2.4.12 Karnaugh Map in Minimization of Boolean Expressions

Boolean expression can be simplified by using Boolean algebraic theorems but there are no specific rule to make the most simplified expression. However, K-map can easily minimize the terms of a Boolean function. Unlike an algebraic method, K-map is a pictorial method and it does not need Boolean algebraic theorem.

K-map is basically a diagram made up of squares. Each of these squares represents a min-term of the variables. If n = numbers of variables then the number of squares in its K-map will be 2^n. K-map is made using the truth table. In fact, it is a special form of be truth table that is folded upon itself like a sphere. Every two adjacent squares of the K-map have a difference of 1 bit including the corners.

K-map can produce **sum of product** (SOP) or **Product of Sum** (POS) expression considering which of the two (0, 1) outputs, are being grouped in it. The grouping of 0's result in **Product of Sum Expression** and the grouping of 1's result in **Sum of Product Expression**. The expression produced by K-map may be the most simplified expression but not unique. There can be more than 1 simplified expression for a single function but they all perform the same.

2.4.13 Rule to Minimization in K-map

1. While grouping, you can make groups of 2^n number, where $n = 0, 1, 2, 3, ...$
2. You can either make groups of 1's (or 0's) but not both.
3. Grouping of 1's lead to sum of product term and groping of 0's lead to product of sum term.
4. While grouping, the groups of 1's should not contain any 0 and the group of 0's should not contain any 1.
5. The function output for 0's grouping should be complemented as F'.

6. Group can be made vertically and horizontally but not diagonally.
7. Group made should be as large as possible even if they overlap.
8. All the like term should be in a group even if they overlap.
9. Uppermost and lowermost squares can be made into a group together as they are adjacent (1-bit difference).

   Same goes for the corner squares.
10. Each group represent a term in the Boolean expression. Larger the group, smaller and simple the term.
11. The product of those literals that remains uncharged in a single group makes the term of the expression.
12. Don't care "X" should also be included while grouping to make a large possible group.

Remark : Karnaugh map of 2 to 4 variables are very easy. However 5 and 6 variable K-map is a little bit complex.

Example 1 : Use the Karnaugh map representation to find a minimal sum of the product expression of the following Boolean function $f(A, B) = \Sigma\, m\,(0, 1, 2)$.

00
01
10
11

Sol. Here $F(A, B) = \Sigma\, m\,(0, 1, 2) = m_0 + m_1 + m_2 = A'B' + A'B + AB'$.

Clearly the given Boolean expression is in ~~POS~~ SOP term. Its K-map for two variable is

A \ B	0	1
0	1	1
1	1	0

A \ B	0	1
0	$A'B'$ m_0	$A'B$ m_1
1	AB' m_2	AB m_3

We made 2 groups of 1's. Each group contains 2 minterms.

In the first group, (m_0, m_2) variable A is changing and B remain unchanged. So the first term of the output expression will be B' (because $B = 0$ in this group). In the second group (m_0, m_1) variable B is changing and variable A remain unchanged so the second term ~~will be~~ of the output expression will be A' (because $A = 0$ in this group).

Now the simplifies expression will be the sum of these two terms as given below :

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

Example 2 : Use the K-map representation to find a minimal sum of the product expression of the Boolean function $F(A, B, C) = \Sigma\, m\,(0, 1, 2, 4, 5, 6)$

Sol. Here $F(A, B, C) = \Sigma\, m\,(0, 1, 2, 4, 5, 6) = m_0 + m_1 + m_2 + m_4 + m_5 + m_6$

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

000
001
010
011
100
101
110
111

Clearly the given Boolean expression is in the SOP POS form. Its K-map for three variable is

A \ BC				
0	1	1	0	1
1	1	1	0	1

A \ BC	00	01	11	10
0	A'B'C' m_0	A'B'C m_1	A'BC m_3	A'BC' m_2
1	AB'C' m_4	AB'C m_5	ABC m_7	ABC' m_6

Here we see that, we can make the groups overlap each other to make them as large as possible and it cover all the 1's.

In the first group (m_0, m_1, m_4, m_5) A and C are changing and B remain unchanged so the first term of the output expression will be B′ (because B = 0 in this group).

In the 2nd group (m_0, m_4, m_2, m_6) A and B are changing and C remain, unchanged so the second term of the output expression will be C′ (because C = 0 in this group).

The sum of the two terms will make the simplified expression of the function and is given below :

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

Example 3 : Minimize the following Boolean expression :

$$F(A, B, C, D) = \Sigma(m_0, m_1, m_2, m_4, m_5, m_6, m_8, m_9, m_{10}, m_{12}, m_{13})$$

Sol. The K-map of the given Boolean expression is

AB \ CD	00	01	11	10
00	1	1	0	1
01	1	1	0	1
11	1	1	0	0
10	1	1	0	1

AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

First of all, we try to make the biggest possible groups as shown in this K-map. Secondly the corners can also be combined to make a group of 4. The remaining last 1 should be combined with the made group to make a big overlapping group.

Group of 8 will give a term of 1 literal that remain unchanged *i.e.* C′.

Corner group of 4 will give term with 2 literals remain unchanged i.e., $B' D'$.

The last group of 4 will give $A' D'$ because they remain unchanged in the group.

So the expression will be $F(A, B, C, D) = C' + B' D' + A' D'$

2.4.14 Another Method for Minimization of Boolean Expression

There are two basic steps for minimizing functions namely, determining prime implicants and then finding subsets such implicants that cover all product terms of a function.

2.4.15 Definition (Implicant) : An implicant of a function is a product term that is included in the function.

For ~~infence~~ instance : $x\,y\,z$ is an implicant of $f(x, y, z) = x\,y$, for $x\,y = x\,y\,z + x\,y\,z'$

2.4.16 Definition (Prime implicant) : A prime implicant of a function is an implicant that is not included in any other implicant of the function.

For instances, xyz is not a prime implicant of $f(x, y, z) = x\,y + x'\,y'\,z'$; because in $x\,y\,z$ is contained in xy.

$x\,y$ is a prime implicant of $f(x, y, z)$ because it is not contained in $x'\,y'\,z'$. So, if an implicant is not prime, then it is possible to obtain prime implicant of by removing some literals from it.

2.4.17 Definition : (Essential prime implicant). If a prime impliant includes a min term that is not included in any other prime impliant, then it called an essential prime impliant.

For example : $f(x, y, z) = x\,y + x'\,y'\,z'$ has two prime implant normally $x\,y$ and $x'\,y'\,z'$. prime implicant $x\,y$ is essential because $x\,y$ contains $x\,y\,z$ and $x\,y\,z'$ which are not contained in any other prime implicant (*i.e.* $x'\,y'\,z'$).

In another function $f(x, y, z) = x\,y + x\,y' + x\,z'$, the prime impliants are $x\,y$, $x\,y'$ and $x\,z'$. Among them $x\,z'$ is not essential because $x\,z' = x\,y'\,z' + x\,y\,z'$ and $x\,y'\,z'$ is in $x\,y'$ and $x\,y\,z'$. and $x\,y'\,z'$ is in $x\,y'$ and $x\,y\,z'$ is in $x\,y$.

2.4.18 Quine-McCluskey Tabular Method (Q