A New Approach to Simplifying Boolean Functions

A. H. M. Ashfak Habib[*1], Md. Abdus Salam[2], Zia Nadir[1], and Hemen Goswami[1]

[1]*Dept. of Computer Science & Engineering,*
Chittagong University of Engineering & Technology (CUET), Bangladesh
[2]*Dept. Electrical and Computer Engineering, College of Engineering, Sultan Qaboos University, Muscat, OMAN.*

المستخلص: هناك العديد من الفوائد في تبسيط الدوال البوليانية قبل تحقيقها وربطها في الدائرة. إن تقليل عدد البوابات في الدائرة الإلكترونية يقلل من كلفة بنائها ويقلل من الحرارة المتولدة من الدائرة المتكاملة ويزيد سرعة عملها وانه لا توجد أية طريقة تقوم بتبسيط الدوال البوليانية الحاوية على اكثر من ستة متغيرات. هذا البحث يستعرض طريقة جديدة للتبسيط بالحل اليدوي ومن الممكن تطبيقها في المسائل الحاوية على عدد كبير من المتغيرات وبكفاءة.

المفردات المفتاحية: طريقة الجدولة، طريقة الخريطة، خريطة كارنف.

Abstract: There are many benefits to simplifying Boolean functions before they are implemented in hardware. A reduced number of gates decreases considerably the cost of the hardware, reduces the heat generated by the chip and, most importantly, increases the speed. But no method is effective for the simplification of Boolean functions, if it involves more than six variables. This paper presents a new manual method of simplification that can be effectively applied to problems with a large number of variables.

Keywords: Tabulation method, Map-tabulation, Karnaugh map

1. Introduction

The complexity of digital logic gates that implement a Boolean function is directly related to the complexity of the algebraic expression from which the function is implemented. Although the truth table representation of a function is unique, expressed algebraically, it can appear in many different forms. Boolean functions may be simplified by various means. We can use the rules of Boolean algebra to simplify an expression, but this method has a few problems. One is that there is no algorithm, which can be used to determine whether or not we have obtained the minimum solution. When should we stop looking for a simplification theorem? Another problem is that we often have to make the expressions more complicated before attempting a simplification. Then we rearrange the terms to obtain advantageous groupings that help to simplify the expressions at a later stage. It is against human nature to climb out of a "local minimum" in the hope of finding a better global solution. But this is exactly what we often have to do.

Another straightforward and simple way of simplification is the "Karnaugh map" method. This method is convenient as long as the number of variables does not exceed five or six. As the number of variables increases, the excessive number of squares prevents a reason exceed five or six. As the number of variables increases, the excessive number of squares prevents a reason able selection of adjacent squares. The obvious disadvantage of the map is that it is essentially a trial and error procedure, which relies on the ability of the human user to recognize certain patterns. For functions of six or move variables, it is difficult to be sure that the best selection has been made.

The tabulation method overcomes this difficulty. It is a specific step-by-step procedure that is guaranteed to produce a simplified standard-form expression for a function. It can be applied to problems with many variables and has the advantage of being suitable for machine computation. However, it is quit tedious and is prone to mistakes because of its routine, and it is a monotonous process. Finally, it is just too cumbersome to manipulate Boolean expressions by hand. Though computer-based tools have been developed for Boolean simplification, we still need knowledge of the basic approach. Because, observing the symmetries in a circuit's function helps to understand its behavior. As CAD tools become even more sophisticated, we need a deeper knowledge of algorithms they apply to use the tools effectively.

This paper assesses previous methods and describes a new method, which uses the advantages of both the map method and the tabulation method and overcomes their disadvantages.

_____
*Corresponding author Email: ashfak@bttb.net.bd

Table 1. General format of simplification table

Group	Input Variables																Simplification of group-1		Simplification of group-2		...	Selected Implicants
	n			n-1				...			2		1				K-map	Simplified terms	K-map	Simplified terms		
	A	B	C	D	.	.	.	...	.	.	U	V	W	X	Y	Z						
1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0						
	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0						
	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	1						
	0	0	0	0	0	0	0	...	0	0	0	0	1	0	1	0						
2	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	1						
	0	0	0	0	0	0	0	...	0	0	0	1	0	0	1	0						
3	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*				
4	0	0	0	0	0	0	0	...	0	0	1	1	0	0	1	0						
	0	0	0	0	0	0	0	...	0	0	1	1	0	1	0	0						
.	.	.	.	.	.	.	.	...	.	.	.	.	.	.	.	.	...	...				
16	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*				
17	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*				
.	.	.	.	.	.	.	.	...	.	.	.	.	.	.	.	.	...	...				
32	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*				
33	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*				
.	.	.	.	.	.	.	.	...	.	.	.	.	.	.	.	.	...	...				
n						m_k											...	...				
						m_l																

2. Description Of New Map-Tabulation Method

Our proposed method, which utilizes both the concepts of the Karnaugh map (Karnaugh, 1953) and the Quine-McCluskey's Tabulation techniques (Mano, 1979), can be termed as the Map-Tabulation (M-T) method. With our approach the given Boolean function should be expressed as the sum of minterms form (a Canonical form), such as:

$$F (A,B,C,D,... .. ...,U,V,W,X,Y,Z) = \Sigma (m_0, m_2, m_3,$$
$$m_{10}, m_{17}, m_{18}, m_{50}, m_{52}, ..., m_k, m_l) \qquad [1]$$

The input variables and the minterms should be divided in to several groups. Each group of input variables will contain maximum of four (4) members, whereas each group of minterms will contain a maximum of sixteen (16) terms. For grouping the input variables we should start from the least significant bit. So the least four bits will form the first group. The 5th to 8th bits will form the second group, and so on. The last group can contain fewer members. Now for grouping the minterms it should be confirmed that the minterms are arranged in the ascending order of their subscripts. If there are n numbers of input variables, the minterms   will be grouped into $2^n / 16 = 2^{(n-4)}$ groups. The minterms among and including m_0 and m_{15} would form the first group, m_{16} to m_{31} forms the second group, m_{32} to m_{47} forms third, and so on. It is not nec

$$F\left(\frac{A,B,C,}{n} \middle| \frac{D,...,...,...,}{n-1} \middle| .... \middle| \frac{...,..., U, V,}{2} \middle| \frac{W, X, Y, Z}{1}\right) = \qquad [2]$$

$$\Sigma\left(\frac{m_0,m_2,m_3,m_{10},}{1} \middle| \frac{m_{17},m_{18},}{2} \middle| \frac{*}{3} \middle| \frac{m_{50},m_{52},}{4} \middle| .... \middle| \frac{m_k,m_l}{n}\right)$$

essary that all these groups must contain elements. Some could form empty groups. If such an empty group occurs, an asterisks (*) symbol will be placed in the group. Two successive groups are separated by a vertical line (|). After grouping the elements the function will become:

Now this function will be rearranged in a tabular form as shown in Table 1. Before placing the functions in the table, all minterms are converted to their equivalent binary. The binary information of cell [1,1] (i.e. group no. of minterms, group no. of variables) will be solved by using

Table 2. Determination of implicants for example 1

Group	Input Variables							Simplification of group-1		Simplification of group-2		Selected Implicants
	2			1				K-map	Simplified terms	K-map	Simplified terms	
	T	U	V	W	X	Y	Z					
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 1 0 0 1 1 0 1 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1				W'YZ' WX √ WY Z' √	WX (1,2) TU' WY'Z' (1,2,4) TU', TV	TU'WX TU'WY'Z' TVWY Z' UVW'X'YZ' TVWXYZ TVWXY'	
2	0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1			0 1 0 1 0 1 1 1 1 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1				XZ WX √ WY'Z' √	W'X'YZ' (4,8) W'XYZ (2,4)	TU'V'W'YZ' TU'VXZ TU'WX TU'WY'Z' TVWY'Z' UVW'X'YZ TVWXYZ TVWXY'
3	* * *			* * * *				*	*		
4	0 1 1 0 1 1 0 1 1 0 1 1 0 1 1			0 0 1 0 0 1 1 1 1 0 0 0 1 1 0 0 1 1 0 1				W'X'YZ' √ W'XYZ # WXY' # WY'Z' √	UV W'XYZ (2,4) TV	
5	* * *			* * * *				*	*		
6	* * *			* * * *				*	*		
7	* * *			* * * *				*	*		
8	1 1 1			0 0 1 0				W'X'YZ' √	WXY' (1,2,4) TV	

a four variable k-map. Simplified terms are placed in the adjacent column. Similarly the binary information of other cell's [2, 1], [3, 1], [4, 1], … , [n,1] are simplified accordingly and the corresponding simplified terms are written. Now all the simplified terms are compared and the common terms are marked. For the uncommon terms all the K-maps are compared . If a loop of 1s produces an uncommon term or common terms of groups which are at least 15 groups apart, matches exactly in position and structure with another loop of another map, then those terms are marked with a different symbol, "#". If no such a match is

found for a term, then the term is listed in the last column along with its higher significant bits. The binary information of the second group of variables is simplified for the common terms and the "#" marked uncommon terms. If the second group of variables contains only one variable, no simplification for the higher terms would be needed. Each variety of the common term is then placed in the last column without combining their higher term. If the second group of variables contains two to four variables, then a K-map of two to four variables should be formed depending on the number of variables of the second group of vari-

Table 3. Determination of implicants for example 2

Group	Input Variables								Simplification of group-1		Simplification of group-2			Selected Implicants
	3	2			1				K-map	Simplified terms	K-map	Simplified terms		
	a	b	c	d	e	f	g	h	i						
1	0 0	0 0	0 0	0 0	0 0	0 0	1 1	1 1	0 1		f'gh √	f'gh (1,2)	b'c'df'gh b'e'f'gh c'd'ef'gh		
2	0 0	0 0	0 0	0 0	1 1	0 0	1 1	1 1	0 1		f'gh √	b'c'd' f'ghi'(3,5,7,9,17)			a'b'c'd'f'gh a'b'e'f'gh a'c'd'ef'gh b'c'd'e f'ghi'
3	0	0	0	1	0	0	1	1	0	-	f'ghi' √				
4	*	*	*	*	*	*	*	*	*	*	*				
5	0	0	1	0	0	0	1	1	0	-	f'ghi' √	b'c', c'd'e'			
6	*	*	*	*	*	*	*	*	*	*	*				
7	0	0	1	1	0	0	1	1	0	-	f'ghi' √				
8	*	*	*	*	*	*	*	*	*	*	*				
9	0	1	0	0	0	0	1	1	0		f'ghi' √	b'c', c'd'e'			
10	*	*	*	*	*	*	*	*	*	*	*				
11	*	*	*	*	*	*	*	*	*	*	*				
12	*	*	*	*	*	*	*	*	*	*	*				
13	*	*	*	*	*	*	*	*	*	*	*				
14	*	*	*	*	*	*	*	*	*	*	*				
15	*	*	*	*	*	*	*	*	*	*	*				
16	*	*	*	*	*	*	*	*	*	*	*				
17	1	0	0	0	0	0	1	1	0	-	f'ghi' √√√	f'ghi'(17) b'c'd'e'	b'c'd'e'f'ghi'		

(Rightmost column: Simplification of variable group-3 for b'c'd'e'f'g hi' → 1)

positions are marked with ×; and are treated as the do not care positions. Then the map is solved and the simplified terms are listed aside. All the simplified terms of the second group of variables are then checked and the terms are marked as before. The binary information of the third group of variables are then simplified for the common terms and the "#" marked terms, in the same manner as explained before. Similarly, all the groups of variables are simplified sequentially. Finally, the simplified Boolean expression is found by adding all the terms of the last column.

ables. If a term is found common in the first, fourth and 15th group of minterms then the zeroth, third and 14th position of the map is marked with one and being simplified. Simplified terms are listed aside. For a "#" marked term the groups of minterms are identified, where the loop of 1s which produces that term is common in position and structure. The higher bits are then simplified by using K-map.

If the term exist in group three and its loop of 1s matches with groups one and four then the second position of the K-map would be marked with one and the zeroth and third

Table 4. Determination of prime implicants

(a)	(b)	(c)	(d)
0000010 2 √	0000_10 (2,6)	00_1_00 (8,12,24,28)	00_11_ _
0001000 8 √	0001_00 (8,12) √	00_1_00 (8,12,24,28)	(12,13,14,15,28,29,30,31)
0000110 6 √	00_1000 (8,24) √	00011_ _ (12,13,14,15)	00_11_ _
0001100 12 √	000_110 (6,14)	√	(12,13,14,15,28,29,30,31)
0011000 24 √	000110_ (12,13) √	00011_ _ (12,13,14,15)	00_11_ _
	00011_0 (12,14) √	√	(12,13,14,15,28,29,30,31)
0001101 13 √	00_1100 (12,28) √	00_110_ (12,13,28,29)	
0001110 14 √	0011_00 (24,28) √	√	
0010101 21 √	0_11000 (24,56) √	00_110_ (12,13,28,29)	
0011100 28 √	00011_1 (13,15) √	√	
0110010 50 √	00_1101 (13,29) √	00_11_0 (12,14,28,30)	
0111000 56 √	000111_ (14,15) √	√	
	00_1110 (14,30) √	00_11_0 (12,14,28,30)	
	00101_1 (21,23) √	√	
	001_101 (21,29) √	0_11_00 (24,28,56,60)	
	001110_ (28,29) √	0_11_00 (24,28,56,60)	
	00111_0 (28,30) √	00_11_1 (13,15,29,31)	
	0_11100 (28,60) √	√	
0001111 15 √	_110010 (50,114)	00_11_1 (13,15,29,31)	
0010111 23 √	0111_00 (56,60) √	√	
0011101 29 √		00_111_ (14,15,30,31)	
0011110 30 √		√	
0111100 60 √		00_111_ (14,15,30,31)	
1110010 114 √		√	
		001_1_1 (21,23,29,31)	
0011111 31 √		001_1_1 (21,23,29,31)	
0110111 55 √	00_1111 (15,31) √	00111_ _ (28,29,30,31)	
0111101 61 √	001_111 (23,31) √	√	
	0_10111 (23,55)	00111_ _ (28,29,30,31)	
	00111_1 (29,31) √	√	
	0_11101 (29,61) √	0_1110_ (28,29,60,61)	
	001111_ (30,31) √	0_1110_ (28,29,60,61)	
	011110_ (60,61) √		

Table 5. Prime-implicants

Decimal	T U V W X Y Z	Terms
2,6	0 0 0 0 _ 1 0	T'U'V'W'YZ'
6,14	0 0 0 _ 1 1 0	T'U'V'XYZ'
50,114	_ 1 1 0 0 1 0	UVW'X'YZ'
23,55	0 _ 1 0 1 1 1	T'VW'XYZ
8,12,24,28	0 0 _ 1 _ 0 0	T'U'WY'Z'
24,28,56,60	0 _ 1 1 _ 0 0	T'VWY'Z'
21,23,29,31	0 0 1 _ 1 _ 1	T'U'VXZ
28,29,60,61	0 _ 1 1 1 0 _	T'VWXY'
12,13,14,15,28,29,30,31	0 0 _ 1 1 _ _	T'U'WX

3. Examples

The Map-Tabulation method is a combination of the K-map and the tabulation methods. Up to four variables can be used and it is exactly the same as the Karnaugh map method. When the number of variables increases it uses some techniques of the tabulation method. It can be used for problems involving any number of variables. This method is different for problems involving five or more input variables. Now we discuss the case for more than four variables.

Example 1:

Table 6. Selection of implicants

Implicants	2	6	8	12	13	14	15	21	23	24	28	29	30	31	50	55	56	60	61	114
T'U'V'W'YZ' √	×	×																		
T'U'V'XYZ'		×				×														
UVW'X'YZ' √															×					×
T'VW'XYZ √									×							×				
T'U'WY'Z' √			×	×						×	×									
T'VWY'Z' √										×	×						×	×		
T'U'VXZ √								×	×			×		×						
T'VWXY' √											×	×						×	×	
T'U'WX √				×	×	×	×				×	×	×	×						
	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

F (T, U, V, W, X, Y, Z) = Σ (2,6,8,12,13,14,15,21,23,24,28, 29,30,31,50,55,56,60,61,114)   [3]

Since there are seven input variables they should be grouped into two groups. The first group contains four least significant bits and rest of the variables will remain in the second group. The number of input variables are n=7. Therefore, the minterms would be grouped into $2^{(7-4)}$ = 8 groups. After grouping, the function would be reformed as follows:

$$F\left(\frac{T,U,V,}{2}\middle|\frac{W,X,Y,Z}{1}\right)=\sum\left(\frac{2,6,8,12,13,14,15,}{1}\middle|\frac{21,23,24,28,29,30,31,}{2}\middle|_3\middle|\frac{50,55,56,60,61}{4}\middle|_5\middle|_6\middle|_7\middle|\frac{114}{8}\right)$$ [4]

Now all the information of this function should be placed into the simplification table as shown in Table 2. At first, the binary information of the first group of minterms and first group of input variables are placed into a 4-variable K-map. Combining the 1s according to the usual procedure, the simplified terms W'YZ', WX and WY'Z' are found and are listed aside. Similarly, with the rest of the combinations of W, X, Y and Z, all the other groups of minterms are simplified accordingly and their results are listed. These simplified terms are compared. Here the term WX is found at row one and two. Therefore check is placed at the right of both terms. Similarly, all other common terms are marked. Now the K-maps are checked for the uncommon terms. The uncommon terms W'YZ' is generated by combining the 1s of the second and sixth positions of the first K-map. No other K-map is found where '1s' exists in the same locations. Therefore W'YZ' along with the higher significant bits of the same group (T'U'V') are written at the column of selected implicants. Similarly, XZ along with T'U'V is placed in the last column. The uncommon term W'XYZ is slightly different. This term is generated by the one of seventh position of the K-map of fourth row. The K-map of the second row also contains a one at the same location. Therefore the symbol "#" is placed on the right of this term. Similarly

the term WXY' is generated by the '1s' of eight and ninth position of the K-map of fourth row. The K-maps of first and second rows also contain '1s' at the same location. Therefore WXY' is also marked with # on the right. For the common terms and the terms marked with #, the binary information of the second group of input variables are simplified. Since the second group of input variables contains three variables, three-variable K-map is used for the simplification. The first term WX is common at the first and second rows, where the value of TUV is 000 and 001 respectively. That is why, 1s are placed at the zeroth and first positions of the map and is solved accordingly. The simplified higher term found from the map is T'U'. So the ultimate simplified term is T'U'WX, and is listed aside. Similarly, all other common terms are listed along with their simplified higher terms. Now the higher significant bits are simplified for the uncommon terms marked with # symbol. The first such term is W'XYZ, and is generated by the one placed at the seventh position of the K-map of fourth row. The seventh position of the K-map of the second row also contains one. These two 1s of the same location of different maps are marked with a circular boundary. At the K-map of higher significant bits the third position is marked with one and the first position is marked as a do not-care location. For simplification, this location can be treated as one or zero. The result of higher bits simplification is T'V. Therefore, finally T'VW'XYZ is written in the last column. Similarly, simplification of higher bits is performed for the other # marked term, and the result is listed. Finally the following simplified terms are found; T'U'V'W'YZ', T'U'VXZ, T'U'WX, T'U'WY'Z', T'VWY'Z', UVW'X'YZ', T'VW'XYZ and T'VWXY'. By adding these terms we get the ultimate simplified function as follows;

F = T'U'V'W'YZ' + T'U'VXZ + T'U'WX + T'U'WY'Z' + T'VWY'Z' + UVW'X'YZ' + T'VW'XYZ + T'VWXY'   [5]

Example 2:

This example is illustrated in DSA method (Rhyne et al., 1977) as,

F (a, b, c, d, e, f, g, h, i) = Σ
(6,7,22,23,38,70,102,134,262)   [6]

Solution

$$F\left(\frac{a,}{3}\left|\frac{b,c,d,e}{2}\right|\frac{f,g,h,i}{1}\right) = \sum \left(\frac{6,7}{1}\left|\frac{22,23,}{2}\right|\frac{38,}{3}\right.$$

$$\left|\frac{*}{4}\right|\frac{70,}{5}\left|\frac{*}{6}\right|\frac{102,}{7}\left|\frac{*}{8}\right|\frac{134,}{9}\left|\frac{*}{10}\right|\frac{*}{11}\left|\frac{*}{12}\right|\frac{*}{13}\left|\frac{*}{14}\right|\frac{*}{15}\left|\frac{*}{16}\right|\frac{262}{17}$$ [7]

Therefore F (a,b,c,d,e,f,g,h,i) = a'b'c'd'f 'gh + a'b'e'f 'gh
 + a'c'd'e'f 'gh + b'c'd'e'f 'ghi'   [8]

4. Verification Of Results

Example 1 outlined above culminates in a simplified Boolean expression. The accuracy of this result can be verified by using other methods of simplification. The practical use of the K-map method is limited to up to six variable functions. But the given function contains seven input variables. Therefore the K-map method will not be convenient for this particular problem. The Quine-McCluskey's tabulation can serve the purpose though it is a monotonous method. The simplification of the given function F (T,U,V,W,X,Y,Z) = Σ (2, 6, 8, 12, 13, 14, 15, 21, 23, 24, 28, 29, 30, 31, 50, 55, 56, 60, 61, 114) , by using the tabulation method is shown in Tables 4-6.

The simplified expression is:

F=T'U'V'W'YZ'+UVW'X'YZ'+T'VW'XYZ +T'U'WY'Z'+T'VWY'Z'+T'U'VXZ+ T'VWXY'+T'U'WX   [9]

This proves that the result obtained earlier, by using the proposed method is correct.

5. Benefits

Our proposed technique is a manual method of simplification of Boolean functions. Therefore it should not be compared with computer aided minimization techniques. CAMP (Biswas, 1986) is completely a computer aided minimization technique. Modified Espresso (Malik et al., 1988), Espresso (Brayton et al., 1985) and MINI (Hong et al., 1974) are more suitable for computer aided minimization than manual minimization. If we consider the manual minimization technique described by Biswas (1971) we will find that the method is almost similar to the Quine-McCluskey's Tabulation method. In this method huge amounts of bit-by-bit comparisons take place, which is quit monotonous and is prone to mistakes. In our proposed method no such bit-by-bit comparison needs to be per-

formed. Now let us consider the Directed Search Algorithm (DSA) (Rhyne et al., 1977). This method can be used as a manual method as well as a computer aided minimization method. It uses the RED-tree procedure. For manual minimization this RED-tree procedure is suitable for functions of five to eight variables. Our proposed method can easily handle functions having more than eight variables.

6. Conclusions

This paper studied the problems of minimizing Boolean expressions and proposed an optimal method of Boolean function simplification. It also evaluated the performance of a new technique. This method overcomes most of the limitations of the other renowned manual methods of simplification. It can be effectively and easily implemented for problems having large numbers of input variables. Complexities arise when the number of minterms for which $F(m_i) = 1$ increase excessively. It is less dependent on the ability of the human user, because it uses a maximum of four variable K-maps. Moreover, it is not as monotonous as the other manual methods. Further research should be conducted to develop an algorithm that uses this new method.

References

Biswas, N. N., 1986, "Computer-Aided Minimization Procedure for Boolean Functions," IEEE Trans. On Computer Aided Design, Vol. CAD-5, no. 2, pp. 303-304.

Biswas, N. N., 1971, "Minimization of Boolean Functions," IEEE Trans. On Computers, pp. 925-929.

Brayton, R. K., Hachtel, G. D., McMullen, C. T., and Sangiovanni-Vincentelli, A. L., (1985), "Logic Minimization Algorithems for VLSI Synthesis," Kluver Academic Publishers.

Hong, S. J., Gain, R. G., and Ostapko, D. L., 1974, "MINI: A Heuristic Approach for Logic Minimization," IBM J. Res. Develop, Vol. 18, pp. 443-458.

Karnaugh, M., 1953, "The Map Method for Synthesis of Combinational Logic Circuits," Trans. AIEE. pt I, vol. 72, no. 9, pp. 593-599.

Malik, A. A., Robert, K., Brayton, A., Newton., R., Alberto, L., and Vincentelli, S., 1988 "A Modified Approach to Two

Mano, M. M., 1979, "Digital Logic and Computer Designm" Prentice-Hall: U.S.A, pp. 72-83,102-110.

Rhyne, V. T., Noe, P. S., McKinney, M. H., and Pooch, U. W, 1977, "A New Technique for the first Minimization of Switching Functions," IEEE Trans. On Computers, Vol. C-26, No. 8, pp. 757-764.