International Journal of Electronics
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713599654

Optimization of Reed-Muller logic functions

L. McKENZIE [a]; A. E. A. Almaini [a]; J. F. Miller [a]; P. Thomson [a]
[a] Department of Electrical, Electronic and Computer Engineering, Napier University, Edinburgh, U.K.

Online Publication Date: 01 September 1993
To cite this Article: McKENZIE, L., Almaini, A. E. A., Miller, J. F. and Thomson, P. (1993) 'Optimization of Reed-Muller logic functions', International Journal of Electronics, 75:3, 451 - 466
To link to this article: DOI: 10.1080/00207219308907124
URL: http://dx.doi.org/10.1080/00207219308907124

INT. J. ELECTRONICS, 1993, VOL. 75, NO. 3, 451-466

Optimization of Reed-Muller logic functions

L. McKENZIE†, A. E. A. ALMAINI†, J. F. MILLER†
and P. THOMSON†

Two algorithms are presented, the first is a technique to determine good, though not necessarily optimum, fixed polarity Reed-Muller expansions of completely specified boolean functions. The second algorithm determines the allocations of the 'don't care' terms of incompletely specified boolean functions resulting in optimum positive polarity Reed-Muller expansions. Additionally, investigations are made into combining these two techniques to determine fixed polarity Reed-Muller expansions of incompletely specified boolean functions. Results are presented which show the effectiveness of the techniques and comparisons are made with existing methods.

1. Introduction

There are certain advantages in using Reed-Muller (RM) expansions to represent switching functions (Besslich 1983, Wu *et al.* 1982). Firstly, functions realized using AND and EXOR gates and hence based on RM expansions, lead to well-structured and easily tested designs. Secondly, investigations have shown that switching functions which do not produce efficient solutions in the boolean domain can often be realized efficiently in the RM domain. Further, many of the techniques of linear algebra, such as matrix and transform operations, may be applied to expressions in the field GF(2).

For each function of n-input variables there exist 2^n fixed polarity RM expansions each with a particular cost, therefore, determining the optimum fixed polarity expansion of a switching function is a non-trivial task. It is possible to perform exhaustive searches to identify optimum expansions of functions with small numbers of input variables. However, as the number of input variables increases, the time and memory allocations required for searches become impractical, even when techniques such as those developed by Besslich (1983) and, more recently, Harking (1990) are employed. At present several non-exhaustive methods exist (Almaini *et al.* 1991, Habib 1990, Wu *et al.* 1982) which determine minimal fixed polarity RM expansions of functions. These techniques do not guarantee that the solution produced is the optimum expansion. This paper describes an improved method for determining minimal fixed polarity RM expansions of functions, without performing exhaustive searches (§2). The method is based on the Tabular Technique (Almaini *et al.* 1991).

Additionally, a non-exhaustive technique is introduced which determines the values of the 'don't care' terms of incompletely specified functions, resulting in optimum zero polarity RM expansions (§3). It is also possible to utilize this algorithm together with the good polarity algorithm to realize good, though not necessarily optimum, Generalized Reed-Muller expansions of incompletely specified functions (§4).

Received 20 November 1992; accepted 23 March 1993.

†Department of Electrical, Electronic and Computer Engineering, Napier University, 219 Colinton Road, Edinburgh, EH14 1DJ, U.K.

2. Gains method

It is, perhaps, appropriate to identify the strengths and weaknesses of the Tabular Technique (TT) (Almaini *et al.* 1991) before detailing the measures taken to improve the quality of the solutions it produces, and so develop the new algorithm.

2.1. *Advantages of TT optimization algorithm*

(*a*) The algorithm provides a good solution within a reasonable time scale.

(*b*) The maximum number of iterations of the algorithm is equal to the number of function variables (n).

As a function variable may only be complemented, and may not be returned to its true state, the number of variables which must be evaluated in the algorithm is reduced by one after each iteration.

2.2. *Disadvantages of TT optimization algorithm*

(*a*) Where the algorithm determines that complementing more than one variable will cause the same reduction in the number of terms in the function, the algorithm arbitrarily chooses a variable. This choice obviously affects the quality of the solution.

(*b*) The algorithm will cease when the number of terms in the function can no longer be reduced by complementing variables. However, the situation may arise where complementing variables alters the structure of the function whilst the number of terms in the function is unchanged. This allows new functions to be evaluated, possibly leading to a more minimal solution.

(*c*) If a variable has been complemented, it cannot be returned to its true state. This eliminates many polarities from the search.

(*d*) The algorithm is biased towards polarity zero as the starting function for the optimization procedure is the zero polarity RM expansion. If the optimum polarity has many complemented variables then many iterations of the algorithm need to be performed thus increasing both the possibility of locating a local rather than a global minimum and the time taken to form a solution.

(*e*) The algorithm will locate and cease on local minima and cannot determine whether a global minimum exists.

The following modifications were implemented in order to improve the quality of the solutions determined by the optimization algorithm.

(i) Introduction of a branching mechanism. If the results of the evaluation procedure indicate that complementing more than one variable would cause the same reduction in the number of terms in the function, then separate functions are generated for each complemented variable.

(ii) A variable may be complemented if it does not alter the number of terms in the function. This transformation will not reduce the number of terms in the function but allows the newly generated function to be evaluated, and generally leads to further iterations of the algorithm.

(iii) Allowing complemented variables to be transformed to their true state in order to improve the quality of the solution obtained. A variable which has been complemented during an early iteration of the algorithm may further reduce the number of terms in the function if returned to its true state.

These modifications address the first three 'disadvantages' of the TT optimization algorithm. The new algorithm is now given.

2.3. *Gains optimization algorithm*

Step 1. Convert the boolean function to zero (positive) polarity RM expansion. The RM function is inserted as the first element in the best polarity list (BPL).

Step 2. For each variable of the function ($x_n \ldots x_1$), and for each function in the BPL, determine the number of pairs of terms which are adjacent in that variable.

Step 3. For each function in the BPL, determine, for each function separately, the number of new terms generated by complementing each function variable in turn (equivalent to counting the number of occurrences in the function of each variable in its true form).

Step 4. For each function in the BPL, calculate whether complementing each variable will increase or decrease the number of terms in the function.

For each variable, $x_n, \ldots x_1$

score_x_j = number of true occurrences of x_j − 2 × (number of pairs of terms adjacent in x_j)

If score_$x_j < 0$, terms will be lost from the function.

If score_$x_j > 0$, terms will be added to the function.

If score_$x_j = 0$, no change in number of terms in the function.

Step 5. Considering all functions in the BPL, find the score_x_j giving the greatest decrease possible in the number of terms in the functions. Let this equal minscore. If minscore > 0 then Step 8, else Step 6.

Step 6. For all functions in the BPL, find functions having variables x_j, with score_x_j = minscore. Generate new functions for each variable x_j, each new function having x_j complemented.

If minscore < 0, then remove all functions from the BPL. Insert new functions into the BPL, removing any duplicates.

If minscore $= 0$, then insert into the BPL any new functions not currently contained in the list.

Step 7. If the BPL contains new functions then go to Step 2, repeat the algorithm for each new function. Otherwise, Step 8.

Step 8. The algorithm determines the fixed polarity RM expansion(s) contained in the BPL to be best (least number of product terms). Optimization process ends.

The following example illustrates the use of the above algorithm in minimizing a function in the Reed–Muller domain.

2.3.1. *Example* 1: *Boolean function* $f(x_5,x_4,x_3,x_2,x_1)=\Sigma m(0,1,7,9,15,16,17,18,20,21,22,25,29)$

Step 1. The function is transformed to the equivalent polarity 0 RM expansion.

Boolean function / RM function (Polarity 0)

x_5	x_4	x_3	x_2	x_1	x_5	x_4	x_3	x_2	x_1	Piterms
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	2
0	0	1	1	1	0	0	1	0	0	4
0	1	0	0	1	0	0	1	1	0	6
0	1	1	1	1	0	0	1	1	1	7
1	0	0	0	0	0	1	0	0	0	8
1	0	0	0	1	0	1	0	0	1	9
1	0	0	1	0	0	1	0	1	0	10
1	0	1	0	0	0	1	0	1	1	11
1	0	1	0	1	0	1	1	0	0	12
1	0	1	1	0	0	1	1	0	1	13
1	1	0	0	1	0	1	1	1	0	14
1	1	1	0	1	0	1	1	1	1	15
					1	0	0	1	0	18
					1	0	0	1	1	19
					1	0	1	0	0	20
					1	0	1	1	0	22
					1	0	1	1	1	23
					1	1	0	1	0	26
					1	1	0	1	1	27
					1	1	1	0	0	28
					1	1	1	0	1	29
					1	1	1	1	0	30
					1	1	1	1	1	31

Best polarity list = (RM expansion: polarity 0)

Step 2. For each variable, find the number of pairs of adjacent terms.

x_5	x_4	x_3	x_2	x_1
2,18	0, 8	0, 4	0, 2	8, 9
4,20	2,10	2, 6	4, 6	6, 7
6,22	4,12	8,12	8,10	10,11
10,26	6,14	9,13	9,11	12,13
12,28	18,26	10,14	12,14	18,19
7,23	20,28	18,22	20,22	14,15
11,27	7,15	11,15	13,15	22,23
13,29	19,27	19,23	28,30	26,27
14,30	22,30	26,30	29,31	28,29
15,31	23,31	27,31	—	30,31
10	10	10	9	10

Step 3. Determine the number of true occurrences of each variable

x_5	x_4	x_3	x_2	x_1
11	14	14	15	10

Step 4. For each variable $x_j (j=1\text{–}5)$,
find score_x_j = number of true occurrences of $x_j - 2 \times$ (number of pairs of terms adjacent in x_j)

	x_5	x_4	x_3	x_2	x_1
	11	14	14	15	10
	-2×10	-2×10	-2×10	-2×9	-2×10
score_x_j	-9	-6	-6	-3	-10

Step 5. minscore $= -10$.

Step 6. score_x_1 = minscore.

Generate new RM expansion with variable x_1 complemented—polarity 1 RM expansion. BPL = (RM expansion: polarity 1).

Step 7. Repeat Steps 2–6 for the new function.

Steps 2–4. For each variable, find the number of pairs of adjacent terms and the number of true occurrences, then calculate score_x_j.

	x_5	x_4	x_3	x_2	$\bar{x}_1$
Number of true occurrences	6	7	8	8	10
$2\times$(number of adjacent pairs)	-2×5	-2×3	-2×5	-2×4	-2×0
score_x_j	-4	$+1$	-2	0	$+10$

Step 5. minscore $= -4$.

Step 6. score_x_5 = minscore.

Generate new RM expansion with variable x_5 complemented—polarity 17 RM expansion. BPL = (RM expansion: polarity 17).

Step 7. Repeat Steps 2–6 for the new function.

Steps 2–4. For each variable, find the number of pairs of adjacent terms and the number of true occurrences, then calculate score_x_j.

	$\bar{x}_5$	x_4	x_3	x_2	$\bar{x}_1$
Number of true occurrences	6	4	4	6	7
$2\times$(number of adjacent pairs)	-2×1	-2×2	-2×2	-2×2	-2×1
score_x_j	$+4$	0	0	$+2$	$+5$

Step 5. minscore = 0.

Step 6. score_x_4 = score_x_3 = minscore.

Generate new RM expansions, the first with variable x_4 complemented-polarity 25 RM expansion, the second with variable x_3 complemented-polarity 21 RM expansion. BPL=(RM expansions: polarity 17, polarity 25, polarity 21).

Step 7. Repeat Steps 2–6 for each new function.

Steps 2–4. For each variable, find the number of pairs of adjacent terms and the number of true occurrences, then calculate score_x_j.

$\bar{x}_5$	$\bar{x}_4$	x_3	x_2	$\bar{x}_1$	$\bar{x}_5$	x_4	$\bar{x}_3$	x_2	$\bar{x}_1$
5	4	4	4	7	6	4	4	4	6
-2×0	-2×2	-2×1	-2×3	-2×3	-2×2	-2×1	-2×2	-2×2	-2×1
+5	0	+2	−2	+1	+2	+2	0	0	+4

Step 5. minscore = −2.

Step 6. score_x_2 = minscore

Generate new RM expansion, with variable x_2 complemented-polarity 27 RM expansion. BPL=(RM expansions: polarity 27).

Step 7. Repeat Steps 2–6 for the new function.

Steps 2–4. For each variable, find the number of pairs of adjacent terms and the number of true occurrences, then calculate score_x_j.

	$\bar{x}_5$	$\bar{x}_4$	x_3	$\bar{x}_2$	$\bar{x}_1$
Number of true occurrences	5	4	3	4	6
2 × (number of adjacent pairs)	-2×1	-2×0	-2×1	-2×1	-2×2
score_x_j	3	4	1	2	2

Step 5. minscore = 1.

Step 8. Best polarity is polarity 27.

RM expansion (polarity 27) contains 8 piterms.

RM function (polarity 27).

$\bar{x}_5$	$\bar{x}_4$	x_3	$\bar{x}_2$	$\bar{x}_1$
0	0	0	1	0
0	0	0	1	1
0	1	0	0	1
1	0	1	0	0
1	0	1	0	1
1	1	0	0	1
1	1	0	1	1
1	1	1	1	1

An exhaustive search shows the polarity 27 RM expansion to contain the least number of product terms. The TT optimization algorithm concluded with the polarity 17 RM expansion, comprising 10 product terms.

The new algorithm increases the likelihood of determining the optimum fixed polarity RM expansion of any given function. It may, however, require more computation time than the existing algorithm, and it is no longer the case that the maximum number of iterations of the algorithm is equal to the number of function variables. Indeed, in a very small number of cases, it is possible that the optimization algorithm will exhaustively search for the best polarity. This is dependent on the inherent structure of the boolean function.

The algorithm has been extensively tested and evaluated and results are shown in Figs 1 and 2. Each test set comprises 1000 randomly generated functions of fixed numbers of variables and minterms. The quality of the solution found by the algorithm is displayed as a percentage calculated from the number of functions for which the algorithm found the optimum polarity (Fig. 1) or a polarity within the 10 best polarities (Fig. 2). The optimum and subsequent nine most minimal polarities, each with a different number of terms, were found by exhaustive search, using the technique developed by Harking (1990). Additionally, results for the TT optimization algorithm (Almaini *et al.* 1991) and the algorithm developed by Habib (1990) are shown for the purposes of comparison. In the case of functions with four variables all methods gave results of 100% when finding a polarity within the 10 best polarities. These results are, therefore, omitted from Fig. 2.

Figure 1 shows that the Gains optimization algorithm on average predicts a solution which is more minimal than any solution predicted by either of the remaining algorithms. It is possible to state that the Gains optimization algorithm will always provide a solution which is at least as minimal as that predicted by the TT optimization algorithm. Also, in all but a very limited number of cases, the Gains algorithm will also provide a better solution than the algorithm developed by Habib (1990). All algorithms are obviously most effective for functions with smaller numbers of variables. This may be partly attributed to the fact that, in general, these functions have fewer numbers of groups of polarities each with differing numbers of piterms (costs). Figure 3 shows how the number of different costs associated with a function increases as the number of variables increases. As an example, a function

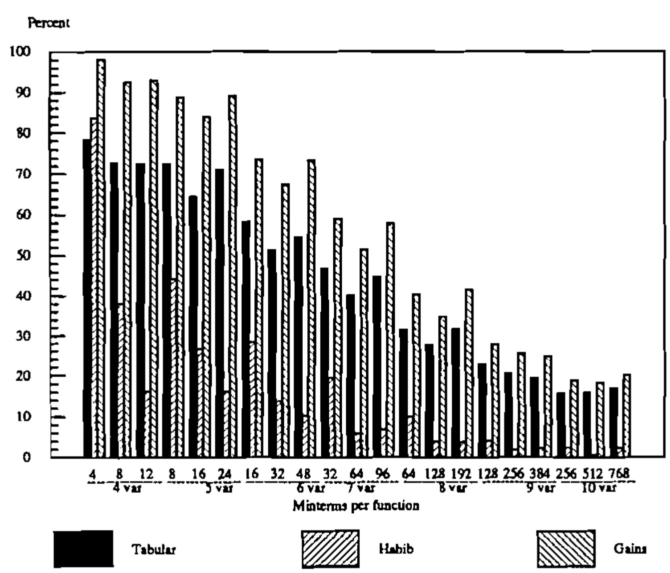

Figure 1. Percentage of cases where method predicts a polarity which is optimum.

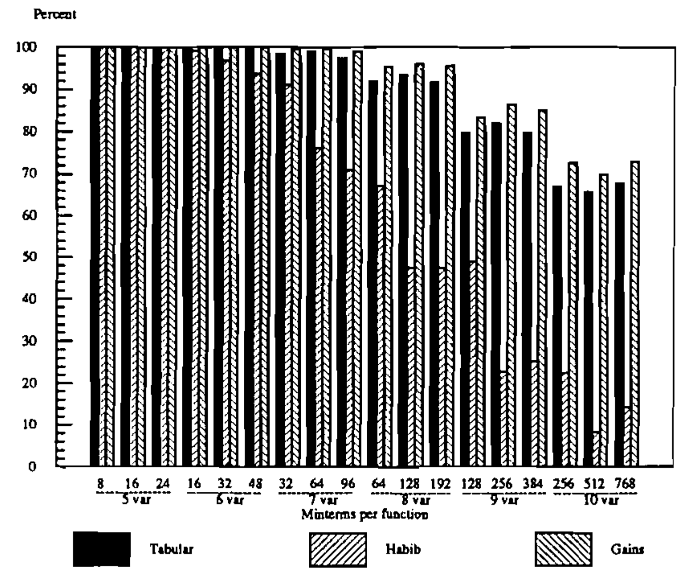

Figure 2. Percentage of cases where method predicts a polarity within first ten minimal polarities.

with 10 variables, and 512 minterms will, on average, have 85 different costs from minimum to maximum. This differs from a small function of four variables and eight minterms, which has on average only six costs from minimum to maximum. As for the previous graphs, each test set comprises of 1000 randomly generated functions.

It is necessary to state that the optimization algorithm developed by Habib (1990) requires that the final solution be observed to determine whether it may be further reduced. As this step is intuitive it was not performed when generating test results, which were gathered from fully automated testing.

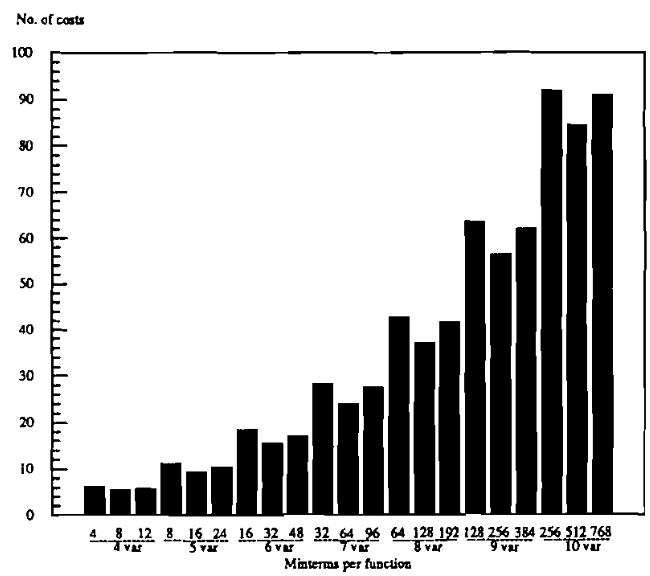

Figure 3. Average number of costs in each set of optimized functions.

The program was implemented in PASCAL and run on APOLLO workstations. Timings for the three algorithms evaluated are comparable, but the method developed by Habib (1990) is always the fastest.

For functions of five variables the algorithms typically require less than half the time required to perform an exhaustive search. Larger functions of 10 variables show that the algorithms typically require less than one tenth the time taken by an exhaustive search. The time taken by any one of the three algorithms to minimize a single five variable function is typically 2 ms, whilst a single 10 variable function can be minimized in approximately 500 ms.

3. Extension of tabular technique to incompletely specified functions

An incompletely specified boolean function has one or more output values which are undefined. When this function is transformed to the RM domain these unspecified or 'don't care' terms are also transformed and their effects distributed over several terms (see the Appendix) (Green 1987, Tran 1987, Varma and Trachtenberg 1991). Just as 'don't care' terms can be utilized to form an efficient solution in sum of products form so these transformed terms can help to realize an efficient RM expansion. Green (1987) has stated that it is not generally the case that an optimum allocation of 'don't care' terms in the boolean domain is also an optimum allocation in the RM domain. There is evidence to support this statement, hence a procedure is required which is suitable for RM functions. The Tabular Technique provides a means of transforming a canonic switching function to a fixed polarity RM expansion. It is obviously useful to transform both completely and incompletely specified functions, and in the case of the latter determine the best assignment of 'don't care' terms. The method proposed here transforms the incompletely specified boolean function to its zero (positive) polarity RM expansion in the usual manner but additionally tracks the transformation of the 'don't care' terms. The resulting function is a RM expansion of specified and unspecified terms. The optimum allocations of the unspecified terms can then be determined.

3.1. *Algorithm to allocate unspecified terms*

Step 1. For the incompletely specified function of n variables and t 'don't care' terms, denote the 'don't care' terms $d_1 \ldots d_t$.
p_i and $p_j d_k$ denote the specified and unspecified terms of the function respectively.

Step 2. Transform the function to the zero (positive) polarity RM expansion, tracking the 'don't care' terms according to the following rules:

	Original (Generated) term	Generated (Original) term	Transformed term
(i)	p_i	p_i	0 (terms cancel)
(ii)	$p_i(d_j \oplus \ldots \oplus d_m)$	$p_i(d_j \oplus \ldots \oplus d_m)$	0 (terms cancel)
(iii)	$p_i(d_j \oplus \ldots \oplus d_m)$	$p_i d_r$	$p_i(d_j \oplus \ldots \oplus d_m \oplus d_r)$
(iv)	p_i	$p_i(d_j \oplus \ldots \oplus d_m)$	$p_i(d_j \oplus \ldots \oplus d_m \oplus 1)$ $\Rightarrow p_i(\bar{d}_j \oplus \ldots \oplus d_m)$
(v)	$p_i(d_j \oplus d_k \oplus \ldots \oplus d_m \oplus d_n \oplus \ldots \oplus d_r)$	$p_i(d_k \oplus \ldots \oplus d_m)$	$p_i(d_j \oplus d_n \oplus \ldots \oplus d_r)$
(vi)	$p_i(d_j \oplus d_k \oplus \ldots \oplus d_m \oplus d_n \oplus \ldots \oplus d_r)$	$p_i(\bar{d}_k \oplus \ldots \oplus d_m)$	$p_i(d_j \oplus d_n \oplus \ldots \oplus d_r \oplus 1)$ $\Rightarrow p_i(\bar{d}_j \oplus d_n \oplus \ldots \oplus d_r)$

Step 3. Form groups of RM unspecified terms, according to the boolean 'don't care' terms which they contain.

Groupings:
Terms dependent on $\dot{d}_1$ are d_1, $\bar{d}_1$

Terms dependent on $\dot{d}_2$, and $\dot{d}_1$ and $\dot{d}_2$ are $d_2, \bar{d}_2, \bar{d}_1 \oplus \bar{d}_2, \bar{d}_1 \oplus d_2,$
$d_1 \oplus \bar{d}_2,\ d_1 \oplus d_2$

⋮ ⋮

Terms dependent on $\dot{d}_t$, and $\dot{d}_t$ and any combination of $\dot{d}_1 \ldots \dot{d}_{t-1}$

Count the number of RM 'don't care' terms in each group $dc_1, \ldots, dc_t$. Let $dc = dc_1 + \ldots + dc_t$.

Step 4. For $d_1 \ldots d_t$ evaluate the effects of setting $d_1(\ldots d_t)$ to zero and then to one. If the RM 'don't care' term is equated to zero then increment the score by one.

Additionally, determine which branches will not lead to the optimum solution and terminate these paths:

Find the maximum score for the current d_i, let this equal maxscore. Find the minimum score for the current d_i, let this equal minscore. Let $dc := dc - dc_i$.

If minscore $+ dc <$ maxscore $+ \left\lceil \frac{dc}{2} \right\rceil$ then terminate all branches with score = minscore.

If $dc = 0$, find maxscore, all scores = maxscore indicate optimum assignments of 'don't care' terms.

Step 5. For all score = maxscore, substitute the appropriate values of $d_1 \ldots d_t$ into the 0 polarity RM expansion to obtain optimum completely specified 0 polarity RM expansions.

3.1.1. *Example* 2: *Boolean function* $f(x_4 x_3 x_2 x_1) = \Sigma m(3, 5, 6, 9, 12, 15) + d(1, 2, 8, 11)$

Step 1. Denote 'don't care' terms $d_1 \ldots d_t$
Boolean function:

x_4	x_3	x_2	x_1	
0	0	1	1	
0	1	0	1	
0	1	1	0	
1	0	0	1	
1	1	0	0	
1	1	1	1	
0	0	0	1	d_1
0	0	1	0	d_2
1	0	0	0	d_3
1	0	1	1	d_4

Step 2. Transform the function to the zero (positive) polarity RM expansion ($*_i$ denotes equivalent terms between associated generated terms and functions).

x_4

Boolean function

	x_4	x_3	x_2	x_1	
	0	0	1	1	
	0	1	0	1	
	0	1	1	0	
$*_1$	1	0	0	1	
	1	1	0	0	
	1	1	1	1	
	0	0	0	1	d_1
	0	0	1	0	d_2
	1	0	0	0	d_3
$*_2$	1	0	1	1	d_4

Generated terms

	x_4	x_3	x_2	x_1	
$*_2$	1	0	1	1	
	1	1	0	1	
	1	1	1	0	
$*_1$	1	0	0	1	d_1
	1	0	1	0	d_2

x_3

Transformed function

	x_4	x_3	x_2	x_1	
	0	0	1	1	
$*_1$	0	1	0	1	
$*_2$	0	1	1	0	
	1	0	0	1	$\bar{d}_1$
$*_3$	1	1	0	0	
$*_4$	1	1	1	1	
	0	0	0	1	d_1
	0	0	1	0	d_2
	1	0	0	0	d_3
	1	0	1	1	$\bar{d}_4$
$*_5$	1	1	0	1	
$*_6$	1	1	1	0	
	1	0	1	0	d_2

Generated terms

	x_4	x_3	x_2	x_1	
	0	1	1	1	
$*_5$	1	1	0	1	$\bar{d}_1$
$*_1$	0	1	0	1	d_1
$*_2$	0	1	1	0	d_2
$*_3$	1	1	0	0	d_3
$*_4$	1	1	1	1	$\bar{d}_4$
$*_6$	1	1	1	0	d_2

x_2

Transformed function

	x_4	x_3	x_2	x_1	
$*_1$	0	0	1	1	
	0	1	0	1	$\bar{d}_1$
	0	1	1	0	$\bar{d}_2$
	1	0	0	1	$\bar{d}_1$
	1	1	0	0	$\bar{d}_3$
$*_2$	1	1	1	1	d_4
	0	0	0	1	d_1
	0	0	1	0	d_2
	1	0	0	0	d_3
$*_3$	1	0	1	1	$\bar{d}_4$
	1	1	0	1	d_1
$*_4$	1	1	1	0	$\bar{d}_2$
$*_5$	1	0	1	0	d_2
$*_6$	0	1	1	1	

Generated terms

	x_4	x_3	x_2	x_1	
$*_6$	0	1	1	1	$\bar{d}_1$
$*_3$	1	0	1	1	$\bar{d}_1$
$*_4$	1	1	1	0	$\bar{d}_3$
$*_1$	0	0	1	1	d_1
$*_5$	1	0	1	0	d_3
$*_2$	1	1	1	1	d_1

x_1

Transformed function

	x_4	x_3	x_2	x_1	
$*_1$	0	0	1	1	$\bar{d}_1$
	0	1	0	1	$\bar{d}_1$
	0	1	1	0	$\bar{d}_2$
$*_2$	1	0	0	1	$\bar{d}_1$
	1	1	0	0	$\bar{d}_3$
$*_3$	1	1	1	1	$d_4 \oplus d_1$
	0	0	0	1	d_1
	0	0	1	0	d_2
	1	0	0	0	d_3
$*_4$	1	0	1	1	$\bar{d}_1 \oplus \bar{d}_4$
$*_5$	1	1	0	1	d_1
	1	1	1	0	$\bar{d}_2 \oplus \bar{d}_3$
	1	0	1	0	$d_2 \oplus d_3$
$*_6$	0	1	1	1	d_1

Generated terms

	x_4	x_3	x_2	x_1	
$*_6$	0	1	1	1	$\bar{d}_2$
$*_5$	1	1	0	1	$\bar{d}_3$
$*_1$	0	0	1	1	d_2
$*_2$	1	0	0	1	d_3
$*_3$	1	1	1	1	$\bar{d}_2 \oplus \bar{d}_3$
$*_4$	1	0	1	1	$d_2 \oplus d_3$

Transformed function

x_4	x_3	x_2	x_1	
0	0	1	1	$\bar{d}_1 \oplus \bar{d}_2$
0	1	0	1	$\bar{d}_1$
0	1	1	0	$\bar{d}_2$
1	0	0	1	$\bar{d}_1 \oplus d_3$
1	1	0	0	$\bar{d}_3$
1	1	1	1	$d_1 \oplus d_4 \oplus \bar{d}_2 \oplus \bar{d}_3$
0	0	0	1	d_1
0	0	1	0	d_2
1	0	0	0	d_3
1	0	1	1	$\bar{d}_1 \oplus \bar{d}_4 \oplus d_2 \oplus d_3$
1	1	0	1	$d_1 \oplus \bar{d}_3$
1	1	1	0	$\bar{d}_2 \oplus \bar{d}_3$
1	0	1	0	$d_2 \oplus d_3$
0	1	1	1	$d_1 \oplus \bar{d}_2$

The incompletely specified boolean function has been transformed to the equivalent zero (positive) polarity RM expansion.

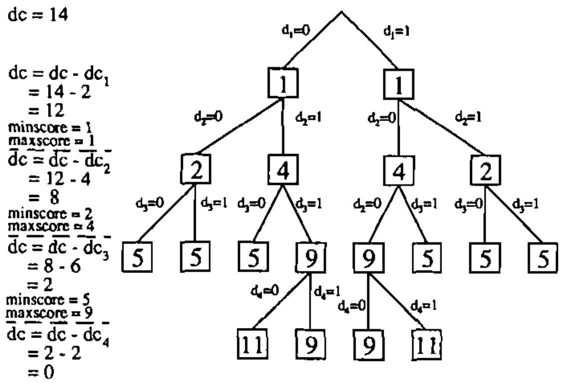

Figure 4. Non-exhaustive process of determining optimum assignments of 'don't care' terms.

Step 3. Group all terms

Terms dependent on $\dot{d}_1$	: d_1, $\bar{d}_1$
Terms dependent on $\dot{d}_1$ and $\dot{d}_2$	: d_2, $\bar{d}_2$, $\bar{d}_1 \oplus d_2, d_1 \oplus \bar{d}_2$
Terms dependent on $\dot{d}_1$, $\dot{d}_2$ and $\dot{d}_3$	: d_3, $\bar{d}_3$, $\bar{d}_1 \oplus d_3, d_1 \oplus \bar{d}_3$, $d_2 \oplus d_3$, $\bar{d}_2 \oplus \bar{d}_3$
Terms dependent on $\dot{d}_1$, $\dot{d}_2$, $\dot{d}_3$ and $\dot{d}_4$	: $\bar{d}_1 \oplus d_2 \oplus d_3 \oplus \bar{d}_4$, $d_1 \oplus \bar{d}_2 \oplus \bar{d}_3 \oplus d_4$

Count the number of terms in each group.

$$dc_1 = 2 \quad dc_2 = 4 \quad dc_3 = 6 \quad dc_4 = 2$$

Total number of 'don't care' terms $= dc = 14$.

Step 4. Evaluate the effects of setting each 'don't care' term first to zero and then to one. Additionally, determine which branches may be terminated.
This step is illustrated in Fig. 4.

Optimum solutions:

$d_1 = 0 \quad d_2 = 1 \quad d_3 = 1 \quad d_4 = 0$

$d_1 = 1 \quad d_2 = 0 \quad d_3 = 0 \quad d_4 = 1$

Step 5. Substitute assignments for the 'don't care' terms into the incompletely specified RM expansion, to determine the optimum positive polarity expansions of an incompletely specified boolean function.

Boolean functions

$f(x_4 x_3 x_2 x_1) = \Sigma m(3,5,6,9,12,15,\mathbf{2},\mathbf{8})$ $\qquad$ $f(x_4 x_3 x_2 x_1) = \Sigma m(3,5,6,9,12,15,\mathbf{1},\mathbf{11})$

Optimum RM expansions (polarity 0).

x_4	x_3	x_2	x_1
0	1	0	1
0	0	1	0
1	0	0	0

x_4	x_3	x_2	x_1
0	1	1	0
1	1	0	0
0	0	0	1

The optimum allocation of the 'don't care' terms in the boolean domain is to assign all 'don't care' terms the value one.

This example illustrates the use of the algorithm to determine the optimum assignments of the unspecified terms of a function without performing an exhaustive search. Although this method was developed as an extension to the Tabular Technique it works equally well when the principle is applied to boolean matrix transformations (Habib 1990, Harking 1990). The terms are transformed to the Reed–Muller domain using matrix manipulations, then the 'don't care' terms are assigned using the tree structure.

4. Generalized Reed–Muller expansions of incompletely specified functions

The search for the optimum fixed polarity RM expansion of an incompletely specified function incorporates the need to find the best assignment of 'don't care' terms with the search for the optimum polarity expansion. An exhaustive search is impractical for all but very small functions with low numbers of unspecified terms. The number of possible combinations to be evaluated during an exhaustive search is a function of both the number of variables n, and the number of 'don't care' terms t. Indeed, it may be formulated as 2^{n+t}.

One approach to finding a solution to this problem is by considering first, only the completely specified terms of the function and determining the optimum polarity. Then, for the new fixed polarity RM expansion, find the best allocation of the 'don't care' terms. Alternatively, a function may be minimized by determining the optimum assignment of 'don't care' terms for the positive polarity RM expansion, then finding the optimum polarity for the now fully specified function. Unfortunately, neither approach guarantees that one will find the optimum RM expansion of a given function, and although the number of combinations has been reduced to 2^n+2^t, both approaches are impractical for all functions except those with limited numbers of variables and unspecified terms.

In order to obtain a satisfactory solution to this problem, within a reasonable time, it is possible to utilize the Gains method and the technique for finding optimum assignments of the 'don't care' terms of a function, as detailed in §2 and §3, respectively. As previously stated there are two approaches to obtaining a minimal fixed polarity expansion of an incompletely specified function. In the first approach the Gains method is used to determine a good fixed polarity expansion, considering only the specified terms of the function. The 'don't care' terms are then transformed to this polarity, their effects are tracked using the rules detailed previously. The two portions of the fixed polarity RM expansion are then united and the optimum allocation of unspecified terms is determined. The second approach requires that the optimum assignment of the 'don't care' terms is found for the positive polarity RM expansion using the method described in §3. The Gains method is then employed to determine a good polarity for the now fully specified function.

Both these approaches have been tested and the results are shown in Fig. 5. Each test set comprised 1000 randomly generated functions of fixed numbers of variables, and specified and unspecified minterms. The quality of the solution found by the algorithm is displayed as a percentage calculated from the number of functions for which the algorithm found the optimum polarity. The optimum polarities, for functions formed for all combinations of 'don't care' terms, were found by

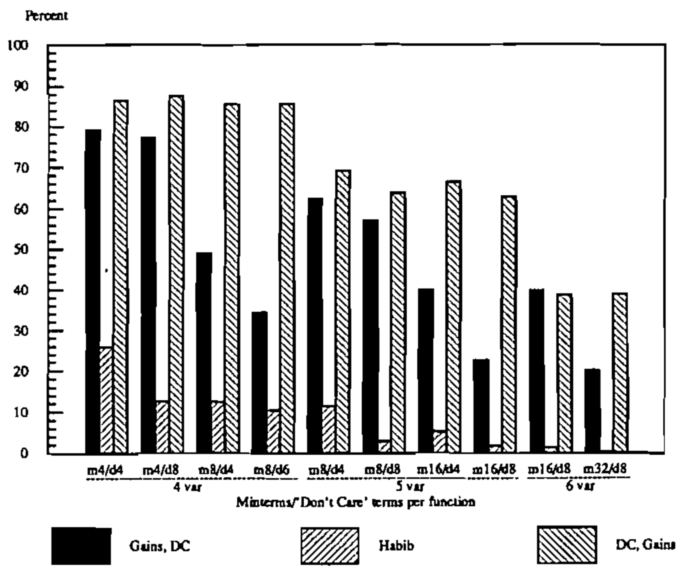

Figure 5. Percentage of cases where method predicts a solution which is optimum.

exhaustive search using the technique developed by Harking (1990). Additionally, results are given for the algorithm developed by Habib (1990). Again it is necessary to state that the optimization algorithm developed by Habib (1990) requires that the final solution be observed to determine whether it may be further reduced. This step was not performed when generating test results.

The results, for the range of functions tested, show that employing the second approach is, in general, more likely to produce the optimum solution. That is, the best fixed polarity RM expansion of an incompletely specified boolean function can be determined by first finding the optimum allocation of 'don't care' terms (§ 3) then finding a good fixed polarity (§ 2).

The program was implemented in PASCAL and run on APOLLO workstations. The timings for the three algorithms evaluated are comparable. As is the case for the results of § 2, the method developed by Habib (1990) is always the fastest, but is less likely to produce optimum solutions (Fig. 5).

5. Conclusions

The Gains algorithm provides a means of determining good fixed polarity RM expansions of a fully specified boolean function within a reasonable time scale. As previously stated it addresses the first three 'disadvantages' of the original Tabular Technique optimization algorithm. The remaining 'disadvantages', namely that the algorithm is biased towards polarity zero and that the algorithm will locate and cease on local minima, have not yet been overcome. In order to address the fourth shortcoming, a type of pre-treatment may prove effective. This would involve determining whether it is best to initially convert the boolean function to either the positive (zero) polarity or the negative $(2^n - 1)$ polarity RM expansion. The optimization algorithm would then be employed. This technique is currently being investigated. The problem of determining which solutions are local minima and which are global minima uncovers the most serious shortcoming of the algorithm, and is the most difficult to overcome. The progression from a local to a global minimum would involve allowing the number of terms in the function to increase. This directly opposes the whole motivation of the algorithm, namely that functions should decrease or at least remain the same size for each iteration of the algorithm. Hence it is possible to locate local minima and the algorithm does not provide any means of determining whether a global minimum exists and, if so, at which polarity. This is an area for future work.

The algorithm detailed is § 3 provides a non-exhaustive means of determining the allocations of unspecified terms of boolean functions, resulting in the optimum positive polarity RM expansion. It may be possible to improve the eficiency of this algorithm by considering the order in which the groupings of the RM 'don't care' terms are formed (Step 3), and also the order in which terms are evaluated (Step 4). That is, it is not always the case that it is best to commence with d_1 and proceed to d_t. This is especially true when allocating 'don't care' terms for fixed polarity expansions other than polarity zero.

The procedure detailed in § 4 shows the use of the previous algorithms in determining sub-optimum fixed polarity expansions of incompletely specified boolean functions. It is felt that an area worthy of further investigation is that of determining good fixed polarities whilst considering both specified and unspecified terms. That is, an algorithm which will determine the best polarity of functions containing RM 'don't care' terms.

ACKNOWLEDGMENT

This project was supported by the U.K. Science and Engineering Research Council, under Grant GR/F 74288.

Appendix

Transform the boolean function f given by

$$f=\sum_{i=0}^{2^n-1} a_i m_i+\sum_{i=0}^{2^n-1} d_{i m_i}$$

to the zero polarity RM expansion given by

$$g=\oplus\sum_{i=0}^{2^n-1} b_i p_i \oplus \sum_{i=0}^{2^n-1} d'_i p_i$$

$$a_i,\ b_i,\ d_i \text{ and } d'_i \in \{0,1\}$$

where elements b_i are related to a_i by

$$b=\mathbf{T}_n a, \quad \mathbf{T}_n=\begin{bmatrix} \mathbf{T}_{n-1} & 0 \\ \mathbf{T}_{n-1} & \mathbf{T}_{n-1} \end{bmatrix} \quad \mathbf{T}_1=\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

and RM unspecified terms d'_i are related to boolean unspecified terms d_i by

$$d'=\mathbf{T}_n d$$

For example, for three variables

$$f=\Sigma m(1,3,7)+\Sigma d(0,4,6)$$

$$b=\mathbf{T}_3(0\ 1\ 0\ 1\ 0\ 0\ 0\ 1)^{\mathrm{T}}=(0\ 1\ 0\ 0\ 0\ 1\ 0\ 1)^{\mathrm{T}}$$

$$d'=\mathbf{T}_3(d_1\ 0\ 0\ 0\ d_2\ 0\ d_3\ 0)^{\mathrm{T}}$$

$$=(d_1\ d_1\ d_1\ d_1\ d_1\oplus d_2\ d_1\oplus d_2\ d_1\oplus d_2\oplus d_3\ d_1\oplus \mathrm{d}_2\oplus d_3)^{\mathrm{T}}$$

Hence,

$$g=(d_1\ 1\oplus d_1\ d_1\ d_1\ d_1\oplus d_2\ 1\oplus d_1\oplus d_2\ d_1\oplus d_2\oplus d_3\ 1\oplus d_1\oplus d_2\oplus d_3)^{\mathrm{T}}$$

References

ALMAINI, A. E. A., THOMSON, P., and HANSON, D., 1991, Tabular Techniques for Reed-Muller logic. *International Journal of Electronics*, **70**, 23-34.

BESSLICH, P. W., 1983, Efficient computer method for ExOR logic design. *Proceedings of the Institution of Electrical Engineers*, Pt E, **130**, 203-206.

GREEN, D. H., 1987, Reed-Muller expansions of incompletely specified functions. *Proceedings of the Institution of Electrical Engineers*, Pt E, **134**, 228-236.

HABIB, M. K., 1990, Boolean matrix representation for the conversion of minterms to Reed-Muller coefficients and the minimization of exclusive-OR switching functions. *International Journal of Electronics*, **68**, 493-506.

HARKING, B., 1990, Efficient algorithm for canonical Reed-Muller expansions of Boolean functions. *Proceedings of the Institution of Electrical Engineers*, Pt E, **137**, 366-370.

TRAN, A., 1987, Graphical method for the conversion of minterms to Reed-Muller coefficients and the minimisation of exclusive-OR switching functions. *Proceedings of the Institution of Electrical Engineers*, Pt E, **134**, 93-99.

VARMA, D., and TRACHTENBERG, E. A., 1991, Computation of Reed-Muller expansions of incompletely specified Boolean functions from reduced representations. *Proceedings of the Institution of Electrical Engineers*, Pt E, **138**, 85-92.

WU, X., CHEN, X., and HURST, S. L., 1982, Mapping of Reed-Muller coefficients and the minimisation of exclusive-OR switching functions. *Proceedings of the Institution of Electrical Engineers*, Pt E, **129**, 15-20.