

جامعة المستقبل
AL MUSTAQBAL UNIVERSITY



قسم الامن

السيبران

ي

DEPARTMENT OF CYBER SECURITY

SUBJECT:

COMPUTER ORGANIZATION & LOGIC DESIGN

CLASS:

FIRST

LECTURE: (5)

(LOGIC GATES)

LECTURER:

MSC :MUNTATHER AL-MUSSAWEE

Definition of SOP

When we add two or multiple product terms by a boolean addition, the output expression is a sum-of-products (SOP).

For example, the expression $a'bc' + a'bd' + a'bc'd$ shows a SOP expression. It can also have a single variable term within the expression like $a + bc + a'b$. These logical expressions are simplified in a way that they must not contain redundant information while creating the minimal version of it.

Domain of a boolean expression

The group of variables, either complimented or uncomplimented, comprised in a boolean expression, is known as the domain. Let's suppose, we have an expression $a'b + ab'c$ then the domain of this expression would be the set of the variables a, b, c .

Implementation of the SOP form

It is mainly implemented by an AND-OR logic where the product of the variables are first produced by AND gate and then added by the OR gates. For example, the expression “ $ab'+bcd+ac$ ” can be expressed by the logic circuit shown in figure 1.1 where the output P of the OR gate is the SOP expression.

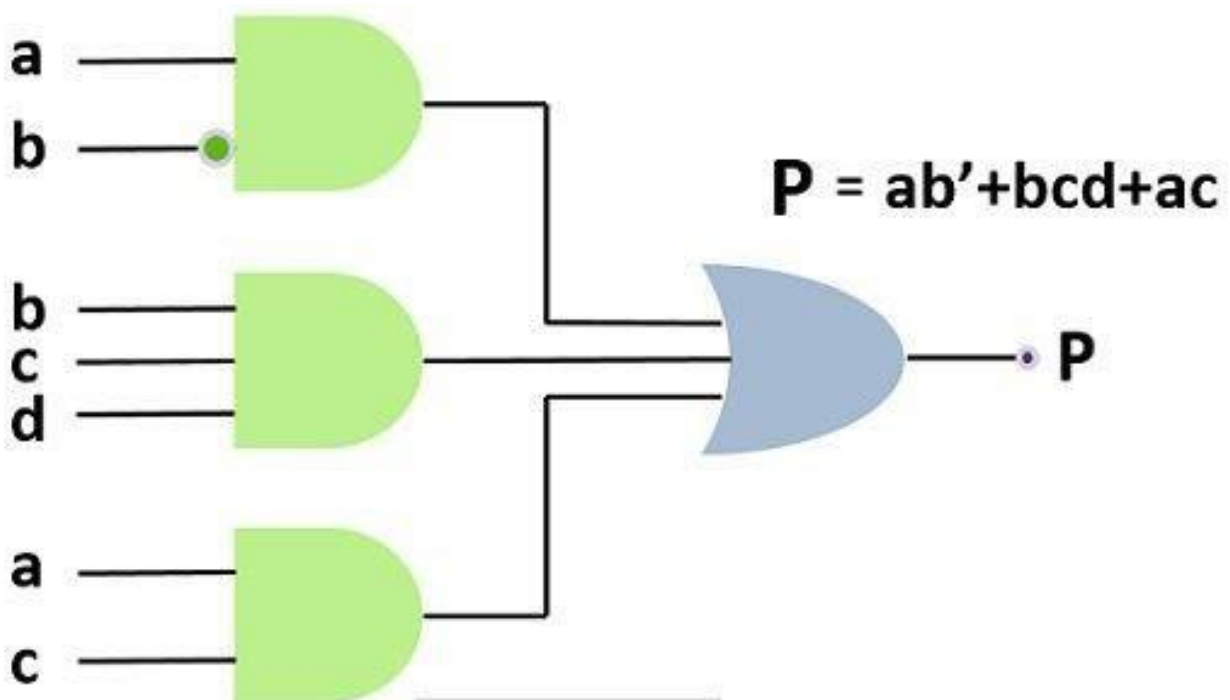


Figure 1.1

Steps for converting the product term into standard SOP

Here the standard SOP or canonical SOP refers to an expression in which all the variables of the domain are present. For generating, standard SOP from the product term the boolean rule “ $A+A'=1$ ” (the output is ‘1’ when a variable added to its complement) is used and below given steps are followed.

Each non-standard term is multiplied by a term constructed by the addition of the absent variable and its complement. This produces two product terms, as we know that anything can be multiplied by '1' without changing its value.

Repeat the step '1' until all the domain variables are present in the expression R.

Example

The term $ab'c+a'b'+abc'd$ converted into the standard SOP or canonical SOP by multiplying the part of the term by the missing term. Such as $a'b'$ is multiplied with the $c+c'$. Similarly, the whole expression is converted in its canonical form by the following given steps.

$$\begin{aligned} &= ab'c + a'b'(c+c') + abc'd \\ &= ab'c + a'b'c + a'b'c' + abc'd \\ &= ab'c(d+d') + a'b'c + a'b'c' + abc'd \\ &= ab'cd + ab'cd' + a'b'c(d+d') + a'b'c' + abc'd \\ &= ab'cd + ab'cd' + a'b'cd + a'b'cd' + a'b'c'(d+d') + abc'd \\ &= ab'cd + ab'cd' + a'b'cd + a'b'cd' + a'b'c'd + a'b'c'd' + abc'd \end{aligned}$$

Now there are various terms which are used while generating the reduced logic function such as **minterm**, **maxterm**, **k-map** (**Karnaugh Map**), which we will elucidate further in the article.

Minterm

In these terms, the input variables making up a boolean expression is the dot product of each other, and it is also known as minterm or product term. Mn following table presents the minterms of the variables.

Inputs			Output	Minterms
A	B	C	X	M_n
0	0	0	0	$m_0 = A'B'C'$
0	0	1	1	$m_1 = A'B'C$
0	1	0	1	$m_2 = A'BC'$
0	1	1	0	$m_3 = A'BC$
1	0	0	0	$m_4 = AB'C'$
1	0	1	0	$m_5 = AB'C$
1	1	0	1	$m_6 = ABC'$
1	1	1	1	$m_7 = ABC$

Figure 1.2

SOP expression from a Truth table

Suppose, we have a truth table (as shown in figure 1.3) in which each term of the input variables are written as the product of all the terms. To determine the input combinations that exist, we need to select the output having value 1 and convert the binary into relevant product term. Here, we will consider '0' as the variable and '1' as the compliment of the variable.

Inputs			Output
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figure 1.3

$$Y(A,B,C,D) = \sum m (0,2,3,6,7)$$

Only those product terms are selected where the output value is 1.

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

So, in accordance with the truth table, the boolean function (Canonical SOP form) in the minterm is :

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

0 is variable and 1 is the complement of the variable.

Definition of POS

POS (Product of Sums) is the representation of the boolean function in which the variables are first summed, and then the boolean product is applied in the sum terms. For example, $(a'+b).(a+b'+c)$ is POS expression where we can see that the variables are added then each bigger term is the product of the other.

Implementation of the POS form

It just needs the variables to be inserted as the inputs to the OR gate. The terms generated by the OR gates are inserted in the AND gate. The sum term is formed by an OR operation, and product of two or multiple sum terms is created by an AND operation. To understand the POS implementation refer the below given figure 2.1 of the expression $(a+b).(b+c+d).(a+c)$.

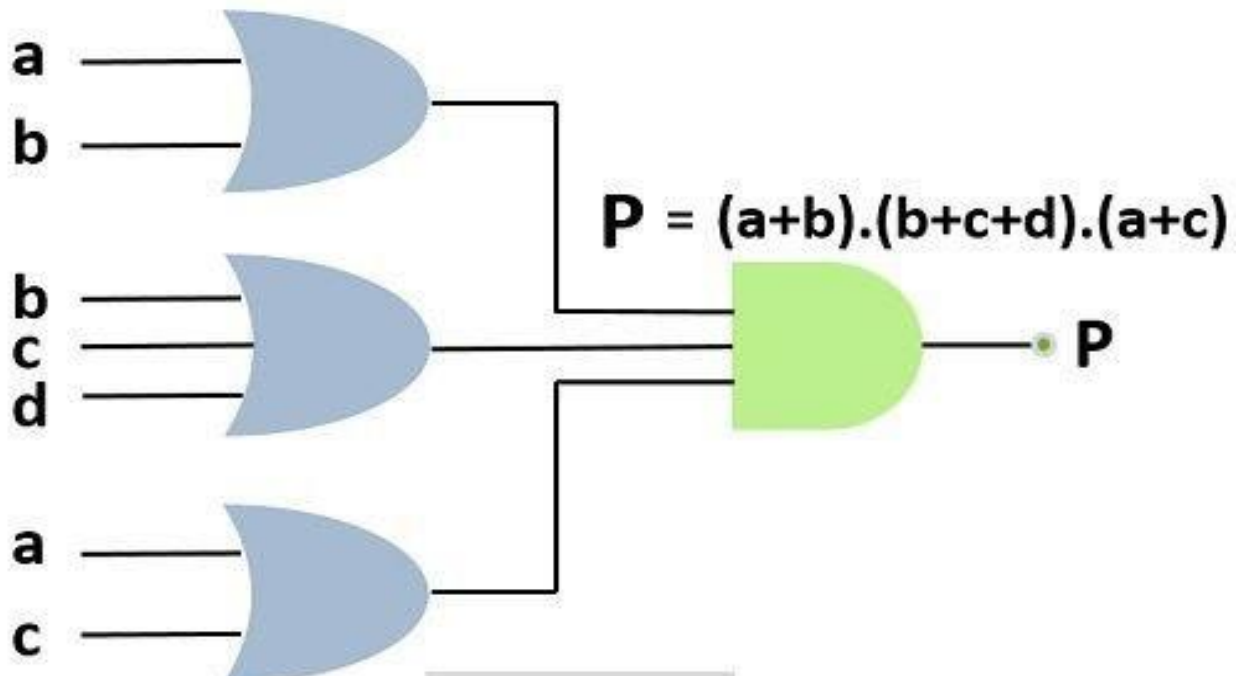


Figure 2.1

Steps for converting the product term into standard POS

Similar to the previous explanation, the standard or canonical POS is in which sum terms does not include all of the variables in the domain of the expression. Here, also we use the boolean algebra rule 8 “A.A’=0” (a variable multiplied by its complement is 0) to convert a term in a standard form, and the method is given below.

In the very first step, each non-standard term added with a term comprised of the product of the absent variable and its complement. This will produce two sum term, and '0' can be added to any term without changing its value.

Then, rule 12 (i.e. $A+BC=(A+B)(A+C)$) is applied to the terms.

The first two steps are redone again and again until all the sum terms involve all the variables present in the domain either in the complemented and uncomplemented form.

Example

The term $(a+b'+c)(b'+c+d')(a+b'+c'+d)$ is translated into the standard POS or canonical POS by adding each term with the missing term (which is a product of its complement). Such as "a+b'+c" is summed with the d.d'. In this way, the entire expression is converted in its canonical form by the following given steps.

$$\begin{aligned} &= (a+b'+c).(b'+c+d').(a+b'+c'+d) \\ &= (a+b'+c+d.d').(b'+c+d').(a+b'+c'+d) \\ &= (a+b'+c+d).(a+b'+c+d').(b'+c+d'+a.a').(a+b'+c'+d) \\ &= (a+b'+c+d).(a+b'+c+d').(b'+c+d'+a).(b'+c+d'+a').(a+b'+c'+d) \\ &= (a+b'+c+d).(a+b'+c+d').(a'+b'+c+d').(a+b'+c'+d) \end{aligned}$$

Maxterm

These are the terms in which the input variables are present in summation form, alternately called as sum term. The below-given table represents the M_n , maxterms of the variables.

Inputs			Output	Maxterms
A	B	C	X	M_n
0	0	0	0	$m_0 = A+B+C$
0	0	1	1	$m_1 = A+B+C'$
0	1	0	1	$m_2 = A+B'+C$
0	1	1	0	$m_3 = A+B'+C'$
1	0	0	0	$m_4 = A'+B+C$
1	0	1	0	$m_5 = A'+B+C'$
1	1	0	1	$m_6 = A'+B'+C$
1	1	1	1	$m_7 = A'+B'+C'$

Figure 2.2

POS expression from a Truth table

To find the POS expression with the help of a truth table (figure 2.3), record the binary values having the output 0. Translate each binary value to the related sum term where each value '1' is substituted with the corresponding variable complement and each 0 is with the corresponding variable.

Inputs			Output
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figure 2.3

$$Y(A,B,C,D) = \prod m(1,4,5)$$

Only those sum terms are selected where the output value is 0.

$$Y = m_1 + m_4 + m_5$$

So, in accordance with the truth table, the boolean function (Canonical POS form) in the maxterm is :

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

1 is variable and 0 is the complement of the variable.

Key Differences Between SOP and POS

SOP (Sum of product) generates expression in which all the variables in a domain are first multiplied then added. On the contrary, the POS (Product of Sum) represents the boolean expression having variables summed then multiplied with each other.

Minterms or product terms are mainly used in the SOP which associates with the high (1) value. Conversely, in POS, Maxterms or sum terms are employed, which produces a low (0) value.

In the SOP, method, the value '1' is replaced by the variable and '0' by its complement. In contrast, when it comes to POS a '0' is substituted by the variable and '1' by its complement.

At last, all the terms are added with each other in case of SOP. As against, in POS, the terms are multiplied with each other in the last step of the process.