

2.8 CANONICAL AND STANDARD FORMS

Logical functions are generally expressed in terms of different combinations of logical variables with their true forms as well as the complement forms. Binary logic values obtained by the logical functions and logic variables are in binary form. An arbitrary logic function can be expressed in the following forms.

(i) Sum of the Products (SOP)

(ii) Product of the Sums (POS)

Product Term. In Boolean algebra, the logical product of several variables on which a function depends is considered to be a product term. In other words, the AND function is referred to as a product term or standard product. The variables in a product term can be either in true form or in complemented form. For example, ABC' is a product term.

Sum Term. An OR function is referred to as a sum term. The logical sum of several variables on which a function depends is considered to be a sum term. Variables in a sum term can also be either in true form or in complemented form. For example, $A + B + C'$ is a sum term.

Sum of Products (SOP). The logical sum of two or more logical product terms is referred to as a sum of products expression. It is basically an OR operation on AND operated variables. For example, $Y = AB + BC + AC$ or $Y = A'B + BC + AC'$ are sum of products expressions.

Product of Sums (POS). Similarly, the logical product of two or more logical sum terms is called a product of sums expression. It is an AND operation on OR operated variables. For example, $Y = (A + B + C)(A + B' + C)(A + B + C')$ or $Y = (A + B + C)(A' + B' + C')$ are product of sums expressions.

Standard form. The standard form of the Boolean function is when it is expressed in sum of the products or product of the sums fashion. The examples stated above, like $Y = AB + BC + AC$ or $Y = (A + B + C)(A + B' + C)(A + B + C')$ are the standard forms.

However, Boolean functions are also sometimes expressed in nonstandard forms like $F = (AB + CD)(A'B' + C'D')$, which is neither a sum of products form nor a product of sums form. However, the same expression can be converted to a standard form with help of various Boolean properties, as:

$$F = (AB + CD)(A'B' + C'D') = A'B'CD + ABC'D'$$

2.8.1 Minterm

A product term containing all n variables of the function in either true or complemented form is called the minterm. Each minterm is obtained by an AND operation of the variables in their true form or complemented form. For a two-variable function, four different combinations are possible, such as, $A'B'$, $A'B$, AB' , and AB . These product terms are called the fundamental products or standard products or minterms. In the minterm, a variable will possess the value 1 if it is in true or uncomplemented form, whereas, it contains the value 0 if it is in complemented form. For three variables function, eight minterms are possible as listed in the following table in Figure 2.9.

A	B	C	Minterm
0	0	0	$A'B'C'$
0	0	1	$A'B'C$
0	1	0	$A'BC'$
0	1	1	$A'BC$
1	0	0	$AB'C'$
1	0	1	$AB'C$
1	1	0	ABC'
1	1	1	ABC

Figure 2-9

So, if the number of variables is n , then the possible number of minterms is 2^n . The main property of a minterm is that it possesses the value of 1 for only one combination of n input variables and the rest of the $2^n - 1$ combinations have the logic value of 0. This means, for the above three variables example, if $A = 0$, $B = 1$, $C = 1$ *i.e.*, for input combination of 011, there is only one combination $A'BC$ that has the value 1, the rest of the seven combinations have the value 0.

Canonical Sum of Product Expression. When a Boolean function is expressed as the logical sum of all the minterms from the rows of a truth table, for which the value of the function is 1, it is referred to as the *canonical sum of product expression*. The same can be expressed in a compact form by listing the corresponding decimal-equivalent codes of the minterms containing a function value of 1. For example, if the canonical sum of product form of a three-variable logic function F has the minterms $A'BC$, $AB'C$, and ABC' , this can be expressed as the sum of the decimal codes corresponding to these minterms as below.

$$\begin{aligned} F(A,B,C) &= (3,5,6) \\ &= m_3 + m_5 + m_6 \\ &= A'BC + AB'C + ABC' \end{aligned}$$

where $\Sigma (3,5,6)$ represents the summation of minterms corresponding to decimal codes 3, 5, and 6.

The canonical sum of products form of a logic function can be obtained by using the following procedure:

1. Check each term in the given logic function. Retain if it is a minterm, continue to examine the next term in the same manner.
2. Examine for the variables that are missing in each product which is not a minterm. If the missing variable in the minterm is X, multiply that minterm with $(X+X')$.
3. Multiply all the products and discard the redundant terms.

Here are some examples to explain the above procedure.

Example 2.6. Obtain the canonical sum of product form of the following function:

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

Solution. The given function contains two variables A and B. The variable B is missing from the first term of the expression and the variable A is missing from the second term of the expression. Therefore, the first term is to be multiplied by $(B + B')$ and the second term is to be multiplied by $(A + A')$ as demonstrated below.

$$\begin{aligned} F(A, B) &= A + B \\ &= A.1 + B.1 \\ &= A(B + B') + B(A + A') \\ &= AB + AB' + AB + A'B \\ &= AB + AB' + A'B \quad (\text{as } AB + AB = AB) \end{aligned}$$

Hence the canonical sum of the product expression of the given function is

$$F(A, B) = AB + AB' + A'B.$$

Example 2.7. Obtain the canonical sum of product form of the following function.

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

Solution. Here neither the first term nor the second term is minterm. The given function contains three variables A, B, and C. The variables B and C are missing from the first term of the expression and the variable A is missing from the second term of the expression. Therefore, the first term is to be multiplied by $(B + B')$ and $(C + C')$. The second term is to be multiplied by $(A + A')$. This is demonstrated below.

$$\begin{aligned} F(A, B, C) &= A + BC \\ &= A(B + B')(C + C') + BC(A + A') \\ &= (AB + AB')(C + C') + ABC + A'BC \\ &= ABC + AB'C + ABC' + AB'C' + ABC + A'BC \\ &= ABC + AB'C + ABC' + AB'C' + A'BC \quad (\text{as } ABC + ABC = ABC) \end{aligned}$$

Hence the canonical sum of the product expression of the given function is

$$F(A, B, C) = ABC + AB'C + ABC' + AB'C' + A'BC.$$

2.8.2 Maxterm

A sum term containing all n variables of the function in either true or complemented form is called the maxterm. Each maxterm is obtained by an OR operation of the variables in their true form or complemented form. Four different combinations are possible for a two-variable function, such as, $A' + B'$, $A' + B$, $A + B'$, and $A + B$. These sum terms are called the standard sums or maxterms. Note that, in the maxterm, a variable will possess the value 0, if it is in true or uncomplemented form, whereas, it contains the value 1, if it is in complemented form. Like minterms, for a three-variable function, eight maxterms are also possible as listed in the following table in Figure 2.10.

A	B	C	Maxterm
0	0	0	$A + B + C$
0	0	1	$A + B + C'$
0	1	0	$A + B' + C$
0	1	1	$A + B' + C'$
1	0	0	$A' + B + C$
1	0	1	$A' + B + C'$
1	1	0	$A' + B' + C$
1	1	1	$A' + B' + C'$

Figure 2-10

So, if the number of variables is n , then the possible number of maxterms is 2^n . The main property of a maxterm is that it possesses the value of 0 for only one combination of n input variables and the rest of the $2^n - 1$ combinations have the logic value of 1. This means, for the above three variables example, if $A = 1$, $B = 1$, $C = 0$ *i.e.*, for input combination of 110, there is only one combination $A' + B' + C$ that has the value 0, the rest of the seven combinations have the value 1.

Canonical Product of Sum Expression. When a Boolean function is expressed as the logical product of all the maxterms from the rows of a truth table, for which the value of the function is 0, it is referred to as the *canonical product of sum expression*. The same can be expressed in a compact form by listing the corresponding decimal equivalent codes of the maxterms containing a function value of 0. For example, if the canonical product of sums form of a three-variable logic function F has the maxterms $A + B + C$, $A + B' + C$, and $A' + B + C'$, this can be expressed as the product of the decimal codes corresponding to these maxterms as below,

$$\begin{aligned}
 F(A,B,C) &= \Pi(0,2,5) \\
 &= M_0 M_2 M_5 \\
 &= (A + B + C)(A + B' + C)(A' + B + C')
 \end{aligned}$$

where $\Pi(0,2,5)$ represents the product of maxterms corresponding to decimal codes 0, 2, and 5. The canonical product of sums form of a logic function can be obtained by using the following procedure.

1. Check each term in the given logic function. Retain it if it is a maxterm, continue to examine the next term in the same manner.
2. Examine for the variables that are missing in each sum term that is not a maxterm. If the missing variable in the maxterm is X, add that maxterm with $(X.X')$.
3. Expand the expression using the properties and postulates as described earlier and discard the redundant terms.

Some examples are given here to explain the above procedure.

Example 2.8. Obtain the canonical product of the sum form of the following function.

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

Solution. In the above three-variable expression, C is missing from the first term, A is missing from the second term, and B is missing from the third term. Therefore, CC' is to be added with first term, AA' is to be added with the second, and BB' is to be added with the third term. This is shown below.

$$\begin{aligned} F(A, B, C) &= (A + B')(B + C)(A + C') \\ &= (A + B' + 0)(B + C + 0)(A + C' + 0) \\ &= (A + B' + CC')(B + C + AA')(A + C' + BB') \\ &= (A + B' + C)(A + B' + C')(A + B + C)(A' + B + C)(A + B + C')(A + B' + C') \\ & \text{[using the distributive property, as } X + YZ = (X + Y)(X + Z)\text{]} \\ &= (A + B' + C)(A + B' + C')(A + B + C)(A' + B + C)(A + B + C') \\ & \text{[as } (A + B' + C)(A + B' + C') = A + B' + C\text{]} \end{aligned}$$

Hence the canonical product of the sum expression for the given function is

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

Example 2.9. Obtain the canonical product of the sum form of the following function.

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

Solution. In the above three-variable expression, the function is given at sum of the product form. First, the function needs to be changed to product of the sum form by applying the distributive law as shown below.

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

Now, in the above expression, C is missing from the first term and B is missing from the second term. Hence CC' is to be added with the first term and BB' is to be added with the second term as shown below.

$$\begin{aligned} F(A, B, C) &= (A + B')(A + C) \\ &= (A + B' + CC')(A + C + BB') \\ &= (A + B' + C)(A + B' + C')(A + B + C)(A + B' + C) \\ & \text{[using the distributive property, as } X + YZ = (X + Y)(X + Z)\text{]} \end{aligned}$$

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

$$[\text{as } (A + B' + C) (A + B' + C) = A + B' + C]$$

Hence the canonical product of the sum expression for the given function is
 $F(A, B, C) = (A + B' + C) (A + B' + C') (A + B + C)$.

2.8.3 Deriving a Sum of Products (SOP) Expression from a Truth Table

The sum of products (SOP) expression of a Boolean function can be obtained from its truth table summing or performing OR operation of the product terms corresponding to the combinations containing a function value of 1. In the product terms the input variables appear either in true (uncomplemented) form if it contains the value 1, or in complemented form if it possesses the value 0.

Now, consider the following truth table in Figure 2.11, for a three-input function Y. Here the output Y value is 1 for the input conditions of 010, 100, 101, and 110, and their corresponding product terms are $A'BC'$, $AB'C'$, $AB'C$, and ABC' respectively.

Inputs			Output Y	Product terms	Sum terms
A	B	C			
0	0	0	0		$A + B + C$
0	0	1	0		$A + B + C'$
0	1	0	1	$A'BC'$	
0	1	1	0		$A + B' + C'$
1	0	0	1	$AB'C'$	
1	0	1	1	$AB'C$	
1	1	0	1	ABC'	
1	1	1	0		$A' + B' + C'$

Figure 2-11

The final sum of products expression (SOP) for the output Y is derived by summing or performing an OR operation of the four product terms as shown below.

$$Y = A'BC' + AB'C' + AB'C + ABC'$$

In general, the procedure of deriving the output expression in SOP form from a truth table can be summarized as below.

1. Form a product term for each input combination in the table, containing an output value of 1.
2. Each product term consists of its input variables in either true form or complemented form. If the input variable is 0, it appears in complemented form and if the input variable is 1, it appears in true form.
3. To obtain the final SOP expression of the output, all the product terms are OR operated.

2.8.4 Deriving a Product of Sums (POS) Expression from a Truth Table

As explained above, the product of sums (POS) expression of a Boolean function can also be obtained from its truth table by a similar procedure. Here, an AND operation is performed on the sum terms corresponding to the combinations containing a function value of 0. In the sum terms the input variables appear either in true (uncomplemented) form if it contains the value 0, or in complemented form if it possesses the value 1.

Now, consider the same truth table as shown in Figure 2.11, for a three-input function Y. Here the output Y value is 0 for the input conditions of 000, 001, 011, and 111, and their corresponding product terms are $A + B + C$, $A + B + C'$, $A + B' + C'$, and $A' + B' + C'$ respectively.

So now, the final product of sums expression (POS) for the output Y is derived by performing an AND operation of the four sum terms as shown below.

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

In general, the procedure of deriving the output expression in POS form from a truth table can be summarized as below.

1. Form a sum term for each input combination in the table, containing an output value of 0.
2. Each product term consists of its input variables in either true form or complemented form. If the input variable is 1, it appears in complemented form and if the input variable is 0, it appears in true form.
3. To obtain the final POS expression of the output, all the sum terms are AND operated.

2.9.5 Conversion between Canonical Forms

From the above example, it may be noted that the complement of a function expressed as the sum of products (SOP) equals to the sum of products or sum of the minterms which are missing from the original function. This is because the original function is expressed by those minterms that make the function equal to 1, while its complement is 1 for those minterms whose values are 0. According to the truth table given in Figure 2.11:

$$\begin{aligned} F(A,B,C) &= \Sigma (2,4,5,6) \\ &= m_2 + m_4 + m_5 + m_6 \\ &= A'BC' + AB'C' + AB'C + ABC' \end{aligned}$$

This has the complement that can be expressed as

$$\begin{aligned} F'(A,B,C) &= (0,1,3,7) \\ &= m_0 + m_1 + m_3 + m_7 \end{aligned}$$

Now, if we take complement of F' by DeMorgan's theorem, we obtain F as

$$\begin{aligned} F(A,B,C) &= (m_0 + m_1 + m_3 + m_7)' \\ &= m_0' m_1' m_3' m_7' \\ &= M_0 M_1 M_3 M_7 \\ &= \Pi(0,1,3,7) \\ &= (A + B + C)(A + B + C') (A + B' + C') (A' + B' + C'). \end{aligned}$$

The last conversion follows from the definition of minterms and maxterms as shown in the tables in Figures 2.9 and 2.10. It can be clearly noted that the following relation holds true $m'_j = M_j$.

That is, the maxterm with subscript j is a complement of the minterm with the same subscript j , and vice versa.

This example demonstrates the conversion between a function expressed in sum of products (SOP) and its equivalent in product of maxterms. A similar example can show the conversion between the product of sums (POS) and its equivalent sum of minterms. In general, to convert from one canonical form to other canonical form, it is required to interchange the symbols Σ and π , and list the numbers which are missing from the original form.

Note that, to find the missing terms, the total 2^n number of minterms or maxterms must be realized, where n is the number of variables in the function.