Generalised k-variable-mixed-polarity Reed–Muller expansions for system of Boolean functions and their minimisation

B.J.Falkowski and C.-H.Chang

Abstract: A lookup table based method to minimise generalised partially-mixed-polarity Reed–Muller (GPMPRM) expansions with k mixed polarity variables is presented. The developed algorithm can produce solutions based on the desired cost criteria for the systems of completely specified functions. A heuristic approach based on the exclusion rule is adopted to extract the best dual polarity variables from any fixed polarity Reed–Muller (FPRM) expansion. The obtained experimental results compared favourably with the recently published results and outperform those generated by the exact minimal FPRM expansion minimisers.

1 Introduction

The classical approach to analysis, synthesis and testing of digital circuits is based on the description by the operators of Boolean algebra. However, for many years, an alternative representation based on the operations of modulo-2 arithmetic has been developed [1–25]. The algebra corresponding to this second approach, being an example of a Galois field (GF), supports such familiar methods for digital signal processing operations as matrices and fast transforms [5, 7, 9–11, 13]. Any Boolean function can be represented in the modulo-2 algebra. The modulo-2 sum-of-products expression is known in the literature [2–12, 16–25] as the complement-free ring sum or Reed–Muller expansion. For a given Boolean function, each Reed–Muller expansion is unique and is its canonical form.

It has long been known that, for some applications, the logic circuits using exclusive OR (EXOR) gates are more economical than the design based on other gates. Such situations happen frequently for many useful functions applied in arithmetic and telecommunication circuits, having a high content of so-called linear part (EXOR part of the function). Some of the examples of such functions are adders and parity checkers. With the advent of cellular field programmable gate arrays (FPGAs) and the introduction of new programmable logic devices (PLDs), for example, Xilinx lookup-based and Actel 1020 multiplexer-based FPGAs, and Signetics LHS501 folded NAND devices, propagation delay and gate area are no longer major concerns in exclusive sum-of-products (ESOP) [5, 11, 24] implementation of logic circuits. What is more, the circuits built around the EXOR gates are easily testable [16, 18]. Fault detection of any logical circuit by verification of its Reed–Muller coefficients was considered in [16]. The upper bound on the number of Reed–Muller coefficients to be verified for detection of all multiple terminal stuck-at-faults and all single input bridging faults is shown to be n, where n is the number of input variables [16]. Recently, important problems of Boolean matching and symmetry detection were solved in the Reed–Muller domain [22, 23].

Unfortunately, the ESOP of a Boolean function exists in many forms, and exact minimal solutions have been found practically only for functions with less than six variables [15]. Special interest and attention have been focused on two of the canonical subfamilies of ESOP, the fixed polarity Reed–Muller (FPRM) expansion [5–10, 12, 15, 17, 18, 21, 23] and the Kronecker Reed–Muller (KRM) expansion [11, 13, 24]. The former has 2^n alternative forms and the latter has 3^n alternative forms. Owing to the high computation complexity, no exact minimisation technique for a canonical form more general than KRM expansion has been proposed [24]. There is another Reed–Muller canonical expansion known as the generalised Reed–Muller (GRM) expansion which consists of a total of $2^{n2^{n-1}}$ alternative forms [3, 5, 8, 10, 11, 22, 24, 25]. GRM expansion can be considered as a combination of two subfamilies of Reed–Muller expansions: the inconsistent mixed polarity Reed–Muller (IMPRM) expansion [3] and the FPRM expansion. Although minimal GRM expansion is expected to be closer to the minimal ESOP than the minimal KRM expansion, due to the greater number of alternative forms, an exhaustive search for a minimal GRM is also computationally unfeasible even for a very small number of variables [5, 10, 11, 22, 24]. Recently, Wu *et al.* and Zeng *et al.* [24, 25] proposed another subfamily of GRM called the generalised partially mixed-polarity Reed–Muller (GPMPRM) expansion. GPMPRM is a superset of FPRM, which has $n2^{n-1}2^{2^{n-1}} - (n-1)2^n$ alternative forms. Based on the number of alternative forms, it is believed that the minimal GPMPRM expansion is still much closer to the minimal ESOP than the minimal KRM expansion.

The definition of GPMPRM expansion from [24] with only one mixed polarity variable was extended to k mixed polarity variables in [25]. It should be stressed, however, that the authors of [25] have not found an efficient exact algorithm for $k > 1$, the task which is solved in the current

IEE Proceedings online no. 20000588

DOI: 10.1049/ip-cds:20000588

Paper first received 11th May 1998 and in revised form 22nd March 2000

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Republic of Singapore

paper. The extension to k mixed polarity variables further reduces the gap between the minimal GPMPRM and the minimal ESOP since the total number of alternative forms is ${}^nC_k 2^{n-k} 2^{k2^{n-1}} - ({}^nC_k - 1)2^n$ based on the new definition, where ${}^nC_k$ is the number of combinations of selecting k out of n objects. The lookup table based approach for the exact minimisation of FPRM expansion developed by the authors in [2, 7] is modified to generate a minimal GPMPRM expansion of k mixed polarity variables. We also show that the method based on the exclusion rule used for the extraction of single mixed polarity variable [14] can also be applied successfully to the general case of k mixed polarity variables. Heuristic minimisation of GPMPRM expansions for multiple output Boolean functions is also presented. For a system of Boolean functions, the direct minimisation of GPMPRM expansion for each output independently which can be performed easily by the single output minimisation algorithm would not maximise the possibility of shared products or literals by different outputs. The method proposed for the minimisation of GPMPRM expansions for multiple output functions has considered the reduction of total number of unique products or literals by appropriate choice of polarities. To speed up the processing time, a quasi-minimisation method is proposed by assuming that the same Reed–Muller product term of different outputs has identical polarities for the corresponding variables. Contrary to all algorithms known from the literature, our algorithm for the minimisation of GPMPRM expansions is adaptable to different cost criteria, for instance, the total number of unique products, the total number of unique literals and the linear combination of both criteria. Based on the size of the tackled problem, our algorithm can use a different size of the lookup table to trade the space complexity problem into the processing time complexity problem. Experimental results show that, even without considering all possible combinations of k variables as the mixed polarity variables, for most functions reported in [24], which considered all combinations of one mixed polarity variable, the quality of the results obtained by our algorithm is either the same or better. Moreover, our algorithm for multiple output function requires a much lower computation time than that of [24] which minimises only a selected output of the same function.

2 Basic definitions

An n-variable Boolean function can be expressed as a canonical Reed–Muller expansion [2–12, 16–25] of 2^n terms as follows:

$$F(x_n, x_{n-1}, \ldots, x_1) = \bigoplus_{j=0}^{2^n-1} a_j \prod_{i=1, j_i=1}^{n} x_i^{\omega_i} \qquad (1)$$

where $\oplus$ denotes the modulo-2 addition, $a_j \in (0, 1)$ is called a Reed–Muller coefficient and $j_i \in (0, 1)$ is the ith bit of the binary representation of j, with j_i being the least significant digit. $\omega_i \in (0, 1)$ is the polarity bit of the variable x_i. $x_i^{\omega_i} = x_i$ when $\omega_i = 0$, and $x_i^{\omega_i} = \bar{x}_i$ when $\omega_i = 1$. When each literal ($x_i^{\omega_i}$, $i = 1, 2, \ldots, n$) throughout the expression (eqn. 1) has a consistent polarity bit value, such an expression is known as a fixed polarity Reed–Muller (FPRM) expansion [2, 5–10, 12, 15, 17–19, 21, 23]. If all the literals in eqn. 1 can have either polarity bit value in any combination, it is known as the generalised Reed–Muller (GRM) expansion [3, 5, 10, 11, 22, 24]. Since there are $n2^{n-1}$ literals in the complete expression (eqn. 1), there are $2^{n2^{n-1}}$ possible GRM expansions, including 2^n FPRM expansions.

In [24], a strong constraint is placed in the definition of GRM expansion to obtain the generalised partially mixed-polarity Reed–Muller (GPMPRM) expansion. It is a subset of GRM expansions that encloses the FPRM expansions. However, the requirement that the polarities of all but one variable have to be fixed can be relaxed. A more general definition of GPMPRM expansion, the possibility which was mentioned in [25], is given as follows:

Definition 1: The generalised partially mixed-polarity Reed–Muller (GPMPRM) expansions are obtained by allowing the $k2^{n-1}$ literals of k variables in expression (eqn. 1) to freely assume either polarity while maintaining consistent fixed polarities for all the literals of the remaining variables.

Under this new definition, for an n-variable completely specified Boolean function, there are ${}^nC_k 2^{n-k} 2^{k2^{n-1}} - ({}^nC_k - 1)2^n$ alternative GPMPRM forms. The proof of this fact can be constructed in a similar way to that shown in [24].

Definition 2: The first-order Boolean derivative (also known as Boolean difference) of an n-variable Boolean function is defined as [1]:

$$\frac{\partial F(X)}{\partial x_i} = F_{\bar{x}_i} \oplus F_{x_i} = F(x_n, \ldots, x_{i+1}, x_i = 0, x_{i-1}, \ldots x_1) \oplus F(x_n, \ldots, x_{i+1}, x_i = 1, x_{i-1}, \ldots x_1) \qquad (2)$$

Since $\partial F(X)/\partial x_i$ is itself a $(n-1)$-variable Boolean function, a higher-order derivative can be similarly defined. Hence the kth derivative is:

$$\frac{\partial^k F(X)}{\partial x_1 \partial x_2 \ldots \partial x_k} = \frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_{k-1}}\left(\cdots \frac{\partial}{\partial x_2}\left(\frac{\partial F(X)}{\partial x_1}\right)\right)\right) \qquad (3)$$

It should be noted that the order of evaluation of the higher-order derivative is unimportant, i.e.

$$\frac{\partial^2 F(X)}{\partial x_i \partial x_j} = \frac{\partial^2 F(X)}{\partial x_j \partial x_i}$$

Based on the definition of Boolean derivative, the FPRM expansion of a Boolean function $F(X)$ in a fixed polarity number $\omega = \omega_n \omega_{n-1} \ldots \omega_2 \omega_1$ can be expressed as:

$$F(X) = F(X)|_{X=\omega} \oplus \left.\frac{\partial F(X)}{\partial x_1}\right|_{X=\omega} \cdot x_1^{\omega_1} \oplus \left.\frac{\partial F(X)}{\partial x_2}\right|_{X=\omega} \cdot x_2^{\omega_2} \oplus \left.\frac{\partial^2 F(X)}{\partial x_1 \partial x_2}\right|_{X=\omega} \cdot x_1^{\omega_1} x_2^{\omega_2} \oplus \ldots \oplus \left.\frac{\partial^n F(X)}{\partial x_1 \partial x_2 \ldots \partial x_n}\right|_{X=\omega} \cdot x_1^{\omega_1} x_2^{\omega_2} \ldots x_n^{\omega_n} \qquad (4)$$

Comparing eqns. 1 and 4, we have:

$$a_j = \left.\frac{\partial^{\|j\|} F(X)}{\partial Z}\right|_{X=\omega} \qquad (5)$$

where $\partial Z = \Pi_{i=1, j_i=1}^{n} \partial x_i$, and the symbol $X = \omega$ denotes that the function $F(X)$ or Boolean derivative is evaluated for the set of variables $x_n x_{n-1} \ldots x_1 = \omega_n \omega_{n-1} \ldots \omega_1$. The symbol $\|j\|$ is the Hamming weight of the integer j, and it represents the number of 1s in the binary representation of j.

The ordered set of all 2^n FPRM expansion coefficients $[a_0\ a_1 \ldots a_{2^n-1}]$ in some chosen polarity number ω is called the polarity vector, denoted by $A^{\omega}(F)$. $A^{\omega}(F)$ can also be represented by a decimal number $\Sigma_{i=0}^{2^n-1} a_i \times 2^i$ with a_0 as the least significant bit and a_{2^n-1} the most significant bit of its binary equivalent. By arranging the 2^n polarity vectors $A^{\omega}(F)$ in ascending order of ω, a polarity coefficient matrix $PC(F)$ [2, 7, 9] is formed. The element a_{ij} in row i ($i = 0, 1, ..., 2^n - 1$) and column j ($j = 0, 1, ..., 2^n - 1$) of the polarity coefficient matrix $PC(F)$ is the coefficient a_j of the FPRM expansion with polarity number $\omega = i$.

The goal of this paper is to find efficiently an optimal GPMPRM expansion with a minimal number of products and literals.

3 Minimisation of GPMPRM expansions

Recently, an algorithm has been developed that utilises only a subset of Walsh coefficients to reveal all the information carried by the polarity coefficient matrix of any three variable Boolean functions [6]. Each class of the functions is associated with a specific subroutine that computes the optimal polarities, optimal weights, optimal fixed polarity Reed–Muller expansions etc. without resorting to an exhaustive search. Direct extension of the method in [6] to handle larger Boolean functions with the number of variables $n > 3$ is unmanageable due to the increasing number of different classes. Nevertheless, exact optimal generation of FPRM expansions for large n have been solved by reducing the polarity coefficient matrix into submatrices of smaller dimension such that each submatrix is a polarity coefficient matrix of a subfunction obtained by either Shannon's decomposition or Boolean difference with respect to some variables [2, 7]. A similar approach to [2] can be applied to the minimisation of GPMPRM expansions by selecting an optimal FPRM expansion for each subfunction, with the exception that the k mixed polarity variables may have different polarities for different subfunctions.

Lemma 1: The polarity coefficient matrix $PC(F)$ of an n-variable completely specified Boolean function $F(X)$, can be partitioned into four submatrices of order 2^{n-1} as [2, 7, 9]:

$$PC(F) = \begin{bmatrix} PC\left(F_{\bar{x}_n}\right) & PC\left(\frac{\partial F(X)}{\partial x_n}\right) \\ PC\left(F_{x_n}\right) & PC\left(\frac{\partial F(X)}{\partial x_n}\right) \end{bmatrix} \quad (6)$$

Let us notice that

$$PC\left(\frac{\partial F(X)}{\partial \bar{x}_n}\right) = PC\left(\frac{\partial F(X)}{\partial x_n}\right)$$

In general, we can apply eqn. 2 recursively to partition the polarity coefficient matrix of order 2^n into q^2 submatrices of order 2^k, where $q = 2^{n-k}$, i.e.

$$PC(F) = \begin{bmatrix} PC\left(f_{0,0}\right) & PC\left(f_{0,1}\right) & \ldots PC\left(f_{0,q-1}\right) \\ PC\left(f_{1,0}\right) & PC\left(f_{1,1}\right) & \ldots PC\left(f_{1,q-1}\right) \\ \vdots & \vdots & \vdots \\ PC(f_{q-1,0}) & PC(f_{q-1,1}) & \ldots PC(f_{q-1,q-1}) \end{bmatrix} \quad (7)$$

where the k-variable subfunction, $f_{i,0} = F_Y$ and $f_{i,j} = \partial^{\|j\|}F_Y/\partial Z$ for all $i = 0, 1, ..., q - 1$ and $j = 1, ..., q - 1$. Y is the set of literals $x_{k+r}^{1-i_r}$ for all values of $r \in \{1, 2, ..., n - k\}$ satisfying $j_r = 0$, when i and j are expressed as binary $(n - k)$-tuples. Similarly, Z is the set of variables x_{k+r} for all values of $r \in \{1, 2, ..., n - k\}$ satisfying $j_r = 1$.

Each submatrix in $PC(F)$ is a polarity coefficient matrix of a k-variable subfunction. Furthermore, the subfunction, $f_{i,j} = f_{l,j}$ when $i_r = l_r$ for all $r = 1, 2, ..., n - k$ satisfying $j_r = 0$. The total number of unique subfunctions is equal to 3^{n-k}.

Example 1: Consider the five variable Boolean function $F(x_5, x_4, x_3, x_2, x_1) = \Sigma m(8, 10, 11, 16, 17, 19, 23, 24, 26, 27) = x_4\bar{x}_3\bar{x}_1 \vee x_4\bar{x}_3x_2x_1 \vee x_5\bar{x}_4\bar{x}_3\bar{x}_2 \vee x_5\bar{x}_4x_2x_1$. Applying eqn. 7 with $k = 3$, we have:

$$f_{0,0} = F_{\bar{x}_5\bar{x}_4} = 0.\ f_{1,0} = F_{\bar{x}_5x_4} = \bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1$$

$$f_{2,0} = F_{x_5\bar{x}_4} = \bar{x}_3\bar{x}_2 \vee x_2x_1$$

$$f_{3,0} = F_{x_5x_4} = \bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1 = f_{1,0}$$

$$f_{0,1} = f_{1,1} = \frac{\partial F_{\bar{x}_5}}{\partial x_4} = \bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1$$

$$f_{2,1} = f_{3,1} = \frac{\partial F_{x_5}}{\partial x_4} = \bar{x}_3\bar{x}_2x_1 \vee \bar{x}_3x_2\bar{x}_1 \vee x_3x_2x_1$$

$$f_{0,2} = f_{2,2} = \frac{\partial F_{\bar{x}_4}}{\partial x_5} = \bar{x}_3\bar{x}_2 \vee x_2x_1$$

$$f_{1,2} = f_{3,2} = \frac{\partial F_{x_4}}{\partial x_5} = 0$$

$$f_{0,3} = f_{1,3} = f_{2,3} = f_{3,3} = \frac{\partial^2 F(X)}{\partial x_5 \partial x_4} = \bar{x}_3\bar{x}_2 \vee x_2x_1$$

$$PC(F) = \begin{bmatrix} PC(0) & PC(\bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1) & PC(\bar{x}_3\bar{x}_2 \vee x_2x_1) & PC(\bar{x}_3\bar{x}_2 \vee x_2x_1) \\ PC(\bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1) & PC(\bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1) & PC(0) & PC(\bar{x}_3\bar{x}_2 \vee x_2x_1) \\ PC(\bar{x}_3\bar{x}_2 \vee x_2x_1) & PC(\bar{x}_3\bar{x}_2x_1 \vee \bar{x}_3x_2\bar{x}_1 \vee x_3x_2x_1) & PC(\bar{x}_3\bar{x}_2 \vee x_2x_1) & PC(\bar{x}_3\bar{x}_2 \vee x_2x_1) \\ PC(\bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1) & PC(\bar{x}_3\bar{x}_2x_1 \vee \bar{x}_3x_2\bar{x}_1 \vee x_3x_2x_1) & PC(0) & PC(\bar{x}_3\bar{x}_2 \vee x_2x_1) \end{bmatrix}$$

Considering the second row ($i = 1$), the eight possible FPRM expansions can be written as:

$$PC\left(\bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1\right) \oplus \bar{x}_4 PC\left(\bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1\right) \oplus x_5 PC(0) \oplus x_5\bar{x}_4 PC\left(\bar{x}_3\bar{x}_2 \vee x_2x_1\right)$$

A GPMPRM expansion with a lower implementation cost than these eight FPRM expansions can be obtained by selecting the minimal FPRM expansion for each of the subfunctions $\bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1$, $\bar{x}_3\bar{x}_1 \vee \bar{x}_3x_2x_1$ and $\bar{x}_3\bar{x}_2 \vee x_2x_1$. The GPMPRM is given by:

$$\begin{aligned} F(X) &= \left(\bar{x}_3 \oplus \bar{x}_3\bar{x}_2x_1\right) \oplus \bar{x}_4\left(\bar{x}_3 \oplus \bar{x}_3\bar{x}_2x_1\right) \\ &\oplus x_5\bar{x}_4\left(x_2x_1 \oplus \bar{x}_3 \oplus \bar{x}_3x_2\right) \\ &= \bar{x}_3 \oplus \bar{x}_3\bar{x}_2x_1 \oplus \bar{x}_4\bar{x}_3 \oplus \bar{x}_4\bar{x}_3\bar{x}_2x_1 \\ &\oplus x_5\bar{x}_4x_2x_1 \oplus x_5\bar{x}_4\bar{x}_3 \oplus x_5\bar{x}_4\bar{x}_3x_2 \end{aligned}$$

From the above example, a GPMPRM expansion can be derived from each row of the decomposed polarity coefficient matrix given by eqn. 7. For each value of row index i, the most significant $n - k$ variables in Y form the fixed polarity variables of the GPMPRM expansion. Their polarities are determined by the corresponding bit values of i, where i is expressed as a binary $(n - k)$-tuple. The least significant k variables have mixed polarity. The fixed polarity variables associated with the submatrix $PC(f_{i,j})$ are given by $\Pi_{1 \le r \le n-k, j_r=1} x_{k+r}^{i_r}$. The GPMPRM expansion generated in this manner has the advantage that the polarities of the fixed polarity variables need to be specified only once, and only k polarity bits of the mixed polarity variables need to be specified for every value of j.

For simplicity, in the sequel, we assume that the k mixed polarity variables of a GPMPRM expansion are $x_1, x_2, ..., x_k$ and the fixed polarity variables are $x_{k+1}, x_{k+2}, ..., x_n$. Selection of a different set of mixed polarity variables affects both the indexing of the variables, which can be corrected by reordering the input variables of the function, and the cost of the final GPMPRM expansion. A heuristic approach for selecting the best mixed polarity variables will be presented in the next Section.

Theorem 1: Let $w_p(f, \phi)$ be the number of product terms of the minimal FPRM expansion of the k-variable function f with polarity number $\phi = \omega_k\ \omega_{k-1}\ ...\ \omega_1$. The number of product terms, $w_p(F)$ of the minimal GPMPRM expansion for an n-variable Boolean function F, generated from eqn. 7 is given by:

$$w_p(F) = \min_{0 \le i < q} \left[\sum_{j=0}^{q-1} \left(\min_{0 \le \phi_j < 2^k} w_p(f_{i,j}, \phi_j) \right) \right] \quad (8)$$

where $q = 2^{n-k}$.

Proof: Let $\phi_{j\min}$ be an optimal polarity of an FPRM expansion of $PC(F)$. Then, $w_p(f_{i,j}, \phi_{j\min}) = \min_{0 \le \phi_j < 2^k} w_p(f_{i,j}, \phi_j)$. By replacing $PC(f_{i,j})$ for all $i, j = 0, 1, ..., q-1$ in eqn. 7 with their respective optimal polarity FPRM expansions, a GPMPRM expansion with fixed polarity number $i = \omega_n\ \omega_{n-1}\ ...\ \omega_{k+1}$ is derived. The number of product terms of this GPMPRM expansion is given by $\Sigma_{j=0}^{q-1} w_p(f_{i,j}, \phi_{j\min})$. Since there are q different polarity numbers for the $(n-k)$ fixed polarity variables, the number of product terms of the minimal GPMPRM expansion is given by eqn. 8.

Theorem 2: Let $w_l(f, \phi)$ be the number of product terms of the minimal FPRM expansion of the k-variable function f with polarity number $\phi = \omega_k\ \omega_{k-1}\ ...\ \omega_1$. The number of literals, $w_l(F)$ of the minimal GPMPRM expansion for an n-variable Boolean function F, generated from eqn. 7 is given by:

$$w_l(F) = \min_{0 \le i < q} \left\{ \sum_{j=0}^{q-1} \left[\min_{0 \le \phi_j < 2^k} \left(w_l(f_{i,j}, \phi_j) + \|j\| \times w_p(f_{i,j}, \phi_j) \right) \right] \right\} \quad (9)$$

where $q = 2^{n-k}$ and $\|j\|$ is the Hamming weight of j.

Proof: Each submatrix $PC(f_{i,j})$ has the dimension of $2^k \times 2^k$. The number of literals contributed by any polarity vector $A^{\phi_j}(f_{i,j})$ of $PC(f_{i,j})$ for $0 \le \phi_j < 2^k$, is given by $w_l(f_{i,j}, \phi_j)$. For $0 \le j < q$, the product terms in the GPMPRM expansion are contributed by the conjunction of the fixed polarity term $\Pi_{1 \le r \le n-k, j_r = 1}\ x_{r+k}^{i_r}$ and the optimal polarity vector of $PC(f_{i,j})$. Since the number of literals in the fixed polarity term is equal to $\|j\|$, the total number of literals contributed by each subfunction $f_{i,j}$ is equal to $\min_{0 \le \phi_j < 2^k}(w_l(f_{i,j}, \phi_j) + \|j\| \times w_p(f_{i,j}, \phi_j))$. Therefore, the number of literals of a GPMPRM expansion with fixed polarity number $i = \omega_{n-k}\ \omega_{n-k-1}\ ...\ \omega_{k+1}$ derived in this manner is given by

Table 1: Lookup table, $\{\phi_{\min}(f_{i,j}), w_{p\min}(f_{i,j}), A^{\phi_{\min}}(f_{i,j})\}$ for $k = 3$

{0, 0, 0}	{7,1, 128}	{6, 1, 128}	{6, 1, 64}	{5, 1, 128}	{5, 1, 32}	{4, 2, 96}	{4, 2, 144}
{4, 1, 128}	{5, 2, 96}	{4, 1, 32}	{5, 2, 144}	{4, 1, 64}	{6, 2, 144}	{7, 2, 144}	{4, 1, 16}
{3, 1, 128}	{3, 1, 8}	{2, 2, 72}	{2, 2, 132}	{1, 2, 40}	{1, 2, 130}	{0, 4, 150}	{7, 3, 104}
{0, 4, 120}	{0, 4, 135}	{0, 4, 210}	{6, 3, 44}	{0, 4, 180}	{5, 3, 74}	{7, 2, 24}	{3, 3, 145}
{2, 1, 128}	{3, 2, 72}	{2, 1, 8}	{3, 2, 132}	{0, 4, 108}	{0, 4, 147}	{0, 4, 198}	{6, 3, 56}
{0, 2, 40}	{1, 4, 150}	{0, 2, 130}	{6, 3, 104}	{0, 4, 228}	{6, 2, 24}	{4, 3, 74}	{2, 3, 145}
{2, 1, 64}	{6, 2, 132}	{7, 2, 132}	{2, 1, 4}	{0, 4, 156}	{3, 3, 98}	{7, 2, 36}	{5, 3, 133}
{0, 4, 216}	{6, 2, 36}	{2, 3, 98}	{4, 3, 133}	{0, 2, 20}	{7, 3, 148}	{6, 3, 148}	{0, 2, 65}
{1, 1, 128}	{3, 2, 40}	{0, 4, 106}	{0, 4, 149}	{1, 1, 8}	{3, 2, 130}	{0, 4, 166}	{5, 3, 88}
{0, 2, 72}	{2, 4, 150}	{0, 4, 226}	{5, 2, 24}	{0, 2, 132}	{5, 3, 104}	{4, 3, 44}	{1, 3, 145}
{1, 1, 32}	{5, 2, 130}	{0, 4, 154}	{3, 3, 100}	{7, 2, 130}	{1, 1, 2}	{7, 2, 66}	{6, 3, 131}
{0, 4, 184}	{5, 2, 66}	{0, 2, 18}	{7, 3, 146}	{1, 3, 100}	{4, 3, 131}	{5, 3, 146}	{0, 2, 33}
{0, 2, 96}	{3, 4, 232}	{0, 4, 202}	{3, 2, 36}	{0, 4, 172}	{3, 2, 66}	{0, 2, 6}	{7, 3, 134}
{0, 4, 232}	{1, 3, 22}	{0, 2, 66}	{1, 5, 233}	{0, 2, 36}	{1, 5, 158}	{4, 3, 134}	{1, 3, 97}
{0, 2, 144}	{3, 3, 104}	{2, 3, 56}	{1, 3, 133}	{1, 3, 88}	{2, 3, 131}	{3, 3, 134}	{0, 2, 9}
{0, 2, 24}	{1, 5, 182}	{2, 3, 146}	{1, 3, 73}	{1, 3, 148}	{2, 3, 41}	{1, 5, 107}	{0, 2, 129}
{0, 1, 128}	{1, 4, 106}	{2, 2, 40}	{1, 4, 149}	{1, 2, 72}	{1, 4, 226}	{3, 4, 150}	{4, 2, 24}
{0, 1, 8}	{1, 4, 166}	{2, 2, 130}	{4, 3, 88}	{1, 2, 132}	{4, 3, 56}	{4, 3, 104}	{0, 3, 145}
{1, 2, 96}	{1, 4, 202}	{2, 4, 232}	{2, 2, 36}	{1, 4, 232}	{1, 2, 66}	{0, 3, 22}	{0, 5, 233}
{1, 4, 172}	{1, 2, 6}	{2, 2, 66}	{6, 3, 134}	{1, 2, 36}	{5, 3, 134}	{0, 5, 158}	{0, 3, 97}
{0, 1, 32}	{1, 4, 154}	{4, 2, 130}	{2, 3, 100}	{1, 4, 184}	{1, 2, 18}	{4, 2, 66}	{6, 3, 146}
{6, 2, 130}	{6, 2, 66}	{0, 1, 2}	{6, 3, 194}	{0, 3, 100}	{4, 3, 146}	{4, 3, 194}	{1, 2, 33}
{1, 2, 144}	{2, 3, 44}	{2, 3, 104}	{0, 3, 133}	{1, 2, 24}	{3, 3, 146}	{0, 5, 182}	{0, 3, 73}
{0, 3, 88}	{2, 3, 134}	{2, 3, 194}	{1, 2, 9}	{0, 3, 148}	{0, 5, 107}	{3, 3, 41}	{1, 2, 129}
{0, 1, 64}	{2, 4, 156}	{2, 4, 216}	{2, 2, 20}	{4, 2, 132}	{1, 3, 98}	{4, 2, 36}	{5, 3, 148}
{5, 2, 132}	{5, 2, 36}	{0, 3, 98}	{4, 3, 148}	{0, 1, 4}	{5, 3, 164}	{4, 3, 164}	{2, 2, 65}
{2, 2, 144}	{1, 3, 74}	{2, 2, 24}	{3, 3, 148}	{1, 3, 104}	{0, 3, 131}	{0, 5, 214}	{0, 3, 41}
{0, 3, 56}	{1, 3, 134}	{0, 3, 146}	{0, 5, 109}	{1, 3, 164}	{2, 2, 9}	{3, 3, 73}	{2, 2, 129}
{3, 2, 144}	{3, 2, 24}	{0, 3, 74}	{2, 3, 148}	{0, 3, 44}	{1, 3, 146}	{0, 3, 134}	{0, 5, 121}
{0, 3, 104}	{0, 5, 151}	{0, 3, 194}	{1, 3, 41}	{0, 3, 164}	{2, 3, 73}	{3, 2, 9}	{3, 2, 129}
{0, 1, 16}	{3, 3, 152}	{2, 3, 152}	{4, 2, 65}	{1, 3, 152}	{4, 2, 33}	{5, 3, 97}	{4, 2, 129}
{0, 3, 152}	{4, 3, 97}	{5, 2, 33}	{5, 2, 129}	{6, 2, 65}	{6, 2, 129}	{7, 2, 129}	{0, 1, 1}

$$\sum_{j=0}^{q-1}\left[\min_{0\le\phi_j<2^k}\left(w_l\left(f_{i,j},\phi_j\right)+\|j\|\times w_p\left(f_{i,j},\phi_j\right)\right)\right]$$

Eqn. 9 is obtained by selecting the expansion with the smallest number of literals among the expansions of q different fixed polarities.

For a small number of mixed polarity variables k, typically $k \le 5$, a precomputed lookup table consisting of the fields of $w_{\min}(f_{i,j})$,$\phi_{\min}(f_{i,j})$ and $A^{\phi_{\min}}(f_{i,j})$ can be used. If the minimisation objective is the number of product terms, $w_{\min}(f_{i,j}) = \min_{0\le\phi<2^k} w_p(f_{i,j}, \phi)$ and $\phi_{\min}(f_{i,j})$ and $A^{\phi_{\min}}(f_{i,j})$ are the corresponding optimal polarity number and optimal polarity vector of the subfunction $f_{i,j}$, respectively. Each polarity vector in the lookup table is represented by a decimal number $\Sigma_{r=0}^{2^k-1} 2^r a_r$ where a_r is the rth coefficient of the expansion. The same decimal number representation is used to uniquely characterise the index to the lookup table $f_{i,j}$. In the continuation, the same symbol $f_{i,j}$ will be used to denote either the Boolean function or its decimal equivalent. That is

$$f_{i,j}=\sum_{r=0}^{2^k-1}2^r m_r$$

where $m_r \in (0, 1)$ is the value of the rth minterm of the k-variable subfunction, $f_{i,j}$. Based on this representation $\overline{f_{i,j}} = 2^{2^k} - 1 - f_{i,j}$ and all subfunctions $f_{i,j}$ from $PC(F)$ for $j \neq 0$ can be calculated by the modulo-2 sum of $\oplus_{k\subseteq_0 j}\, f_{k\oplus i,0}$, where the notation $k \subseteq_0 j$ means the set of integers k belongs to the zero subnumber of j [9]. The conditions for which $k \subseteq_0 j$ are given by $k_r = 0$ if $j_r = 0$ and $k_r = 0$ or 1 if $j_r = 1$ for all values of $r = 1, 2, ..., n - k$. The lookup table for $k = 3$ is shown in Table 1. In Table 1, the entries are arranged in ascending order of $f_{i,j}$, from left to right of each row. The leftmost entry in the first row corresponds to $f_{i,j} = 0$, and the leftmost entry in the second row corresponds to $f_{i,j} = 8$ etc. If the minimisation objective is the number of literals, an additional field $\min_{0\le\phi<2^k} w_l(f_{i,j}, \phi)$ is included in the lookup table. Since $A^{\phi_j}(\overline{f_{i,j}})$ and $A^{\phi_j}(f_{i,j})$ differ only in the constant term a_0, $w_l(\overline{f_{i,j}}, \phi_j) = w_l(f_{i,j}, \phi_j)$. However, from theorem 2, the constant term of each subfunction contributes $\|j\|$ literals to the final GPMPRM expansion and cannot be neglected. When there are more than one optimal polarities for a subfunction, the optimal polarity for the polarity vector with $a_0 = 0$ is chosen.

Our approach to the minimisation of GPMPRM expansion can be viewed as a partition of the polarity coefficient matrix into 2^{n-k} by 2^{n-k} submatrices after the selection of the k mixed polarity variables. Each submatrix is represented by a single decimal number indexed into various lookup tables. By accumulating the weights obtained from the lookup table for every column in a row, a row weight is obtained and compared to the value of a global variable storing the optimal row weight. At the beginning, the optimal row weight is set to the row weight of the first row. As subsequent rows are scanned, the optimal row weight is updated if a smaller row weight is detected. Based on theorems 1 and 2, the algorithm for the fast computation of the minimal GPMPRM expansion is shown in Fig. 1. The principle of operations is illustrated by Example 2. As the example uses the names of the variables in Fig. 1, it will be presented after the pseudocodes have been explained.

In Fig. 1, *table1* and *table2* are lookup tables for different options of minimisation specified by the Boolean variable *optimise_nof_products*. Each product term of a GPMPRM is considered as a concatenation of two products (i.e., the products of the fixed polarity variables and the mixed polarity variables). The presence of a mixed polarity product is indicated by a 1 in the binary k-tuple of *opt_GPMPRM*[j], while its associated fixed polarity variables is indicated by the 1s in the binary $(n - k)$-tuple of j. The polarities of the fixed polarity variables of the final GPMPRM expansion are stored in the bit fields of the variable *optimal_fixed_polarity*, and the polarities of the mixed polarity variables in the products terms of *opt_GPMPRM*[j] are stored in the bit fields of *optimal_mixed_polarity*[j].

```
GPMPRM
{
    first_time = TRUE;
    for (each selection of k variables) {
        for (i = 0 to 2^(n-k)-1) {
            weight = 0;
            for (j = 0 to 2^(n-k)-1) {
                if (optimize_nof_products) {
                    lookup(table1, f_i,j, w_pmin(f_i,j));
                    weight = weight + w_pmin(f_i,j);
                } else {
                    lookup(table2, f_i,j, w_pmin(f_i,j), w_lmin(f_i,j));
                    weight = weight + w_lmin(f_i,j) + ||j|| w_pmin(f_i,j);
                }
            }
            if (first_time) {
                optimal_weight = weight;
                optimal_fixed_polarity = i;
                for (j = 0 to 2^(n-k)-1) {
                    if (optimize_nof_products) lookup(table1, f_i,j, φ_min(f_i,j), A^φmin(f_i,j));
                    else lookup(table2, f_i,j, φ_min(f_i,j), A^φmin(f_i,j));
                    optimal_mixed_polarity[j] = φ_min(f_i,j);
                    optimal_GPMPRM[j] = A^φmin(f_i,j);
                }
                first_time = FALSE;
            }
            if (weight < optimal_weight) {
                optimal_weight = weight;
                optimal_fixed_polarity = r;
                for (j = 0 to 2^(n-k)-1) {
                    if (optimize_nof_products) lookup(table1 f_i,j, φ_min(f_i,j), A^φmin(f_i,j));
                    else lookup(table2,f_i,j, φ_min(f_i,j), A^φmin(f_i,j));
                    optimal_mixed_polarity[j] = φ_min(f_i,j);
                    optimal_GPMPRM[j] = A^φmin(f_i,j);
                }
            }
        }
    }
}
```

Fig. 1 *Generation of minimal GPMPRM expansion*

If we consider only one arbitrary set of k mixed polarity variables, the outer loop of **GPMPRM** can be removed and the resulting GPMPRM expansion is optimum with respect to a given set of mixed polarity variables. Such a constraint is frequently encountered in practice as it may be more cost effective to restrict the privilege of dual polarities to only some specific variables.

Example 2: Consider the five variable Boolean function F from Example 1. By expressing the subfunctions in the decimal number representation as described before, the polarity coefficient matrix of F can be written as:

$$PC(F)=\begin{bmatrix} PC(0) & PC(13) & PC(139) & PC(139)\\ PC(13) & PC(13) & PC(0) & PC(139)\\ PC(139) & PC(134) & PC(139) & PC(139)\\ PC(13) & PC(134) & PC(0) & PC(139)\end{bmatrix}$$

From Table 1, $w_p(0) = 0$, $w_p(13) = 2$, $w_p(139) = 3$ and $w_p(134) = 4$. For $\omega_5\omega_4 = 00$, $w_p(F) = 0 + 2 + 3 + 3 = 8$. For $\omega_5\omega_4 = 01$, $w_p(F) = 2 + 2 + 0 + 3 = 7$. For $\omega_5\omega_4 = 10$, $w_p(F) = 3 + 4 + 3 + 3 = 13$. For $\omega_5\omega_4 = 11$, $w_p(F) = 2 + 4 + 0 + 3 = 9$.

Since $w_{\min}(F) = 7$ is the minimal weight, the fixed polarity literals are chosen to be x_5 and $\bar{x}_4$. From Table 1, *optimal_mixed_polarity*[0] = *optimal_mixed_polarity*[1] = $\phi_{\min}(13) = 6$, *optimal_mixed_polarity*[2] = $\phi_{\min}(0) = 0$

optimal_mixed_polarity[3] = $\phi_{min}(139)$ = 4. Also, $A^6(13)$ = 144 = 10010000_2 or $\bar{x}_3 \oplus \bar{x}_3\bar{x}_2x_1$, $A^0(0) = 0$ and $A^4(139)$ = 88 = 01011000_2 or $x_2\, x_1 \oplus \bar{x}_3 \oplus \bar{x}_3x_2$. The minimal GPMPRM expansion obtained is the same as that given in Example 1.

4 Selection of mixed polarity variables

The algorithm presented in the previous Section assumes that the mixed polarity variables are the k least significant variables. If the mixed polarity variables are not the least significant variables, they can always be reordered such that they become the k least significant variables. However, the ordering of the input variables has an effect on the cost of the final GPMPRM expansion. To extract the best mixed polarity variables such that our minimisation will yield good quality result for a given function, we have investigated several reduction rules used in the minimisation of EXOR expressions. To avoid high computation complexity, simplification rules that cause a temporary expansion in the dimension of the initial representation are avoided.

Of the different reduction rules used for EXOR minimisation [19], exclusion only operates on two EXOR product terms with fixed polarity variables. It is therefore the most suitable candidate for extraction of mixed polarity variables based on an initial FPRM expansion of the function. The exclusion rule with $f = x_i$ has been used in [14] for extracting the single mixed polarity variable for minimisation of GPMPRM expansion with one mixed polarity variable. In what follows, we will show that the exclusion rule can also be a good heuristic for extracting multiple dual polarity variables.

Consider the application of exclusion rule $fg \oplus g = \bar{f}g$ on the following FPRM expansion, where g is a product term that does not contain the variable x_i and x_j.

$$g \oplus gx_i^{\omega_i} \oplus gx_j^{\omega_j} \oplus gx_i^{\omega_i}x_j^{\omega_j} = (g \oplus gx_i^{\omega_i}) \oplus$$
$$(gx_j^{\omega_j} \oplus gx_i^{\omega_i}x_j^{\omega_j}) = gx_i^{\overline{\omega_i}} \oplus gx_i^{\overline{\omega_i}}x_j^{\omega_j} = gx_i^{\overline{\omega_i}}x_j^{\overline{\omega_j}}$$

Substitution of $g \oplus gx_i^{\omega_i} \oplus gx_j^{\omega_j} \oplus gx_i^{\omega_i}x_j^{\omega_j}$ by $gx_i^{\overline{\omega_i}}x_j^{\overline{\omega_j}}$ saves three product terms. In the above example, the exclusion rule has been applied twice to extract the mixed polarity variable x_i in the first step. The same result can also be obtained by applying the exclusion rule twice to extract the mixed polarity variable x_j in the first step. This implies that there are two pairs of FPRM product terms different only in the variable x_i and two pairs of FPRM product terms different only in the variable x_j in the original FPRM expansion. In general, for each of the k variables, if there are 2^{k-1} pairs of product terms of an FPRM expansion different only in that variable, there exist 2^k FPRM product terms that can be reduced to a single mixed polarity term containing these k variables with their polarities all inverted.

The heuristic for the selection of k best dual polarity variables with a trivial modification of the procedure presented in [14] is given as follows:

(i) Form an array G of binary n-tuples, such that the decimal equivalent of any element $j \in G$ when $a_j = 1$ for $j = 0, 1, ..., 2^n - 1$.

(ii) Form an integer array S of dimension n. Initialise all elements s_i to 0 for i = 1, 2, ..., n.

(iii) If, for a pair of numbers $\{a, b\}$ in G, the absolute difference $|a - b| = 2^{i-1}$, increment the element s_i in S by 1. Repeat for all pairs of numbers in G.

(iv) Select k elements with biggest values from S. Their indices are the indices of the k mixed polarity variables.

It should be noted that, due to the difference in minimisation approach, there is no need to sort the elements in G in ascending order of magnitude, as opposed to the procedure given in [14]. As the above extraction algorithm has a computation complexity of $O(|G|^2)$ where $|G|$ is the cardinality of the array G, it is more beneficial to use the optimal polarity FPRM expansion in Step 1.

Example 3: Consider the five-variable Boolean function F in Example 1. The minimal FPRM expansion obtained by the algorithm in [2, 7] is given by: $F = \bar{x}_3 \oplus \bar{x}_3\bar{x}_2x_1 \oplus \bar{x}_4\bar{x}_3 \oplus \bar{x}_4\bar{x}_3\bar{x}_2x_1 \oplus x_5\bar{x}_4x_1 \oplus x_5\bar{x}_4\bar{x}_2x_1 \oplus x_5\bar{x}_4\bar{x}_3\bar{x}_2$. The set of FPRM product terms in decimal number representation is given by {4, 7, 12, 15, 25, 27, 30}. Following the extraction procedure, we have $s_1 = s_3 = s_5 = 0$, $s_2 = 1$ for the pair {25, 27}, and $s_4 = 2$ for the pairs {4, 12} and {7, 15}. Thus, x_4 and x_2 must be selected as the mixed polarity variables. For $k = 3$, if we reorder the variables by interchanging the variables x_4 and x_1 such that x_4 becomes the least significant variable and x_2, the second least significant variable, we have:

$$PC(F) = \begin{bmatrix} PC(10) & PC(2) & PC(1) & PC(68) \\ PC(8) & PC(2) & PC(69) & PC(68) \\ PC(11) & PC(70) & PC(1) & PC(68) \\ PC(77) & PC(70) & PC(69) & PC(68) \end{bmatrix}$$

The minimal GPMPRM expansion generated by Procedure GPMPRM is given by:

$$F = \bar{x}_3x_4 \oplus x_1\bar{x}_3\bar{x}_2x_4 \oplus x_5\bar{x}_3\bar{x}_2\bar{x}_4 \oplus x_5x_1x_2\bar{x}_4$$

With fixed polarity literals x_5 and x_1, a saving of three product terms is achieved as compared with the GPMPRM expansion obtained in Example 2.

If only one mixed polarity variable is allowed as in algorithms from [14, 24], the minimal GPMPRM expansion generated will have five product terms with x_4 selected as the optimal mixed polarity variable. The minimal GPMPRM expansion is given by:

$$F = \bar{x}_3x_4 \oplus x_1\bar{x}_3\bar{x}_2x_4 \oplus x_5\bar{x}_3\bar{x}_2\bar{x}_4 \oplus x_5x_1\bar{x}_4 \oplus x_5x_1\bar{x}_2\bar{x}_4$$

5 Minimisation for multiple output functions

Many ESOP minimisers use a two-phase method when dealing with multiple output functions. Each output of the multiple output functions is first treated as an independent single output function. After applying the minimisation procedure, each output is further minimised according to some predetermined order based on the previously obtained expressions. Such an approach, however, cannot guarantee global minimality. Particularly for the cube-based methods [9, 18, 23], the second phase employs an iterative improvement technique which has both the final result and complexity relying greatly on the ordering of the outputs and the sorting of cubes which has been experimentally demonstrated in [23]. The minimal polarity for one output is not likely to be optimum for the complete system of functions as there may be replication of identical product terms in a number of outputs.

To perform global minimisation for a system of completely specified functions, common terms for each polarity must be sought. Let $F_1, F_2, ..., F_m$ be the outputs of a system of m completely specified n-variable Boolean functions F. Furthermore, let $(f_r)_{i,j}$ denote a subfunction $f_{i,j}$ of F_r where $1 \le r \le m$, $0 \le j \le 2^{n-k} - 1$ and the binary representation of $i = \omega_n\, \omega_{n-1}\, ...\, \omega_{k+1}$ is the polarity number of the fixed polarity variables $x_n, x_{n-1}, ..., x_{k+1}$. The subfunctions

$(f_r)_{i,j}$ for all $r = 1, 2, ..., m$ form a system of k-variable Boolean functions $F_{i,j}$. Under the assumption that the dual polarity variables $x_k, x_{k-1}, ..., x_1$ assume the same optimal polarity $\phi_j = \omega_k\ \omega_{k-1}\ ...\ \omega_1$ for all outputs of the subfunctions $(f_r)_{i,j}$ with the same value of j, theorem 3 provides a multiple output variant of theorems 1 and 2 for determining the weight of the GPMPRM expansion $w(F)$ with product terms shared by more than one output counted only once.

Theorem 3: Let $w(F_{i,j}, \phi_j)$ denote the total number of unique product terms (w_p) or literals (w_l) of the FPRM expansions for a system of k-variable functions, $F_{i,j}$ in polarity ϕ_j. The weight vector $W(F_{i,j}) = [w(F_{i,j}, 0)\ w(F_{i,j}, 1)\ ...\ w(F_{i,j}, 2^k - 1)]^T$ is given by:

$$W(F_{i,j}) = \frac{1}{2^{m-1}}\Big\{\sum W\big((f_g)_{i,j}\big) + \sum W\big((f_g)_{i,j} \oplus (f_h)_{i,j}\big) + \sum W\big((f_g)_{i,j} \oplus (f_h)_{i,j} \oplus (f_e)_{i,j}\big) + \dots + W\big((f_1)_{i,j} \oplus (f_2)_{i,j} \oplus \dots \oplus (f_m)_{i,j}\big)\Big\} \quad (10)$$

where $g, h, e \in \{1, 2, ..., m\}, g \neq h, h \neq e, g \neq e$.

It follows that the weight in terms of the total number of unique products w_p or literals w_l of the GPMPRM expansion of F with polarity number i for the fixed polarity variables is given by:

$$w(F, i) = \sum_{j=0}^{q-1} w(F_{i,j}, \phi_{j\min}) \quad (11)$$

where $w(F_{i,j}, \phi_{j\min}) = \min_{0 \le \phi_j < 2^k}[w(F_{i,j}, \phi_j)]$ and $q = 2^{n-k}$.

Proof: Consider any arbitrary polarity ϕ_j, $0 \le \phi_j \le 2^k - 1$. Let the polarity vectors of the m subfunctions $(f_1)_{i,j}, (f_2)_{i,j}, ..., (f_m)_{i,j}$ be $A_1, A_2, ..., A_m$. Any nonzero GPMPRM coefficient a_t of F that appears in at least one of $A_1, A_2, ..., A_m$ contributes a weight of 1 to $w_p(F_{i,j}, \phi_j)$ or $\|t\|$ to $w_l(F_{i,j}, \phi_j)$ for $j2^k \le t < (j + 1)2^k$, and $0 \le j \le q - 1$. Let p_t be the contribution of the nonzero coefficient a_t to $w_p(F_{i,j}, \phi_j)$ and l_t be the contribution of a_t to $w_l(F_{i,j}, \phi_j)$. $p_t = 0$ iff $a_t = 0$ in all the polarity vectors $A_1, A_2, ..., A_m$ and 1 otherwise. $l_t = 0$ iff $t = 0$ or $a_t = 0$ in all the polarity vectors $A_1, A_2, ..., A_m$ and $\|t\|$ otherwise. Since $\|0\| = 0$, $l_t = \|t\| \times p_t$.

Define $y_{t1}, y_{t2}, ..., y_{tm} \in (0, 1)$ to be the variables representing the logical value of any coefficient a_t in $A_1, A_2, ..., A_m$, respectively. Then $p_t = y_{t1} \vee y_{t2} \vee ... \vee y_{tm}$. It can be shown by induction that, for any Boolean variable x_i, $2^{m-1}(\vee_{i=1}^{m} x_i) = \Sigma_{i=1}^{m} x_i + \Sigma$(all combinations of EXORing of two variables) + Σ(all combinations of EXORing of three variables) + ... + Σ(all combinations of EXORing of $m - 1$ variables) + (EXORing of m variables), where Σ means arithmetic summation. Thus,

$$\begin{aligned} w_p(F_{ij}, \phi_j) &= \sum_{t=0}^{2^k-1} p_t \\ &= \frac{1}{2^{m-1}}\Big\{\sum_{t=0}^{2^k-1}\sum_{g=1}^{m} y_{tg} + \sum_{t=0}^{2^k-1}\sum (y_{tg} \oplus y_{th}) + \sum_{t=0}^{2^k-1}\sum (y_{tg} \oplus y_{th} \oplus y_{te}) + \dots + \sum_{t=0}^{2^k-1} y_{t1} \oplus y_{t2} \oplus \dots \oplus y_{tm}\Big\} \\ &= \frac{1}{2^{m-1}}\Big\{\sum_{g=1}^{m}\sum_{t=0}^{2^k-1} y_{tg} + \sum\sum_{t=0}^{2^k-1} (y_{tg} \oplus y_{th}) + \sum\sum_{t=0}^{2^k-1} (y_{tg} \oplus y_{th} \oplus y_{te}) + \dots + \sum_{t=0}^{2^k-1} (y_{t1} \oplus y_{t2} \oplus \dots \oplus y_{tm})\Big\} \\ &= \frac{1}{2^{m-1}}\Big\{\sum_{g=1}^{m} w_p(f_g) + \sum w_p(f_g \oplus f_h) + \sum w_p(f_g \oplus f_h \oplus f_e) + \dots + w_p(f_1 \oplus f_2 \oplus \dots \oplus f_m)\Big\} \end{aligned}$$

Similarly,

$$\begin{aligned} w_l(F^\alpha, \phi_j) &= \sum_{t=0}^{2^k-1} \|t\| \times p_t \\ &= \frac{1}{2^{m-1}}\Big\{\sum_{g=1}^{m} w_l(f_g) + \sum w_l(f_g \oplus f_h) + \sum w_l(f_g \oplus f_h \oplus f_e) + \dots + w_l(f_1 \oplus f_2 \oplus \dots \oplus f_m)\Big\} \end{aligned}$$

Hence,

$$W(F_{ij}) = \frac{1}{2^{m-1}}\Big\{\sum_{g=1}^{m} W\big((f_g)_{ij}\big) + \sum W\big((f_g)_{i,j} \oplus (f_h)_{i,j}\big) + \dots + W\big((f_1)_{i,j} \oplus (f_2)_{i,j} \oplus \dots \oplus (f_m)_{i,j}\big)\Big\}$$

where $W(F_{i,j}) = [w(F_{i,j}, 0)\ w(F_{i,j}, 1)\ ...\ w(F_{i,j}, 2^k - 1)]^T$.

For each value of j, there is a polarity number $\phi_{j\min}$ for which the weight $w_l(F_{i,j}, \phi_j)$ is the minimum among the elements in $W(F_{i,j})$. From theorems 1 and 2, we have $w(F, i) = \Sigma_{j=0}^{q-1} w(F_{i,j}, \phi_{j\min})$.

It should be noted that the weight of the GPMPRM expansion for any fixed polarity number i obtained by theorem 3 is not the absolute minimum as we have assumed that all output functions have identical mixed polarity literals for the same product. It is possible to achieve a lower weight in term of the total number of unique products or literals by applying theorem 3 to all ${}^mC_r$ systems of r ($r = 1, 2, ..., m$) k-variable functions selected from $F_{i,j}$ for every value of j. The system of functions that has the minimum weight among the ${}^mC_r$ systems will have the same $\phi_{j\min}$ for their mixed polarity literals. The process repeats with these functions being removed from $F_{i,j}$ until $\phi_{j\min}$ for all functions have been found.

To apply theorem 3 to obtain global minimisation of multiple output functions, the lookup tables used in the algorithm GPMPRM have to be extended. Three lookup tables NP, NL and RM are indexed by the decimal number representation of a k-variable subfunction f. Each record of

NP and NL contains a weight vector $W_p(f)$ and $W_l(f)$, respectively, and each record of RM consists of a polarity coefficient matrix of f, $PC(f)$. To conserve memory space, $PC(f)$ is stored as a $2^k \times 1$ column vector with each element being a decimal number representation of the polarity vector A^{ϕ} where $\phi = 0, 1, ..., 2^k - 1$. Since $\overline{A^{\phi}} = 1 \oplus A^{\phi}$, only the records of half of the total number of k-variable Boolean functions are required for the lookup table RM

Corollary 1: It follows from theorem 3 that, for a system of m n-variable completely specified Boolean functions F, the weights in terms of the total number of products, $w_p(F, i)$ and the total number of literals, $w_l(F, i)$ of the GPMPRM expansion in an arbitrary polarity number i for the fixed polarity variables are given by:

$$w_p(F,i) = \frac{1}{2^{m-1}}\left\{\sum_{j=0}^{q-1}\left[\min_{0\le\phi_j<2^k}\left(\sum_{s=1}^{2^m-1} NP\left[\underset{s_r=1}{\oplus}(f_r)_{i,j}\right]\right)\right]\right\} \quad (12)$$

$$w_l(F,i) = \frac{1}{2^{m-1}}\left\{\sum_{j=0}^{q-1}\left[\min_{0\le\phi_j<2^k}\left(\sum_{s=1}^{2^m-1} NL\left[\underset{s_r=1}{\oplus}(f_r)_{i,j}\right] + \|j\|\times\sum_{s=1}^{2^m-1} NP\left[\underset{s_r=1}{\oplus}(f_r)_{i,j}\right]\right)\right]\right\} \quad (13)$$

where s_r is the rth ($r = 1, 2, ..., m$) bit in binary m-tuple of the integer s. $(f_r)_{i,j}$ is the subfunction $f_{i,j}$ of the rth output, F_r. $NP[\cdot]$ and $NL[\cdot]$ are weight vectors from the lookup tables NP and NL, respectively. The function $\min_{0\le\phi_j<2^k}(\cdot)$ gives the minimum value among elements of the $2^k \times 1$ column vector and the row number ϕ_j for which this minimum value $\phi_{j\min}$ occurs.

Corollary 2: The coefficient of the GPMPRM expansion for each output r is given by the binary 2^n-tuple A_r as follows:

$$A_r = \sum_{j=0}^{q-1} 2^j\left\{c_j \oplus RM\left[\min\left((f_r)_{i,j}, \overline{(f_r)_{i,j}}\right)\right][\phi_{j\min}]\right\} \quad (14)$$

where $c_j = 1$ if $(f_r)_{i,j} \ge 2^{2^k-1}$ and 0 otherwise. The binary equivalent of A_r is equal to $a_{2^n-1} \ldots a_1\, a_0$ and can be formed as a concatenation of the binary 2^k-tuples of $c_j \oplus RM[\min(f_r)_{i,j}, \overline{(f_r)_{i,j}}][\phi_{j\min}]$ for $j = 0, 1, ..., 2^{n-k} - 1$. The polarity number for the mixed polarity variables in the jth tuple is given by $\phi_{j\min}$.

Based on corollaries 1 and 2, the algorithm GPMPRM in Fig. 1 can be easily modified to achieve a global minimisation for multiple output functions. The resulting GPMPRM expansions are either optimal or quasioptimal due to the simplification explained earlier.

Example 4: Consider the minterm lists of a 5-variable 3-output function F taken from [14]:

$$f_1(x_5, x_4, x_3, x_2, x_1) = \Sigma m(6, 7, 11, 14, 16, 18, 20, 23, 30, 31)$$

$$f_2(x_5, x_4, x_3, x_2, x_1) = \Sigma m(2, 10, 11, 17, 19, 21, 22, 26)$$

$$f_3(x_5, x_4, x_3, x_2, x_1) = \Sigma m(1, 5, 7, 10, 14, 18, 22, 25, 29, 31)$$

Consider the case of x_3, x_2 and x_1 as the mixed polarity variables and the fixed polarity number $i = 1$ (i.e. the fixed polarity literals are x_5 and $\bar{x}_4$). The calculation of the GPMPRM expansion by corollaries 1 and 2 is shown as follows:

For $j = 0$, $(f_1)_{1,0} = (f_1)_{\bar{x}_5x_4} = 72$, $(f_2)_{1,0} = (f_2)_{\bar{x}_5x_4} = 12$ and $(f_3)_{1,0} = (f_3)_{\bar{x}_5x_4} = 68$. $(f_1)_{1,0} \oplus (f_2)_{1,0} = 68$, $(f_1)_{1,0} \oplus (f_3)_{1,0} = 12$, $(f_2)_{1,0} \oplus (f_3)_{1,0} = 72$, and $(f_1)_{1,0} \oplus (f_2)_{1,0} \oplus (f_3)_{1,0} = 0$. From the lookup table NP, $NP[72] = [2\ 3\ 4\ 6\ 3\ 2\ 6\ 4]^T$, $NP[12] = [2\ 2\ 4\ 4\ 1\ 1\ 2\ 2]^T$, $NP[68] = [2\ 1\ 4\ 2\ 2\ 1\ 4\ 2]^T$ and $NP[0] = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$. From theorem 3, $W(F_{1,0}) = \frac{1}{4}\{NP[72] + NP[12] + NP[68] + NP[68] + NP[12] + NP[72] + NP[0]\} = [3\ 3\ 6\ 6\ 3\ 2\ 6\ 4]^T$. Therefore, $\phi_{0\min} = 5$ and $w_p(F_{1,0}, \phi_{0\min}) = 2$. From the lookup table RM; $RM[(f_1)_{1,0}][\phi_{0\min}] = RM[72][5] = 72 = 01001000_2 = x_2\bar{x}_1 \oplus \bar{x}_3x_2$. $RM[(f_2)_{1,0}][\phi_{0\min}] = RM[12][5] = 64 = 01000000_2 = \bar{x}_3x_2$. $RM[(f_3)_{1,0}][\phi_{0\min}] = RM[68][5] = 8 = 00001000_2 = x_2\bar{x}_1$.

For $j = 1$, $(f_1)_{1,1} = \partial(f_1)_{\bar{x}_5}/\partial x_4 = 136$, $(f_2)_{1,1} = \partial(f_2)_{\bar{x}_5}/\partial x_4 = 8$, $(f_3)_{1,1} = \partial(f_3)_{\bar{x}_5}/\partial x_4 = 230$. $(f_1)_{1,1} \oplus (f_2)_{1,1} = 128$, $(f_1)_{1,1} \oplus (f_3)_{1,1} = 110$, $(f_2)_{1,1} \oplus (f_3)_{1,1} = 238$. $(f_1)_{1,1} \oplus (f_2)_{1,1} \oplus (f_3)_{1,1} = 102$. From the lookup table NP, $NP[136] = [1\ 2\ 2\ 4\ 1\ 2\ 2\ 4]^T$, $NP[8] = [2\ 4\ 4\ 8\ 1\ 2\ 2\ 4]^T$, $NP[230] = [3\ 5\ 5\ 6\ 4\ 5\ 5\ 6]^T$, $NP[128] = [1\ 2\ 2\ 4\ 2\ 4\ 4\ 8]^T$, $NP[110] = [4\ 5\ 5\ 6\ 3\ 5\ 5\ 6]^T$, $NP[238] = [3\ 3\ 3\ 2\ 3\ 3\ 3\ 2]^T$ and $NP[102] = [2\ 3\ 3\ 2\ 2\ 3\ 3\ 2]^T$. From theorem 3, $W(F_{1,1}) = [4\ 6\ 6\ 8\ 4\ 6\ 6\ 8]^T$. Therefore, $\phi_{1\min} = 0$ or 4, and $w_p(F_{1,1}, \phi_{1\min}) = 4$. Select $\phi_{1\min} = 0$, from the lookup table RM, $RM[(f_1)_{1,1}][\phi_{1\min}] = RM[136][0] = 1 \oplus_d RM[119][0] = 8 = 00001000_2 = x_2x_1$. $RM[(f_2)_{1,1}][\phi_{1\min}] = RM[8][0] = 136 = 10001000_2 = x_2x_1 \oplus x_3x_2x_1$. $RM[(f_3)_{1,1}][\phi_{1\min}] = RM[230][0] = 1 \oplus_d RM[25][0] = 134 = 10000110_2 = x_1 \oplus x_2 \oplus x_3x_2x_1$.

For $j = 2$, $(f_1)_{1,2} = \partial(f_1)_{x_4}/\partial x_5 = 136$, $(f_2)_{1,2} = \partial(f_2)_{x_4}/\partial x_5 = 8$, $(f_3)_{1,2} = \partial(f_3)_{x_4}/\partial x_5 = 230$. $(f_1)_{1,2} \oplus (f_2)_{1,2} = 128$, $(f_1)_{1,2} \oplus (f_3)_{1,2} = 110$, $(f_2)_{1,2} \oplus (f_3)_{1,2} = 238$. $(f_1)_{1,2} \oplus (f_2)_{1,2} \oplus (f_3)_{1,2} = 102$. Since $W(F_{1,2}) = W(F_{1,1})$, $\phi_{2\min} = 0$ or 4, and $w_p(F_{1,2}, \phi_{2\min}) = 4$. Select $\phi_{2\min} = 0$, $RM[(f_1)_{1,2}][\phi_{2\min}] = x_2x_1$. $RM[(f_2)_{1,2}][\phi_{2\min}] = x_2x_1 \oplus x_3x_2x_1$. $RM[(f_3)_{1,2}][\phi_{2\min}] = x_1 \oplus x_2 \oplus x_3x_2x_1$.

For $j = 3$, $(f_1)_{1,3} = \partial^2 f_1/\partial x_5\partial x_4 = 221$, $(f_2)_{1,3} = \partial^2 f_2/\partial x_5\partial x_4 = 102$, $(f_3)_{1,3} = \partial^2 f_3/\partial x_5\partial x_4 = 0$. $(f_1)_{1,3} \oplus (f_2)_{1,3} = 187$, $(f_1)_{1,3} \oplus (f_3)_{1,3} = 221$, $(f_2)_{1,3} \oplus (f_3)_{1,3} = 102$. $(f_1)_{1,3} \oplus (f_2)_{1,3} \oplus (f_3)_{1,3} = 187$. From the lookup table NP, $NP[221] = [3\ 3\ 2\ 3\ 3\ 3\ 2\ 3]^T$, $NP[102] = [2\ 3\ 3\ 2\ 2\ 3\ 3\ 2]^T$, and $NP[187] = [3\ 2\ 3\ 3\ 3\ 2\ 3\ 3]^T$. From theorem 3, $W(F_{1,3}) = [4\ 4\ 4\ 4\ 4\ 4\ 4\ 4]^T$. Hence, $\phi_{3\min} = 0, 1, ..., 7$ and $w_p(F_{1,3}, \phi_{3\min}) = 4$. Select $\phi_{3\min} = 0$, from the lookup table RM, $RM[(f_1)_{1,3}][\phi_{3\min}] = RM[221][0] = 1 \oplus_d RM[34][0] = 11 = 00001011_2 = 1 \oplus x_1 \oplus x_2x_1$. $RM[(f_2)_{1,3}][\phi_{3\min}] = RM[102][0] = 6 = 00000110_2 = x_1 \oplus x_2$. $RM[(f_3)_{1,3}][\phi_{3\min}] = RM[0][0] = 0 = 00000000_2 = 0$.

By eqn. 11, $w_p(F, 1) = w_p(F_{1,0}, \phi_{0\min}) + w_p(F_{1,1}, \phi_{1\min}) + w_p(F_{1,2}, \phi_{2\min}) + w_p(F_{1,3}, \phi_{3\min}) = 14$. By corollary 2, the GPMPRM expansion for each output is given as follows:

$$A_1 = (x_2\bar{x}_1 \oplus \bar{x}_3x_2) \oplus \bar{x}_4(x_2x_1) \oplus x_5(x_2x_1) \oplus x_5\bar{x}_4(1 \oplus x_1 \oplus x_2x_1).$$

$$A_2 = \bar{x}_3x_2 \oplus \bar{x}_4(x_2x_1 \oplus x_3x_2x_1) \oplus x_5(x_2x_1 \oplus x_3x_2x_1) \oplus x_5\bar{x}_4(x_1 \oplus x_2).$$

$$A_3 = x_2\bar{x}_1 \oplus \bar{x}_4(x_1 \oplus x_2 \oplus x_3x_2x_1) \oplus x_5(x_1 \oplus x_2 \oplus x_3x_2x_1)$$

6 Experimental results

The complexity of the lookup table based method GPMPRM depends on the total number of unique subfunctions. In the worst case, there are 3^{n-k} subfunctions, and the computation complexity is of the order $O({}^nC_k\, 3^{n-k})$ since there are ${}^nC_k$ selections of k mixed polarity variables. With the heuristic algorithm for the selection of k mixed polarity variables, the complexity is reduced to $O(p^2 3^{n-k})$ where p is the number of FPRM terms. If the computation complexity is evaluated with the ratio of the number of alternative GPMPRM expansions to the order of computation complexity, the ratio is greater than 1 and grows rapidly as n and k increase. In contrast, the ratio tends to 1 as n becomes larger for the fast algorithm in [24] where the GPMPRM expansion is defined for only one mixed polarity variable. For multiple output function, under the assumption that each output has the same mixed polarity literals for the same GPMPRM product, the computation complexity is $O(p^2\, 2^m\, 3^{n-k})$ where m is the number of outputs.

The presented algorithm is implemented on the HP Apollo Series 715 workstation with $k = 3$ mixed polarity variables that are selected in an optimal way. The current implementation can calculate the minimal GPMPRM expansions for multiple output Boolean functions based on either the minimal number of products w_p or literals w_l, although theoretically any cost function of the form $aw_l + bw_p$ is possible where a and b are integer constants. The quality of the results for a minimal GPMPRM expansion with three mixed polarity variables always outperforms the procedure with only one mixed polarity variable. A range of benchmark examples have been tested. Some multiple output two level examples in pla format from MCNC benchmarks minimised with w_p and w_l as cost functions are compared in Table 2 with the results obtained from the exact FPRM minimisers [2, 4]. There are many exact minimisers for FPRM expansions developed by different authors, but to compare the execution time for FPRM and GPMPRM on the same machine we used exact FPRM minimisers developed by us [2, 7]. The columns labelled time (GPMPRM) and time (FPRM) are the user (usr) and

Table 2: Benchmark results for GPMPRM, FPRM and ESOP

Function	n	m	GPMPRM		FPRM		time GPMPRM (s)		time FPRM (s)		EXMIN-2			EXORCISM-MV-2		
			w_p	w_l	w_p	w_l	usr	sys	usr	sys	C_T	C_L	time[a]	C_T	C_L	time[b]
5xp1	7	10	59	198	61	224	5.15	0.04	4.41	0.04	34	186	13	33	178	13.6
9sym	9	1	136	504	173	636	46.87	0.01	47.32	0.01	53	433	25	51	425	39.0
con1	7	2	13	37	17	48	0.24	0.02	0.23	0.02	–	–	–	9	37	0.4
misex1	8	7	16	51	20	68	7.81	0.01	7.62	0.03	–	–	–	12	89	1.4
rd53	5	3	20	45	20	45	0.01	0.04	0.02	0.04	15	60	2	14	57	1.3
rd73	7	3	63	189	63	189	2.05	0.01	2.05	0.02	42	221	20	38	191	24.6
rd84	8	4	107	352	107	352	30.97	0.02	30.88	0.04	59	330	45	57	317	168.2
sao2	10	4	70	365	100	707	387.14	0.02	386.30	0.04	29	310	8	28	286	10.9
squar5	5	8	23	56	23	56	0.08	0.01	0.06	0.06	–	–	–	19	82	2.9
xor5	5	1	5	5	5	5	0.02	0.01	0.00	0.01	–	–	–	5	10	0.2
Z9sym	9	1	136	504	173	636	60.48	0.02	61.00	0.04	–	–	–	–	–	–
clip	9	5	181	825	206	995	136.14	0.05	134.14	0.04	68	517	55	65	490	66.9

[a] cpu seconds on SPARC Station 1+. [b] cpu seconds on SPARC Station 1.

Table 3: Benchmark results of single output for GPMPRM, GPMPRM1, GPMPRM2 and CGRMIN

Function	n	GPMPRM			GPMPRM1		GPMPRM2	CGRMIN	
		w_p	w_l	time	w_p	time	w_p	w_p	time
5xp11	7	8	33	0.05	9	0.22	9	12	–
5xp13	7	16	47	0.04	14	0.22	–	19	–
5xp15	7	7	13	0.03	6	0.20	–	7	–
9sym	9	136	504	0.02	139	3.9	139	173	1851.3
con12	7	5	12	0.04	9	0.22	6	12	1.2
rd532	5	5	5	0.05	5	0.03	5	5	1.1
rd732	7	7	7	0.06	7	0.22	7	7	49.6
rd842	8	8	8	0.06	8	0.83	8	8	364.2
xor5	5	5	5	0.03	5	0.00	–	5	–
sao22	10	34	174	0.05	37	16.58	37	58	642.1
sao23	10	28	156	0.05	35	16.52	34	47	905.9
z4m11	7	15	56	0.08	22	0.2	15	25	–
z4m12	7	9	22	0.04	–	–	9	–	–
z4m13	5	5	8	0.05	–	–	5	–	–
z4m14	3	3	3	0.05	–	–	3	–	–
majority	5	6	17	0.06	–	–	6	–	–

'–' indicates data not available

system (sys) execution time in seconds of our current implementation of GPMPRM and the exact FPRM minimiser [2, 7], respectively. For most functions, our results for GPMPRM expansions are significantly better than the exact FPRM expansions. The savings in the total number of literals for GPMPRM expansion are more prominent than the savings in the number of product terms. For completeness with recent results on exclusive or sum-of-products (ESOPs), we also included the results for heuristic minimisation of ESOP expression for multiple-valued functions obtained by the EXMIN-2 and EXORCISM-MV-2 minimiser [20]. It should be noticed that each variable in ESOP can have an arbitrary polarity in different terms, and that the same sets of literals can be repeated in different terms. In Table 2, the symbols C_T and C_L have the following meaning [20]: C_T is the total number of multiple-valued terms in the solution, and C_L is the total number of input multiple-valued wires to the AND and EXOR gates in the solution. As expected, the most general ESOP forms for multiple-valued functions are more compact in number of terms and connections when compared to the binary case. It is, however, obtained at the expense of the final circuit realisation as a multiple-valued circuit require more silicon area and larger number of transistors. Table 3 summarises the comparison between the quality and system execution time of our algorithm and those of the exact minimisers with one mixed polarity variable, GPMPRM1 [24] and GPMPRM2 [14], and an exact FPRM minimiser, CGRMIN [18] for single output functions. Our results for the majority of the functions are either the same or better than that for GPMPRM1 and GPMPRM2, and outperform CGRMIN. Moreover, the processing time of our algorithm is remarkably lower than that for GPMPRM1 and CGRMIN. It should, however, be noticed that Table 3 uses time taken from [18, 24] directly, so the time is also influenced by the different workstations used in each of the experiments.

7 Conclusion

This paper solves the open problem stated in [25] on how to minimise GPMPRM expansions with k mixed polarity variables. For such a case, this expansion has ${}^nC_k 2^{n-k} 2^{k2^{n-1}} - ({}^nC_k - 1)2^n$ alternative forms which is closer to the ESOP than the original definition. An efficient lookup table based method is presented in this paper for the heuristic minimisation of GPMPRM expansions with $k < 6$ mixed polarity variables for multiple output functions. The complexity of the minimisation problem varies with the value of k, with $k = 0$ and n represent the extreme cases of the FPRM and GRM expansions, respectively. In general, the size of GPMPRM is smaller than the size of FPRM but the size of GPMPRM is probably much greater than the size of GRM or pseudo Kronecker expansions [5]. A comparison of these expansions is the topic of our current research study. As the table lookup operation involves constant time computational complexity, increasing the value of k speeds up processing for a more complex minimisation problem at the expense of higher storage requirements. The value of k is limited by the exponential complexity of the lookup table. Due to the inherent nature of the NP problem, the presented algorithm is highly efficient for up to ten input variables with $k = 3$. It is also adaptable to various cost functions which is what is lacking in the existing minimiser [24, 25].

8 References

1 AKERS, S.: 'On a theory of Boolean functions', *SIAM J.*, 1959, **7**, pp. 487–498
2 CHANG, C.H., and FALKOWSKI, B.J.: 'Flexible optimization of fixed polarity Reed-Muller expansions for multiple output completely and incompletely specified Boolean functions'. Proceedings of the IEEE/IEICE Asia and South Pacific design automation conference (ASP-DAC '95), Makuhari, Chiba, Japan, 1995, pp. 335–340
3 COHN, M.: 'Inconsistent canonical forms of switching functions', *IRE Trans. Electron. Comput.*, 1962, **11**, pp. 284–285
4 CSANKY, L., PERKOWSKI, M.A., and SCHAEFER, I.: 'Canonical restricted mixed-polarity exclusive-OR sums of products and the efficient algorithm for their minimization', *IEE Proc., E*, 1993, **140**, (1), pp. 69–77
5 DAVIO, P., DESCHAMPS, J.P., and THAYSE, A.: 'Discrete and switching functions' (McGraw-Hill, New York, 1978)
6 FALKOWSKI, B.J., and CHANG, C.H.: 'Generation of fixed polarity Reed-Muller expansions from subset of Walsh spectral coefficients for completely specified Boolean functions'. Proceedings of the 5th workshop on *Spectral techniques*, Beijing, China, 1994, pp. 214–219
7 FALKOWSKI, B.J., and CHANG, C.H.: 'An exact minimizer of fixed polarity Reed–Muller expansions', *Int. J. Electron.*, 1995, **79**, (4), pp. 389–409
8 FALKOWSKI, B.J., and CHANG, C.H.: 'Minimization of partially-mixed-polarity Reed–Muller expansions for multiple output incompletely specified Boolean functions'. Proceedings of the 6th workshop on *Post binary ultra-large scale integration*, Antigonish, Nova Scotia, Canada, 1997, pp. 45–46
9 FISHER, L.T.: 'Unateness properties of AND-exclusive-OR logic circuits', *IEEE Trans. Comput.*, 1974, **23**, (2), pp. 166–172
10 GREEN, D.H.: 'Families of Reed–Muller canonical forms', *Int. J. Electron.*, 1991, **70**, pp. 259–280
11 GREEN, D.H.: 'Reed–Muller canonical forms with mixed polarity and their manipulations', *IEE Proc., E*, 1990, **137**, (1), pp. 103–113
12 LIN, T.L., and TRAN, A.: 'Minimization of multiple-output exclusive-OR switching functions', *Int. J. Electron.*, 1993, **75**, (4), pp. 665–674
13 LUI, P.K., and MUZIO, J.C.: 'Boolean matrix transforms for the minimization of modulo-2 canonical expansions', *IEEE Trans. Comput.*, 1992, **41**, (3), pp. 342–347
14 MAREK-SADOWSKA, M., GUNER, G., KRISHNAMURTHY, T., GARGESHWARI, S., and SHARMA, CH.: 'Minimized generalized partially mixed polarity Reed–Muller expansion'. Proceedings of the 3rd international workshop on *Applications of the Reed–Muller expansion in circuit design* (Reed-Muller '97), Oxford, UK, 1997, pp. 35–44
15 PERKOWSKI, M.A., and CHRZANOWSKA-JESKE, M.: 'An exact algorithm to minimize mixed-radix exclusive sums of products for incompletely specified Boolean functions'. Proceedings of the IEEE international symposium on *Circuits and systems* (ISCAS '90), New Orleans, Louisiana, USA, 1990, pp. 1652–1655
16 REDDY, S.M.: 'Easily testable realizations for logic functions', *IEEE Trans.*, 1972, **C-21**, (11), pp. 1183–1188
17 SALUJA, K.K., and ONG, E.H.: 'Minimization of Reed–Muller canonic expansion', *IEEE Trans. Comput.*, 1979, **28**, (2), pp. 535–537
18 SARABI, A., and PERKOWSKI, M.A.: 'Fast exact and quasi-minimal minimization of highly testable fixed-polarity AND/XOR canonical networks'. Proceedings of the 29th ACM/IEEE design automation conference, 1992, pp. 30–35
19 SAUL, J., ESCHERMANN, B., and FROESSL, J.: 'Two-level logic circuits using EXOR sums of products', *IEE Proc., E*, 1993, **40**, (6), pp. 348–356
20 SONG, N., and PERKOWSKI, M.A.: 'Minimization of exclusive sum-of-products expressions for multiple-valued input, incompletely specified functions', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 1996, **15**, (4), pp. 385–395
21 TRAN, A.: 'Graphical method for the conversion of minterms to Reed–Muller coefficients and the minimisation of exclusive-OR switching functions', *IEE Proc., E*, 1987, **134**, (2), pp. 93-99
22 TSAI, C.C., and MAREK-SADOWSKA, M.: 'Generalized Reed–Muller forms as a tool to detect symmetries', *IEEE Trans. Comput.*, 1996, **45**, (1), pp. 33–40
23 TSAI, C.C., and MAREK-SADOWSKA, M.: 'Minimization of fixed polarity AND/XOR canonical network', *IEE Proc., E*, 1994, **141**, (6), pp. 369–374
24 WU, H., PERKOWSKI, M.A., ZENG, X., and ZHUANG, N.: 'Generalized partially-mixed-polarity Reed–Muller expansion and its fast computation', *IEEE Trans. Comput.*, 1996, **45**, (9), pp. 1084–1088
25 ZENG, X., PERKOWSKI, M.A., WU, H., and SARABI, A.: 'A new exact algorithm for highly testable generalized partially-mixed-polarity Reed–Muller forms'. Proceedings of the IFIP WG10.5 workshop on *Applications of the Reed–Muller expansion in circuit design*, Makuhari, Chiba, Japan, 1995, pp. 231–239