Efficient Calculation of Fixed-Polarity Polynomial Expressions for Multiple-Valued Logic Functions*

Dragan Janković Radomir S. Stanković Rolf Drechsler

Faculty of Electronics
University of Niš
18 000 Niš, Yugoslavia

Institute of Computer Science
University of Bremen
28359 Bremen, Germany

Abstract

This paper presents a tabular technique for calculation of fixed-polarity polynomial expressions for MV functions. The technique is derived from a generalization of the corresponding methods for Fixed-Polarity Reed-Muller (FPRM) expressions for switching functions. All useful features of tabular techniques for FPRMs, as for example, simplicity of involved operations and high possibilities for parallelization of the calculation procedure, are preserved. The method can be extended to the calculation of coefficients in Kronecker expressions for MV functions.

1 Introduction

Fixed-Polarity Reed-Muller (FPRM) expressions are a way for optimization of *Positive-polarity Reed-Muller* (PPRM) expressions [16]. FPRMs are derived by allowing to freely choose the negative $\overline{x}_i$ or the positive x_i literal, but not both, for each variable in a given switching function $f(x_1, \ldots, x_n)$. The assignment of literals to variables is uniquely specified by the polarity $p = (p_1, \ldots, p_n)$, $p_i \in \{0, 1\}$, where $p_i = 1$ and $p_i = 0$ determine the assignment of negative and positive literals to the i-th variable, respectively. The complexity of FPRMs is usually estimated through the number of non-zero coefficients. For a given function f, the FPRM with the minimum number of coefficients is considered as the optimum FPRM for f.

There are several methods for the generation of FPRMs for a given function f and the required polarity p using different data structures to represent f [8, 20, 12, 14, 3, 15]. The term *Tabular Techniques* (TTs) usually refers to methods derived for minterm representations [1, 2, 19].

TTs exploit linearity of Reed-Muller expressions, which permits to determine the value of a coefficient c_i in FPRM expressions for f as the EXOR sum of the contributions of each true minterm in f to c_i [19]. Main features of TT methods, that can be seen as advantages over other methods, can be summarized as follows:

1. In TTs, FPRM coefficients are calculated through 1-minterms. Contribution of each minterm to the FPRM spectrum for f can be determined separately and independently of the contribution of other minterms. Thus, TTs possess an inherent possibility for efficient parallelization of the calculation procedure [19].

2. Processing of each minterm is simple, since there are no arithmetic operations. Instead, processing of a minterm reduces to convertion of a minterm into another minterms by using some relatively simple processing rules.

The concept of Reed-Muller expressions can be extended to MV functions in several ways, depending on generalization of AND and XOR operations. Fixed polarity Reed-Muller expressions have been extended to the ternary field $GF(3)$ in [10] and in general to a prime field $GF(q)$ in [9, 7]. In [11, 7] different methods for calculation of FPRM expressions over $GF(4)$ have been proposed while the method for computation of polarity matrices for Reed-Muller expressions based on prime finite fields has been introduced in [13].

In this paper, the TT method for calculation of FPRMs for switching functions proposed in [19] is generalized to MV functions. However, instead of minterms, the proposed method uses MV disjoint cubes. A similar approach has been considered in [6] for the binary case. Furthermore, we propose some improvements to the overall algorithms resulting in a reduced computational complexity.

Then, we briefly discuss extensions of the proposed method to determination of Kronecker expressions for MV functions [17]. We study further generalizations to the expressions permitting the use of an extended set of complements for MV variables, as for example *Reed-Muller-Fourier* (RMF) expressions [18]. We also provide a modification of the proposed method enabling to determine for a given function f a fixed polarity polynomial expression for a polarity p_i directly from the corresponding expression

*This work was supported in part by DFG grant Dr 287/9-1.

Table 1. Additions and multiplication in $GF(4)$

$+$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

for f for an arbitrary polarity p_j, without returning necessarily to the positive-polarity expression or to some other representation of f. This modification applies for any of the Kronecker polynomial expressions.

For simplicity of notations, presentation in this paper is given by the example of polynomial representations for MV functions defined over Galois field $GF(4)$. Extension to arbitrary Kronecker product based polynomial expressions on arbitrary $GF(p)$, and other related polynomial expressions are straightforward.

2 Reed-Muller Expressions over $GF(4)$

In this section, we introduce some notations and definitions of *Galois Field* (GF) expressions.

Denote by $E(4)$ the set of four elements. For convenience, we will identify the elements of $E(4)$ with the non-negative integers 0, 1, 2, 3.

Definition 1 *$E(4)$ expresses the structure of the Galois field modulo 4, $GF(4)$, under the addition and the multiplication defined as in Table 1.*

The set of elementary functions 1, x, x^2, x^3 is a basis in the space of one-variable functions over $GF(4)$. Therefore, each function f given by the truth-vector

$$\mathbf{F} = [f(0), \ldots, f(3)]^T$$

can be represented by the Reed-Muller expressions for functions over $GF(4)$ (RMGFE) given in the matrix form by

$$f(x) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \cdot (\mathbf{G}_4(1) \cdot \mathbf{F}),$$

where

$$\mathbf{G}_4(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

and all calculations are carried out in $GF(4)$.

In this notation, the basic functions 1, x, x^2, x^3 can be considered as columns of the matrix $\mathbf{G}_4^{-1}(1)$, inverse to $\mathbf{G}_4(1)$.

Extension of this expression to n-variable functions defined over $GF(4)$ is straightforward through the Kronecker product.

Definition 2 *The RMGFE for an n-variable four-valued function f given by its truth-vector*

$$\mathbf{F} = [f(0), \ldots, f(4^n - 1)]^T$$

is

$$f = \left(\bigotimes_{i=1}^{n} \begin{bmatrix} 1 & x_i & x_i^2 & x_i^3 \end{bmatrix} \right) \cdot \left(\left(\bigotimes_{i=1}^{n} \mathbf{G}_4(1) \right) \cdot \mathbf{F} \right),$$

where $\otimes$ denotes the Kronecker product and all calculations are carried out in $GF(4)$.

Optimization of polynomial representations for MV functions is possible by using different complements for variables [13]. For each variable x_i in functions defined over $GF(4)$ there are three complements denoted as ${}^i\bar{x}$, $i = 1, 2, 3$ and defined as ${}^i\bar{x} = x \oplus i$, $i = 1, 2, 3$. The use of complements for a variable requires permutation of columns in the corresponding basic GF matrix. Table 2 shows complements for variables over $GF(4)$ and the corresponding basic transform matrices.

Definition 3 *Each n-variable function f defined over $GF(4)$ can be represented by the Fixed Polarity RMGFE (FPRMGFE) for the polarity $p = (p_1, \ldots, p_n)$*

$$f = \left(\bigotimes_{i=1}^{n} [1, {}^{p_i}\bar{x}_i, ({}^{p_i}\bar{x}_i)^2, ({}^{p_i}\bar{x}_i)^3] \right) \cdot (\bigotimes_{i=1}^{n} \mathbf{G}_4^{<p_i>}) \cdot \mathbf{F},$$

where $\mathbf{G}_4^{<p_i>}$ is the GF transform matrix for the polarity p_i.

3 Tabular Technique for FPRMGFE

The tabular technique for calculation of FPRM expressions proposed in [19] is an improvement of approaches presented in [2, 1]. This TT starts from a table of minterms for a given switching function f. The terms in FPRM expressions for f are generated by performing a set of simple rules over each minterm. Equal product terms that have been already generated from previously processed minterms are deleted by a cancelation procedure. This method processes each minterm separately.

The method consists of three important steps

1. Generation of new product terms from minterm by using some appropriately defined processing rules.
2. Canceling of equal terms.
3. EXOR of all uncancelled terms with polarity p.

In what follows, this technique for switching functions [19] is extended to function defined over $GF(4)$. The proposed TT for MV functions (MVTT) performs the same steps as the TT method for switching functions generalized to MV minterms. However, we redefine processing rules in a way that permits to eliminate summation of product terms with the polarity p resulting in a reduced computational complexity.

Table 2. $GF(4)$ **transform matrices**

$x = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$	$\mathbf{G}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$
${}^1\bar{x} = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 2 \end{bmatrix}$	$\mathbf{G}_4^{<1>} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$
${}^2\bar{x} = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 1 \end{bmatrix}$	$\mathbf{G}_4^{<2>} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 3 & 2 & 0 & 1 \\ 2 & 3 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$
${}^3\bar{x} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \end{bmatrix}$	$\mathbf{G}_4^{<3>} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 3 & 1 & 0 \\ 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

3.1 Generation of New Terms

The rule for generation of new terms from a MV minterm is derived by using the following properties of RMGFE.

From linearity of RMGFE, for arbitrary functions f and g defined over $GF(4)$, it holds

$$GF(f \oplus g) = GF(f) \oplus GF(g),$$

where $GF(f)$ and $GF(g)$ are RMGFE for f and g.

From properties of the Kronecker product, we derive

$$GF^{<p>}(f \oplus g) = GF^{<p>}(f) \oplus GF^{<p>}(g). \qquad (1)$$

Denote by M the decimal index for a minterm $m = (m_1, \ldots, m_n)$ over $GF(4)$. Thus, $M = \sum_{i=1}^{n} m_i 4^{n-i}$.

An n-variable four-valued function f given by the truth-vector $\mathbf{F} = [f(0), f(1), \ldots, f(4^n-1)]^T$ can be represented as a sum of truth-vectors for true MV minterms of f

$$\mathbf{F} = \mathbf{F}_0 \oplus \mathbf{F}_1 \oplus \ldots \oplus = \mathbf{F}_{4^n-1},$$

where $\mathbf{F}_i = [0, 0, \ldots, 0, f(i), 0, \ldots, 0]^T$,

From Equation (1), it follows that

$$GF^{<p>}(\mathbf{F}) = GF^{<p>}(\mathbf{F}_0) \oplus \ldots \oplus GF^{<p>}(\mathbf{F}_{4^n-1}).$$

New terms in RMGFE generated from a given MV minterm m represented by $\mathbf{F}_M$, are determined as $GF^{<p>}(\mathbf{F}_M)$.

Since in each $\mathbf{F}_i$, $i = 0, \ldots, 4^n - 1$, there is a single non-zero element $f(i)$, while $f(j) = 0$ for $j \neq i$, then

$$GF^{<p>}(\mathbf{F}_i) = f(i)\mathbf{g}_i,$$

where $\mathbf{g}_i$ is the i-th column of $\mathbf{G}_4^{<p>}(n)$. Each value in $GF^{<p>}(\mathbf{F}_i)$ can be considered as a contribution of minterm m to the corresponding coefficients in $GF^{<p>}(\mathbf{F}) = \mathbf{C}^{<p>} = [c_0^{<p>}, \ldots, c_{4^n-1}^{<p>}]^T$. These contributions can be calculated as

$$\begin{aligned} c_u^{<p>} &= \bigoplus_{v=0}^{4^n-1} \left(\mathbf{G}_4^{<p>}[u,v] \cdot f(v)\right) \\ &= \bigoplus_{v=0}^{4^n-1} \left(\Pi_{z=1}^{n} \mathbf{G}_4^{<p_z>}[u_z, v_z]\right) \cdot f(v), \end{aligned} \qquad (2)$$

where $u = 0, \ldots, 4^n - 1$, $u = (u_1, \ldots, u_n)$, $v = (v_1, \ldots, v_n)$. $\mathbf{G}_4^{<p>}[u,v]$ denotes the element in the u-th row and the v-th column in $\mathbf{G}_4^{<p>}$, while $\mathbf{G}_4^{<p_z>}[u_z, v_z]$ denotes the element in the u_z-th row and the v_z-th column in $\mathbf{G}_4^{<p_z>}$.

Example 1 *Consider the calculation of the coefficients in FPRMGFE of a two-variable function f for the polarity $p = (2,1)$. By definition,*

$$\mathbf{C}^{<p>} = (\mathbf{G}_4^{<2>} \otimes \mathbf{G}_4^{<1>}) \cdot \mathbf{F}.$$

Contribution of the function value f_3, i.e. of the minterm $m = (03) = x_1 \cdot^3 \bar{x}_2$ to the coefficient c_6 is equal to $\mathbf{G}_4^{<(21)>}[6,3] \cdot f_3 = 1 \cdot f_3$. This contribution can be calculated by using Equation (2) as

$$\mathbf{G}_4^{<2>}[1,0] \cdot \mathbf{G}_4^{<1>}[2,3] \cdot f_3 = 3 \cdot 2 \cdot f_3 = 1 \cdot f_3.$$

Contribution of the minterm $m = (03) = x_1 \cdot^3 \bar{x}_2$ to all coefficients in the FPRMGFE for f is given by

$$\begin{aligned} &GF^{<p>}(\mathbf{F}_3) = \mathbf{G}_4^{<21>} \cdot \mathbf{F}_3 \\ &= \mathbf{G}_4^{<21>} \cdot [0,0,0,f_3,0,0,0,0,0,0,0,0,0,0,0,0]^T \\ &= [0,0,0,0,0,2f_3,f_3,3f_3,0, \\ &\qquad f_3,3f_3,2f_3,0,3f_3,2f_3,f_3]^T. \end{aligned}$$

where multiplication of a constant by a function value is performed in GF(4) and

$$\mathbf{G}_4^{<21>} = \mathbf{G}_4^{<2>} \otimes \mathbf{G}_4^{<1>}.$$

By using Equation (2), we derive a rule for the generation of new terms $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ and determine the contribution $v(\pi)$ of a minterm m as

$$\pi_i \in \{j | \mathbf{G}_4^{<p_i>}[j, m_i] \neq 0\}, \qquad (3)$$

$$v(\pi) = (v_1 \cdot v_2 \cdot \ldots \cdot v_n) \cdot f(M), \qquad (4)$$

where $v_i = \mathbf{G}_4^{<p_i>}[\pi_i, m_i]$, $i = 1, \ldots, n$.

The number of newly generated terms from a minterm m depends on the relationships between bits in the minterm m and the polarity p.

3.2 Cancelling of Terms

In TT for calculation of FPRM expressions, each two newly generated terms that are equal to some previously generated term are deleted, since contribution of each new term is equal 1 and $1 \oplus 1 = 0$. Therefore, in existing TTs, this process is called "cancelling of equal pairs". There are two approaches to cancelling equal pairs, i.e. by using a hash table [2] or an index table with 2^n entries for an n variable function [19]. In a hash table, only uncancelled terms are stored. For each newly generated term we first check if there exists the equal term already in the hash table. If such a term exists, the newly generated term is deleted, otherwise it is included in the hash table. Due to that, the hash table consumes less memory than the index table although in the worst case the hash table can consume 2^n entries as in the index table. Therefore, this approach can be applied for an arbitrary number of variables, but requires some considerable execution time because access to the desired entry in a hash table is on the average slower than in an index table. On the other side, the usage of the index table provides possibilities for parallelization of the related procedures.

In the MV case, we process each minterm of a given function by using Equation (3) to generate new terms and Equation (4) to determine their contributions to the coefficients in a FPRMGFE for f. The contribution of a term takes values in $E(4)$. Therefore, we cannot delete equal terms if they have different contributions, which means if they correspond to different function values. However, the total contribution of a subset of terms can be equal to zero, in which case we delete these terms.

Example 2 *For a function f, assume that from a subset of minterms corresponding to non-zero values for f, we generate three new terms $t_1 = t_2 = t_3$ whose contributions are 2, 3, and 1, respectively. The sum of these contributions is $2 + 3 + 1 = 0$, and thus the terms t_1, t_2, and t_3 can be deleted.*

Let a function f be given by its truth-vector

$$\mathbf{F} = [f_0, f_1, f_2, a, a, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}, a, f_{15}],$$

where $a, f_i \in \{0, 1, 2, 3\}$. Contributions of minterms (03), (10), and (32) to the coefficient $c_6^{<2,1>}$ are $1 \cdot a$, $2 \cdot a$, and $3 \cdot a$, respectively. The sum of these contributions is $1 \cdot a + 2 \cdot a + 3 \cdot a = 0$. Therefore, minterms (03), (10), and (32) do not influence the coefficient $c_6^{<2,1>}$.

3.3 EXOR of Uncanceled Terms

In [19], the last step in TT performs EXOR of uncancelled terms with the given polarity p. The same is done in a similar way in [6]. However, the rule for generation of new terms from a minterm incorporate this step. Thus, this step does not exist in the algorithm to implement MVTT, which improves efficiency of the algorithm in terms of time. Therefore, MVTT consists of the following steps

1. Given a function f by the set of minterms corresponding to non-zero values in f and a polarity $p = (p_1, \ldots, p_n)$. Generate new terms by using the relation above and determine their contributions to the coefficients in the FPRMGFE.

2. Cancel the terms when the sum of their contributions is equal to zero, and store the other terms.

In what follows, we derive an algorithm to perform MVTT by using the index table. However, the same algorithm can be performed by using the hash table.

3.4 MVTT Algorithm Using the Index Table

Let $\mathbf{I}$ be an index table with 4^n entries:

$$\begin{aligned} \mathbf{I} &= (I_0, \ldots, I_{4^n-1}) \\ &= (I_{(0,0,\ldots,0)}, \ldots, I_{(4^n-1,4^n-1,\ldots,4^n-1)}) \end{aligned}$$

The algorithm to perform MVTT for FPRMGFE consists of the following steps:

1. Given a polarity $p = (p_1, p_2, \ldots, p_n)$.

2. Generate $\mathbf{G}_4^{<i>}$, $i \in \{p_1, p_2, \ldots, p_n\}$ (see Table 2).

3. Express a minterm m as a four-valued n-tuple $m = (m_1, m_2, \ldots, m_n)$. Value of m is $f(m)$.

4. For a minterm m generate new terms π_j and determine their contributions $v(\pi_j)$, by using Equation (3) and (4), respectively.

5. For each newly generated term π_j add the value $v(\pi_j)$ to the index table entry $I_{(\pi_1,\ldots,\pi_n)}$.

6. Repeat steps 3, 4, and 5 for all minterms.

Example 3 *Consider the calculation of a FPRMGFE for a two variable function f given by the truth vector $\mathbf{F} = [0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 1, 0, 0, 0]^T$, for the polarity $p = (2, 1)$. By definition*

$$\begin{aligned} \mathbf{C}^{<p>} &= (\mathbf{G}_4^{<2>} \otimes \mathbf{G}_4^{<1>}) \cdot \mathbf{F} \\ &= [0, 0, 0, 0, 3, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1]^T . \end{aligned} \quad (5)$$

We calculate these coefficients by using MVTT as follows: From the non-zero values in the truth-vector $\mathbf{F}$, the function f is given by the minterms

$$f(x_1, x_2) = 1 \, for \, (x_1 x_2) \in \{(30)\}$$

and

$$f(x_1, x_2) = 2 \, for \, (x_1 x_2) \in \{(10), (11), (12), (13)\}.$$

Contribution of the minterms to the coefficients vector $\mathbf{C}^{<2,1>}$ as well as their values in the index table are given in Table 3.

The starting value of the index table (column denoted as sIT) is equal to zero for all the coefficients. The i-th value *in the column denoted as σ. Each minterm represents its contribution to the i-th coefficient in $\mathbf{C}^{<2,1>}$. These contributions are added to the current values in the index table (column IT on the left side) and the new value of the index table is shown in the column denoted by IT on the right side. The last calculated value in the index table includes contributions of all the minterms and it is equal to the coefficient vector (the last column IT in the table). Minterms are processed in the following order:*

$$(10) - 2, (11) - 2, (12) - 2, (13) - 2, (30) - 1$$

Contributions of each minterm given in the form of newly generated terms are shown in Table 4.

Table 4. New minterms

minterm	generated minterms
(10)-2	(11)-2; (12)-2; (13)-2; (21)-1; (22)-1; (23)-1; (31)-2; (32)-2; (33)-2
(11)-2	(10)-3; (13)-3; (20)-1; (22)-1; (30)-2; (33)-2
(12)-2	(11)-1; (12)-2; (13)-3; (21)-2; (22)-3; (23)-1; (31)-3; (32)-1; (33)-2
(13)-2	(11)-2; (12)-1; (13)-3; (21)-3; (22)-2; (23)-1; (31)-1; (32)-3; (33)-2
(30)-1	(11)-1; (12)-1; (13)-1; (21)-1; (22)-1; (23)-1; (31)-1; (32)-1; (33)-1

4 Extensions

The MVTT method introduced in this paper is explained by the example of the calculation of FPGFRME by using the index table and starting from the function represented by minterms. However, this method can also be applied for the calculation of other fixed polarity polynomial expressions for MV functions.

4.1 Calculation of Kronecker Expansions

The method can be extended to expressions for MV function, when the corresponding transform matrices $\mathbf{T}(n)$ have a Kronecker product structure. Such extensions will be explained by the example of Kronecker RM-expressions over $GF(4)$ [17].

Definition 4 *Each n-variable function f defined over $GF(4)$, given by its truth vector $\mathbf{F}$, can be represented by the Kronecker RM Expressions (KRME)*

$$f = \bigotimes_{i=1}^{n} \mathbf{X}_i \cdot \bigotimes_{i=1}^{n} \mathbf{K}_i \cdot \mathbf{F},$$

where

$$\mathbf{X}_i \in \left\{ \left[1, \dot{x}_i, \dot{x}_i^2, \dot{x}_i^3\right], \left[J_0(x_i), J_1(x_i), J_2(x_i), J_3(x_i)\right] \right\},$$

and $\mathbf{K}_i \in \left\{ \mathbf{G}_4^{<i>}, \mathbf{I}_4 \right\}$, $\dot{x}_i$ is the corresponding complement of variable x_i, i.e. ${}^i\bar{x}_i$ while $J_j(x_i), j = 0,1,2,3$ are the characteristic functions defined by

$$J_j(x_i) = \begin{cases} 1, & if x_i = j, \\ 0, & otherwise. \end{cases}$$

Example 4 *Consider a KRME for function f in Example 3 where the 2-Davio $GF(4)$ is used for variable x_1, and the generalized Shannon expansion is used for variable x_2. This KRME is calculated as*

$$\begin{aligned} \mathbf{C} &= \left(\mathbf{G}_4{}^{<2>} \otimes \mathbf{I}_4\right) \cdot \mathbf{F} \\ &= [0,0,0,0,1,0,0,0,0,1,1,1,3,2,2,2] \end{aligned}$$

The difference among the Positive-Polarity RMGFE, FPRMGFE, and KRME is in the expansion rules applied to the variables. Therefore, the modification in the algorithm for implementation of the MVTT for FPRMGFE to calculate the KRMEs consists in the usage of different rules for generation of new terms. In particular, in Equation (3) the matrix $\mathbf{G}$ should be replaced by the corresponding transform matrix $\mathbf{K}_i$.

4.2 MVTT Method Starting from FPRM

The proposed algorithm calculates FPGFRM coefficients starting from the function given by its minterms. This algorithm can be used for the calculation of FPGFRME for a polarity (p') starting from FPGFRME from an arbitrary polarity (p) by changing the rule for the generation of new terms. If a function is given by a fixed polarity expression for the polarity p, and the fixed polarity expression for the polarity p' is required, then the matrix $\mathbf{G}_4^{<p_i>}$ in Equation (3) should be replaced by the matrix $\mathbf{T}^{<p'_i>} \cdot (\mathbf{T}^{<p_i>})^{-1}$.

This modification follows from the property that new terms are considered as contributions of minterms to the calculated coefficient vector, and from the relation

$$\begin{aligned} \mathbf{C}^{<p'>} &= \mathbf{T}^{<p'>}(n) \cdot \mathbf{F} \\ &= \mathbf{T}^{<p'>}(n) \cdot (\mathbf{T}^{<p>}(n))^{-1} \cdot \mathbf{C}^{<p>}. \end{aligned}$$

4.3 MVTT for Extended Polarities

As is pointed out in Section 2, polarity of variables can be considered as a permutation of elements in the vector of possible values for variables. For FPRMGFE in $GF(4)$, we use only 4 out of 4! possible permutations, since other permutations do not reduce the number of non-zero coefficients. However, there are polynomial expressions for MV functions where all the permutations of values for variables reduce the number of non-zero coefficients, as for example *Reed-Muller-Fourier* (RMF) expressions [18]. The proposed method also applies to those expressions. For calculation of coefficients in RMF expressions, the set of possible matrices $\mathbf{G}^{<p_i>}$ in Equation (3) should be extended. For example, for RMF expressions over $GF(4)$, the polarity p may be any permutation of order four.

Table 3. Contributions and index table

		(10)-2		(11)-2		(12)-2		(13)-2		(30)-1	
i	sIT	σ	IT	σ	IT	σ	IT	σ	IT	σ	IT
0	0	0	0	0	0	0	0	0	0	0	**0**
1	0	0	0	0	0	0	0	0	0	0	**0**
2	0	0	0	0	0	0	0	0	0	0	**0**
3	0	0	0	0	0	0	0	0	0	0	**0**
4	0	0	0	3	3	0	3	0	3	0	**3**
5	0	3	3	0	3	1	2	2	0	1	**1**
6	0	3	3	0	3	1	2	2	0	1	**1**
7	0	3	3	3	0	3	3	3	0	1	**1**
8	0	0	0	1	1	0	1	0	1	0	**1**
9	0	1	1	0	1	2	3	3	0	1	**1**
10	0	1	1	0	1	3	2	2	0	1	**1**
11	0	1	1	1	0	1	1	1	0	1	**1**
12	0	0	0	2	2	0	2	0	2	0	**2**
13	0	2	2	0	2	3	1	1	0	1	**1**
14	0	2	2	0	2	1	3	3	0	1	**1**
15	0	2	2	2	0	2	2	2	0	1	**1**

Example 5 *The RMF transform matrix for a four valued function* $\mathbf{R}(1)$ *is given by*

$$\mathbf{R}(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 3 & 3 \end{bmatrix}.$$

The RMF transform matrix which corresponds to the 10-th complement of variable x*,* $^{10}\bar{x}$*, is given by*

$$\mathbf{R}^{<10>}(1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 \\ 2 & 0 & 1 & 1 \\ 1 & 3 & 1 & 3 \end{bmatrix},$$

where the permutations are considered in the lexicographic order ((0123), (0132), (0213), (0231), (0312), (0321), (1023), (1032), (1203), (1230), (1302), (1320), (2013), (2031), (2103), (2130), (2301), (2310), (3012), (3021), (3102), (3120), (3201), (3210)*).*

4.4 Minterms or Disjoint Cubes

The rule for generation of new terms from minterms can be modified into a rule that works with disjoint cubes (the algorithms to generate disjoint cubes are given in [4, 5]) instead of minterms similar as this is done for the binary case in [6]. This modification reduces the time complexity of the related algorithms, since the number of disjoint cubes is usually considerably smaller than the number of minterms. A generalization of that approach to the MV case may be done as follows:

Cubes representing a function f are disjoint if each minterm m, $f(m) \neq 0$, is covered by only one cube. The function f is given by

$$\mathbf{F} = \mathbf{U}_1 \oplus \ldots \oplus \mathbf{U}_t,$$

where $\mathbf{U}_i = \mathbf{F}_{i_1} \oplus \ldots \oplus \mathbf{F}_{i_{4^r}}$, t is the number of disjoint cubes, and r is the order of the cube.

Experimental results given in Table 5 show that implementation of MVTTs is faster over cubes than over minterms. We compare runtimes of our algorithm applied to the calculation of all 4^n FPRMGFE for randomly generated 4-valued functions represented by disjoint cubes and minterms. In the column denoted by *in* the number of variables is given. The number of disjoint cubes and minterms are given in the columns denoted as *cubes* and *mint.*, respectively. The corresponding runtimes in seconds are given in columns denoted by *cruntime* and *mruntime*. The column $(\%)$ shows reduction of the runtimes for function represented by cubes versus function represented by minterms.

The number of minterms or cubes has an important influence on runtimes. Table 6 shows this property. In the column denoted by *cubes*, the number of cubes in 5-variable four-valued randomly generated functions is given, while the corresponding time spend in calculation of all $4^5 = 1024$ FPRMs is given in the column denoted by *runtime*. In this experiment, the savings due to the usage of disjoint cubes instead of minterms range from 60% to 94.57%.

The experiments were carried out on a 60MHz SUN SPARCstation 20 with 128MB of main memory and all the runtimes are given in CPU seconds.

5 Concluding Remarks

We presented a method for calculation of FPRM expressions for functions defined over $GF(4)$, given by minterms or disjoint cubes. The method was extended to calculation

Table 5. MVCTT runtime for functions represented by disjoint cubes and minterms

in	*cubes*	*mint.*	*cruntime*	*mruntime*	*(%)*
3	10	22	0.02	0.05	-60.00
4	10	52	0.28	1.22	-77.05
5	100	328	18.18	72.36	-74.98
6	100	613	157.33	1604.12	-90.19
7	100	1402	2072.59	38169.28	-94.57
8	1000	8155	160599.04	2383390.72	-93.26

of Kronecker expressions over $GF(4)$ and fixed polarity expressions for a required polarity starting from the expression for an arbitrary polarity. The method can be applied also to polynomial expressions where an extended set of complements for variables is allowed.

All useful features of TTs for calculation of FPRM expressions for Boolean functions, as simplicity of involved operations and high possibilities for parallelization [19], are preserved.

Table 6. Number of cubes and runtimes

cubes	*runtime*
1	2.99
100	13.55
200	24.22
300	36.57
400	48.35
500	60.27
600	72.12
700	81.58
800	95.73
900	106.56
1000	120.01

References

[1] Almaini, A.E.A., McKenzie, L., “Tabular techniques for generating Kronecker expansion”, *IEE Proc. Part E*, Vol. 143, No. 4, July 1996, 205-212.

[2] Almaini, A.E.A., Thompson, P., Hanson, D., “Tabular techniques for Reed-Muller logic”, *Int. J. Electronics*, Vol. 70, No.1, 1991, 23-34.

[3] Drechsler, R., Theobald, M., Becker, B., “Fast OFDD-based minimization of fixed polarity Reed-Muller expressions”, *IEEE Trans. on Computers*, Vol.45, No.11, 1996, 1294-1299.

[4] Falkowski, B.J., Schäfer, Perkowski, M.A., “A fast computer methods for the calculation of disjoint cubes for completely and incompletely specified Boolean functions”, in *Proc. 33rd Midwest Symp. on circuits and Systems*, Calgary, Canada, 1990, 1119-1122.

[5] Falkowski, B.J., Perkowski, M.A., “Algorithm for the generation of disjoint cubes for completely and incompletely specified Boolean functions”, *Int. J. Electronics*, Vol.70, No.3, 1991, 533-538.

[6] Falkowski, B.J., Perkowski, M.A., “One more way to calculate Generalized Reed-Muller expansions of Boolean functions”, *Int. J. Electronics*, Vol.71, No.3, 1991, 385-396.

[7] Falkowski, B.J., Rahardja, S., “Efficient computation of quaternary fixed polarity Reed-Muller expansions”, *IEE Proc., Part E*, Vol.142, No.5, 1995, 345-352.

[8] Fisher, L.T., “Unateness properties of AND-EXCLUSIVE-OR logic circuits”, *IEEE Trans. on Computers*, Vol.23, No.2, 1974, 166-172.

[9] Green, D.H, Taylor, I.S., “Modular representation of multiple-valued logic systems”, *IEE Proc., Part E*, Vol.121, No.2, 1974, 166-172.

[10] Green, D.H., “Ternary Reed-Muller switching functions with fixed and mixed polarities”, *Int. J. Electronics*, Vol.67, No.5, 1989, 761-775.

[11] Green, D.H., “Reed-Muller expansions with fixed and mixed polarities over GF(4)”, *IEE Proc., Part E*, Vol.137, No.5, 1990, 380-388.

[12] Harking, B., “Efficient algorithm for canonical Reed-Muller expansions of Boolean functions”, *IEE Proc., Part E*, Vol.137, No.5, 1990, 366-370.

[13] Harking, B., Moraga, C., “Efficient derivation of Reed-Muller expansions in multiple-valued logic systems”, *22nd ISMVL*, Sendai, Japan, 1992, 436-441.

[14] Purwar, S., “An efficient method of computing generalized Reed-Muller expansions from Binary decision diagram”, *IEEE Trans. on Computers*, Vol.40, No.11, 1991, 1298-1301.

[15] Sarabi, A., Perkowski, M.A., “Fast exact and quasy-minimal minimization of highly testable fixed polarity AND/XOR canonical networks”, *Proc. Design Automation Conference*, June 1992, 30-35.

[16] Sasao, T., *Logic Synthesis and Optimization*, Kluwer Academic Publishers, 1999.

[17] Stanković, R.S., Drechsler, R., “Circuit design from Kronecker Galois field decision diagrams for multiple-valued functions”, *27th ISMVL*, Antigonish, Nova Scotia, 1997, 275-280.

[18] Stanković, R.S., Janković, D., Moraga, C., “Reed-Muller-Fourier representations versus Galois field representations for four-valued logic functions”, *Proc. 3rd Int. Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, September 19-20, Oxford, UK, 1997, 269-278.

[19] Tan, E.C., Yang, H., “Fast tabular technique for fixed-polarity Reed-Muller logic with inherent parallel processes”, *Int. J. Electronics*, Vol. 85, No. 85, 1998, 511-520.

[20] Tran, A., “Graphical method for the conversion of minterms to Reed-Muller coefficients and the minimization of EX-OR switching functions”, *IEE Proc. Part E*, Vol. 134, 1987, 93-99.