SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348 – 8549 /doi:10.14445/23488549/IJECE-V8I1P102

Volume 8 Issue 1, 6-11, January 2021
© 2021 Seventh Sense Research Group®

A Comparison of Two Methods for Realizing Minimal Function of Several Logic Variables

T. S. Rathore* and K.S. Sanila#

*Independent Researcher, G-803, Country Park, Dattapada Road, Borivali, Mumbai, India
#School of Electrical Sciences, IIT Goa, Farmagudi, Ponda, Goa 403401, India

Abstract *- It is shown, through two examples, that Rathore's method for finding the minimal realization of a logic function of several variables is simpler and straight forward than Prasad's method. The former has a distinct advantage of giving a complete set of minimal realizations at one stretch over the latter method.*

Keywords: *Boolean functions, Karnaugh map, Minimal realization, Prasad's method, Rathore's method*

I. INTRODUCTION

There has been a lot of interest in minimizing a logic function to its minimal form [1-7]. Recently, Prasad [6] has proposed a method for reducing the logic function of many variables. Essentially, he obtains the minimal function in two major steps. In the first step, a complete set of all prime implicants is obtained, consisting of several sub-steps. Many K-maps of size 4×4 [8] are to be visualized. Many concepts, such as 'prime implicants,' '*K*-don't cares,' tree, leaf, parents, children, etc., are used. Therefore, it is quite lengthy and involved. In the second step, a minimal realization is obtained using some other method [9]-[12]. The author seems to be unaware of Rathore's method (RM) [7] on similar work, as he does not refer to it. In the present work, a comparison of Prasad's method (PM) and RM is given. It is shown that the latter method is straight-forward and simpler than the former one. This is shown by reducing two functions to their minimal forms. One function is given in the example, 2.1 of [6], and the other is taken from [7].

II. EXAMPLES

2.1 Example 1:

Find the minimal realization of the following function f_a, taken from [6], using RM [7].

$$f_a(x_1,x_2,x_3,x_4,x_5,x_6,x_7) = x_1x_2x_3x_4 + x_4x_5x_6x_7$$
$$+x_2'x_3'x_4'x_5 + x_1x_4x_5x_7 + x_1'x_2x_4x_5'x_6'.$$
$$= \begin{cases} (4,5,6,7,15,31,40,41,47,56,57,63,68, \\ 69,70,71,77,79,93,95,109,111,120,121, \\ 122,123,124,125,126,127) \end{cases} \quad (1)$$

To save space, only the first 32 minterms of f_a are shown in Fig.1 and its reduction process to arrive at f_{a1}. Reaming minterms are treated in a similar way to obtain f_{a2}, f_{a3}, f_{a4}. For convenience, a long table in the horizontal direction is divided into 2 Tables, as shown in Fig. 1(a) and 1(b). The final minimal function is shown as f in the last column of Fig. 1(b).

Elimination of LSB from one column to the next column is achieved using a truth table shown in Fig. 2 or relation in (2), or K-map of size 2×1 [8], repeatedly.

$$f = x'y + xz \quad (2)$$

Note that in each subsequent column, the number of rows is reduced by 50%. Only *K*-map of size 2×1 for 1 variable is used repeatedly. Note from (2), if $y = z = w$ then $f = w$. This reduces the number of literals. Hence, one must use this property wherever possible. The following Boolean identities may help to make $y = z$.

$$x + x' = 1 \quad (3)$$
$$1 = 1 + x \quad (4)$$
$$x + y = x\,y' + y = x + x'y \quad (5)$$

For example, in Fig. 1(a), each pair of two rows from top to bottom of column 2 is replaced using (2) in the next column. This eliminates the variable x_7. Equation (5) may give additional minimal realizations.

If there are 2^n (n is an integer) entries in a block are 0 (1), then subsequent n columns will have all entries 0 (1). Such entries can be filled first in the entire table, then proceed to fill other entries as per the procedure suggested in [7] till only one row is left. This makes the method faster. For example, in Example 1, the entries corresponding to decimal numbers 0-3, 8-11, 16-19, 24-27 are 0, and 4-7 are filled in first. Thus, we need not handle all the minterms.

There are following two trivial cases where one would not like to go for the simplification method.

x	f
0	y
1	z

Fig. 2 Truth table or K-map of size 2×1.

No	$f_a(x_1x_2x_3x_4x_5x_6x_7)$	$f_a(x_1x_2x_3x_4x_5x_6)$	$f_a(x_1x_2x_3x_4x_5)$	$f_a(x_1x_2x_3x_4)$	$f_a(x_1x_2x_3)$	$f_a(x_1x_2)$
0	0	0	0	x_5	$x_4'x_5 + x_4x_5x_6x_7$	$f_{a1} = x_3'x_4'x_5 + x_4x_5x_6x_7$
1	0					
2	0	0				
3	0					
4	1	1	1			
5	1					
6	1	1				
7	1					
8	0	0	0	$x_5x_6x_7$		
9	0					
10	0	0				
11	0					
12	0	0	x_6x_7			
13	0					
14	0	x_7				
15	1					
16	0	0	0	0	$x_4x_5x_6x_7$	
17	0					
18	0	0				
19	0					
20	0	0	0			
21	0					
22	0	0				
23	0					
24	0	0	0	$x_5x_6x_7$		
25	0					
26	0	0				
27	0					
28	0	0	x_6x_7			
29	0					
30	0	x_7				
31	1					

(a)

$f_a(x_1\ x_2)$	$f_a(x_1)$	f
$f_{a1} = x_3'x_4'x_5 + x_4x_5x_6x_7$	$x_4x_5x_6x_7 + x_2'x_3'x_4'x_5 + x_2x_4x_5'x_6'$	$x_1x_4x_5x_7 + x_4x_5x_6x_7 + x_1x_2x_3x_4 + x_2'x_3'x_4'x_5 + x_1'x_2x_4x_5'x_6'$
$f_{a2} = x_4x_5'x_6' + x_4x_5x_6x_7$		
$f_{a3} = x_3'x_4'x_5 + x_4x_5x_7$	$x_2'x_3'x_4'x_5 + x_2x_3x_4 + x_4x_5x_7$	
$f_{a4} = x_4x_5x_7 + x_3x_4$		

(b)

Fig. 1 Minimal realization of f_a using RM (a) First 7 columns and (b) remaining three columns

Case 1: A function has a single minterm. This itself is the simplified form.
Case 2: If there is an *n* variable function of two minterms such that all the literals are identical except the one which appears as compliments in the two min terms¸ then one can use the symmetry or repetition property [7] and get the simplification easily as any one of the minterms without this literal.

In RM, there exists an option to switch over to the 4×4 K map reduction process when 4 LSB are left. For example, 16 minterms are obtained from a column of f_a $(x_1x_2x_3x_4)$ for all the 128 minterms and filled in the 4×4 K map, as shown in Fig. 3. Now routine simplification will yield

$f = x_1x_4x_5x_7 + x_4x_5x_6x_7 + x_1x_2x_3x_4 + x_2'x_3'x_4'x_5 + x_1'x_2x_4x_5'x_6'$

which is the same as given in the last column of Table 1(b).

$x_3x_4 \rightarrow$ $x_1x_2 \downarrow$	00	01	11	10
00	x_5	$x_5x_6x_7$	$x_5x_6x_7$	0
01	0	$x_5x_6x_7 + x_6'x_7'$	$x_5x_6x_7 + x_6'x_7'$	0
11	0	x_5x_7	1	0
10	x_5	x_5x_7	x_5x_7	0

Fig. 3 K map for example 1

Prasad's revised method (PRM) [6] is used to eliminate some prime implicants obtained using the flowchart for getting minimal expression. This method proceeds through the following steps.

1. *K*-don't cares replaced with a single *K*.
2. A child node stops progressing if it doesn't have at least one minterm of value 1.

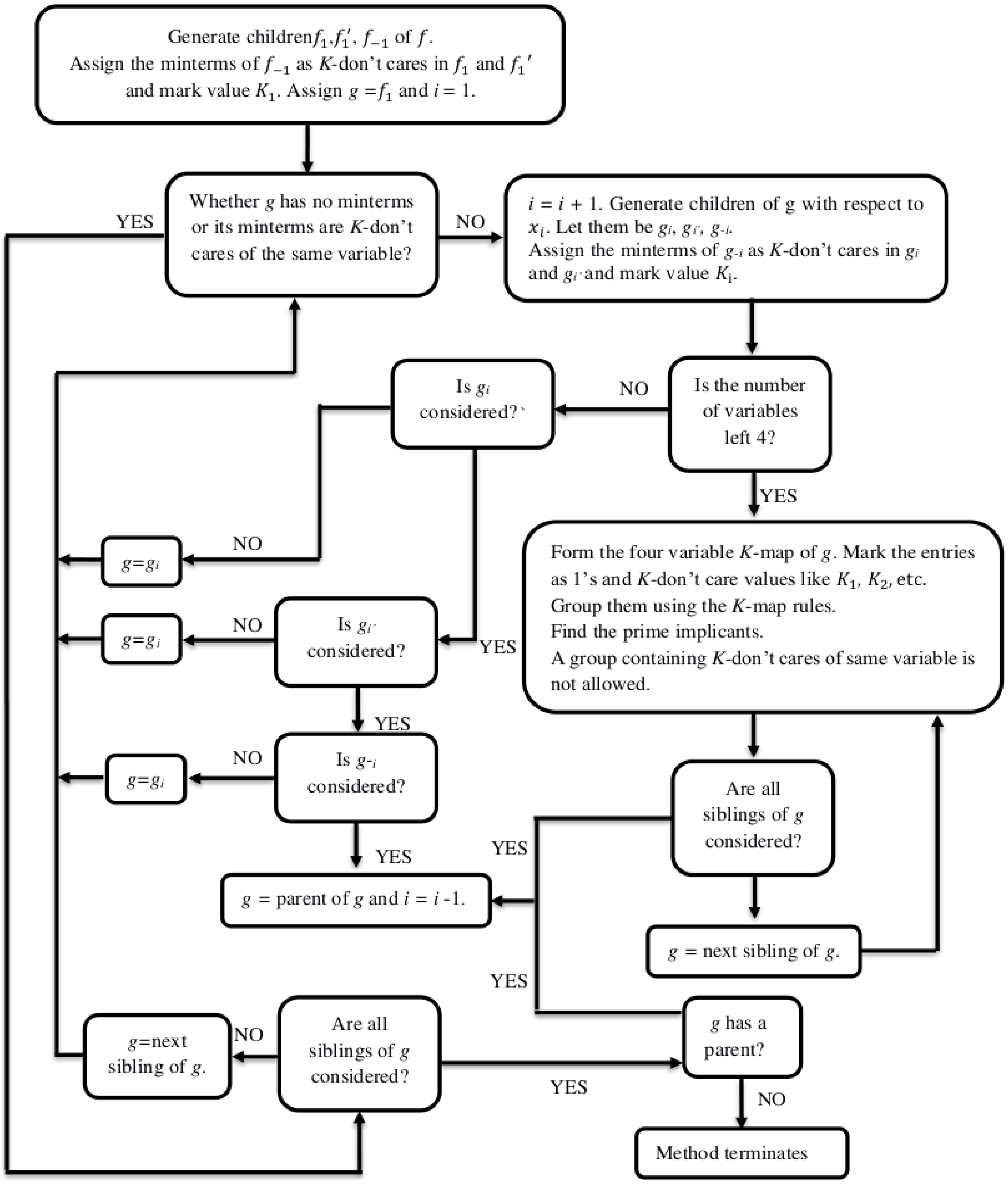
Generate children f_1, f_1', f_{-1} of f.
Assign the minterms of f_{-1} as K-don't cares in f_1 and f_1' and mark value K_1. Assign $g = f_1$ and $i = 1$.
Whether g has no minterms or its minterms are K-don't cares of the same variable?
YES
NO
$i = i + 1$. Generate children of g with respect to x_i. Let them be g_i, $g_{i'}$, g_{-i}.
Assign the minterms of g_{-i} as K-don't cares in g_i and $g_{i'}$ and mark value K_i.
Is g_i considered?
NO
Is the number of variables left 4?
YES
NO
$g=g_i$
Form the four variable K-map of g. Mark the entries as 1's and K-don't care values like K_1, K_2, etc.
Group them using the K-map rules.
Find the prime implicants.
A group containing K-don't cares of same variable is not allowed.
NO
$g=g_i$
Is $g_{i'}$ considered?
YES
YES
NO
$g=g_i$
Is g_{-i} considered?
Are all siblings of g considered?
YES
YES
g = parent of g and $i = i$ -1.
g = next sibling of g.
YES
g=next sibling of g.
NO
Are all siblings of g considered?
g has a parent?
YES
NO
Method terminates

Fig. 4 A general flowchart for PM

3. While generating prime implicants, it should cover at least one 1 and a new 1 that is not present in other prime implicants generated so far.

2.2 Example 2:

Let the function to be minimized by

$$f_b = \sum(0,1,3,4,7,13,15,19,20,22,23,29,31) \qquad (6)$$

The reduction of f_b using RM is given in Fig. 5. From Fig. 5(a), there have to be 9 possible outputs, but they all reduce to 4 only, as shown in Fig. 5(b). If the function given itself is minimal, the method gives all the minimal functions, including the one given. Thus, the RM provides an answer to the question: How many minimal realizations are possible for a given function, and what are they?

	f_b	$f_b(x_1x_2x_3x_4)$	$f_b(x_1x_2x_3)$	$f_b(x_1x_2)$	$f_b(x_1)$
0	1	$1 + x_5$	$x_5 + x_5'x_4' = x_4' + x_4x_5$	$x_4'x_5' + x_4x_5 + x_3'x_5$ --- (i) OR $x_4'x_5' + x_4x_5 + x_3'x_4'$ --- (ii) $= x_3x_4'x_5' + x_4x_5 + x_3'x_4'$ ---(iii)	$x_2'x_4'x_5' + x_2'x_4x_5 + x_2'x_3'x_5 + x_2x_3x_5$ --- (a) OR $x_2'x_4x_5 + x_2'x_4'x_5' + x_2'x_3'x_4' + x_2x_3x_5$ --- (b) OR $x_2'x_4x_5 + x_2'x_3'x_4' + x_2'x_3x_4'x_5' + x_2x_3x_5$ --- (c)
1	1				
2	0	x_5			
3	1				
4	1	x_5'	$x_4'x_5' + x_4x_5$		
5	0				
6	0	x_5			
7	1				
8	0	0	0	x_3x_5	
9	0				
10	0	0			
11	0				
12	0	x_5	x_5		
13	1				
14	0	x_5			
15	1				
16	0	0	x_4x_5	$x_4x_5 + x_3x_5'$ --- (i) OR $x_4x_5 + x_3x_4 + x_3x_4'x_5'$ --- (ii) OR $x_4x_5 + x_3x_4x_5' + x_3x_4'x_5'$ $= x_4x_5 + x_3x_5 + x_3x_4'x_5'$ ---(iii)	$x_2'x_4x_5 + x_2'x_3x_5' + x_2x_3x_5$ --- (d) OR $x_2'x_4x_5 + x_2'x_3x_4'x_5' + x_2'x_3x_4 + x_2x_3x_5$ --- (e) OR $x_2'x_4x_5 + x_2'x_3x_5' + x_2'x_3x_4'x_5' + x_2x_3x_5$ --- (f)
17	0				
18	0	x_5			
19	1				
20	1	x_5'	$x_5' + x_5x_4 = x_4 + x_4'x_5'$		
21	0				
22	1	$1 + x_5'$			
23	1				
24	0	0	0	x_3x_5	
25	0				
26	0	0			
27	0				
28	0	x_5	x_5		
29	1				
30	0	x_5			
31	1				

(a)

	Minimal Expressions for f_b
(a) & (d), (a) & (e)	$f_{b,min1} = x_2'x_4x_5 + x_2x_3x_5 + x_1'x_2'x_4'x_5' + x_1'x_2'x_3'x_5 + x_1x_2'x_3x_5'$
(b) & (d)	$f_{b,min2} = x_2'x_4x_5 + x_2x_3x_5 + x_1'x_2'x_4'x_5' + x_1'x_2'x_3'x_4' + x_1x_2'x_3x_5'$
(c) & (e), (b) & (e)	$f_{b,min3} = x_2'x_4x_5 + x_2x_3x_5 + x_2'x_3x_4'x_5' + x_1'x_2'x_3'x_4' + x_1x_2'x_3x_4$
(a) & (f), (b) & (f), (c) &(d), (c) & (f)	$f_{b,min4} = x_2'x_4x_5 + x_2x_3x_5 + x_2'x_3x_4'x_5' + x_1'x_2'x_3'x_4' + x_1x_2'x_3x_5'$

(b)

Fig. 5 Minimal realization of f_b using RM

Method	Number of variables	Number of the level of computations	Number of 4x4 K-map reductions	Remarks
PM	7	13	27	One additional step is required for PRM
RM		8	0	2×1 K-map reductions are considered in the number of level of computations
PM	5	1	3	Only one minimal expression is obtained after PRM
RM		6	0	All minimal expressions are obtained.

Fig. 6 Computational analysis of f_a and f_b using PM and RM.

Reduction of f_b using PM gives 8 prime implicants:

$$P_1 = x_2'x_4x_5, P_2 = x_2x_3x_5, P_3 = x_1'x_2'x_3'x_5,$$
$$P_4 = x_2'x_3x_4'x_5', P_5 = x_1'x_2'x_4'x_5'\ P_6 = x_1x_2'x_3x_5',$$
$$P_7 = x_1'x_2'x_3'x_4', P_8 = x_1x_2'x_3x_4.$$

It can be further reduced by eliminating P_3, P_5, P_6 since it doesn't group a new one that is not present in the prime implicants generated so far. Therefore, minimal expression of f_b can be written as

$$f_{b,\min} = x_2'x_4x_5 + x_2x_3x_5 + x_2'x_3x_4'x_5' + x_1'x_2'x_3'x_4' + x_1x_2'x_3x_4 \quad (7)$$

III. COMPARISON

1. In RM, all the product terms in the last column, such as in Fig. 1(b), are prime implicants and form a minimal function. Not a single extra prime implicant is generated. While in PM, all of the prime implicants are determined first and then Eliminate those which are in excess to form the minimal function.
2. Relations have been given by (3), (4), and (5), with which readers are already familiar, are used. So, one can easily adapt the method without involving any new concepts. While concepts such as 'prime implicants,' 'K-don't cares,' 'tree,' 'leaf,' 'parent nodes,' 'children,' 'grand-children,' etc. are involved in PM.
3. In PM, a K-map of size 4×4 is used, while in RM, a K-map of size 2×1 for 1 variable, as shown in Fig. 2, is used. Thus, it does not involve a higher K-maps size and, therefore, reduces the effort considerably.
4. In PM, all the prime implicants are found using the generalized K map method. Further, using PRM, some prime implicants are eliminated for getting one minimal function. In RM, the required prime implicants are automatically obtained and therefore no Need to reduce them to form minimal expression. Also, by writing $x + y$ in two possible ways ($x + x'y$ or $xy' + y$), we automatically get the complete set of all the minimal implicants as in Fig. 5 (b). While PM yields only one minimal expression given in (8).
5. The number of levels of computations required to solve an 'n' variable Boolean function in PM, obtained from the flow chart, is

$$N_L = 3^0 + 3^1 + 3^2 + 3^3 + \ldots + 3^{n-5}$$
$$= \frac{3^{n-4} - 1}{2} \quad (7)$$

where $3^0, \cdots, 3^{n-5}$ represent the number of sets of children that need to be considered at each level of the tree. The numeral '3' signifies the number of children for a parent. While RM requires only (n+1) computational levels comprising 2×1 K-map simplifications. Thus, PM requires a larger N_L than RM for $n > 6$. Apart from this, in PM 3^{n-4} number of K-map of size 4×4 needs to be solved to find all the prime implicants. Computational analysis of f_a and f_b is given in Fig. 6.

IV. CONCLUSION

Through two examples, it is shown that RM for reducing a logic function of several variables to its minimal form is straight-forward and simpler than PM. Also, in the former method, a complete set of all the minimal expressions are found simultaneously. This is a distinct advantage of RM over PM.

REFERENCES

[1] Ali M. Rushdi and H. A. Al-Yahya, Derivation of the complete sum of a switching function with the aid of the variable-entered Karnaugh map, Journal of King Saud University: Engineering Sciences. 13(2)(2001), 239-269.

[2] Ali M. Rushdi and H. A. Al-Yahya, Variable-entered Karnaugh map procedures for obtaining the irredundant Disjunctive forms of a switching function from its Complete sum, Journal of King Saud University, Engineering Sciences, 14(1)(2002) 13-27.

[3] A. M. Rushdi, Improved variable-entered Karnaugh map procedures, Computer and Electrical Engineering, 13(1)(1987), 41-52..

[4] A. M. Rushdi & Al-Yahya, H. A., Further improved variable-entered Karnaugh map procedures for obtaining the irredundant forms of an incompletely-specified switching function, Journal of King Abdulaziz University: Engineering Sciences, 13(1)(2001), 111-152.

[5] A. M. Rushdi, Utilization of Karnaugh maps in multi-value qualitative comparative analysis, International Journal of Mathematical, Engineering and Management Sciences. 3(1)(2017), 28-46.

[6] V. C. Prasad, Generalized Karnaugh Map Method for Boolean Functions of Many Variables, IETE J. of Education, (2017). https://doi.org/10.1080/09747338.2017.1293568.

[7] T. S. Rathore, Minimal realizations of logic functions using truth table method with distributed simplification, IETE J. Education, 55 (1)(2014), 26-32.

[8] T. S. Rathore, A Note on the Size of a Karnaugh Map, IETE J. Education, 57(2)(2016). https://DOI: 10.1080/09747338.2016.1261048.

[9] M. P. Marcus, Switching Circuits for Engineers. Englewood Cliffs, NJ: Prentice-Hall, (1969).
[10] A. B. Marcovitz, Introduction to Logic Design. New York: International Edition, McGraw–Hill, (2002).
[11] D. D. Givone, Digital Principles, and design, New York: McGraw-Hill, (2003).
[12] Z. Kohavi and N. K. Jha, Switching and Finite Automata Theory. New York: Cambridge University Press, (2010). Available: http://www.cambridge.org/9780521857482.