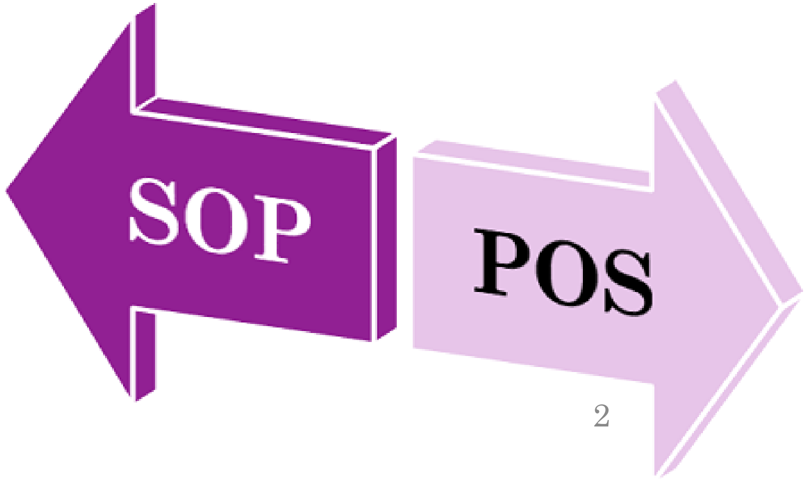
Jiangxi University of Science and Technology

SOP and POS

The difference and example

SOP and POS

- The SOP (Sum of Product) and POS (Product of Sum) are the methods for deducing a particular logic function. In other words, these are the ways to represent the deduced reduced logic function. We can use the deduced logic function in designing a logic circuit.
- The prior difference between the SOP and POS is that the SOP contains the OR of the multiple product terms. Conversely, POS produces a logical expression comprised of the AND of the multiple OR terms. Before understanding SOP and POS, we must learn various related terms so that the entire thing would collectively make some sense.



SOP and POS

- The SOP and POS, both forms are used for representing the expressions and also holds equal importance. In an effort for finding whether your reduced boolean expression is correct or not for designing the logical circuit, one can compare the SOP and POS form of expression and check whether they are equivalent or not. Additionally, for any binary value, the resultant of SOP and POS are either be both 1 or 0 based on the binary value.

Representation of Boolean expression

Boolean expression can be represented by either

(i)Sum of Product(SOP) form or

(ii)Product of Sum (POS form)

e.g.

AB+AC → SOP
(A+B)(A+C) → POS

In above examples both are in SOP and POS respectively but they are not in Standard SOP and POS.

Canonical form of Boolean Expression (Standard form)

- In standard SOP and POS each term of Boolean expression must contain all the literals (with and without bar) that has been used in Boolean expression.
- If the above condition is satisfied by the Boolean expression, that expression is called Canonical form of Boolean expression.

Canonical form of Boolean Expression (Standard form) Contd..

- In Boolean expression **AB+AC** the literal C is mission in the 1st term **AB** and B is mission in 2nd term **AC**. That is why AB+AC is not a Canonical SOP.

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.

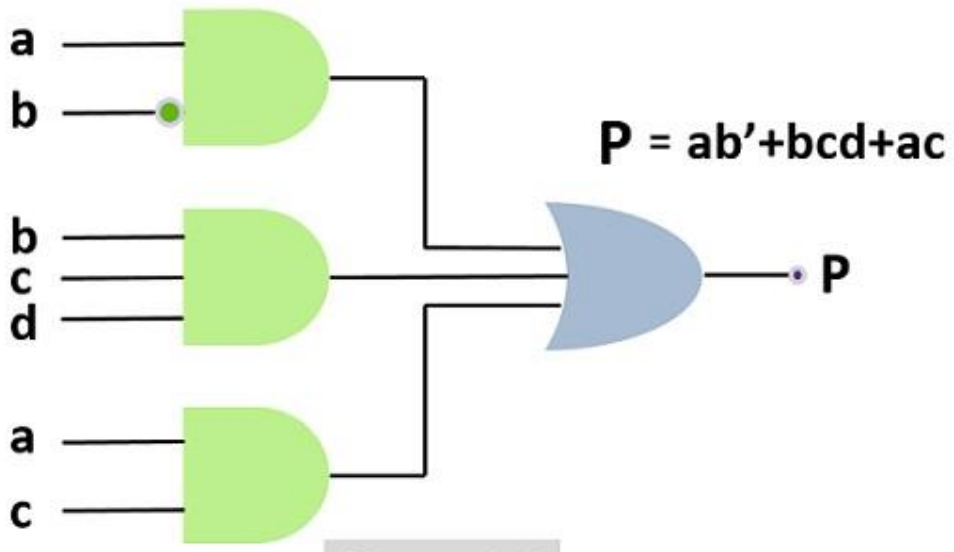


Figure 1.1

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.

Sum of Products (SOP)

- A boolean expression consisting purely of Minterms (product terms) is said to be in canonical sum of products form.
- Example
  lets say, we have a Boolean function F defined on two variables A and B. So, A and B are the inputs for F and lets say, output of F is true i.e., F = 1 when any one of the input is true or 1. Now we draw the truth table for F.

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

- Now we will create a column for the minterm using the variables A and B.
- If input is 0 we take the complement of the variable and if input is 1 we take the variable as is.

A	B	F	Minterm
0	0	0	A'B'
0	1	1	A'B
1	0	1	AB'
1	1	1	AB

To get the desired canonical SOP expression we will add the **minterms** (product terms) for which the **output is 1**.

F = A’B + AB’ + AB

Canonical form of Boolean Expression (Standard form) Contd..

Convert AB+AC in Canonical SOP (Standard SOP)

Sol. AB + AC

AB(C+C’) + AC(B+B’)

ABC+ABC’+ABC+AB’C

ABC+ABC’+AB’C

Distributive law

Canonical form of Boolean Expression (Standard form) Contd..

Convert (A+B)(A+C) in Canonical SOP (Standard SOP)

Sol. $(A+B).(A+C)$

$(A+B)+(C.C') \,.\, (A+C)+(B.B')$

$(A+B+C).(A+B+C').(A+B+C)(A+B'+C)$ **Distributive law**

$(A+B+C).(A+B+C')(A+B'+C)$ **Remove duplicates**

Canonical form of Boolean Expression (Standard form) Contd..

Minterm and Maxterm

Individual term of Canonical Sum of Products (SOP) is called Minterm. In otherwords minterm is a product of all the literals (with or without bar) within the Boolean expression.

Individual term of Canonical Products of Sum (POS) is called Maxterm. In otherwords maxterm is a sum of all the literals (with or without bar) within the Boolean expression.

Converting Sum of Products (SOP) to shorthand notation

- From the previous example we have
  $F = A'B + AB' + AB$
  Now, lets say we want to express the SOP using shorthand notation.
- we have $F = A'B + AB' + AB$

$A'B = (01)_2 = m_1$
$AB' = (10)_2 = m_2$
$AB = (11)_2 = m_3$

We saw the conversion of SOP to shorthand notation

Converting shorthand notation to Sum of Products (SOP)

- Lets say, we have a boolean function F defined on two variables A and B. So, A and B are the inputs for F and lets say, the minterms are expressed as shorthand notation given below.
  $F = \sum(1, 2, 3)$
  our task is to get the SOP.
- F has two input variables A and B and output of F = 1 for m_1, m_2 and m_3 i.e., 2nd, 3rd and 4th combination.
- we have,
  $$\begin{aligned} F &= \sum(1, 2, 3) \\ &= m_1 + m_2 + m_3 \\ &= 01 + 10 + 11 \end{aligned}$$

- To convert from shorthand notation to SOP we follow the given rules. If the variable is 1 then it is taken "as is" and if the variable is 0 then we take its "complement".
- $F = \sum(1, 2, 3)$
  $= A'B + AB' + AB$
- And we have the required SOP

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.

Product of Sums (POS)

- A boolean expression consisting purely of Maxterms (sum terms) is said to be in canonical product of sums form.

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).

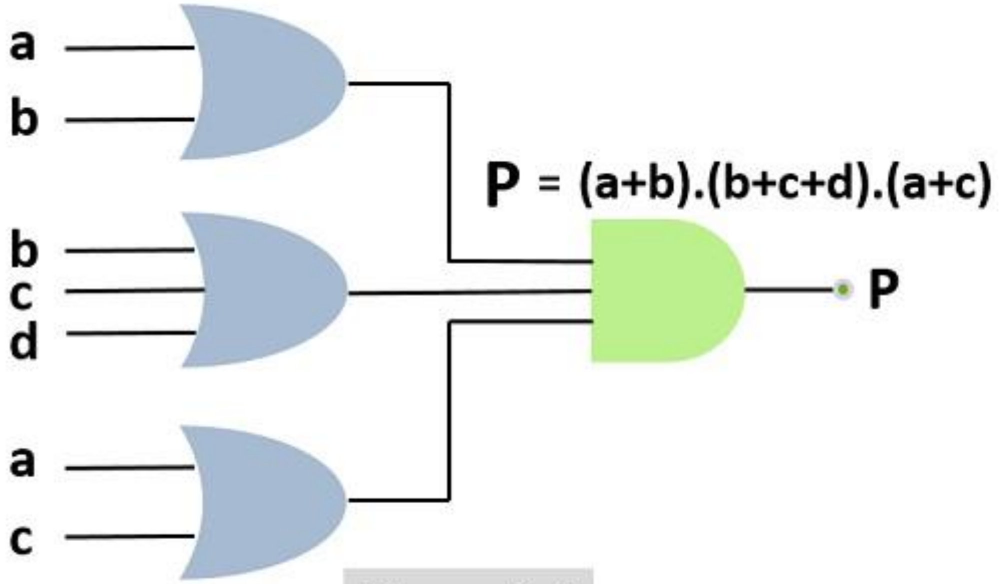


Figure 2.1

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.

$$= (a+b'+c).(b'+c+d').(a+b'+c'+d)$$
$$= (a+b'+c+d.d').(b'+c+d').(a+b'+c'+d)$$
$$= (a+b'+c+d).(a+b'+c+d').(b'+c+d'+a.a').(a+b'+c'+d)$$
$$= (a+b'+c+d).(a+b'+c+d').(b'+c+d'+a).(b'+c+d'+a').(a+b'+c'+d)$$
$$= (a+b'+c+d).(a+b'+c+d').(a'+b'+c+d').(a+b'+c'+d)$$

Product of Sums (POS)

- Example
  Lets say, we have a boolean function F defined on two variables A and B.
- So, A and B are the inputs for F and lets say, output of F is true i.e., F = 1 when only one of the input is true or 1.now we draw the truth table for F

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

Product of Sums (POS)

- Now we will create a column for the maxterm using the variables A and B. If input is 1 we take the complement of the variable and if input is 0 we take the variable as is.

To get the desired canonical POS expression we will multiply the maxterms (sum terms) for which the output is 0.

F = (A+B) . (A'+B')

A	B	F	Maxterm
0	0	0	A + B
0	1	1	A + B'
1	0	1	A' + B
1	1	0	A' + B'

Converting Product of Sums (POS) to shorthand notation

- From the previous example we have
  $F = (A+B) . (A'+B')$
  Now, lets say we want to express the POS using shorthand notation.

we have $F = (A+B) . (A'+B')$

$$A+B = (00)_2 = M_0$$
$$A'+B' = (11)_2 = M_3$$

- Now we express F using shorthand notation.

$F = M_0 . M_3$

This can also be written as $F = \prod(0, 3)$

We saw the conversion of POS to shorthand notation. Lets check the conversion of shorthand notation to POS.

Converting shorthand notation to Product of Sums (POS)

Lets say, we have a boolean function F defined on two variables A and B so, A and B are the inputs for F and lets say, the maxterm are expressed as shorthand notation given below.

$F = \prod(1, 2, 3)$

Our task is to get the POS.

F has two input variables A and B and output of F = 0 for M_1, M_2 and M_3 i.e., 2nd, 3rd and 4th combination.

we have, $F = \prod(1, 2, 3)$
$= M_1 . M_2 . M_3$
$= 01 . 10 . 11$

To convert from shorthand notation to POS we follow the given rules. If the variable is 0 then it is taken as is and if the variable is 1 then we take its complement.

we have, $F = \prod(1, 2, 3)$
$= (A+B') . (A'+B) . (A'+B')$

And we have the required POS.

Minterms & Maxterms for 2 variables
(Derivation of Boolean function from Truth Table)

DIGITAL SYSTEMS DESIGN

x	y	Index	Minterm	Maxterm
0	0	0	$m_0 = x' y'$	$M_0 = x + y$
0	1	1	$m_1 = x' y$	$M_1 = x + y'$
1	0	2	$m_2 = x y'$	$M_2 = x' + y$
1	1	3	$m_3 = x y$	$M_3 = x' + y'$

The minterm m_i should evaluate to 1 for each combination of x and y.
The maxterm is the complement of the minterm

Minterms & Maxterms for 3 variables

x	y	z	Index	Minterm	Maxterm
0	0	0	0	$m_0 = \bar{x}\,\bar{y}\,\bar{z}$	$M_0 = x + y + z$
0	0	1	1	$m_1 = \bar{x}\,\bar{y}\,z$	$M_1 = x + y + \bar{z}$
0	1	0	2	$m_2 = \bar{x}\,y\,\bar{z}$	$M_2 = x + \bar{y} + z$
0	1	1	3	$m_3 = \bar{x}\,y\,z$	$M_3 = x + \bar{y} + \bar{z}$
1	0	0	4	$m_4 = x\,\bar{y}\,\bar{z}$	$M_4 = \bar{x} + y + z$
1	0	1	5	$m_5 = x\,\bar{y}\,z$	$M_5 = \bar{x} + y + \bar{z}$
1	1	0	6	$m_6 = x\,y\,\bar{z}$	$M_6 = \bar{x} + \bar{y} + z$
1	1	1	7	$m_7 = x\,y\,z$	$M_7 = \bar{x} + \bar{y} + \bar{z}$

Maxterm M_i is the complement of minterm m_i

$$M_i = \overline{m_i} \text{ and } m_i = \overline{M_i}$$

Solved Problem

Prob. Find the minterm designation of XY'Z'

Sol. Subsitute 1's for non barred and 0's for barred letters

Binary equivalent = 100

Decimal equivalent = 4

Thus $XY'Z'=m_4$

Purpose of the Index

- **Minterms and Maxterms are designated with an index**
- **The index number corresponds to a binary pattern**
- **The index for the minterm or maxterm, expressed as a binary number, is used to determine whether the variable is shown in the true or complemented form**
- **For Minterms:**
  - **'1' means the variable is "Not Complemented" and**
  - **'0' means the variable is "Complemented".**
- **For Maxterms:**
  - **'0' means the variable is "Not Complemented" and**
  - **'1' means the variable is "Complemented".**

Comparison Chart

Basis for comparison	SOP	POS
Expands to	Sum of Product	Product of Sum
Basic	Form of representation of a boolean expression incorporating minterms	Technique of generating a boolean expression involving maxterms.
Expression includes	Product terms are taken where the input set produces a value 1.	Only Sum terms which generate a value 0.
Method	1 represents the variable and 0 is the complement of it.	0 represents the variable and 1 complement of the variable.
Obtained through	Adding corresponding product terms.	Multiplying the relevant sum terms.
Order of implementation	OR gate is employed after the AND gate.	AND gate is used after the OR gate.

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.

Karnaugh Map (K-map) :

- It is quite similar to a truth table where we have various probable values of the input variables and the outcome for each value. A karnaugh map provides an organized way for simplifying boolean expressions and helps in constructing the simplest minimized SOP and POS expression.

Reference

Digital Design, 5th edition by Morris Mano and Michael Ciletti