Minimization Techniques

Two-variable k-map:

A two-variable k-map can have 2^2=4 possible combinations of the input variables A and B.E ac h of these combinations, , B,A ,AB(in the SOP form) is called a minterm. The minterm may be represented in terms of their decimal designations – m0 for , m1 for B,m2 for A and m3 for AB, assuming that A represents the MSB. The letter m stands for minterm and the subscript represents the decimal designation of the minterm. The presence or absence of a minterm in the expression indicates that the output of the logic circuit assumes logic 1 or logic 0 level for that combination of input variables.

The expression f= ,+ B+A +AB , it can be expressed using min

term as F= m0+m2+m3=∑m(0,2,3)

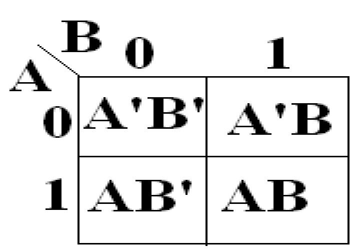
Using Truth Table:

Minterm	Inputs A	B	Output F
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	1

A 1 in the output contains that particular minterm in its sum and a 0 in that column indicates that the particular mintermdoes not appear in the expression for output . this information can also be indicated by a two-variable k-map.

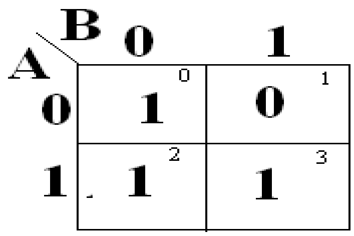
Mapping of SOP Expresions:

A two-variable k-map has 22=4 squares .These squares are called cells. Each square on the k-map represents a unique minterm. The minterm designation of the squares are placed in any square, indicates that the corresponding minterm does output expressions. And a 0 or no entry in any square indicates that the corresponding minterm does not appear in the expression for output.



The minterms of a two-variable k-map

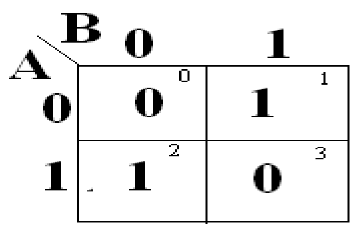
The mapping of the expressions =∑m(0,2,3)is



k-map of ∑m(0,2,3)

EX: Map the expressions f= B+A

F= m_1+m_2=∑m(1,2)The k-map is



Minimizations of SOP expressions:

To minimize Boolean expressions given in the SOP form by using the k-map, look for adjacent adjacent squares having 1's minterms adjacent to each other, and combine them to form larger squares to eliminate some variables. Two squares are said to be adjacent to each other, if their minterms differ in only one variable. (i.e, B & A differ only in one variable. so theymay be combined to form a 2-square to eliminate the variable B.similarly all other.

The necessary condition for adjacency of minterms is that their decimal designations must differ by a power of 2. A minterm can be combined with any number of minterms adjacent to it to form larger squares. Two minterms which are adjacent to each other can be combined to form a bigger square called a 2-square or a pair. This eliminates one variable – the variable that is not common to both the minterms. For EX:

m0 and m1 can be combined to yield,

f_1 = m0+m1= + B= (B+

)= m0 and m2 can be combined to yield,

f_2 = m0+m2= + = (+)=

m1 and m3 can be combined to yield,

f_3= m1+m3= B+AB=B(+)=B

m2 and m3 can be combined to yield,

f_4 = m2+m3=A +AB=A(B+)=A

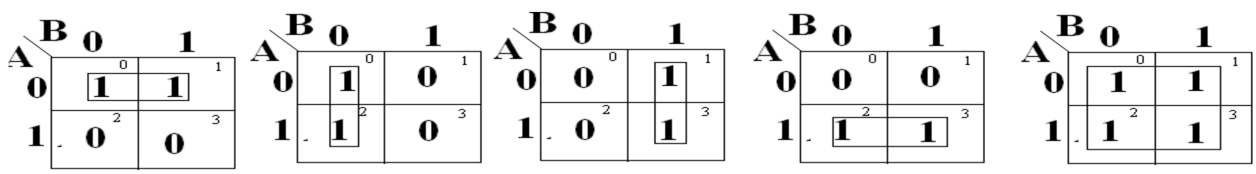
m_0 ,m_1 ,m_2 and m_3 can be combined to yield,

= + +A +AB

= (B+) +A(B+)

= +A

=1



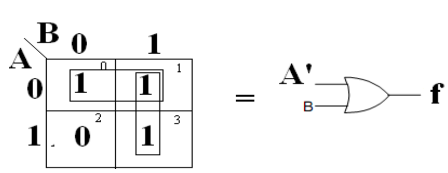
f1= f2= f3=B f4=A f5=1

The possible minterm groupings in a two-variable k-map.

Two 2-squares adjacent to each other can be combined to form a 4-square. A 4-square eliminates 2 variables. A 4-square is called a quad. To read the squares on the map after minimization, consider only those variables which remain constant through the square, and ignore the variables which are varying. Write the non complemented variable if the variable is remaining constant as a 1, and the complemented variable if the variable is remaining constant as a 0, and write the variables as a product term. In the above figure f_1 read as , because, along the square , A remains constant as a 0, that is , as , where as B is changing from 0 to 1.

EX: Reduce the minterm f= +A +AB using mapping Expressed in terms of minterms, the given expression is F=m_0+m_1+m_2+ m_3=m∑(0,1,3)& the figure shows the k-map for f and its reduction . In one 2-square, A is constant as a 0 but B varies from a 0 to a 1, and in the other 2- square, B is constant as a 1 but A varies from a 0 to a 1. So, the reduced expressions is +B.

It requires two gate inputs for realization as



f= +B (k-map in SOP form, and logic diagram.)

The main criterion in the design of a digital circuit is that its cost should be as low as possible. For that the expression used to realize that circuit must be minimal.Since the cost is proportional to number of gate inputs in the circuit in the circuit, an expression is considered minimal only if it corresponds to the least possible number of gate inputs. & there is no guarantee for that k-map in SOP is the real minimal. To obtain real minimal expression, obtain the minimal expression both in SOP & POS form form by using k-maps and take the minimal of these two minimals.

The 1's on the k-map indicate the presence of minterms in the output expressions, where as the 0s indicate the absence of minterms .Since the absence of a minterm in the SOP expression means the presense of the corresponding maxterm in the POS expression of the same .when a SOP expression is plotted on the k-map, 0s or no entries on the k-map represent the maxterms. To obtain the minimal expression in the POS form, consider the 0s on the k-map and follow the procedure used for combining 1s. Also, since the absence of a maxterm in the POS expression means the presence of the corresponding minterm in the SOP expression of the same , when a POS expression is plotted on the k-map, 1s or no entries on the k-map represent theminterms.

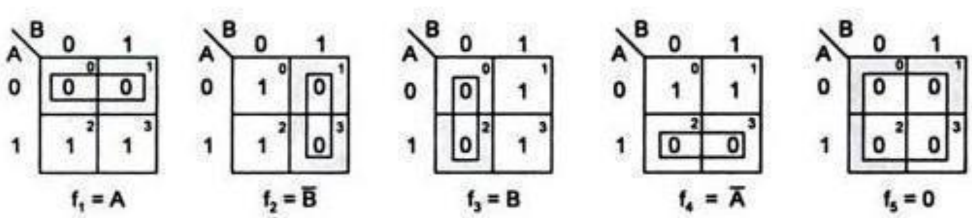
Mapping of POS expressions:

Each sum term in the standard POS expression is called a maxterm. A function in two variables (A, B) has four possible maxterms, A+B,A+ , +B, +

. They are represented as M_0, M_1, M2, and M3respectively. The uppercase letter M stands for maxterm and its subscript denotes the decimal designation of that maxterm obtained by treating the non-complemented variable as a 0 and the complemented variable as a 1 and putting them side by side for reading the decimal equivalent of the binary number so formed.

For mapping a POS expression on to the k-map, 0s are placed in the squares corresponding to the maxterms which are presented in the expression an d1s are placed in the squares corresponding to the maxterm which are not present in the expression. The decimal designation of the squares of the squares for maxterms is the same as that for the minterms. A two-variable k-map & the associated maxterms are asthe maxterms of a two-variable k-map

The possible maxterm groupings in a two-variable k-map



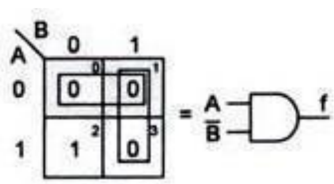
Minimization of POS Expressions:

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

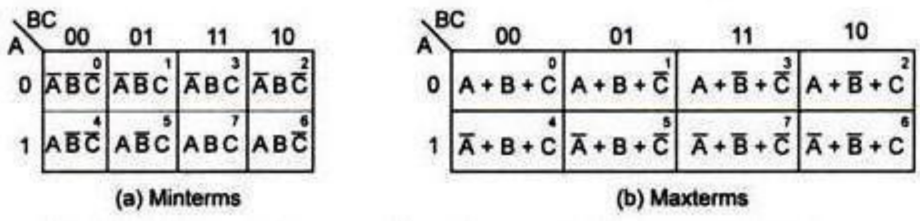
In this given expression ,the maxterm M_2 is absent. This is indicated by a 1 on the k-map. The corresponding SOP expression is $\sum m_2$ or AB‘. This realization is the same as that for the POS form.

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.

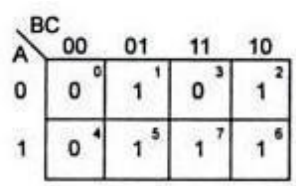
The binary numbers along the top of the map indicate the condition of B and C for each column. The binary number along the left side of the map against each row indicates the condition of A for that row. For example, the binary number 01 on top of the second column in fig indicates that the variable 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

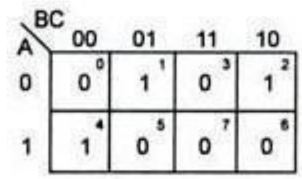


K-map in SOP form

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



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 by$0,m_1,m_2$ and m_3, the variables B and C are changing, and A remains constant as a 0. Algebraically,

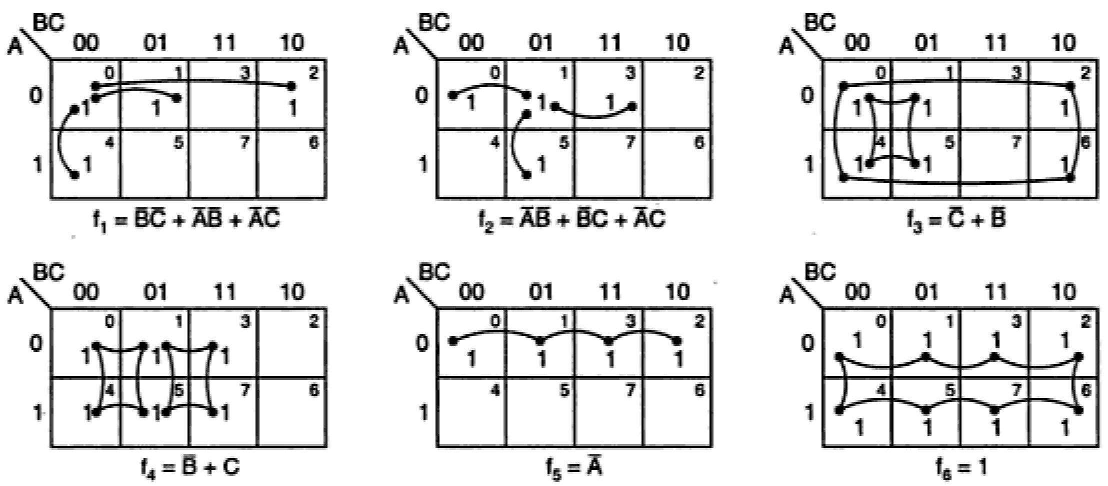
$$f_5= m_0+m_1+m_2+m_3$$

$$= \quad + \quad C+ \quad +$$

$$= \quad (\quad +C)+ \quad B(C+)$$

$$= \quad + \quad B$$

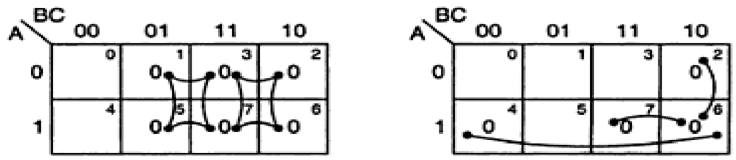
$$= \quad (\quad +B)=$$



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 a0 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) $f_1 = (\overline{C})(\overline{B})$ (b) $f_2 = (\overline{A} + \overline{B})(\overline{B} + C)(\overline{A} + C)$

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

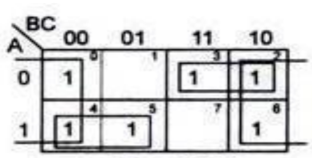
In SOP k-map, the reduction is done as:

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 .

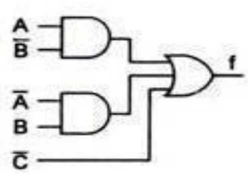
2 m_3 has only one adjacency m_2 , so combine m_3 and m_2 into a square. Along this 2-square A remains constant as 0 and B remains constant as 1 but C varies from 1 to 0. So read it as B.

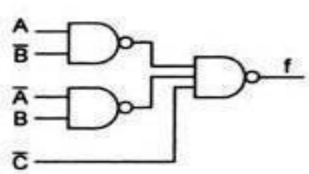
3 m_6 can form a 2-square with m_2 and m_4 can form a 2-square with $m_{0,}$ but observe that by wrapping the map from left to right m_0, m_4 ,m_2 ,m_6 can form a 4-square. Out of these m_2 and m4 have already been combined but they can be utilized again. So make it. Along this 4-square, A is changing 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



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





f_{min}=

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

Four variable k-maps:

Four variable k-map expressions can have 2^4=16 possible combinations of input variables such as , ,------------ABCD with minterm designations m_0,m_1---------------------m_{15} respectively in SOP form & A+B+C+D, A+B+C+ ,---------- + + + with maxterms M_0,M_1,----------

-M_{15} respectively in POS form. It has 2^4=16 squares or cells.The binary number designations of rows & columns are in the gray code. Here follows 01 & 10 follows 11 called Adjacency ordering.

AB \ CD	00	01	11	10
00	0: $\bar{A}\bar{B}\bar{C}\bar{D}$	1: $\bar{A}\bar{B}\bar{C}D$	3: $\bar{A}\bar{B}CD$	2: $\bar{A}\bar{B}C\bar{D}$
01	4: $\bar{A}B\bar{C}\bar{D}$	5: $\bar{A}B\bar{C}D$	7: $\bar{A}BCD$	6: $\bar{A}BC\bar{D}$
11	12: $AB\bar{C}\bar{D}$	13: $AB\bar{C}D$	15: $ABCD$	14: $ABC\bar{D}$
10	8: $A\bar{B}\bar{C}\bar{D}$	9: $A\bar{B}\bar{C}D$	11: $A\bar{B}CD$	10: $A\bar{B}C\bar{D}$

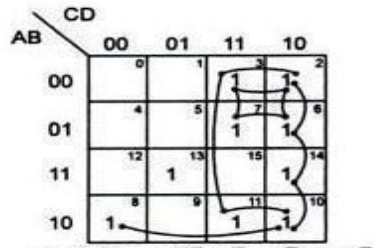
SOP form

AB \ CD	00	01	11	10
00	0: $A+B+C+D$	1: $A+B+C+\bar{D}$	3: $A+B+\bar{C}+\bar{D}$	2: $A+B+\bar{C}+D$
01	4: $A+\bar{B}+C+D$	5: $A+\bar{B}+C+\bar{D}$	7: $A+\bar{B}+\bar{C}+\bar{D}$	6: $A+\bar{B}+\bar{C}+D$
11	12: $\bar{A}+\bar{B}+C+D$	13: $\bar{A}+\bar{B}+C+\bar{D}$	15: $\bar{A}+\bar{B}+\bar{C}+\bar{D}$	14: $\bar{A}+\bar{B}+\bar{C}+D$
10	8: $\bar{A}+B+C+D$	9: $\bar{A}+B+C+\bar{D}$	11: $\bar{A}+B+\bar{C}+\bar{D}$	10: $\bar{A}+B+\bar{C}+D$

POS form

EX: **Reduce using mapping the expression Σ m(2, 3, 6, 7, 8, 10, 11, 13, 14).**

Start with the minterm with the least number of adjacencies. The minterm m_{13} has no adjacency. Keep it as it is. The m_8 has only one adjacency, m_{10}. Expand m_8 into a 2-square with m_{10}. The m_7 has two adjacencies, m_6 and m_3. Hence m_7 can be expanded into a 4-square with m_6, m_3 and m_2. Observe that, m_7, m_6, m_2, and m_3 form a geometric square. The m_{11} has 2 adjacencies, m_{10} and m_3. Observe that, m_{11}, m_{10}, m_3, and m_2 form a geometric square on wrapping the K-map. So expand m_{11} into a 4-square with m_{10}, m_3 and m_2. Note that, m_2 and m_3, have already become a part of the 4-square m_7, m_6, m_2, and m_3. But if m_{11} is expanded only into a 2-square with m_{10}, only one variable is eliminated. So m_2 and m_3 are used again to make another 4-square with m_{11} and m_{10} to eliminate two variables. Now only m_6 and m_{14} are left uncovered. They can form a 2-square that eliminates only one variable. Don't do that. See whether they can be expanded into a larger square. Observe that, m_2, m_6, m_{14}, and m_{10} form a rectangle. So m_6 and m_{14} can be expanded into a 4-square with m_2 and m_{10}. This eliminates two variables.



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

Five variable k-map:

Five variable k-map can have 2^5 =32 possible combinations of input variable as , E,--------ABCDE with minterms m_0, m_1-----m_{31} respectively in SOP & A+B+C+D+E, A+B+C+ ,---------- + + + + with maxterms M_0,M_1, -----------
M_{31} respectively in POS form. It has 2^5=32 squares or cells of the k-map are divided into 2 blocks of
16 squares each.The left block represents minterms from m_0 to m_{15} in which A is a 0, and the right block represents minterms from m_{16} to m_{31} in which A is 1.The 5-variable k-map may contain 2-squares, 4-squares , 8-squares , 16-squares or 32-squares involving these two blocks. Squares are also considered adjacent in these two blocks, if when superimposing one block on top of another, the squares coincide with one another.

Some possible 2-squares in a five-variable map are m_0, m_{16}; m_2, m_{18}; m_5, m_{21}; m_{15}, m_{31}; m_{11}, m_{27}.

Some possible 4-squares are m_0, m_2, m_{16}, m_{18}; m_0, m_1, m_{16}, m_{17}; m_0, m_4, m_{16}, m_{20}; m_{13}, m_{15}, m_{29}, m_{31}; m_5, m_{13}, m_{21}, m_{29}.

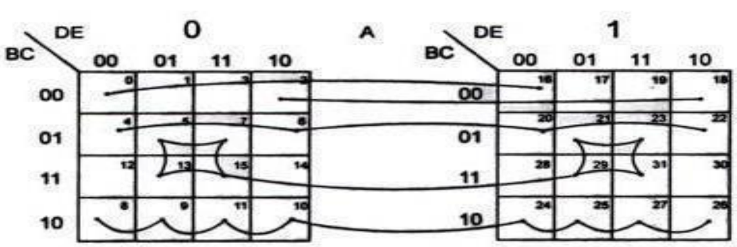
Some possible 8-squares are m_0, m_1, m_3, m_2, m_{16}, m_{17}, m_{19}, m_{18}; m_0, m_4, m_{12}, m_8, m_{16}, m_{20}, m_{28}, m_{24}; m_5, m_7, m_{13}, m_{15}, m_{21}, m_{23}, m_{29}, m_{31}.

The squares are read by dropping out the variables which change. Some possible

Grouping s is

(a) $m_0, m_{16} = \bar{B}\bar{C}\bar{D}\bar{E}$
(b) $m_2, m_{18} = \bar{B}\bar{C}D\bar{E}$
(c) $m_4, m_6, m_{20}, m_{22} = \bar{B}C\bar{E}$
(d) $m_5, m_7, m_{13}, m_{15}, m_{21}, m_{23}, m_{29}, m_{31} = CE$
(e) $m_8, m_9, m_{10}, m_{11}, m_{24}, m_{25}, m_{26}, m_{27} = B\bar{C}$

$M_0, M_{16} = B + C + D + E$
$M_2, M_{18} = B + C + \bar{D} + E$
$M_4, M_6, M_{20}, M_{22} = B + \bar{C} + E$
$M_5, M_7, M_{13}, M_{15}, M_{21}, M_{23}, M_{29}, M_{31} = \bar{C} + \bar{E}$
$M_8, M_9, M_{10}, M_{11}, M_{24}, M_{25}, M_{26}, M_{27} = \bar{B} + C$



Ex: F=∑m(0,1,4,5,6,13,14,15,22,24,25,28,29,30,31) is SOP

POS is F=πM(2,3,7,8,9,10,11,12,16,17,18,19,20,21,23,26,27)

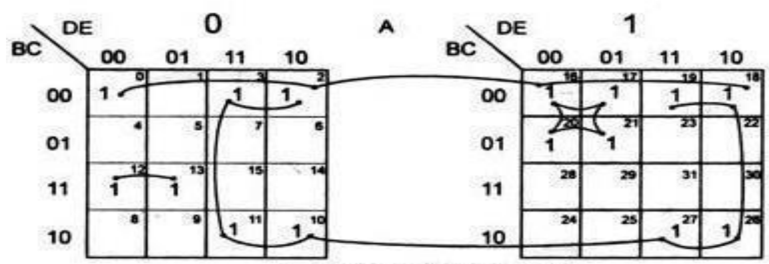
The real minimal expression is the minimal of the SOP and POS forms.

The reduction is done as

1. There is no isolated 1s
2. M_{12} can go only with m_{13}. Form a 2-square which is read asA‘BCD‘
3. M_0 can go with m_2,m_{16} and m_{18} . so form a 4-square which is read as B‘C‘E‘
4. M_{20},m_{21},m_{17} and m_{16} form a 4-square which is read as AB‘D‘
5. M2,m3,m18,m19,m10,m11,m26 and m27 form an 8-square which is read as C‘d
6. Write all the product terms in SOP form.

So the minimal expression is

F_{min}= A‘BCD‘+B‘C‘E‘+AB‘D‘+C‘D(16 inputs)



$f = \bar{A}BC\bar{D} + \bar{B}\bar{C}\bar{E} + A\bar{B}\bar{D} + \bar{C}D$

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

1. There are no isolated 0s

M_1 can go only with M_5. So, make a 2-square, which is read as $(A + B + D + \bar{E})$.

3. M_4 can go with M_5, M_7, and M_6 to form a 4-square, which is read as $(A + B + \bar{C})$.

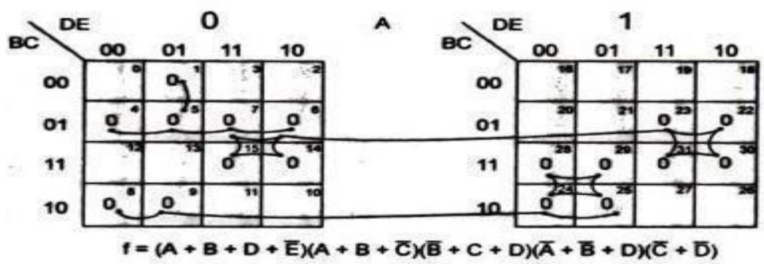
4. M_8

5. M_{28}

6. M_{30}

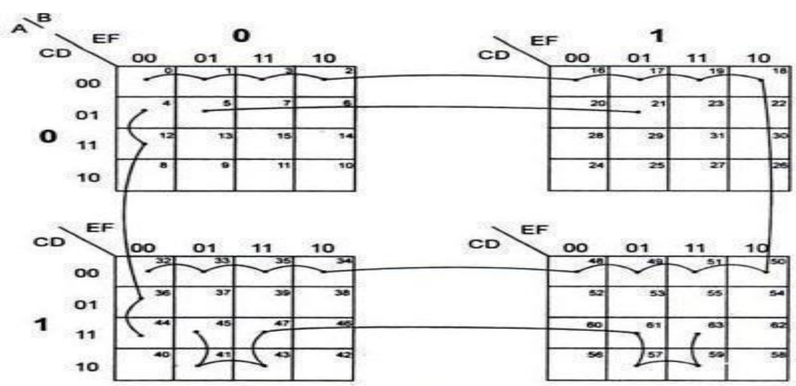
7. Sum terms in POS form. So the minimal expression in POS is

F_{min}= A‘BcD‘+B‘C‘E‘+AB‘D‘+C‘D



Six variable k-map:

Six variable k-map can have 2^6 =64 combinations as , ,------------ABCDEF with minterms m_0, m_1-----m_{63} respectively in SOP & (A+B+C+D+E+F),----------- (+ + + + +) with maxterms M_0,M_1, -----------M_{63} respectively in POS form. It has 2^6=64 squares or cells of the k-map are divided into 4 blocks of 16 squares each.



Some possible groupings in a six variable k-map

Don't care combinations: For certain input combinations, the value of the output is unspecified either because the input combinations are invalid or because the precise value of the output is of no consequence. The combinations for which the value of experiments are not specified are called don‘t care combinations are invalid or because the precise value of the output is of no consequence. The combinations for which the value of expressions is not specified are called don‘t care combinations or Optional Combinations, such expressions stand incompletely specified. The output is a don‘t care for these invalidcombinations.

Ex:In XS-3 code system, the binary states 0000, 0001, 0010,1101,1110,1111 are unspecified. & never occur called don‘t cares.

A standard SOP expression with don‘t cares can be converted into a standard POS form by keeping the don‘t cares as they are & writing the missing minterms of the SOP form as the maxterms of the POS form viceversa.

Don‘t cares denoted by‗X‘ or ‗φ‘

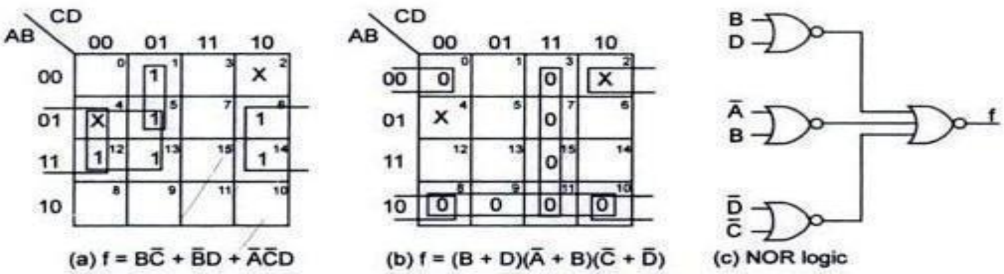
Ex:f=∑m(1,5,6,12,13,14)+d(2,4)

Or f=πM(0,3,7,9,10,11,15).πd(2,4)

SOP minimal form f_{min}= +B +

POS minimal form f_{min}=(B+D)(+B)(+D)

= + + + + (+



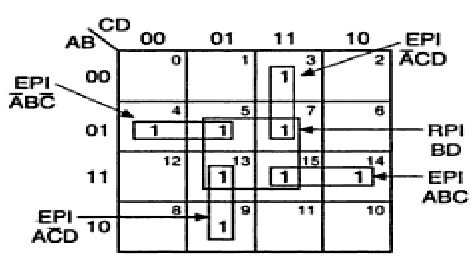
Prime implicants, Essential Prime implicants, Redundant prime implicants:

Each square or rectangle made up of the bunch of adjacent minterms is called a subcube. Each of these subcubes is called a Prime implicant (PI). The PI which contains at leastone which cannot be covered by any other prime implicants is called as Essential Prime implicant (EPI).The PI whose each 1 is covered at least by one EPI is called a Redundant Prime implicant (RPI). A PI which is neither an EPI nor a RPI is called a Selective Prime implicant (SPI).

The function has unique MSP comprising EPI is

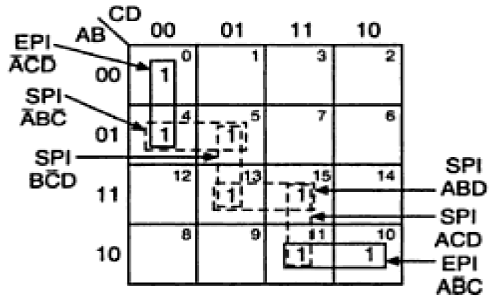
F(A,B,C,D)= CD+ABC+A D + B

The RPI _BD‘ may be included without changing the function but the resulting expression would not be in minimal SOP(MSP) form.



Essential and Redundant Prime Implicants

F(A,B,C,D)=∑m(0,4,5,10,11,13,15) SPI are marked by dotted squares, shows MSP form of a function need not be unique.



Essential and Selective Prime Implicants

Here, the MSP form is obtained by including two EPI's & selecting a set of SPI's to cover remaining uncovered minterms 5,13,15. & these can be covered as

(A) (4,5) &(13,15)---------- B +ABD
(B) (5,13) & (13,15)-------- B D+ABD
(C) (5,13) & (15,11)------- B D+ACD

F(A,B,C,D)= +A C-------- EPI's + B +ABD

(OR) F(A,B,C,D)= +A C-------- EPI's + B D+ABD

(OR) F(A,B,C,D)= +A C -------- EPI's + B D+ACD

False PI's Essential False PI's, Redundant False PI's & Selective False PI's:

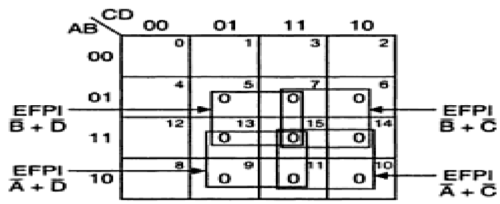
The maxterms are called falseminterms. The PI's is obtained by using the maxterms are called False PI's (FPI). The FPI which contains at least one _0' which can't be covered by only other FPI is called an Essential False Prime implicant (ESPI)

F(A,B,C,D)= ∑m(0,1,2,3,4,8,12)

=π M(5,6,7,9,10,11,13,14,15)

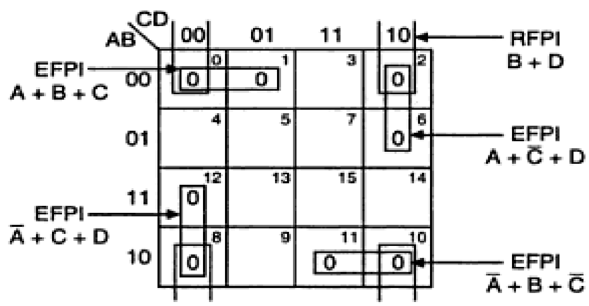
F_{min}= (+)(+)(+)(+)

All the FPI, EFPI's as each of them contain atleast one _0' which can't be covered by any other FPI



Essential False Prime implicants

Consider Function F(A,B,C,D)= π M(0,1,2,6,8,10,11,12)



Essential and Redundant False Prime Implicants

Mapping when the function is not expressed in minterms (maxterms):

An expression in k-map must be available as a sum (product) of minterms (maxterms). However if not so expressed, it is not necessary to expand the expression algebraically into its minterms (maxterms). Instead, expansion into minterms (maxterms) can be accomplished in the process of entering the terms of the expression on the k-map.

Limitations of Karnaugh maps:

- Convenient as long as the number of variables does not exceed six.
- Manual technique, simplification process is heavily dependent on the humanabilities.

Quine-Mccluskey Method:

It also known as *Tabular method.* It is more systematic method of minimizing expressions of even larger number of variables. It is suitable for hand computation as well as computation by machines i.e., programmable. . The procedure is based on repeated application of the combining theorem.

PA+P =P (P is set of literals) on all adjacent pairs of terms, yields the set of all PI‘s from which a minimal sum may be selected.

Consider expression

∑m(0,1,4,5)= + C+A +A C

First, second terms & third, fourth terms can be combined

$$(+) + (C +) = + A$$

Reduced to

$$(+) =$$

The same result can be obtained by combining m_0& m_4 & m_1&m_5 in first step & resulting terms in the second step .

Procedure:

- Decimal Representation
- Don‘t cares
- PI chart
- EPI
- Dominating Rows & Columns
- Determination of Minimal expressions in complexcases.

Branching Method:

EXAMPLE 3.29 Obtain the set of prime implicants for the Boolean expression $f = \Sigma\, m(0, 1, 6, 7, 8, 9, 13, 14, 15)$ using the tabular method.

Solution

Group the minterms in terms of the number of 1s present in them and write their binary designations. The procedure to obtain the prime implicants is shown in Table 3.3.

Table 3.3 Example 3.29

	Column 1		Column 2		Column 3
	Minterm	**Binary designation**		**A B C D**	**A B C D**
Index 0	0	0 0 0 0 ✓	0, 1 (1)	0 0 0 – ✓	0, 1, 8, 9 (1, 8) – 0 0 – Q
Index 1	1	0 0 0 1 ✓	0, 8 (8)	– 0 0 0 ✓	... ...
	8	1 0 0 0 ✓	1, 9 (8)	– 0 0 1 ✓	... ...
Index 2	6	0 1 1 0 ✓	8, 9 (1)	1 0 0 – ✓	6, 7, 14, 15 (1, 8) – 1 1 – P
	9	1 0 0 1 ✓	6, 7 (1)	0 1 1 – ✓	
Index 3	7	0 1 1 1 ✓	6, 14 (8)	– 1 1 0 ✓	
	13	1 1 0 1 ✓	9, 13 (4)	1 – 0 1 S	
	14	1 1 1 0 ✓	7, 15 (8)	– 1 1 1 ✓	
Index 4	15	1 1 1 1 ✓	13, 15 (2)	1 1 – 1 R	
			14, 15 (1)	1 1 1 – ✓	

Comparing the terms of index 0 with the terms of index 1 of column 1, $m_0(0000)$ is combined with $m_1(0001)$ to yield 0, 1 (1), i.e. 000 –. This is recorded in column 2 and 0000 and 0001 are checked off in column 1. $m_0(0000)$ is combined with $m_8(1000)$ to yield 0, 8 (8), i.e. – 000. This is recorded in column 2 and 1000 is checked off in column 1. Note that 0000 of column 1 has already been checked off. No more combinations of terms of index 0 and index 1 are possible. So, draw a line below the last combination of these groups, i.e. below 0, 8 (8), – 000 in column 2. Now 0, 1 (1), i.e. 000 – and 0, 8 (8), i.e. – 000 are the terms in the first group of column 2.

Comparing the terms of index 1 with the terms of index 2 in column 1, $m_1(0001)$ is combined with $m_9(1001)$ to yield 1, 9 (8), i.e. – 001. This is recorded in column 2 and 1001 is checked off in column 1 because 0001 has already been checked off. $m_8(1000)$ is combined with $m_9(1001)$ to yield 8, 9 (1), i.e. 100 –. This is recorded in column 2. 1000 and 1001 of column 1 have already been checked off. So, no need to check them off again. No more combinations of terms of index 1 and index 2 are possible. So, draw a line below the last combination of these groups, i.e. 8, 9 (1),

-- 001 in column 2. Now 1, 9 (8), i.e. – 001 and 8, 9 (1), i.e. 100– are the terms in the second group of column 2.

Similarly, comparing the terms of index 2 with the terms of index 3 in column 1,

$m_6(0110)$ and $m_7(0111)$ yield 6, 7 (1), i.e. 011–. Record it in column 2 and check off 6(0110) and 7(0111).

$m_6(0110)$ and $m_{14}(1110)$ yield 6, 14 (8), i.e. –110. Record it in column 2 and check off 6(0110) and 14(1110).

$m_9(1001)$ and $m_{13}(1101)$ yield 9, 13 (4), i.e. 1–01. Record it in column 2 and check off 9(1001) and 13(1101).

So, 6, 7 (1), i.e. 011–, and 6, 14 (8), i.e. –110 and 9, 13 (4), i.e. 1–01 are the terms in group 3 of column 2. Draw a line at the end of 9, 13 (4), i.e. 1–01.

Also, comparing the terms of index 3 with the terms of index 4 in column 1,

$m_7(0111)$ and $m_{15}(1111)$ yield 7, 15 (8), i.e. –111. Record it in column 2 and check off 7(0111) and 15(1111).

$m_{13}(1101)$ and $m_{15}(1111)$ yield 13, 15 (2), i.e. 11–1. Record it in column 2 and check off 13 and 15.

$m_{14}(1110)$ and $m_{15}(1111)$ yield 14, 15 (1), i.e. 111–. Record it in column 2 and check off 14 and 15.

So, 7, 15 (8), i.e. –111, and 13, 15 (2), i.e. 11–1 and 14, 15 (1), i.e. 111– are the terms in group 4 of column 2. Column 2 is completed now.

Comparing the terms of group 1 with the terms of group 2 in column 2, the terms 0, 1 (1), i.e. 000– and 8, 9 (1), i.e. 100– are combined to form 0, 1, 8, 9 (1, 8), i.e. –00–. Record it in group 1 of column 3 and check off 0, 1 (1), i.e. 000–, and 8, 9 (1), i.e. 100– of column 2. The terms 0, 8 (8), i.e. –000 and 1, 9 (8), i.e. –001 are combined to form 0, 1, 8, 9 (1, 8), i.e. –00–. This has already been recorded in column 3. So, no need to record again. Check off 0, 8 (8), i.e. –000 and 1, 9 (8), i.e. –001 of column 2. Draw a line below 0, 1, 8, 9 (1, 8), i.e. –00–. This is the only term in group 1 of column 3. No term of group 2 of column 2 can be combined with any term of group 3 of column 2. So, no entries are made in group 2 of column 2.

Comparing the terms of group 3 of column 2 with the terms of group 4 of column 2, the terms 6, 7 (1), i.e. 011–, and 14, 15 (1), i.e. 111– are combined to form 6, 7, 14, 15 (1, 8), i.e. –11–. Record it in group 3 of column 3 and check off 6, 7 (1), i.e. 011– and 14, 15 (1), i.e. 111– of column 2. The terms 6, 14 (8), i.e. –110 and 7, 15 (8), i.e. –111 are combined to form 6, 7, 14, 15 (1, 8), i.e. –11–. This has already been recorded in column 3; so, check off 6, 14 (8), i.e. –110 and 7, 15 (8), i.e. –111 of column 2.

Observe that the terms 9, 13 (4), i.e. 1–01 and 13, 15 (2), i.e. 11–1 cannot be combined with any other terms. Similarly in column 3, the terms 0, 1, 8, 9 (1, 8), i.e. –00– and 6, 7, 14, 15 (1, 8), i.e. –11– cannot also be combined with any other terms. So, these 4 terms are the prime implicants.

The terms, which cannot be combined further, are labelled as P, Q, R, and S. These form the set of prime implicants.

EX:

Obtain the minimal expression for f = Σ m(1, 2, 3, 5, 6, 7, 8, 9, 12, 13, 15) using the tabular method.

Solution

The procedure to obtain the set of prime implicants is illustrated in Table 3.4.

Table 3.4 Example 3.30

Step 1		Step 2	Step 3	
Index 1	1 ✓	1, 3 (2) ✓	1, 3, 5, 7 (2, 4)	T
	2 ✓	1, 5 (4) ✓	1, 5, 9, 13 (4, 8)	S
	8 ✓	1, 9 (8) ✓	2, 3, 6, 7 (1, 4)	R
Index 2	3 ✓	2, 3 (1) ✓	8, 9, 12, 13 (1, 4)	Q
	5 ✓	2, 6 (4) ✓	5, 7, 13, 15 (2, 8)	P
	6 ✓	8, 9 (1) ✓		
	9 ✓	8, 12 (4) ✓		
	12 ✓	3, 7 (4) ✓		
Index 3	7 ✓	5, 7 (2) ✓		
	13 ✓	5, 13 (8) ✓		
Index 4	15 ✓	6, 7 (1) ✓		
		9, 13 (4) ✓		
		12, 13 (1) ✓		
		7, 15 (8) ✓		
		13, 15 (2) ✓		

The non-combinable terms P, Q, R, S and T are recorded as prime implicants.

$$P \rightarrow 5, 7, 13, 15\ (2, 8) = X\ 1\ X\ 1 = BD$$

(Literals with weights 2 and 8, i.e. C and A are deleted. The lowest minterm is $m_5(5 = 4 + 1)$. So, literals with weights 4 and 1, i.e. B and D are present in non-complemented form. So, read it as BD.)

$$Q \rightarrow 8, 9, 12, 13\ (1, 4) = 1\ X\ 0\ X = A\bar{C}$$

(Literals with weights 1 and 4, i.e. D and B are deleted. The lowest minterm is m_8. So, literal with weight 8 is present in non-complemented form and literal with weight 2 is present in complemented form. So, read it as $A\bar{C}$.)

$$R \rightarrow 2, 3, 6, 7\ (1, 4) = 0\ X\ 1\ X = \bar{A}C$$

(Literals with weights 1 and 4, i.e. D and B are deleted. The lowest minterm is m_2. So, literal with weight 2 is present in non-complemented form and literal with weight 8 is present in complemented form. So, read it as $\bar{A}C$.)

$$S \rightarrow 1, 5, 9, 13\ (4, 8) = X\ X\ 0\ 1 = \bar{C}D$$

(Literals with weights 4 and 8, i.e. B and A are deleted. The lowest minterm is m_1. So, literal with weight 1 is present in non-complemented form and literal with weight 2 is present in complemented form. So, read it as $\bar{C}D$.)

$$T \rightarrow 1, 3, 5, 7\ (2, 4) = 0\ X\ X\ 1 = \bar{A}D$$

(Literals with weights 2 and 4, i.e. C and B are deleted. The lowest minterm is 1. So, literal with weight 1 is present in non-complemented form and literal with weight 8 is present in complemented form. So, read it as $\bar{A}D$.)

The prime implicant chart of the expression

$$f = \Sigma\, m(1, 2, 3, 5, 6, 7, 8, 9, 12, 13, 15)$$

is as shown in Table 3.5. It consists of 11 columns corresponding to the number of minterms and 5 rows corresponding to the prime implicants P, Q, R, S, and T generated. Row R contains four ×s at the intersections with columns 2, 3, 6, and 7, because these minterms are covered by the prime implicant R. A row is said to cover the columns in which it has ×s. The problem now is to select a minimal subset of prime implicants, such that each column contains at least one × in the rows corresponding to the selected subset and the total number of literals in the prime implicants selected is as small as possible. These requirements guarantee that the number of unions of the selected prime implicants is equal to the original number of minterms and that, no other expression containing fewer literals can be found.

Table 3.5 Example 3.30: Prime implicant chart

		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	2	3	5	6	7	8	9	12	13	15
$^*P \rightarrow 5, 7, 13, 15\ (2, 8)$				×		×				×	×
$^*Q \rightarrow 8, 9, 12, 13\ (1, 4)$							×	×	×	×	
$^*R \rightarrow 2, 3, 6, 7\ (1, 4)$		×	×		×	×					
$S \rightarrow 1, 5, 9, 13\ (4, 8)$	×			×				×		×	
$T \rightarrow 1, 3, 5, 7\ (2, 4)$	×		×	×		×					

In the prime implicant chart of Table 3.5, m_2 and m_6 are covered by R only. So, R is an essential prime implicant. So, check off all the minterms covered by it, i.e. m_2, m_3, m_6, and m_7. Q is also an essential prime implicant because only Q covers m_8 and m_{12}. Check off all the minterms covered by it, i.e. m_8, m_9, m_{12}, and m_{13}. P is also an essential prime implicant, because m_{15} is covered only by P. So check off m_{15}, m_5, m_7, and m_{13} covered by it. Thus, only minterm 1 is not covered. Either row S or row T can cover it and both have the same number of literals. Thus, two minimal expressions are possible.

$$P + Q + R + S = BD + A\bar{C} + \bar{A}C + \bar{C}D$$

or

$$P + Q + R + T = BD + A\bar{C} + \bar{A}C + \bar{A}D$$