3.1 Objectives

- To write Boolean functions in their standard Min and Max terms format.
- To simplify Boolean expressions using Karnaugh Map.

3.2 Sum of Products & Product of Sums

Any Boolean expression can be simplified in many different ways resulting in different forms of the same Boolean function. All Boolean expressions, regardless of their forms, can be converted into one of two standard forms; the sum-of-product form and the product-of-sum forms. Standardization makes the evaluation, simplification and implementation of Boolean expression more systematic and easier.

The Sum-of-Product (SOP):

Writing functions in SOP form means that the inputs of each term are multiplied using AND function, then all terms are added together using OR function. The variables in each term are not necessarily all the variables of the function. For example, a SOP of F(A,B,C) may contain a term that contains only the variable A but not B nor C, in such case the term is not in its ***standard SOP form***. ***Standard SOP term*** must contain all the function variables. From Boolean algebra thermos (X+X'=1), then if the term is multiplied by (X+X'), it becomes in the standard SOP form, but its value is not affected.

Example 3.1:

The following function is written in the SOP form:

F(A,B,C)=A+BC'+A'BC

The inputs to the function F are A, B and C. In each term the inputs are ANDed then all terms are ORed to form the function F.

Note that the last term A'BC contains all the inputs of the function (A, B and C), so, this term is written in standard form. But the second term BC' is not in standard form because the input A does not exist, then multiply by (A'+A). The same is done for the remaining term as follows:

F(A,B,C)=A(B+B')(C+C')+BC'(A+A')+A'BC

F(A,B,C)=ABC+ABC'+AB'C+AB'C'+ ABC'+A'BC'+A'BC

From Boolean algebra,

(A+A=A)

then all similar terms in the equation will be reduced to one term. Now the function F becomes

F(A,B,C)=ABC+ABC'+AB'C+AB'C'+ A'BC'+A'BC

The Product-of-Sum (POS):

Writing functions in POS form means that the inputs of each term are Added together using OR function then all terms are multiplied together using AND function. The variables in each term are not necessarily all the variables of the function. For example, a POS of F(A,B,C) may contain a term that contains only the variable A but not B nor C, in such case the term is not in its ***standard POS form***. ***Standard POS term*** must contain all the function variables. From Boolean algebra thermos (X.X'=0), then if the term is added to (X.X'), it becomes in the standard POS form, but its value is not affected.

Example 3.2:

The following function is written in the POS form:
F(A,B,C)=A.(B+C').(A'+B+C')

The inputs to the function F are A, B and C. In each term the inputs are ORed then all terms are ANDed to form the function F.

Note that the last term (A'+B+C') contains all the inputs of the function (A, B and C), so, this term is written in standard form. But the second term (B+C') is not in standard form because the input A does not exist, then add (A'.A). The same is done for the remaining term as follows:

F(A,B,C)=[A+(B.B')+(C.C')].[(B+C')+(A.A')].(A'+B+C')

F(A,B,C)=[(A+B+C).(A+B+C').(A+B'+C).(A+B'+C')].[(A+B+C').(A'+B+C')].(A'+B+C')

From Boolean algebra,

(A.A=A)

then all similar terms in the equation will be reduced to one term. Now the function F becomes

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

3.3 Minterms

Writing a function in its **minterm** format is equivalent to writing the function in its standard SOP format such that the value of the function at these terms is 1. So that if we have the truth table relating the input variables to the function F, then we can determine which cases result in F=1 and write the minterm form of the function.

Example 3.3:

Using the following truth table, write the function F in its minterm format.

3.4 Maxterms

Writing a function in its maxterm format is equivalent to writing the function in its standard POS format such that the value of the function at these terms is 0. So that if we have the truth table relating the input variables to the function F, we can determine which cases result in F=0 and write the maxterm form of the function.

Example 3.4:

Using the truth table above, write the function F in its maxterm format.

The function F is equal to 0 in the un-highlighted cases above, which are the cases of 1,3,4 and 5 or (A+B+C'), (A+B'+C'), (A'+B+C) and (A'+B+C') respectively.

The function F is written in maxterms as follows:

The function F is equal to 1 in the highlighted cases, which are the cases of 0,2,6 and 7 or A'B'C', A'BC', ABC' and ABC respectively.

	A	B	C	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

The function F is written in minterm format as follows:

F(A,B,C)= A'B'C'+A'BC'+ ABC'+ ABC

Or

F(A,B,C)=∑(0,2,6,7)

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

Or

F(A,B,C)= Π(1,3,4,5)

Note: From Example 3.3 and 3.4, the maxterm is the complement of the minterm.

3.5 Karnaugh Map

The Karnaugh map (K-map) provides a systematic way of simplifying Boolean algebra expressions. This can be done without thoroughly searching the basic theorems of Boolean algebra. Instead all the possible combinations of the variables written in the standard form for **POS (product of sums) or SOP (sums of products)** are plotted in cells arranged in a rectangle or square. Adjacent cells share a redundant Boolean variable. The simplification of the original Boolean expression comes form grouping the logical one's (minterms) or 0's (maxterms). This eliminates the redundant variable and simplifies the original Boolean expression.

The circulation must be done according to the following rules:

1. A group must contain either 1, 2, 4, 8, 16…..cells.
2. Each cell in a group must be adjacent to one or more cells in the same group, but all cells do not have to be adjacent to each other.
3. ALWAYS include the most possible of 1's in a group in accordance to rule (1).
4. Each 1 on the K-map must be included in at least one group. The 1's in a group can be included in another group as long as the overlapping groups include non common 1's.

Example 3.5 shows how to use the K-map to simplify functions.

Example 3.5:

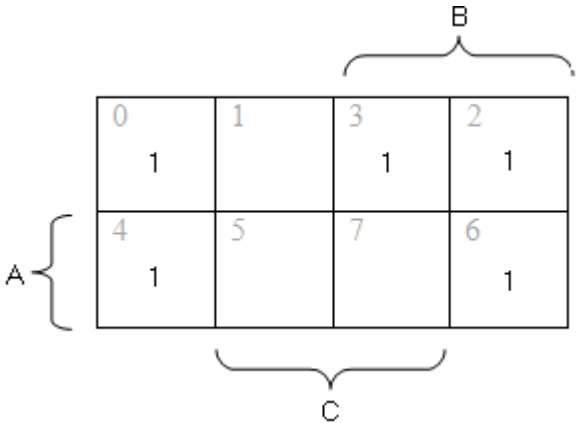
Simply the following function using K-map

F(A,B,C)=A'B+ABC'+B'C'

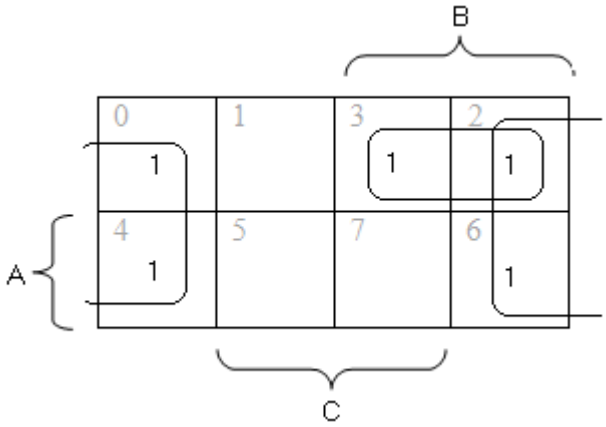
In order to use the K-map the function should be written in its min or maxterms format.

F(A,B,C)=A'BC+A'BC'+ABC'+AB'C'+A'B'C'

F(A,B,C)=∑(0,2,3,4,6)



To write the function as simplified SOP then circle the 1's in K-map



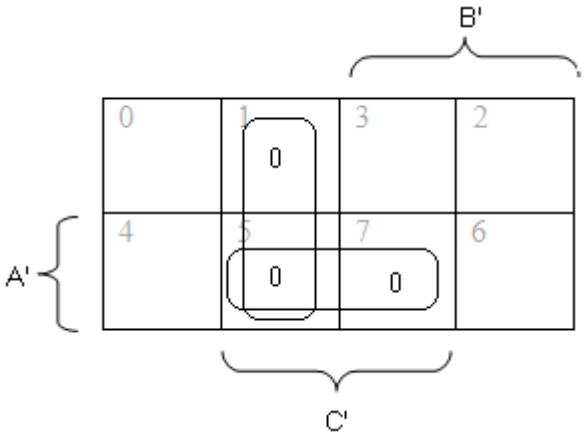
From K-map

F(A,B,C)=C'+BA'

F is written in maxterm as follows

F(A,B,C)= Π(1,5,7)

To write the function as POS then circle the 0's in K-map



From K-map

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

Pre-Lab 3

1- Which of following function is SOP and which is POS:

a) F1(A,B,C) = A'+A'.B+B.C

b) F2(A,B,C) = A'.B'.C+C'

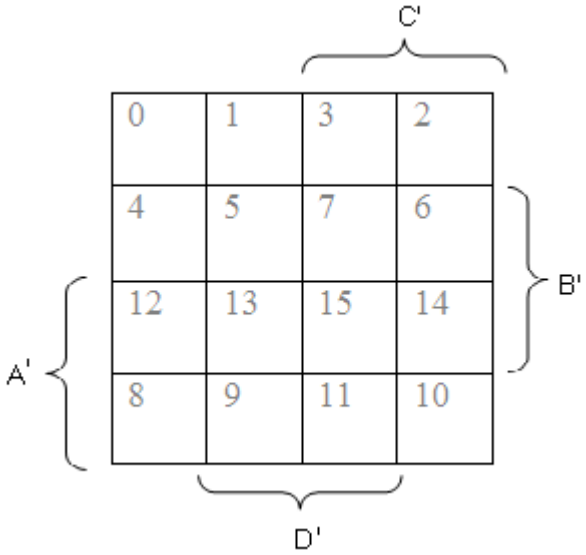
c) F3(A,B,C) = (A'+C').(B+A').(A+B'+C)

2- Given the function

F1 (A, B, C, D) = (A' +B+C).(A+B+D').(B'+C+D)

a- Write the function in standard POS form.

b- Using the K-map shown below, write the function in minterm form.



3- Design a logic function that detects if the number of 1s in a 4-bit binary number is less than 3. (i.e. if ∑1s< 3 then F is HIGH).use K-map to write the function **F(A,B,C,D) in SOP**. Build the logic circuit using MultiSIM.

Lab Work 3

Implement the circuit you designed in question 3.

Homework 3

1. Design a logic circuit to produce a HIGH output only if the input, represented by a 4-bit binary number, is greater than 12 or less than 3.
   a. Develop the truth table.
   b. Using K-map, write the function F(A,B,C,D) in SOP format by circling the 1's.
   c. Using K-map, write the function F(A,B,C,D) in POS format by circling the 0's.