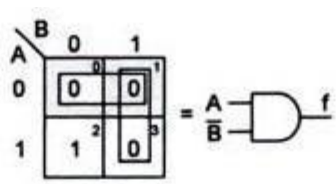
Minimization of POS Expressions:

To obtain the minimal expression in POS form, map the given POS expression on to the K-map and combine the adjacent 0s into as large squares as possible. Read the squares putting the complemented variable if its value remains constant as a 1 and the non-complemented variable if its value remains constant as a 0 along the entire square (ignoring the variables which do not remain constant throughout the square) and then write them as a sum term.

Various maxterm combinations and the corresponding reduced expressions are shown in figure. In this f_1 read as A because A remains constant as a 0 throughout the square and B changes from a 0 to a 1. f_2 is read as B' because B remains constant along the square as a 1 and A changes from a 0 to a 1. f_5
Is read as a 0 because both the variables are changing along the square.

Ex: Reduce the expression f=(A+B)(A+B')(A'+B') using mapping.

The given expression in terms of maxterms is f=πM(0,1,3). It requires two gates inputs for realization of the reduced expression as



F=AB'

K-map in POS form and logic diagram

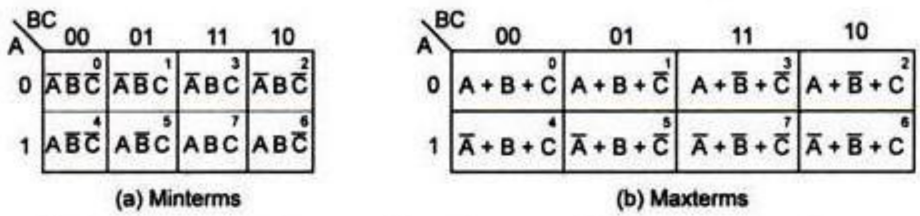
In this given expression ,the maxterm M_2 is absent. This is indicated by a 1 on the k-map. The corresponding SOP expression is $\sum m_2$ or AB'. This realization is the same as that for the POS form.

Three-variable K-map:

A function in three variables (A, B, C) expressed in the standard SOP form can have eight possible combinations: A B C , AB C,A BC ,A BC,AB C ,AB C,ABC , and ABC. Each one of these combinations designate d by m0,m1,m2,m3,m4,m5,m6, and m7, respectively, is called a minterm. A is the MSB of the minterm designator and C is the LSB.

In the standard POS form, the eight possible combinations are:A+B+C, A+B+C , A+B +C,A+B + C ,A + B+ C,A + B + C ,A + B + C,A + B + C . Each oneof these combinations designated by M_0, M1, M2, M3, M4, M5, M6, and M7respectively is called a maxterm. A is the MSB of the maxterm designator and C is the LSB.

A three-variable k-map has, therefore, 8(=2^3) squares or cells, and each square on the map represents a minterm or maxterm as shown in figure. The small number on the top right corner of each cell indicates the minterm or maxterm designation.



The three-variable k-map.

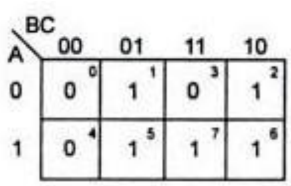
The binary numbers along the top of the map indicate the condition of B and C for each column. The binary number along the left side of the map against each row indicates the condition of A for that row. For example, the binary number 01 on top of the second column in fig indicates that the variable B appears in complemented form and the variable C in non-complemented form in all the minterms in that column. The binary number 0 on the left of the first row indicates that the variable A appears in complemented form in all the minterms in that row, the binary numbers along the top of the k-map are not in normal binary order. They are, infact, in the Gray code. This is to ensure that twophysically adjacent squares are really adjacent, i.e., their minterms or maxterms differ by only one variable.

Ex: Map the expression f=: C+ + + +ABC

In the given expression , the minterms are : C=001=m_1 ; =101=$m_{5;}$ =010=$m_{2;}$

=110=m_6;ABC=111=m_7.

So the expression is f=∑m(1,5,2,6,7)= ∑m(1,2,5,6,7). The corresponding k-map is



K-map in SOP form

Ex: Map the expression f= (A+B+C),(+ +) (+ +)(A + +)(+ +)

In the given expression the maxterms are :A+B+C=000=M_0; + + =101=M_5; + + = 111=M_7; A + + =011=M_3; + + =110=M_6.

So the expression is f = π M (0,5,7,3,6)= π M (0,3,5,6,7). The mapping of the expression is

A \ BC	00	01	11	10
0	0 (0)	1 (1)	0 (3)	1 (2)
1	1 (4)	0 (5)	0 (7)	0 (6)

K-map in POS form.

Minimization of SOP and POS expressions:

For reducing the Boolean expressions in SOP (POS) form plotted on the k-map, look at the 1s (0s) present on the map. These represent the minterms (maxterms). Look for the minterms (maxterms) adjacent to each other, in order to combine them into larger squares. Combining of adjacent squares in a k-map containing 1s (or 0s) for the purpose of simplification of a SOP (or POS)expression is called *looping*. Some of the minterms (maxterms) may have many adjacencies. Always start with the minterms (maxterm) with the least number of adjacencies and try to form as large as large a square as possible. The larger must form a geometric square or rectangle. They can be formed even by wrapping around, but cannot be formed by using diagonal configurations. Next consider the minterm (maxterm) with next to the least number of adjacencies and form as large a square as possible. Continue this till all the minterms (maxterms) are taken care of . A minterm (maxterm) can be part of any number of squares if it is helpful in reduction. Read the minimal expression from the k-map, corresponding to the squares formed. There can be more than one minimal expression.

Two squares are said to be adjacent to each other (since the binary designations along the top of the map and those along the left side of the map are in Gray code), if they are physically adjacent to each other, or can be made adjacent to each other by wrapping around. For squares to be combinable into bigger squares it is essential but not sufficient that their minterm designations must differ by a power of two.

General procedure to simplify the Boolean expressions:

1. Plot the k-map and place 1s(0s) corresponding to the minterms (maxterms) of the SOP (POS) expression.
2. Check the k-map for 1s(0s) which are not adjacent to any other 1(0). They are isolated minterms(maxterms) . They are to be read as they are because they cannot be combined even into a 2-square.
3. Check for those 1s(0S) which are adjacent to only one other 1(0) and make them pairs (2 squares).
4. Check for quads (4 squares) and octets (8 squares) of adjacent 1s (0s) even if they contain some 1s(0s) which have already been combined. They must geometrically form a square or a rectangle.
5. Check for any 1s(0s) that have not been combined yet and combine them into bigger squares if possible.
6. Form the minimal expression by summing (multiplying) the product the product (sum) terms of all the groups.

Reading the K-maps:

While reading the reduced k-map in SOP (POS) form, the variable which remains constant as 0 along the square is written as the complemented (non-complemented) variable and the one which remains constant as 1 along the square is written as non-complemented (complemented) variable and the term as a product (sum) term. All the product (sum) terms are added (multiplied).

Some possible combinations of minterms and the corresponding minimal expressions readfrom the k-maps are shown in fig: Here f_6 is read as 1, because along the 8-square no variable remains constant. F_5 is read as , because, along the 4-square formed by0,m_1,m_2 and m_3 , the variables B and C are changing, and A remains constant as a 0. Algebraically,

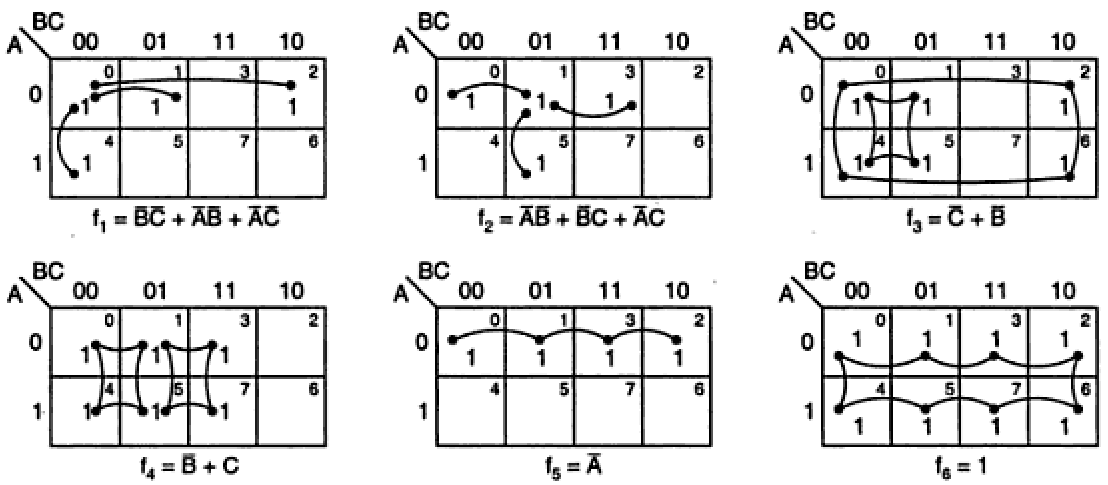
$$f_5= m_0+m_1+m_2+m_3$$

= + C+ +

= (+C)+ B(C+)

= + B

= (+B)=

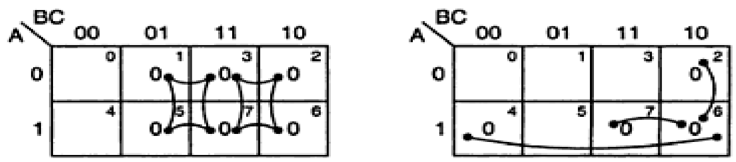


f_3 is read as + , because in the 4-square formed by m0,m2,m6, and m4, the variable A and B are changing , where as the variable C remains constant as a 0. So it is read as . In the 4-square formed by m_0, m1, m4, m_5, A and C are changing but B remains constant as a 0. So it is read as . So, the resultant expression for f_3 is the sum of these two, i.e., + .

f_1 is read as + + ,because in the 2-square formed by m_0 and m_4 , A is changing from a 0 to a 1. Whereas B and C remain constant as a 0. So it s read as . In the 2-square formed by m_0 and m_1, C is changing from a 0 to a 1, whereas A and B remain constant as a 0. So it is read as .In the 2-square formed by m_0 and m_2 , B is changing from a 0 to a 1 whereas A and C remain constant as a 0. So, it is read as . Therefore, the resultant SOP expression is

+ +

Some possible maxterm groupings and the corresponding minimal POS expressions read from the k-map are



(a) $f_1 = (\bar{C})(\bar{B})$ (b) $f_2 = (\bar{A} + \bar{B})(\bar{B} + C)(\bar{A} + C)$

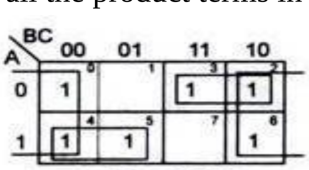
In this figure, along the 4-square formed by M_1, M3, M7, M5, A and B are changing from a 0 to a 1, where as C remains constant as a 1. SO it is read as . Along the 4-squad formed by M_3, M2, M7, and M_6, variables A and C are changing from a 0 to a 1. But B remains constant as a 1. So it is read as . The minimal expression is the product of these two terms , i.e., f_1 = ( )( ).also in this figure, along the 2-square formed by M_4 and M6 , variable B is changing from a 0 to a 1, while variable A remains constant as a 1 and variable C remains constant as a 0. SO, read it as

+C. Similarly, the 2-square formed by M_7 andM_6 is read as + , while the 2-square formed by M_2 and M_6 is read as +C. The minimal expression is the product of these sum terms, i.e, f_2 =(+)+**(** +)+(+C)

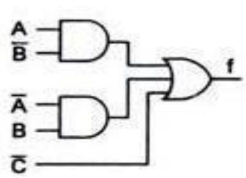
Ex:Reduce the expression f=∑m(0,2,3,4,5,6) using mapping and implement it in AOI logic as well as in NAND logic.The Sop k-map and its reduction , and the implementation of the minimal expression using AOI logic and the corresponding NAND logic are shown in figures below

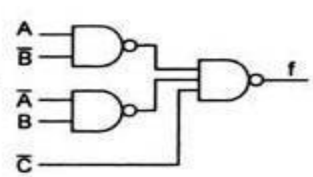
In SOP k-map, the reduction is done as:

1. m_5 has only one adjacency m_4 , so combine m_5 and m_4 into a square. Along this 2-square A remains constant as 1 and B remains constant as 0 but C varies from 0 to 1. So read it as A .
2. m_3 has only one adjacency m_2 , so combine m_3 and m_2 into a square. Along this 2-square A remains constant as 0 and B remains constant as 1 but C varies from 1 to 0. So read it as B.
3. m_6 can form a 2-square with m_2 and m_4 can form a 2-square with m_0, but observe that by wrapping the map from left to right m_0, m_4 ,m_2 ,m_6 can form a 4-square. Out of these m_2 andm4 have already been combined but they can be utilized again. So make it. Along this 4-square, A is changing from 0 to 1 and B is also changing from 0 to 1 but C is remaining constant as 0. so read it as .
4. Write all the product terms in SOP form. So the minimal SOP expression is

f_{min}= 

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





k-map **AOI logic** **NAND logic**