\ &= \sum(7,6,1,0,5,2) \\ \text{SOP} &= \sum(0,1,2,5,6,7) \end{aligned}$$

$$\begin{aligned} \text{POS} &= \text{Complement of SOP} \\ &= \Pi(3,4) \\ &= \Pi(011,100) \\ &= (A+\bar{B}+\bar{C})(\bar{A}+B+C) \end{aligned}$$

**FINDING THE CANONICAL PRODUCT OF SUM (POS) FORM FOR AN EXPRESSION:**

**Step 1 :** When any switching expression is to be expressed in canonical POS forms, then each term of the expression should be examined and if it is a maxterm , then it should be kept as it is.

Eg : $A+B+\bar{C}$ - all the 3 variables are present – so it is a maxterm and hence kept as it is.

**Step 2 :** If any particular variable does not occur in any sum term, then for each variable A,B or C which does not occur, add that term by $A\bar{A}$, $B\bar{B}$ or $C\bar{C}$ as the case may be.

Eg : $A+\bar{B}$ – here C is missing so add this term by ($C\bar{C}$).

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

**Step 3 :** Convert the sum terms into product of sums.

**Step 4 :** Eliminate the repeated terms.

**Example** : Find the canonical POS and SOP for the given switching functions.

$$\begin{aligned} f(A,B,C) &= (\bar{A}).(B+\bar{C}) \\ &= (\bar{A}+B\bar{B}+C\bar{C}).(B+\bar{C}+A\bar{A}) \quad \text{(step 2)} \end{aligned}$$

$= ((\overline{A}+B)(\overline{A}+\overline{B})+C\overline{C}).(A+B+\overline{C}).(\overline{A}+B+\overline{C})$ (Step 3)

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

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

Eliminating the repeated terms, we get (step 4)

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

$=$ (100) (110) (101) (111) (001)

$= \Pi(4,6,5,7,1)$

$= \Pi(1,4,5,6,7)$

To get SOP , multiply each variable by the absent variable.

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

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

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

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

$= \overline{A}BC + \overline{A}B\overline{C} + \overline{A}\,\overline{B}\,\overline{C}$ (deleted $\overline{A}B\overline{C}$ as it has repeated twice)

$=$ 011+010+000

$= \sum(3,2,0)$

SOP $= \sum(0,2,3)$

**Example :** From the truth table given below , express the function $f$ in sum of minterms and product of maxterms . Obtain the switching function f(A,B,C) in canonical SOP and POS from and prove that they both minimize to same value.

| Decimal value | A | B | C | $f$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 0 |

Solution:

The decimal value for which the function $f$ assumes value ‘1’ are 0,1,4,5

$= \sum(0,1,4,5)$

$f$=000+001+100+101 = **Sum Of Minterms**

$f(A,B,C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + A\overline{B}C$ = **SOP form**

The above form can be minimized to

$= \bar{A}\,\bar{B}(\bar{C}+C) + A\bar{B}(C+\bar{C})$ (Since $(C+\bar{C}) = 1$)

$= \bar{A}\,\bar{B} + A\bar{B}$

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

$= \bar{B}.1 = \mathbf{\bar{B}}$

The decimal value for which the function *f* assumes value '0' are 2,3,6,7

$= \Pi(2,3,6,7)$

=(010)(011)(110)(111) = **Product Of Maxterms**

$f(A,B,C) = (A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)(\bar{A}\;\bar{B}\;\bar{C})$ = **POS form**

The above form can be minimized to

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

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

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

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

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

Hence the minimization of both canonical SOP and POS gave the same value ($\mathbf{\bar{B}}$).

## KARNAUGH'S MAP – CONSTRUCTION AND PROPERTIES

- The Karnaugh map is a **modified from of Venn diagram** of a switching function with 4 or less variables in the canonical SOP form.
- When a venn diagram is redrawn using rectangles and squares and complemented and uncomplemented variables are represented by 0's and 1's, in the columns or rows which it represents, then the diagram is called **Karnaugh Map**.
- Each square of K map is denoted by a binary number or its decimal number.
- If the function has n variables then there must be $2^n$ squares. For example if there are 4 variables then the map has $2^4$=16 squares.
- To **construct a Karnaugh map** of a switching function, first the function is represented in the sum of products form.
- **Example** : $f(A,B,C) = \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$

$= 000+001+010+011+111$

$= \sum(0,1,2,3,7)$

The above function is represented on Karnaugh Map by marking the squares by **'1'**

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $1_0$ | $1_2$ | 6 | 4 |
| 1 | $1_1$ | $1_3$ | $1_7$ | 5 |

→ The decimal numbers are usually written in small numerals at the bottom right corner of the squares representing a decimal number. **Example** : $f(A,B,C,D) = \sum(0,1,3,7,11,15)$

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $1_0$ | 4 | 12 | 8 |
| 01 | $1_1$ | 5 | 13 | 9 |
| 11 | $1_3$ | $1_7$ | $1_{15}$ | $1_{11}$ |
| 10 | 2 | 6 | 14 | 10 |

The simplification of the switching function can be done by combining the '1' cells into pairs, quartets and octets as the case may be.

- Minterms of adjacent squares in the map are identical except for one variable, which appears complemented in one square and uncomplemented in the adjacent square.
- Eg : In the above K map take the first row - first two cells having the decimal value 0,4 i.e 0000 ($\bar{A}\ \bar{B}\ \bar{C}\ \bar{D}$)and 1000($A\bar{B}\ \bar{C}\ \bar{D}$) here only one variable 'A' appears complemented in one square and uncomplemented in another square.
- According to this definition of adjacency
  - ➢ The square of the extreme ends of the same horizontal row are adjacent. Eg : 0(0000), 8(1000)
  - ➢ The square of the top and bottom squares of a column are adjacent. Eg : 0(0000), 2(0010)
  - ➢ The four corner squares of a map are adjacent.
  - ➢ Eg : 0000,1000 ,0010,1010 - i.e [0000,1000] [0000,0010] [1000, 1010] [0010,1010]

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $1_0$ | $1_2$ | 6 | 4 |
| 1 | $1_1$ | $1_3$ | $1_7$ | 5 |

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $1_0$ | 4 | 12 | 8 |
| 01 | $1_1$ | 5 | 13 | 9 |
| 11 | $1_3$ | $1_7$ | $1_{15}$ | $1_{11}$ |
| 10 | 2 | 6 | 14 | 10 |

**Examples** :

1. Simply the Boolean function $f$(A,B,C) = $\sum$ ( 3,4,6,7)

***Solution***

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 2 | $1_6$ | $1_4$ |
| 1 | 1 | $1_3$ | $1_7$ | 5 |

- Pairs are (6,4) and (3,7)
- Before minimization the function is written as $\overline{A}$BC+A$\overline{B}\,\overline{C}$+AB$\overline{C}$+ABC
- After minimization the function is written as BC+A$\overline{C}$.
- 6 is written as AB$\overline{C}$ and the adjacent square 4 is written as A$\overline{B}\,\overline{C}$. The common term is A$\overline{C}$.
- 3 is written as $\overline{A}$BC and the adjacent square 7 is written as ABC. The common term is BC.
- Hence BC+A$\overline{C}$.

To check :

$$f = \sum(3,4,6,7)$$

$$f = \overline{A}BC + A\overline{B}\,\overline{C} + AB\overline{C} + ABC$$

$$= \overline{A}BC + ABC + A\overline{B}\,\overline{C} + AB\overline{C} \text{ (rearranged)}$$

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

2. Simplify the switching function $f$(A,B,C,D) = $\sum$(0,1,2,3,8,9,10,11)

***Solution***

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $1_0$ | 4 | 12 | $1_8$ |
| 01 | $1_1$ | 5 | 13 | $1_9$ |
| 11 | $1_3$ | 7 | 15 | $1_{11}$ |
| 10 | $1_2$ | 6 | 14 | $1_{10}$ |

**<u>a. Simplification Using K Map :</u>**

- '1' is marked in the respective squares.
- See the common variable in all these squares.
- In 0(0000), 1(0001), 2(0010), 3(0011) – the common variable is $\bar{A}\,\bar{B}$
- In 8(1000), 9(1001), 10(1010), 11 (1011) – the common variable is A$\bar{B}$
- So $\bar{A}\,\bar{B}$ + A$\bar{B}$
  = $\bar{B}(\bar{A}$+A)  = $\bar{B}$

The function in minimized form is $\bar{B}$

**<u>b.Simplification using Boolean Algebra :</u>**

$$f = \bar{A}\,\bar{B}\,\bar{C}\,\bar{D} + \bar{A}\,\bar{B}\,\bar{C}\,D + \bar{A}\,\bar{B}\,C\,\bar{D} + \bar{A}\,\bar{B}\,C\,D + A\,\bar{B}\,\bar{C}\,\bar{D} + A\,\bar{B}\,\bar{C}\,D + A\,\bar{B}\,C\,\bar{D} + A\,\bar{B}\,CD$$
$$= \bar{A}\,\bar{B}\,\bar{C}(\bar{D}+D) + \bar{A}\,\bar{B}\,C(\bar{D}+D) + A\,\bar{B}\,\bar{C}(\bar{D}+D) + A\,\bar{B}\,C(\bar{D}+D)$$
$$= \bar{A}\,\bar{B}(\bar{C}+C) + A\,\bar{B}(\bar{C}+C)$$
$$= \bar{B}(\bar{A}+A) \qquad = \bar{B}.$$

**<u>IMPLICANTS :</u>**

- When a switching function of four or less than four variables is represented on a K map, then the set of adjacent minterms or the simplified product term obtained by combining the minterms of set are called implicants of the switching function
- Prime-Implicant : An implicant is called a prime-implicant of the switching function if it is not a subset of any other implicant of the switching function.

- Essential Prime-Implicant : A prime-implicant which includes a ‘1’ cell , which is not included in any other prime-implicant , on the K map, is known as an essential prime-implicant of the switching function.

## MINIMIZATION IN SOP FORM USING KARNAUGH MAP :

1. Minimize the Boolean function $f$ (A,B,C)= $\bar{A}\,\bar{B}\,\bar{C}+\bar{A}B\bar{C}+\bar{A}BC$ using K map.
   **Solution :**

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $1_0$ | $1_2$ | 6 | 4 |
| 1 | 1 | $1_3$ | 7 | 5 |

$f$ (A,B,C)= $\bar{A}\,\bar{B}\,\bar{C}+\bar{A}B\bar{C}+\bar{A}BC$
= 000+010+011
= $\sum(0,2,3)$

Mark ‘1‘is made in 3 cells.

- Take first row (0,2) , the two cells are adjacent. The value is 000 and 010, only variable B appears complemented in 0(0**0**0) and uncomplemented in 2(0**1**0). Hence this can be written as $\bar{A}\,\bar{C}$ (common terms).
- Take second column (2,3), the two cells are adjacent as only one variable (C) appears complemented in one and uncomplemented in another. Hence this can be written as $\bar{A}B$ (common terms).
- Thus the Boolean function $\bar{A}\,\bar{B}\,\bar{C}+\bar{A}B\bar{C}+\bar{A}BC$ is minimized to $\bar{A}\,\bar{C}+\bar{A}B$.

2. Minimize the Boolean function $f$ (A,B,C)= $\bar{A}\,\bar{B}\,\bar{C}+\bar{A}B\bar{C}+A\bar{B}\,\bar{C}+AB\bar{C}$ using K map.

   **Solution :**

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $1_0$ | $1_2$ | $1_6$ | $1_4$ |
| 1 | $1_1$ | 3 | $1_7$ | 5 |

$f$ (A,B,C)= $\bar{A}\,\bar{B}\,\bar{C}+\bar{A}B\bar{C}+A\bar{B}\,\bar{C}+AB\bar{C}$
= 000 + 010 +100+110
=$\sum$ (0,2,4,6)

‘1’ is marked in the first row.

- The cells are adjacent . The common term is $\bar{C}$ and the terms that appears complemented and uncomplemented forms are A and B.
- Thus the Boolean function

$\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + AB\overline{C}$ is minimized to $\overline{C}$.

3. Minimize the Boolean function $f$(A,B,C,D)= $\sum$(1,3,6,7,9,13,14,15)

Solution :

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 4 | 12 | 8 |
| 01 | $1_{1}$ | 5 | $1_{13}$ | $1_{9}$ |
| 11 | $1_{3}$ | $1_{7}$ | $1_{15}$ | 11 |
| 10 | 2 | $1_{6}$ | $1_{14}$ | 10 |

- Adjacent minterms are combined to form the following sub- cubes.
  (1,3) (13,9) (6,7,14,15)
- (1,3) can be represented as $\overline{A}\,\overline{B}\,D$ (common terms in representing 0,3)
- (13,9) can be represented as $A\overline{C}D$
- (6,7,14,15) can be represented as BC

Thus the Boolean function

$f$(A,B,C,D) is minimized to $\overline{A}\,\overline{B}\,D + A\overline{C}D + BC$.

**<u>MINIMIZATION IN POS FORM USING KARNAUGH MAP :</u>**

1. Obtain the minimal POS expression for the switching function given below using K map . $f$(A,B,C,D)= $\Pi$(1,2,4,5,6,7,8,9,10,11,13,14)

**Solution :**

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | $0_{4}$ | 12 | $0_{8}$ |
| 01 | $0_{1}$ | $0_{5}$ | $0_{13}$ | $0_{9}$ |
| 11 | 3 | $0_{7}$ | 15 | $0_{11}$ |
| 10 | $0_{2}$ | $0_{6}$ | $0_{14}$ | $0_{10}$ |

- First form the sub-cubes by combining the adjacent max terms .
  (4,5,7,6) (8,9,10,11) (1,5,9,13) (2,6,14,10)
- Consider (4,5,6,7) – the common terms are A+ $\overline{B}$
- Here the value of variable C and D changes.
- ( 8,9,10,11) indicates $\overline{A}$+B

- (1,5,9,13) indicates $C + \overline{D}$
- (2,6,14,10) indicates $\overline{C}$ +D

Thus the minimal POS expression for the switching function is given by the product of above four sum terms as

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

### 2.2.6 DON'T CARE COMBINATIONS:

- When the variables are not mutually independent, the function may assign '1' for some combinations and '0' for other combinations.
- The combinations for which the value of the function is not specified with certainty is called **don't care combinations**.
- These values are denoted by $\phi$ or D on K map.
- Example : Consider the following Boolean function together with the don't-care minterms :

$$f(A,B,C) = \sum (0,2,6)$$
$$d(A,B,C) = \sum (4,5)$$

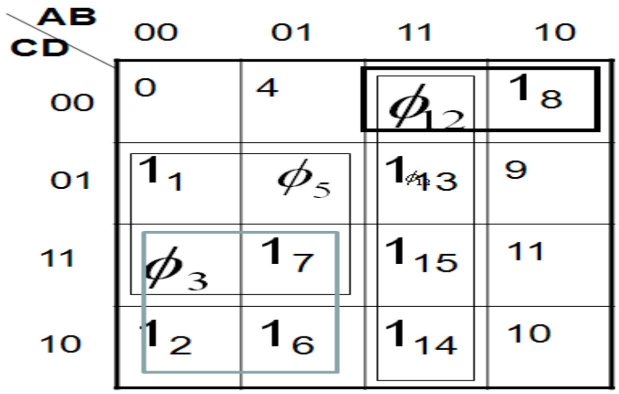
| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $1_0$ | $1_2$ | $1_6$ | $\phi_4$ |
| 1 | 1 | 3 | 7 $\phi_5$ | $\phi_5$ |

- The 1's and $\phi$ 's are combined to enclose the maximum number of adjacent squares.
- The simplified expression is $\overline{C}$
- If the don't care minterms are not included the simplified expression will be $\overline{A}\,\overline{C}$ + AB$\overline{C}$
- Thus the expression $\overline{C}$ represents the Boolean function $f$ (A,B,C) = $\sum$ (0,2,4,6)

**Example** : Minimize the multiple – output switching function given below, using a four- variable K map.

$$f(A,B,C,D)= \sum (1,2,6,7,8,13,14,15) + \sum_{\phi} (3,5,12)$$

**Solution :**




- Sometimes certain designs require only some minterms to be defined and few other can be either ‘0’s or ‘1’s.
- The $\phi$ terms represented are called unspecified states.