

SIMPLIFICATION AND MINIMIZATION OF BOOLEAN FUNCTIONS

3.1 Introduction

The complexity of digital logic gates to implement a Boolean function is directly related to the complexity of algebraic expression. Also, an increase in the number of variables results in an increase of complexity. Although the truth table representation of a Boolean function is unique, its algebraic expression may be of many different forms. Boolean functions may be simplified or minimized by algebraic means as described in lecture 2. However, this minimization procedure is not unique because it lacks specific rules to predict the succeeding step in the manipulative process. The map method, first proposed by Veitch and slightly improvised by Karnaugh, provides a simple, straightforward procedure for the simplification of Boolean functions. The method is called Veitch diagram or Karnaugh map, which is regarded as a pictorial representation of a truth table.

The Karnaugh map provides a systematic method for simplification and manipulation of a Boolean expression. The map is a diagram consisting of squares. For n variables on a Karnaugh map there are 2^n numbers of squares. Each square or cell represents one of the minterms. Since any Boolean function can be expressed as a sum of minterms, it is possible to recognize a Boolean function graphically in the map from the area enclosed by those squares whose minterms appear in the function. It is also possible to derive alternative algebraic expressions or simplify the expression with a minimum number of variables or literals and sum of products or product of sums terms, by analyzing various patterns. In fact, the map represents a visual diagram of all possible ways a function can be expressed in a standard form and the simplest algebraic expression consisting of a sum of products or product of sums can be selected. Note that the expression is not necessarily unique.

3.2 TWO-VARIABLE KARNAUGH MAPS

A two-variable Karnaugh map is shown in Figure (3.1).

X	Y	Minterms	
0	0	$x'y'$	m0
0	1	$x'y$	m1
1	0	xy'	m2
1	1	xy	m3

		y		0	1
x	0	$x'y'$	m0	$x'y$	m1
	1	xy'	m2	xy	m3

Figure (3.1) : Two variable karnaugh map.

Example 3.1: Simplify the following Boolean expression using Karnaugh map:

$$F(x,y)=x'y+x'y'$$

Solution:

		y		0	1
x	0	1	1		
	1				

$F(x,y) = x'$

Example 3.2: Simplify the following Boolean expression using Karnaugh map:

$$F(x,y)=xy+x'y$$

Solution:

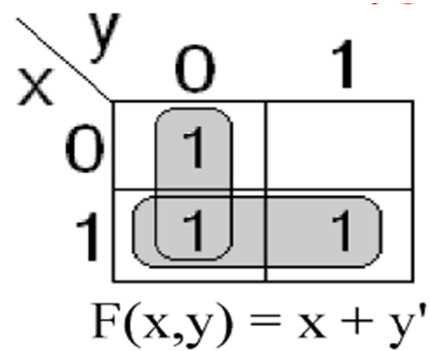
		y		0	1
x	0		1		
	1		1		

$F(x,y) = y$

Example 3.3: Simplify the following Boolean expression using Karnaugh map:

$$F(x,y)=x'y'+xy'+xy$$

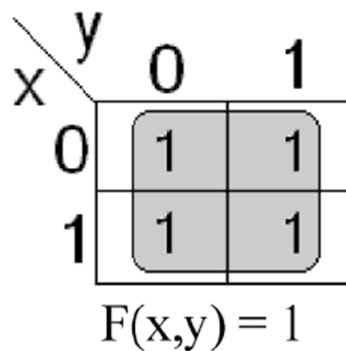
Solution:



Example 3.4: Simplify the following Boolean expression using Karnaugh map:

$$F(x,y)=xy+x'y+xy'+x'y'$$

Solution:



3.3 THREE-VARIABLE KARNAUGH MAPS

A three-variable Karnaugh map is shown in Figure (3.2).

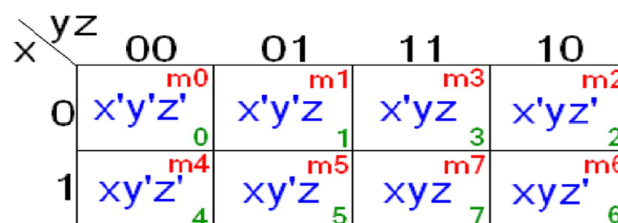


Figure (3.2) : Three variable karnaugh map.

Example 3.5: Simplify the following Boolean expression using Karnaugh map:

$$F(x,y,z) = \Sigma(3,4,6,7)$$

Solution:

x \ yz	yz			
	00	01	11	10
0			1	
1	1		1	1

$$F(x,y,z) = xz' + yz$$

Example 3.6: Simplify the following Boolean expression using Karnaugh map:

$$F(x,y,z) = \Sigma(0,1,2,4,5,6)$$

Solution:

x \ yz	yz			
	00	01	11	10
0	1	1		1
1	1	1		1

$$F(x,y,z) = y' + z'$$

3.4 FOUR-VARIABLE KARNAUGH MAPS

A four-variable Karnaugh map is shown in Figure (3.3).

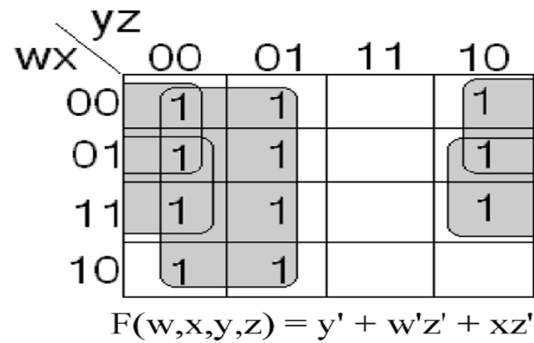
wx \ yz	yz			
	00	01	11	10
00	m0 $w'x'y'z'$ 0	m1 $w'x'y'z$ 1	m3 $w'x'yz$ 3	m2 $w'x'yz'$ 2
01	m4 $w'xy'z'$ 4	m5 $w'xy'z$ 5	m7 $w'xyz$ 7	m6 $w'xyz'$ 6
11	m12 $wxy'z'$ 12	m13 $wxy'z$ 13	m15 $wxyz$ 15	m14 $wxyz'$ 14
10	m8 $wx'y'z'$ 8	m9 $wx'y'z$ 9	m11 $wx'yz$ 11	m10 $wx'yz'$ 10

Figure (3.2) : Four variable karnaugh map.

Example 3.7: Simplify the following Boolean expression using Karnaugh map:

$$F(w,x,y,z) = \Sigma(0,1,2,4,5,6,8,9,12,13,14)$$

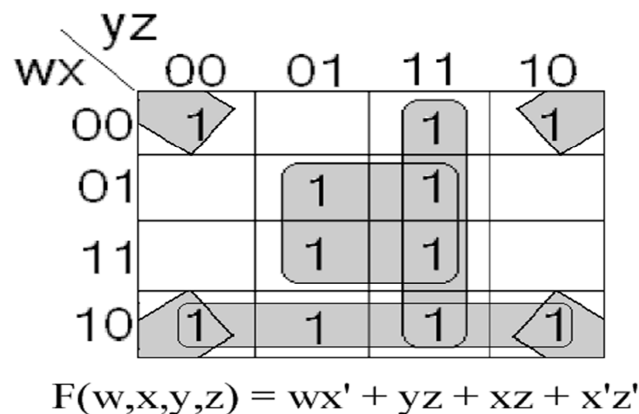
Solution:



Example 3.8: Simplify the following Boolean expression using Karnaugh map:

$$F(w,x,y,z) = \Sigma(0,2,3,5,7,8,9,10,11,13,15)$$

Solution:



3.5 DON'T-CARE COMBINATIONS

In certain digital systems, some input combinations never occur during the process of a normal operation because those input conditions are guaranteed never to occur. Such input combinations are called *Don't-Care Combinations*. The function output may be either 1 or 0 and these functions are called incompletely specified functions. These input combinations can be plotted on the Karnaugh map for further simplification of the function. The don'tcare combinations are represented by d or x or Φ .

Example 3.8: Simplify the following Boolean expression using Karnaugh map:

$$F(w,x,y,z) = \Sigma(1,3,7,11,15) + x(0,2,5,8)$$

Solution:

		yz			
		00	01	11	10
wx	00	×	1	1	×
	01		×	1	
	11			1	
	10	×		1	

$$F(w,x,y,z) = w'x' + yz$$