

## 2.2 REPRESENTATION OF LOGIC FUNCTION

The use of switching devices like transistors give rise to a special case of the Boolean algebra called as switching algebra. In switching algebra, all the variables assume one of the two values which are 0 and 1.

In Boolean algebra, 0 is used to represent the 'open' state or 'false' state of logic gate. Similarly, 1 is used to represent the 'closed' state or 'true' state of logic gate.

A Boolean expression is an expression which consists of variables, constants (0-false and 1-true) and logical operators which results in true or false.

A Boolean function is an algebraic form of Boolean expression. A Boolean function of  $n$ -variables is represented by  $f(x_1, x_2, x_3, \dots, x_n)$ . By using Boolean laws and theorems, we can simplify the Boolean functions of digital circuits. A brief note of different ways of representing a Boolean function is shown below.

- Sum-of-Products (SOP) Form
- Product-of-sums (POS) form
- Canonical forms

There are two types of canonical forms:

- Sum-of-min terms or Canonical SOP
- Product-of- max terms or Canonical POS

Boolean functions can be represented by using NAND gates and also by using K-map (Karnaugh map) method. We can standardize the Boolean expressions by using by two standard forms.

SOP form – Sum Of Products form

POS form – Product Of Sums form

Standardization of Boolean equations will make the implementation, evolution and simplification easier and more systematic.

### 2.2.1 Sum of Product (SOP) Form

The sum-of-products (SOP) form is a method (or form) of simplifying the Boolean expressions of logic gates. In this SOP form of Boolean function representation, the

variables are operated by AND (product) to form a product term and all these product terms are ORed (summed or added) together to get the final function.

A sum-of-products form can be formed by adding (or summing) two or more product terms using a Boolean addition operation. Here the product terms are defined by using the AND operation and the sum term is defined by using OR operation.

The sum-of-products form is also called as Disjunctive Normal Form as the product terms are ORed together and Disjunction operation is logical OR. Sum-of-products form is also called as Standard SOP.

SOP form representation is most suitable to use them in FPGA (Field Programmable Gate Arrays).

Examples

$$AB + ABC + CDE$$

$$(AB)^- + ABC + CDE^-$$

SOP form can be obtained by

- Writing an AND term for each input combination, which produces HIGH output.
- Writing the input variables if the value is 1, and write the complement of the variable if its value is 0.
- OR the AND terms to obtain the output function.

Ex: Boolean expression for majority function  $F = A'BC + AB'C + ABC' + ABC$

Truth table:

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

Now write the input variables combination with high output.  $F = AB + BC + AC$ .

Checking

By Idempotence law, we know that

$$([ABC + ABC]) + ABC = (ABC + ABC) = ABC$$

Now the function  $F = A'BC + AB'C + ABC' + ABC$

$$= A'BC + AB'C + ABC' + ([ABC + ABC]) + ABC$$

$$= (ABC + ABC') + (ABC + AB'C) + (ABC + A'BC)$$

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

$$= AB + BC + AC.$$

### 2.2.2 Product of Sums (POS) Form

The product of sums form is a method (or form) of simplifying the Boolean expressions of logic gates. In this POS form, all the variables are ORed, i.e. written as sums to form sum terms.

All these sum terms are ANDed (multiplied) together to get the product-of-sum form. This form is exactly opposite to the SOP form. So this can also be said as “Dual of SOP form”.

Here the sum terms are defined by using the OR operation and the product term is defined by using AND operation. When two or more sum terms are multiplied by a Boolean OR operation, the resultant output expression will be in the form of product-of-sums form or POS form.

The product-of-sums form is also called as Conjunctive Normal Form as the sum terms are ANDed together and Conjunction operation is logical AND. Product-of-sums form is also called as Standard POS.

Examples

$$(A+B) * (A + B + C) * (C +D)$$

$$(A+B)^{-} * (C + D + E)^{-}$$

POS form can be obtained by

- Writing an OR term for each input combination, which produces LOW output.
- Writing the input variables if the value is 0, and write the complement of the variable if its value is 1.

- AND the OR terms to obtain the output function.

Ex: Boolean expression for majority function  $F = (A + B + C) (A + B + C') (A + B' + C) (A' + B + C)$

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

Now write the input variables combination with high output.  $F = AB + BC + AC$ .

Checking

By Idempotence law, we know that

$$[(A + B + C) (A + B + C)] (A + B + C) = [(A + B + C)] (A + B + C) = (A + B + C)$$

Now the function

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

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

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

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

$$= [(A + B) + (C * C')] [(B + C) + (A * A')] [(A + C) + (B * B')]$$

$$= [(A + B) + 0] [(B + C) + 0] [(A + C) + 0] = (A + B) (B + C) (A + C)$$

### 2.2.3 Canonical Form (Standard SOP and POS Form)

Any Boolean function that is expressed as a sum of minterms or as a product of max terms is said to be in its “canonical form”.

It mainly involves in two Boolean terms, “minterms” and “maxterms”.

When the SOP form of a Boolean expression is in canonical form, then each of its product term is called ‘minterm’. So, the canonical form of sum of products function is also

known as “minterm canonical form” or Sum-of-minterms or standard canonical SOP form.

Similarly, when the POS form of a Boolean expression is in canonical form, then each of its sum term is called ‘maxterm’. So, the canonical form of product of sums function is also known as “maxterm canonical form or Product-of sum or standard canonical POS form”.

### Min terms

A minterm is defined as the product term of  $n$  variables, in which each of the  $n$  variables will appear once either in its complemented or un-complemented form. The min term is denoted as  $m_i$  where  $i$  is in the range of  $0 \leq i < 2^n$ .

A variable is in complemented form, if its value is assigned to 0, and the variable is un-complimented form, if its value is assigned to 1.

For a 2-variable ( $x$  and  $y$ ) Boolean function, the possible minterms are:

$x'y'$ ,  $x'y$ ,  $xy'$  and  $xy$ .

For a 3-variable ( $x$ ,  $y$  and  $z$ ) Boolean function, the possible minterms are:

$x'y'z'$ ,  $x'y'z$ ,  $x'yz'$ ,  $x'yz$ ,  $xy'z'$ ,  $xy'z$ ,  $xyz'$  and  $xyz$ .

- 1 – Minterms = minterms for which the function  $F = 1$ .
- 0 – Minterms = minterms for which the function  $F = 0$ .

Any Boolean function can be expressed as the sum (OR) of its 1- min terms. The representation of the equation will be

- $F(\text{list of variables}) = \Sigma(\text{list of 1-min term indices})$

Ex:  $F(x, y, z) = \Sigma(3, 5, 6, 7)$

The inverse of the function can be expressed as a sum (OR) of its 0- min terms. The representation of the equation will be

- $F(\text{list of variables}) = \Sigma(\text{list of 0-min term indices})$

Ex:  $F'(x, y, z) = \Sigma(0, 1, 2, 4)$

Examples of canonical form of sum of products expressions (min term canonical form):

i)  $Z = XY + XZ'$

ii)  $F = XYZ' + X'YZ + X'YZ' + XY'Z + XYZ$

In standard SOP form, the maximum possible product terms for  $n$  number of variables are given by  $2^n$ . So, for 2 variable equations, the product terms are  $2^2 = 4$ . Similarly, for 3 variable equations, the product terms are  $2^3 = 8$ .

### Max terms

A max term is defined as the product of  $n$  variables, within the range of  $0 \leq i < 2^n$ . The max term is denoted as  $M_i$ . In max term, each variable is complimented, if its value is assigned to 1, and each variable is un-complimented if its value is assigned to 0.

For a 2-variable ( $x$  and  $y$ ) Boolean function, the possible max terms are:

$x + y$ ,  $x + y'$ ,  $x' + y$  and  $x' + y'$ .

For a 3-variable ( $x$ ,  $y$  and  $z$ ) Boolean function, the possible maxterms are:

$x + y + z$ ,  $x + y + z'$ ,  $x + y' + z$ ,  $x + y' + z'$ ,  $x' + y + z$ ,  $x' + y + z'$ ,  $x' + y' + z$  and  $x' + y' + z'$ .

- 1 – Max terms = max terms for which the function  $F = 1$ .
- 0 – max terms = max terms for which the function  $F = 0$ .

Any Boolean function can be expressed the product (AND) of its 0 – max terms. The representation of the equation will be

- $F(\text{list of variables}) = \Pi (\text{list of 0-max term indices})$

Ex:  $F(x, y, z) = \Pi(0, 1, 2, 4)$

The inverse of the function can be expressed as a product (AND) of its 1 – max terms.

The representation of the equation will be

- $F(\text{list of variables}) = \Pi (\text{list of 1-max term indices})$

Ex:  $F'(x, y, z) = \Pi(3, 5, 6, 7)$

Examples of canonical form of product of sums expressions (max term canonical form):

i.  $Z = (X + Y)(X + Y')$

ii.  $F = (X' + Y + Z')(X' + Y + Z)(X' + Y' + Z')$

In standard POS form, the maximum possible sum terms for  $n$  number of variables are given by  $2^n$ . So, for 2 variable equations, the sum terms are  $2^2 = 4$ . Similarly, for 3 variable equations, the sum terms are  $2^3 = 8$ .

Table for  $2^n$  min terms and  $2^n$  max terms

The below table will make you understand about the representation of the mean terms and max terms of 3 variables.

Variables			Min terms	Max terms
A	B	C	$m_i$	$M_i$
0	0	0	$A' B' C' = m_0$	$A + B + C = M_0$
0	0	1	$A' B' C = m_1$	$A + B + C' = M_1$
0	1	0	$A' B C' = m_2$	$A + B' + C = M_2$
0	1	1	$A' B C = m_3$	$A + B' + C' = M_3$
1	0	0	$A B' C' = m_4$	$A' + B + C = M_4$
1	0	1	$A B' C = m_5$	$A' + B + C' = M_5$
1	1	0	$A B C' = m_6$	$A' + B' + C = M_6$
1	1	1	$A B C = m_7$	$A' + B' + C' = M_7$

### Conversions of Canonical Forms

We can represent the one canonical formed equation in other canonical form i.e. we can represent the SOP form of equation in POS form and POS form equation in SOP form. To convert the canonical equations, we interchange the  $\Sigma$  and  $\Pi$  symbols after listing out the index numbers of the equations, which are excluded from the original form of equation.

The important thing to remember about Boolean functions is that, the SOP and POS forms are Duals to each other. There are 2 steps to follow to convert the canonical form of the equations. They are

Step 1: Interchanging the operational symbols,  $\Sigma$  and  $\Pi$  in the equation.

Step 2: Use the De Morgan's principle of Duality to the index numbers of the Boolean function or writing the indexes of the terms that are not presented in the given form of equation.

#### 2.2.4 Conversion of SOP form to POS form

To convert the SOP form into POS form, first we should change the  $\Sigma$  to  $\Pi$  and then write the numeric indexes of missing variables of the given Boolean function.

Example:

The SOP function

$F = \sum A, B, C (0, 2, 3, 5, 7) = A' B' C' + A B' C' + A B' C + ABC' + ABC$  is written in POS form by

Step 1: changing the operational sign to  $\Pi$

Step 2: writing the missing indexes of the terms, 001, 100 and 110. Now write the sum form for these noted terms.

$$001 = (A + B + C) \quad 100 = (A + B' + C') \quad 110 = (A + B' + C')$$

Writing down the new equation in the form of POS form,

$$F = \Pi A, B, C (1, 4, 6) = (A + B + C) * (A + B' + C') * (A + B' + C')$$

Conversion of POS form to SOP form

To convert the POS form into SOP form, first we should change the  $\Pi$  to  $\Sigma$  and then write the numeric indexes of missing variables of the given Boolean function.

Ex: The POS function  $F = \Pi A, B, C (2, 3, 5) = A B' C' + A B' C + ABC'$  is written in SOP form by

Step 1: changing the operational sign to  $\Sigma$

Step 2: writing the missing indexes of the terms, 000, 001, 100, 110, and 111. Now write the product form for these noted terms.

$$000 = A' * B' * C' \quad 001 = A' * B' * C \quad 100 = A * B' * C'$$

$$110 = A * B * C' \quad 111 = A * B * C$$

Writing down the new equation in the form of SOP form,

$$F = \Sigma A, B, C (0, 1, 4, 6, 7) = (A' * B' * C') + (A' * B' * C) + (A * B' * C') + (A * B * C') + (A * B * C)$$

Conversion of SOP form to standard SOP form or Canonical SOP form

We can include all the variables in each product term of the SOP form equation, which doesn't have all the variables by converting into standard SOP form. The normal SOP form function can be converted to standard SOP form by using the Boolean algebraic law,  $(A + A' = 1)$  and by following the below steps.

Step 1:

By multiplying each non-standard product term with the sum of its missing variable and its complement, which results in 2 product terms

Step 2:

By repeating the step 1, until all resulting product terms contain all variables



By these two steps we can convert the SOP function into standard SOP function. In this process, for each missing variable in the function, the number of product terms will double.

Example:

Convert the non standard SOP function  $F = x y + x z + y z$

Sol:

$$F = x y + x z + y z$$

$$= x y (z + z') + x (y + y') z + (x + x') y z$$

$$= x y z + x y z' + x y z + x y' z + x y z + x' y z$$

$$= x y z + x y z' + x y' z + x' y z$$

The standard SOP form is  $F = x y z + x y z' + x y' z + x' y z$

### 2.2.5 Conversion of POS form to standard POS form or Canonical POS form

We can include all the variables in each product term of the POS form equation, which doesn't have all the variables by converting into standard POS form. The normal POS form function can be converted to standard POS form by using the Boolean algebraic law,  $(A * A' = 0)$  and by following the below steps.

Step 1:

By adding each non-standard sum term to the product of its missing variable and its complement, which results in 2 sum terms

Step 2:

Applying Boolean algebraic law,  $A + BC = (A + B) * (A + C)$

Step 3:

By repeating the step 1, until all resulting sum terms contain all variables

By these three steps we can convert the POS function into standard POS function.

Example:

$$F = (A' + B + C) * (B' + C + D') * (A + B' + C' + D)$$

In the first term, the variable D or D' is missing, so we add  $D * D' = 1$  to it. Then

$$(A' + B + C + D * D') = (A' + B + C + D) * (A' + B + C + D')$$

Similarly, in the second term, the variable A or A' is missing, so we add  $A * A' = 1$  to it.

Then

$$(B' + C + D' + A * A') = (A + B' + C + D') * (A' + B' + C + D')$$

The third term is already in the standard form, as it has all the variables. Now the standard POS form equation of the function is

$$F = (A' + B + C + D) * (A' + B + C + D') * (A + B' + C + D') * (A' + B' + C + D') * (A + B' + C' + D)$$

