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

Highly efficient exhaustive search algorithm for optimizing canonical Reed-Muller expansions of boolean functions

J. F. Miller [a]; P. Thomson [a]
[a] Department of Electrical, Electronic and Computer Engineering, Napier University, Edinburgh, UK

Online Publication Date: 01 January 1994
To cite this Article: Miller, J. F. and Thomson, P. (1994) 'Highly efficient exhaustive search algorithm for optimizing canonical Reed-Muller expansions of boolean functions', International Journal of Electronics, 76:1, 37 - 56
To link to this article: DOI: 10.1080/00207219408925904
URL: http://dx.doi.org/10.1080/00207219408925904

INT. J. ELECTRONICS, 1994, VOL. 76, NO. 1, 37-56

Highly efficient exhaustive search algorithm for optimizing canonical Reed–Muller expansions of boolean functions

J. F. MILLER† and P. THOMSON†

A new method is presented for calculating fixed polarity Reed-Muller expansions from the boolean minterms. Direct transformation equations of Reed-Muller expansions with polarity are derived. A highly efficient and flexible exhaustive search algorithm is presented which can obtain an optimum polarity more quickly if a sub-optimum polarity is obtained first. Also exact formulae are presented for optimum polarities for any three-variable logic function.

Notation

$$\dot{x}_i \equiv x_i \text{ or } \bar{x}_i$$

A single subscript k represents the kth bit, $k=0$ is least significant. p represents decimal polarity. m represents decimal minterm. m_i is the ith minterm, m_{ik} is the kth bit of minterm m_i. b_j is the jth bit of the RM coefficient vector for a given boolean function. b_{ij} is the jth bit of the RM coefficient vector corresponding to minterm m_i. n represents the number of variables.

1. Introduction

It is now a well documented fact that logic functions may be specified either by the usual boolean notation or alternatively in the Reed-Muller (RM) form (Green 1986, 1987, Almaini 1989). Further, there is now a reasonable body of work which states how a logic function, specified by its minterms, may be converted to the Reed-Muller form (Tran 1987, Almaini *et al.* 1991), and the advantages of doing this (Reddy 1972). Besides the advantage of testability outlined in the paper by Reddy, the use of the RM form allows a further 2^n fixed specifications of any given logic function if n is the number of variables required to completely specify the original function in the Boolean form. The advantage of this is that these 2^n different specifications may be examined in order to determine which of these is optimum in terms of some previously defined cost function (related to the number of gates used) (Almaini *et al.* 1991). Each one of these 2^n new specifications is referred to as an **RM fixed polarity form**, and therefore one wishes to know which of these polarities should be used to optimize the function in the Reed-Muller form by minimizing the cost function. The solution to this problem will give either a unique polarity, or a small set of polarities which represent the Reed-Muller function with the minimum number of exclusive-ORs (EX-ORs).

Algorithms have been devised which perform an exhaustive search over the 2^n polarities to determine the polarities which result in the minimum number of EX-ORs in the RM expansions (Besslich 1983, Harking 1990, Almaini *et al.* 1991, Lui and Muzio 1991). However as the number of variables grows these techniques can

Received 14 June 1993; accepted 3 August 1993.
†Department of Electrical, Electronic and Computer Engineering, Napier University, 219 Colinton Road, Edinburgh EH14 1DJ, UK.

0020-7217/94 $10.00

become impractical due to the computation time and computer memory requirements. The most efficient exhaustive searches (Harking 1990, Lui and Muzio 1991) have time complexities of O[3^n]. In response to this there have been a number of algorithms developed which find a sub-optimum polarity in a much reduced amount of time (Wu *et al.* 1982, Habib 1990, Almaini *et al.* 1991, McKenzie *et al.* 1993).

As yet there has appeared in the literature no formulae which explicitly give an optimum polarity for a given number of variables irrespective of the number of product terms. In this paper explicit formulae are given for an optimum polarity for fixed polarity RM expansions of boolean functions of three variables. Formulae are derived which relate the coefficients of the product terms in the RM expansions (henceforth referred to as b-coefficients) to the polarity numbers of the expansion. From these formulae an algorithm is derived for exhaustive search which in the worst case (it is functionally dependent) compares favourably with other efficient search methods (Harking 1990) but for many functions can complete an exhaustive search much faster and moreover computation time is further reduced significantly for functions if a sub-optimum polarity has been obtained using, for instance, the tabular technique (Almaini *et al.* 1991). The algorithm is flexible enough to allow a rough indication of how long a full exhaustive search is likely to take. This gives the user the option of deciding not to do an exhaustive search if the cost savings for the sub-optimum search are sufficient.

2. Foundations

Theorem 1

Every boolean function f: $\{0, 1\}^n \rightarrow \{0, 1\}$

$$f(x_{n-1}, \ldots, x_0) = \sum_{i=0}^{2^n-1} a_i m_i \tag{1}$$

where m_i are the minterms and $a_i = 1$ or 0 indicate the presence or absence of minterms respectively, can also be expressed in Reed–Muller form (RM) as follows

$$f(\dot{x}_{n-1}, \ldots, \dot{x}_0) = \oplus \sum_{i=0}^{2^n-1} b_i \phi_i \tag{2}$$

where

$$\phi_i = \prod_{k=0}^{n-1} \dot{x}_k^{i_k} \tag{3}$$

where $\dot{x}_k = x_k$ or $\bar{x}_k$, $x_k \in \{0, 1\}$, $a_i b_i \in \{0, 1\}$.
ϕ_i are known as product terms and b_i determine whether a product term is present or not. $\oplus$ indicates the XOR operation and multiplication is assumed to be the AND operation.

Proof

The proof has been given by Akers (1959).

Definitions

A RM function $f(\dot{x}_{n-1}, \ldots, \dot{x}_0)$ is said to have **fixed polarity** if throughout the expression each variable $\dot{x}_k$ is either x_k or $\bar{x}_k$ exclusively. If for some variables x_k and

$\bar{x}_k$ both occur when the function is said to have **mixed polarity**. In this paper we will only be dealing with fixed polarity RM functions.

Associated with each of the 2^n possible fixed polarity expressions for a particular function is an integer $p \in \mathrm{N}$ ($\mathrm{N} = \{0, 1, 2, \ldots\}$) $0 \leq p \leq 2^n - 1$ whose binary digits p_k are defined.

$$p_k = \delta(\dot{x}_k, \bar{x}_k) \tag{4}$$

where

$$\delta(a, b) = \begin{cases} 0 & a \neq b \\ 1 & a = b \end{cases} \tag{5}$$

Example 1

Consider the boolean function,

$$\begin{aligned} f(x_2, x_1, x_0) &= \textstyle\sum(0, 2, 5, 7) \\ &= \bar{x}_2\bar{x}_1\bar{x}_0 + \bar{x}_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 x_0 \\ &= \sum_{i=0}^{7} a_i m_i \end{aligned}$$

$$a_1, a_3, a_4, a_6 = 0 \quad a_0, a_2, a_5, a_7 = 1$$

This function can also be written in polarity 0 RM form

$$f(x_2, x_1, x_0) = 1 \oplus x_0 \oplus x_2$$

so that

$$b_2, b_3, b_5, b_6, b_7 = 0 \quad b_0, b_1, b_4, = 1$$

Changing the polarity to 3 gives

$$f(x_2, \bar{x}_1, \bar{x}_0) = \bar{x}_0 \oplus x_2$$

so that

$$b_0, b_2, b_3, b_5, b_6, b_7 = 0 \quad b_1, b_4 = 1$$

Boolean polarity

Two boolean functions with different minterm sets may effectively represent the same function provided one function is obtained from the other by complementing certain columns of the truth table representing that function. This is equivalent to substituting for one or more variables $\dot{x}_k$ with $\bar{\dot{x}}_k$. As an example consider

$$f(x_2, x_1, x_0) = \Sigma(0, 2, 4) = \bar{x}_2\bar{x}_1\bar{x}_0 + \bar{x}_2 x_1 \bar{x}_0 + x_2\bar{x}_1\bar{x}_0$$

Substituting

$$x'_2 = \bar{x}_2, \quad x'_1 = x_1, \quad x'_0 = \bar{x}_0$$

the above becomes

$$f(x'_2, x'_1, x'_0) = x'_2\bar{x}'_1x'_0 + x'_2x'_1x'_0 + \bar{x}'_2\bar{x}'_1x'_0$$

$$f(x'_2, x'_1, x'_0) = \Sigma(1, 5, 7)$$

$f(x'_2, x'_1, x'_0)$ is said to be the original function $f(x_2, x_1, x_0)$ represented in boolean polarity 5.

The concept of boolean polarity allows one to group logic functions together which are polarity maps of each other.

Lemma 1 establishes that boolean polarity and Reed-Muller polarity are really synonymous. It is also clear that logic functions with an odd number of minterms which XOR to a particular sum *cannot* be polarity mapped into each other. This fact is important in Corollary 1.

A polarity p boolean function is defined as follows.

$$f^p_{\mathrm{B}}(x'_{n-1}, x'_{n-2}, \ldots, x'_0) = \sum_{i=0}^{2^n-1} a_i t_i(p) \tag{6}$$

$$t_i(p) = \prod_{k=0}^{n-1} x_k(i_k) \oplus p_k$$

where

$$x'_j = x_j \oplus p_j, \quad x_k(0) = \bar{x}_k, \quad x_k(1) = x_k$$

3. Theory

Lemma 1

The p polarity RM function generated from a boolean function in zero polarity is identical to the zero polarity RM function generated from the p polarity boolean function.

Proof

As we have seen, a zero polarity boolean function can be written

$$f^0_{\mathrm{B}}(x_{n-1}, \ldots, x_0) = \sum_{i=0}^{2^n-1} a_i t_i(0) \tag{7}$$

$$t_i(0) = \prod_{k=0}^{n-1} x_k(i_k), \quad x_k(0) = \bar{x}_k, \quad x_k(1) = x_k$$

Now f^p_{RM} is calculated from the boolean function f^0_{B} by replacing variables $\bar{x}_k$ by $1 \oplus x_k$ (or x_k by $1 \oplus \bar{x}_k$) depending on the polarity as follows.

$$f^{p}_{\mathrm{RM}} = \oplus \sum_{i=0}^{2^n-1} a_i t_i(p) \tag{8}$$

$$t_i(p) = \prod_{k=0}^{n-1} z_k(i_k), \quad z_k(0) = 1 \oplus x_k \oplus p_k, \quad z_k(1) = x_k \oplus p_k$$

The zero polarity RM function f' is given by definition as

$$f'(x'_{n-1}, \ldots, x'_0) = \sum_{i=0}^{2^n-1} a_i t_i(0) \tag{9}$$

where

$$t_i(0) = \prod_{k=0}^{n-1} x'_k(i_k), \quad x'_k(0) = 1 \oplus x'_k, \quad x'_k(1) = x'_k$$

If x'_k is defined to be equal to $x_k \oplus p_k$ then we are using the p polarity boolean representation for the function. Hence substituting $x'_k = x_k \oplus p_k$ into (9) gives

$$f'(x_{n-1} \oplus p_{n-1}, \ldots, x_0 \oplus p_0) = \sum_{i=0}^{2^n-1} a_i t_i(0) \tag{10}$$

$$t_i(0) = \prod_{k=0}^{n-1} x'_k(i_k), \quad x'_k(0) = 1 \oplus x_k \oplus p_k, \quad x'_k(1) = x_k \oplus p_k$$

Equation (10) is seen to be identical to (8).

Example 2

Let a zero polarity boolean function be (x_2 most significant)

$$f^{0}_{\mathrm{B}}(x_2, x_1, x_0) = \bar{x}_2 \bar{x}_1 x_0 + x_2 \bar{x}_1 \bar{x}_0 (\Sigma(1, 4))$$

The RM function derived from this in polarity 4 is

$$f^{4}_{\mathrm{RM}}(\bar{x}_2, x_1, x_0) = (1 \oplus x_0 \oplus \bar{x}_2)(1 \oplus x_1) \tag{11}$$

The same boolean function represented in polarity 4 is

$$f^{4}_{\mathrm{B}}(x'_2, x'_1, x'_0) = x'_2 \bar{x}'_1 x'_0 + \bar{x}'_2 \bar{x}'_1 \bar{x}'_0 (\Sigma(1 \oplus 4, 4 \oplus 4))$$

where

$$x'_2 = \bar{x}_2, \quad x'_1 = x_1, \quad x'_0 = x_0 \tag{12}$$

The zero polarity RM function derived from this is

$$f^{4}_{\mathrm{RM}}(x'_2, x'_1, x'_0) = x'_2(1 \oplus x'_1) x'_0 \oplus (1 \oplus x'_2)(1 \oplus x'_1)(1 \oplus x'_0)$$

$$= (1 \oplus x'_0 \oplus x'_2)(1 \oplus x'_1) \tag{13}$$

Clearly (11) and (13) together with (12) are identical.

Theorem 2

The coefficients $b_i(p)$ of the RM fixed polarity p expansion of a logic function are related to the coefficients b_i in polarity zero as follows:

$$b_i(p) = \oplus \sum_{\substack{k=0 \\ ki=0}}^{i+k=2^n-1} b_{i+k}\left(\prod_{j=0}^{n-1} p_j^{k_j}\right) \tag{14}$$

where k_j are the bits of k and p_j are the bits of the polarity p. ki is the bitwise AND of integers k and i.

Proof

The p polarity RM expansion of the zero polarity RM polynomial

$$f = \oplus \sum_{i=0}^{2^n-1} b_i \prod_{k=0}^{n-1} x_k^{i_k}$$

is given by (replacing x_k by $x_k \oplus p_k$)

$$f(p) = \oplus \sum_{i=0}^{2^n-1} b_i \prod_{k=0}^{n-1} (x_k^{i_k} \oplus p_k^{i_k})$$

Expanding this we obtain

$$f(p) = b_0 \oplus b_1(x_0 \oplus p_0) \oplus b_2(x_1 \oplus p_1) \oplus b_3(x_1 \oplus p_1)(x_0 \oplus p_0)$$

$$\oplus \cdots \oplus b_{2^n-1}(x_{n-1} \oplus p_{n-1})(x_{n-2} \oplus p_{n-2}) \cdots (x_0 \oplus p_0)$$

Rearranging and collecting the constant term and terms depending on x_0, x_1, x_1x_0, $\ldots$, $x_{n-1}x_{n-2}\cdots x_0$ respectively, it is seen that

$$f(p) = (b_0 \oplus b_1 p_0 \oplus b_2 p_1 \oplus b_3 p_1 p_0 \oplus \cdots \oplus b_{2^n-1} p_{n-1} p_{n-2} \cdots p_0)$$

$$\oplus (x_0)(b_1 \oplus b_3 p_1 \oplus b_5 p_2 \oplus b_7 p_2 p_1 \oplus \cdots \oplus b_{2^n-1} p_{n-1} p_{n-2} \cdots p_1)$$

$$\vdots$$

$$\oplus (x_{n-1} x_{n-2} \cdots x_1)(b_{2^n-2} \oplus b_{2^n-1} p_0)$$

$$\oplus (x_{n-1} x_{n-2} \cdots x_0) b_{2^n-1}$$

It is clear from this that the ith product term $\prod_{k=0}^{n-1} x_k^{i_k}$ is multiplied by the coefficient $b_i(p)$ which is a function of the zero-polarity b-coefficients b_j such that the bitwise AND of j and i is equal to i. Hence (14) is obtained. □

Example 3

The equations for $b_i(p)$ for three variables are given by

$b_0(p) = b_0 \oplus b_1 p_0 \oplus b_2 p_1 \oplus b_3 p_1 p_0 \oplus b_4 p_2 \oplus b_5 p_2 p_0 \oplus b_6 p_2 p_1 \oplus b_7 p_2 p_1 p_0$

$b_1(p) = b_1 \oplus b_3 p_1 \oplus b_5 p_2 \oplus b_7 p_2 p_1$

$b_2(p) = b_2 \oplus b_3 p_0 \oplus b_6 p_2 \oplus b_7 p_2 p_0$

$b_3(p) = b_3 \oplus b_7 p_2$

$b_4(p) = b_4 \oplus b_5 p_0 \oplus b_6 p_1 \oplus b_7 p_1 p_0$

$b_5(p) = b_5 \oplus b_7 p_1$

$b_6(p) = b_6 \oplus b_7 p_0$

$b_7(p) = b_7$

In the next section it will be seen how the coefficients b_i in zero-polarity relate to the minterms of the original boolean function. It will also be seen how for three variables (14) can be solved to provide an *exact* solution for the bits of the optimum polarity (Corollary 1, 2).

Theorem 3

A boolean function f in n variables

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

has the product term coefficients b_i for the RM form in polarity 0,

$$b = aT_n \tag{15}$$

where

$$b = (b_{2^n - 1}, \ldots, b_1, b_0) \quad a = (a_{2^n - 1}, \ldots, a_1, a_0)$$

and T_n is a $2^n \times 2^n$ binary matrix defined as follows:

$$T_n = \begin{bmatrix} T_{n-1} & 0 \\ T_{n-1} & T_{n-1} \end{bmatrix} \quad T_0 = (1) \tag{16}$$

Matrix multiplication is carried out using AND for multiplication and XOR for addition of binary elements.

Proof

The proof has been given by Besslich (1983) and Green (1987).

Example 4

Let f be the three variable function given in Example 2

$$a=(00010010)$$

then,

$$(b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)=(00010010)\begin{bmatrix} 1000\,0000 \\ 1100\,0000 \\ 1010\,0000 \\ 1111\,0000 \\ 1000\,1000 \\ 1100\,1100 \\ 1010\,1010 \\ 1111\,1111 \end{bmatrix}$$

$$b=(01011010) \quad \text{and} \quad f=x_2x_1\oplus x_2\oplus x_1x_0\oplus x_0$$

Note the operation in (15) may be carried out a minterm at a time. Consequently one may assign a b coefficient vector for each minterm. The final set of b coefficients representing the function would be all the b coefficient vectors for each minterm XORed together. With this in mind (15) is written

$$B(n)=A(n)T_n \tag{17}$$

where $A(n)$ is an $2^n\times 2^n$ binary matrix with non-zero elements $a_{ij'}=a_i, j'=2^n-1-i$, $0\le i\le r-1$. $B(n)$ is the $2^n\times 2^n$ binary matrix with elements $b_{ij'}$, $j'=2^n-1-j$, $0\le j\le 2^n-1$. j is the matrix column number.

b_j is then written

$$b_j=\oplus\sum_{i=0}^{r-1} b_{ij}$$

where r is the number of minterms.

Define also the minterm table M (or truth table) as follows. M is the $r\times n$ binary matrix whose elements are $m_{i\,n-j}$ are the $(n-j)$th bits of m_i, $1\le j\le n$, $0\le i\le r-1$, i.e.

$$m_i=(m_{i\,n-1}, m_{i\,n-2}, \ldots, m_{i\,0})$$

With these preliminaries over, we proceed to the next Lemma.

Lemma 2

The single minterm b coefficients $b_{ij'}$ of the RM function in polarity zero are related to the minterm components m_{ik} as follows:

$$b_{ij'}=\prod_{k=0}^{n-1}(1-m_{ik}j_k), \quad j'=2^n-1-j \tag{18}$$

where n is the number of variables. $m_i, j\in \mathrm{N}$ with $0\le m_i$, and $j\le 2^n-1$.

Proof

By induction on n

$$n=1 \quad m_0=0 \quad m_1=1 \quad j=0, 1$$

From (15) and (16) it is seen that

$$\begin{bmatrix} b_{01} & b_{00} \\ b_{11} & b_{10} \end{bmatrix} = \underset{A}{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}} \underset{T_1}{\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \tag{19}$$

Now examining (18)

$$b_{i,1-j} = 1 - m_{i0} j_0$$

Setting these values out in matrix form

$$b_{i,1-j} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

where in row 1 $i=0$, in row 2 $i=1$, in column 1 $j=1$ and in column 2 $j=0$. This is identical to (19). Thus (18) is true for $n=1$. It now has to be shown that (18) holds for $n+1$ if it is assumed to hold for n.

From (17)

$$B(n+1) = A(n+1)T_{n+1} = \begin{bmatrix} 0 & A(n) \\ A' & 0 \end{bmatrix} \begin{bmatrix} T_n & 0 \\ T_n & T_n \end{bmatrix}$$

$$B(n+1) = \begin{bmatrix} A(n)T_n & A(n)T_n \\ A'T_n & 0 \end{bmatrix} \tag{20}$$

where in row 1 $i_n=0$, in row 2 $i_n=1$, in column 1 $j_n=1$ and in column 2 $j_n=0$. Here A' is a $2^n \times 2^n$ matrix with non-zero coefficients $a'_{ij} = a_{2^n+i}$, $0 \le i \le 2^n - 1$, $j' = 2^n - 1 - i$, and the rows and columns of matrix $B(n+1)$ have been identified with the values of i_n and j_n. Now the induction assumption is that for n variables,

$$B(n) = A(n)T_n = \prod_{k=0}^{n-1} (1 - m_{ik} j_k)$$

Noting that $A'T_n$ is a matrix involving only n variables (since $i_n=1, j_n=0$) it is clear from the form of (20) that

$$b_{ij'}(n+1) = (1 - m_{i_n} j_n) \prod_{k=0}^{n-1} (1 - m_{ik} j_k)$$

□

Theorem 4

Let f be an n variable boolean function with minterms $\{m_i\}$, $0 \le i \le r-1$ where r is the number of minterms. The b coefficients of f in Reed-Muller form in polarity p, $b_{j'}(n, p)$ are given by

$$b_{j'}(n, p) = \oplus \sum_{i=0}^{r-1} b_{j'}(m_i, n, p) \tag{21}$$

where

$$b_{j'}(m_i, n, p) = \prod_{k=0}^{n-1} y_{ik} \quad j' = 2^n - 1 - j \quad 0 \le j \le 2^n - 1 \tag{22}$$

and

$$y_{ik} = \begin{cases} \bar{m}_{ik}\bar{p}_k \oplus m_{ik}p_k & j_k = 1 \\ 1 & j_k = 0 \end{cases}$$

Proof

From the definition of boolean polarity, new minterms m_i' are given by $m_i' = m_i \oplus p$ where p is the new polarity. Thus using Lemma 2 it can be seen that

$$b_{j'}(m_i', n) \equiv b_{j'}(m_i \oplus p, n) \equiv b_{j'}(m_i, n, p) = \prod_{k=0}^{n-1} [1 - (m_{ik} \oplus p_k) j_k]$$

Define

$$y_{ik} = 1 - (m_{ik} \oplus p_k) j_k$$

and it can be seen that

$$1 - m_{ik} \oplus p_k = \bar{m}_{ik}\bar{p}_k \oplus m_{ik}p_k$$

Therefore

$$y_{ik} = \begin{cases} \bar{m}_{ik}\bar{p}_k \oplus m_{ik}p_k & j_k = 1 \\ 1 & j_k = 0 \end{cases}$$

Hence

$$b_{j'}(m_i, n, p) = \prod_{k=0}^{n-1} y_{ik}$$

From the remarks preceding Lemma 2 it was seen that the final b coefficient vector for a function $b_i(n)$ was obtained by XORing (or summation modulo 2) the rows of matrix $B(n) = [b_{mj'}]$. □

Example 5

$$n = 4 \quad f = \Sigma(1, 5, 7, 12)$$

The minterm table representing this function is

$$M = i \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \overset{j'}{\begin{array}{cccc} 3 & 2 & 1 & 0 \end{array}} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \qquad Y = i \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \overset{j'}{\begin{array}{cccc} 3 & 2 & 1 & 0 \end{array}} \begin{bmatrix} \bar{p}_3 & \bar{p}_2 & \bar{p}_1 & p_0 \\ \bar{p}_3 & p_2 & \bar{p}_1 & p_0 \\ \bar{p}_3 & p_2 & p_1 & p_0 \\ p_3 & p_2 & \bar{p}_1 & \bar{p}_0 \end{bmatrix} = (y_{ij})$$

j		
0	$b_{15}(4, p) = \oplus \sum_{i=0}^{r-1} 1$	$j_0 = j_1 = j_2 = j_3 = 0$
1	$b_{14}(4, p) = y_{00} \oplus y_{10} \oplus y_{20} \oplus y_{30}$	$j_0 = 1, j_1 = j_2 = j_3 = 0$
2	$b_{13}(4, p) = y_{01} \oplus y_{11} \oplus y_{21} \oplus y_{31}$	$j_1 = 1, j_0 = j_2 = j_3 = 0$
3	$b_{12}(4, p) = y_{00}y_{01} \oplus y_{10}y_{11} \oplus y_{20}y_{21} \oplus y_{30}y_{31}$	$j_0 = j_1 = 1, j_2 = j_3 = 0$
	$\vdots$	
15	$b_0(4, p) = y_{00}y_{01}y_{02}y_{03} \oplus y_{10}y_{11}y_{12}y_{13} \oplus y_{20}y_{21}y_{22}y_{23} \oplus y_{30}y_{31}y_{32}y_{33}$	$j_0 = j_1 = j_2 = j_3 = 1$

An interesting feature of Lemma 1 is that if a boolean function f is mapped to a new function f' by EX-ORing each minterm by polarity p the cost structures (i.e., the number of terms in the RM expansion for each polarity) of the two functions are rearrangements of one another.

In addition any permutation of the minterm table representing a boolean function will merely rearrange the cost structures of the associated RM expansions.

These facts imply that large groups of functions are intimately related to one another.

Corollary 1

If a three-variable function has an odd number of minterms r and the minterm table consists of columns containing an odd number of ones then zero is an optimum polarity.

Proof

By assumption

$$\forall k \quad \oplus \sum_{i=0}^{r'} m_{ik} = 1$$

where $r' = r - 1$ r is odd this implies

$$\oplus \sum_{i=0}^{r'} \bar{m}_{ik} = 0$$

Now

$$b_7=1, \quad b_6=\oplus\sum_{i=0}^{r'}\bar{m}_{i0}=0$$

$$b_5=\oplus\sum_{i=0}^{r'}\bar{m}_{i1}=0$$

$$b_3=\oplus\sum_{i=0}^{r'}\bar{m}_{i2}=0$$

Therefore, the equations which determine how b_i varies with polarity p ($b_i(p)$) as seen in Example 3 reduce to

$$\begin{aligned} &b_7(p)=1 \quad && b_4(p)=b_4\oplus p_1p_0 \\ &b_6(p)=p_0 \quad && b_2(p)=b_2\oplus p_2p_0 \\ &b_5(p)=p_1 \quad && b_1(p)=b_1\oplus p_2p_1 \\ &b_3(p)=p_2 \quad && b_0(p)=b_0\oplus b_1p_0\oplus b_2p_1\oplus b_4p_2\oplus p_2p_1p_0 \end{aligned} \tag{23}$$

Clearly $p=0$ automatically generates three zeros in the set of six coefficients. The only way that $b_4(p)$, $b_2(p)$, $b_1(p)$ can be forced to zero if b_4, b_2 or b_1 is 1 is by setting two bits of the polarity p to 1, but this causes a loss of a zero overall as two of $\{b_6(p), b_5(p), b_3(p)\}$ become 1.

Simple arguments like this show that $p=0$ must be an optimum polarity. It is clear from Lemma 1 that an optimum polarity can be obtained for all three-variable functions having an odd number of minterms by using the following rule:

> EX-OR the columns of the minterm table representing the function and then EX-OR with 7 (bitwise). The answer obtained is an optimum polarity. □

The three-variable case where the function has an even number of minterms is dealt with in the following corollary.

Corollary 2

If a three-variable function has an even number of minterms then the bits p_k of the optimum polarities are defined as follows: b_i are the zero-polarity b-coefficients and b_c is defined by $b_c=4b_3+2b_5+b_6$ and the discriminant D is given by $D=b_1b_6\oplus b_2b_5\oplus b_3b_4$.

$b_c=0$ Solutions of $b_1p_0 \oplus b_2p_1 \oplus b_4p_2 = b_0$

$$b_c=1 \quad p_0=\begin{cases} b_0 \oplus b_2b_4 & D=1 \\ 0 \text{ or } 1 & D=0 \end{cases}, \quad p_1=b_4, \quad p_2=b_2$$

$$b_c=2 \quad p_0=b_4, \quad p_1=\begin{cases} b_0 \oplus b_1b_4 & D=1 \\ 0 \text{ or } 1 & D=0 \end{cases}, \quad p_2=b_1$$

$$b_c=3 \quad p_0=\begin{cases} b_0 \oplus b_2b_4 & D=1 \\ 0 \text{ or } 1 & D=0 \end{cases}, \quad p_1=\begin{cases} b_0 \oplus b_1b_4 & D=1 \\ b_4 \oplus p_0 & D=0 \end{cases}, \quad p_2=b_1 \text{ or } b_2$$

$$b_c=4 \quad p_0=b_2, \quad p_1=b_1, \quad p_2=\begin{cases} b_0 \oplus b_1b_2 & D=1 \\ 0 \text{ or } 1 & D=0 \end{cases}$$

$$b_c=5 \quad p_0=\begin{cases} b_0 \oplus b_2b_4 & D=1 \\ 0 \text{ or } 1 & D=0 \end{cases}, \quad p_1=b_1 \text{ or } b_4, \quad p_2=\begin{cases} b_0 \oplus b_2b_1 & D=1 \\ b_1 \oplus p_0 & D=0 \end{cases}$$

$$b_c=6 \quad p_0=b_2 \text{ or } b_4, \quad p_1=\begin{cases} b_0 \oplus b_1b_4 & D=1 \\ 0 \text{ or } 1 & D=0 \end{cases}, \quad p_2=\begin{cases} b_0 \oplus b_1b_2 & D=1 \\ b_1 \oplus p_1 & D=0 \end{cases}$$

$$b_c=7 \quad p_0=\begin{cases} 0 \text{ or } 1 \; (b_4 \oplus p_1, \; b_2 \oplus p_2) & D=1 \\ b_0 \oplus b_2b_4 & D=0 \end{cases}$$

$$p_1=\begin{cases} b_4 \oplus p_0 \; (0 \text{ or } 1, \; b_1 \oplus p_2) & D=1 \\ b_0 \oplus b_1b_4 & D=0 \end{cases}$$

$$p_2=\begin{cases} b_2 \oplus p_0 \; (b_1 \oplus p_1, \; 0 \text{ or } 1) & D=1 \\ b_0 \oplus b_1b_2 & D=0 \end{cases} \qquad (24)$$

Proof

Equations (14) give for three variables and an even number of minterms

$$b_7(p)=0 \qquad b_4(p)=b_4 \oplus b_5p_0 \oplus b_6p_1$$

$$b_6(p)=b_6 \qquad b_2(p)=b_2 \oplus b_3p_0 \oplus b_6p_2$$

$$b_5(p)=b_5 \qquad b_1(p)=b_1 \oplus b_3p_1 \oplus b_5p_2$$

$$b_3(p)=b_3 \qquad b_0(p)=b_0 \oplus b_1p_0 \oplus b_2p_1 \oplus b_3p_1p_0 \oplus b_4p_2 \oplus b_5p_2p_0 \oplus b_6p_2p_1 \qquad (25)$$

Since the equations on the left-hand side above are independent of polarity they can be ignored. The remaining equations are then analysed for all possible values taken by b_3, b_5, b_6. Consequently there are eight cases to be considered. However the symmetry of the equations mean that essentially only four cases are required. The cases are $b_c=0$, $b_c=1$, $b_c=3$, $b_c=7$.

Considering $b_c=0$, it is seen that the only bit $b_i(p)$ which varies with polarity is $b_0(p)$; consequently setting this to zero the first equation of (24) is obtained.

Now examine $b_c=1$: substituting into (25) gives

$$b_4(p)=b_4 \oplus p_1$$

$$b_2(p)=b_2 \oplus p_2$$

$$b_1(p)=b_1$$

$$b_0(p)=b_0 \oplus b_1 p_0 \oplus b_2 p_1 \oplus b_4 p_2 \oplus p_2 p_1$$

Clearly to obtain a maximum number of $b_i(p)$ to be zero

$$p_1=b_4, \quad p_2=b_2, \quad b_1 p_0 \oplus b_2 b_4=b_0$$

Hence the second equation of (24) is obtained.

Consider $b_c=3$: (25) becomes

$$b_4(p)=b_4 \oplus p_1 \oplus p_0$$

$$b_2(p)=b_2 \oplus p_2$$

$$b_1(p)=b_1 \oplus p_2$$

$$b_0(p)=b_0 \oplus b_1 p_0 \oplus b_2 p_1 \oplus p_2 p_0 \oplus b_4 p_2 \oplus p_2 p_1$$

Let $b_2=b_1$. Clearly it is favourable to set $p_2=b_2$. Hence the last of the equations becomes

$$b_0(p)=b_0 \oplus b_2 b_4$$

Now, noting the first equation above, the optimum solution is

$$p_0=0 \text{ or } 1, \quad p_1=b_4 \oplus p_0, \quad p_2=b_2$$

Now let $b_2 \neq b_1$. Either $p_2=b_2$ or $p_2=b_1$ can be chosen. Let $p_2=b_2$, then substituting into the last of the equations, we obtain

$$b_0(p)=b_0 \oplus b_2 b_4 \oplus p_0$$

Thus, an optimum is achieved when

$$p_0=b_0 \oplus b_2 b_4, \quad p_1=b_4 \oplus b_0 \oplus b_2 b_4=b_0 \oplus b_1 b_4$$

Similar arguments for the case when $p_2=b_1$ give us the fourth equation of (24).

Finally consider $b_c=7$. The equations become

$$b_4(p)=b_4 \oplus p_1 \oplus p_0$$

$$b_2(p)=b_2 \oplus p_2 \oplus p_0$$

$$b_1(p)=b_1 \oplus p_2 \oplus p_1$$

$$b_0(p)=b_0 \oplus b_1 p_0 \oplus b_2 p_1 \oplus b_4 p_2 \oplus p_0 p_1 \oplus p_0 p_2 \oplus p_1 p_2$$

As before the aim is to make as many of the $b_i(p)$ to be zero as possible. Hence from the first two equations:

$$p_1=b_4 \oplus p_0$$

$$p_2=b_2 \oplus p_0$$

Then

$$b_1 p=b_1 \oplus b_2 \oplus b_4$$

and

$$b_0 p=b_0 \oplus b_2 \oplus b_4 \oplus (1 \oplus b_1 \oplus b_2 \oplus b_4) p_0$$

Clearly

$$\text{if } b_1 \oplus b_2 \oplus b_4=1 \quad p_0=0 \text{ or } 1$$

and

$$\text{if } b_1 \oplus b_2 \oplus b_4=0 \quad p_0=b_0 \oplus b_2 b_4$$

makes $b_0(p)$ vanish.

Taking account of the symmetry of these equations it can be seen that the last of equations (24) is obtained. □

4. Exhaustive search algorithm

In this section an algorithm will be presented which determines the optimum polarity by exhaustive search. The basic idea of this algorithm is to calculate the costs of each RM expansion associated with every polarity. Equations (14) enable one to determine a formula for how a particular bit of the b-coefficient vector varies with polarity. For some functions the particular zero polarity b-coefficients b_j required for the formula for $b_i(p)$ are all zero. This means $b_i(p)=b_i$ and the algorithm does not have to calculate how $b_i(p)$ changes. In fact for functions of n variables having an even number of minterms,

$$b_i(p)=b_i \forall p \quad \textit{when } \bar{\imath}=0 \text{ or } 2^k,\ k=0,\ 1, \ldots, n$$

Thus for all functions of an even number of minterms of n variables there are at least n columns of the $b_i(p)$ matrix which are fixed. In addition (14) implies that for certain values of i, $b_i(p)$ for different polarities may be identical. This would occur when the bits of the polarity are missing in the formula for $b_i(p)$. This can be illustrated by examining $b_{10}(p)$ for a four-variable function.

According to (14)

$$b_{10}(p)=b_{10}\oplus b_{10+1}p_0\oplus b_{10+4}p_2\oplus b_{10+1+4}p_2p_0$$

$$b(p)=b_{10}\oplus b_{11}p_0\oplus b_{14}p_2\oplus b_{15}p_2p_0 \tag{26}$$

Since $b_{10}(p)$ only depends on bits p_0 and p_2 of p

$$b_{10}(0)=b_{10}(2)=b_{10}(8)=b_{10}(10)$$

$$b_{10}(1)=b_{10}(3)=b_{10}(9)=b_{10}(11)$$

$$b_{10}(4)=b_{10}(6)=b_{10}(12)=b_{10}(14)$$

$$b_{10}(5)=b_{10}(7)=b_{10}(13)=b_{10}(15) \tag{27}$$

Thus assuming b_{10} is known, $b_{10}(p)$ can be constructed for all values of p by calculating $b_{10}(11)$, $b_{10}(14)$, $b_{10}(15)$.

It was seen in (14) how the sequence of numbers k such that k AND $i=0$ generates the sequence of b-coefficients which are involved in the formula $b_i(p)$. In addition the sequence of numbers k' such that k' and $\bar{i}=0$ generates the polarities for which $b_i(p)$ is invariant. It is these two sequences of numbers which can be used to generate $b_i(p)$ for all p as seen above in (26)-(27).

From these arguments it is clear that the number n_i of $b_i(p)$ that have to be calculated to construct $b_i(p)\forall p$ is given by

$$n_i=2^j-1 \tag{28}$$

where j is the number of zero bits in integer i. To calculate $b_i(p)$ for all i and p, n_i has to be multiplied by the number of integers with j zeros (${}^nC_j$) and then summed over j. Thus the total number of $b_i(p)$ which have to be calculated N is given by:

$$N=\sum_{j=0}^{n}{}^nC_j(2^j-1)=3^n-2^n \tag{29}$$

Clearly it is advantageous to map the input logic function by a polarity which is known to give a significant increase in the number of zeros in the associated b-vector. This polarity can be furnished by any algorithm which computes a good polarity quickly (O[2^n]). The mapped function is then analysed by the exhaustive search and an optimum polarity obtained. The optimum polarity for the original function is then obtained by EX-ORing the aforementioned polarities. This technique allows a program to obtain a sub-optimum polarity quickly, advise as to the associated cost and then compute an optimum polarity in a shorter period of time than would have been taken if no sub-optimum mapping had been carried out. The algorithm which accomplishes this is set out below.

Algorithm

Step 1. Get the number of variables n.
Get the minterms.
Calculate maximum polarity (2^n-1).

Step 2. Calculate zero-polarity b-coefficient vector *bvec0_bits*.

Step 3. Calculate a sub-optimum polarity p_s and map *bvec0_bits* by this polarity to create a new *bvec0_bits*.

Step 4. For each bit of the b-coefficient vector, *examine_bit*, do the following:

(*a*) Calculate the array *formula_total* (k in equation (14) and the number of such terms, *num_terms_total*.

(*b*) Calculate the array *formula* which is a subset of *formula_total* in that terms involving b_j which are zero are not present. Obtain the number of such terms, *num_terms_formula*.

(*c*) Calculate the *copy_numbers* array which has *num_copies* elements. (These are the polarities which cause no change in $b_i(p)$).

(*d*) If *num_terms_formula* is not zero then do the following
 for each term in array *formula_total* EX-OR all the terms in formula together for the given *examine_bit* and obtain the result; increment the array of costs corresponding to each polarity using the *copy_numbers* array
else
 calculate the residual cost by summing *bvec0_bits* [*examine_bit*].

Step 5. Determine the polarity p_0 which gives the minimum cost from the complete costs array. The optimum polarity is $p_s \oplus p_0$.

Example 6

Let the logic function f be defined by the following minterms

$$f = \Sigma(3, 4, 6, 11, 13, 15)$$

The number of variables is 4. Maximum polarity is 15.
The zero-polarity b-vector is (*bvec0_bits*)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	1	1	1	0	1	0	0	0	0	1	0	0	0

Let the sub-optimal polarity predictor gives a good polarity $b_s = 8$.
The new *bvec0_bits* are

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0

Consider bit 3 of the b-vector (*examine_bit* = 3).

$$b_3(p) = b_3 \oplus b_7 p_2 \oplus b_{11} p_3 \oplus b_{15} p_2 p_3$$

$$= b_{3+0} \oplus b_{3+4} p_2 \oplus b_{3+8} p_3 \oplus b_{3+12} p_2 p_3 \qquad (30)$$

This equation is represented by the array *formula_total* with elements

$formula\_total\,[0]=0,\quad formula\_total\,[1]=4,\quad \ldots,\quad formula\_total\,[3]=12$

Now $b_3=1$, $b_7=1$, $b_{11}=0$, $b_{15}=0$ so

$$b_3(p)=1\oplus p_2 \tag{31}$$

This equation is represented by the array *formula* with elements

$$formula\,[0]=b_3 \text{ (by definition)}, formula\,[1]=4 \tag{32}$$

The *copy_numbers* array is generated in a similar way to *formula_total* but using the complement of 3(12). This gives:

copy_numbers [0] = 0
copy_numbers [1] = 1
copy_numbers [2] = 2
copy_numbers [3] = 3

Now the polarities for which the array *formula* has to be decoded are considered.

The general decoding procedure is as follows.

Count all elements of *formula* [i] $i>0$ such that polarity AND *formula* [i] $=$*formula* [i]. Add $b_{examine\_bit}$ (0) to the final count; if the answer is even then the final result is 0 otherwise it is 1.

Clearly $b_i(i)=b_i(0)$, hence
$b_3(3)=b_3$ (*formula_total* [0] + *copy_numbers* [3]) $=b_3=1$ *costs* [3] = *costs* [3] + 1
$b_3(2)=b_3$ (*formula_total* [0] + *copy_numbers* [2]) $=b_3=1$ *costs* [2] = *costs* [2] + 1
$b_3(1)=b_3$ (*formula_total* [0] + *copy_numbers* [1]) $=b_3=1$ *costs* [1] = *costs* [1] + 1
$b_3(0)=b_3$ by definition.

Next polarity 7 is considered (7 = *formula_tot* [1] + 3)
$b_3(7)=b_3$ (*formula_total* [1] + *copy_numbers* [3]) $=0$ *costs* [7] = *costs* [7] + 0
This result is calculated by decoding the information stored in *formula* (equations (31), (32)).

The remaining terms are then copied giving

$$b_3(6)=b_3\ (formula\_total\,[1]+copy\_numbers\,[2])=0$$

$$\vdots$$

$$b_3(4)=b_3\ (formula\_total\,[1]+copy\_numbers\,[0])=0$$

Also the costs array is updated.

Next polarities 11 (*formula_total* [2] + 3) and 15 (*formula_total* [3] + 3) are considered in a similar way and the costs associated with each polarity calculated.

So it is seen how the decoding of the array *formula* is only carried out for three polarities (7, 11, 15) and from these three calculations $b_3(p)$ is calculated for all polarities p.

It turns out for this example that 8 of the b-vector bits do not depend on polarity at all. For these bits a residual sum is calculated. The bits are 7, 9, 10, 11, 12, 13, 14, 15. The residual sum is the sum of the corresponding zero polarity b-vector bits. Hence the residual sum equals 1. Bits 0, 1, 2, 3, 4, 5, 6, and 8 do change with polarity according to the following equations.

$$b_0(p)=p_1p_0\oplus p_2p_3\oplus p_3p_2$$

$$b_1(p)=p_0\oplus p_2\oplus p_2p_1$$

$$b_2(p)=p_0\oplus p_2p_0$$

$$b_3(p)=1\oplus p_2$$

$$b_4(p)=p_0\oplus p_1p_0\oplus p_3$$

$$b_5(p)=1\oplus p_1$$

$$b_6(p)=p_0$$

$$b_8(p)=p_2$$

For this example when the costs are sorted it is found that the best polarity is 0. Consequently the optimum polarity for the original function is $8\oplus 0=8$.

It is possible to modify the above procedure so that no decoding of the formula array takes place and a weighted sum is calculated which relates to how many calculations will have to be performed. This is related to how long the full exhaustive search is likely to take. So potentially a program could give advice relating to the likely computation time. This can be very useful. The algorithm has been programmed in C and run on a HP 720 workstation. Some timing figures for exhaustive search are presented in the table for the function $f=\Sigma(0, 1, 2)$ for various numbers of variables. Times are also given for exhaustive search without employing a sub-optimum polarity predictor. The sub-optimum polarity predictor used was the Tabular Technique (Almaini *et al.* 1991).

Timings will vary with the function and with the accuracy of the non-exhaustive sub-optimum polarity predictor. However, the method presented in this paper compares favourably with other efficient exhaustive searches (Harking 1990) even in the worst possible cases and for some functions the algorithm proved to be orders of magnitude better. A great advantage of the method described in this paper is that it can be used 'in line' with other methods of predicting a good polarity so that an exhaustive search need only be undertaken if it is felt that the cost savings are great enough. Since the calculation of the b-coefficient vector bits are totally independent the algorithm would lend itself to a parallel processing technique.

Number of variables	CPU time (s)	CPU time (s)†
9	0·2	0·3
10	0·5	1·1
11	1·8	4·8
12	6·3	21·7
13	23·5	101·3
14	89·6	480·6
15	362·6	2383·9

†Without sub-optimum polarity predictor.

For incompletely specified functions the algorithm could be used in association with an algorithm which determines an optimum allocation of 'don't care' terms to produce a completely specified reduced function (McKenzie *et al.* 1993).

5. Conclusion

The concept of boolean polarity has been introduced. A new method for calculating how the coefficients of the product terms of the fixed polarity Reed–Muller expansions of boolean functions vary with polarity has been introduced. Exact formulae have been given for the optimum polarity (minimizing cost) for any three-variable logic functions. A new extremely efficient algorithm for calculating the optimum polarity for RM expansions has been presented. The algorithm can work 'in-line' with a fast method of finding a sub-optimum polarity. This has been found to further reduce the exhaustive search time. The authors believe the algorithms presented in this paper to be the fastest exhaustive search method available which takes a minterm specification of the logic function.

References

Akers, S. B., 1959, On a theory of boolean functions. *Journal of the SIAM*, 7, 487–498.

Almaini, A. E. A., 1989, *Electronic Logic Systems*, second edition (Prentice-Hall), Chap. 12.

Almaini, A. E. A., Thomson, P., and Hanson, D., 1991, Tabular techniques for Reed–Muller logic. *International Journal of Electronics*, **70**, 23–24.

Besslich, Ph. W., 1983, Efficient computer method for EX-OR logic design. *Proceedings of the Institution of Electrical Engineers*, Pt E, *Computers and Digital Techniques*, **130**, 203–206.

Green, D. H., 1986, *Modern Logic Design* (Addison-Wesley); 1987, Reed–Muller expansions of incompletely specified functions. *Proceedings of the Institution of Electrical Engineers*, Pt E, *Computers and Digital Techniques*, **134**, 228–236.

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

Harking, B., 1990, Efficient algorithm for canonical Reed–Muller expansion of boolean functions. *Proceedings of the Institution of Electrical Engineers*, Pt E, *Computers and Digital Techniques*, **137**, 366–370.

Lui, P. K., and Muzio, J. C., 1991, Boolean matrix transforms for the parity spectrum and minimisation of modulo_2 canonical expansions. *Proceedings of the Institution of Electrical Enghineers*, Pt E, *Computers and Digital Techniques*, **138**, 411–417.

McKenzie, L., Almaini, A. E. A., Miller, J. F., and Thomson, P., 1993, Optimisation of Reed–Muller Logic Functions. *International Journal of Electronics*, **75**, 451–466.

Reddy, S. M., 1972, Easily testable realisations for logic functions. *Transactions of the Institute of Electrical and Electronics Engineers*, **C21**, 1183–1188.

Tran, A., 1987, Graphical method for the conversion of minterms to Reed–Muller coefficients and the minimisation of EX-OR switching functions. *Proceedings of the Institution of Electrical Engineers*, Pt E, *Computers and Digital Techniques*, **134**, 93–99.

Wu, X., Chen, X., and Hurst, S. L., 1982, Mapping of Reed–Muller coefficients and the minimisation of EX-OR switching functions. *Proceedings of the Institution of Electrical Engineers*, Pt E, *Computers and Digital Techniques*, **129**, 15–20.