A New Exact Algorithm for Highly Testable Generalized Partially-Mixed-Polarity Reed-Muller Forms

Xiaoqiang Zeng, Marek Perkowski, Haomin Wu,

Portland State University,
Dept. of Electrical Engineering
Portland, Oregon 97207
Tel: 503-725-5411,
Fax: 503-725-4882
e-mail: mperkows@ee.pdx.edu

Andisheh Sarabi

Viewlogic Systems, Inc.
47211 Lakeview Blvd.
Fremont, CA 94538
Tel: (510) 659-4001x330
Fax: (510) 659-4010
e-mail: andisheh@wcdf.viewlogic.com

Abstract

Generalized Partially-Mixed-Polarity Reed-Muller (GPMPRM) expansions are a canonical sub-family of Generalized Reed-Muller (GRM) expansions and super-family of Fixed-Polarity Reed-Muller (FPRM) expansions. The main motivation to study GPMPRM forms is their very high testability, similar to FPRM forms and better than GRMs, which are also highly testable. Since GPMPRM sub-family of ESOP is much larger than FPRM expansions, the minimal form of this expansion will be much closer to the minimal ESOP than the minimal form of FPRM expansion, with the same order of testability. We give an improved algorithm to find exact GPMPRM forms. By comparing the ratio of the number of forms to the number of operations of the algorithm used to compute the minimal forms with that of the existing algorithms, we conclude that the algorithm is highly efficient. Similar to the expansions based on Kronecker matrix products, this algorithm can be efficiently implemented in hardware using only EXOR operations.

1 Introduction

The problems of finding an exact or high-quality approximate ESOP expressions for Boolean functions are central in this EXOR based synthesis [4, 11, 13, 14, 16]. In an attempt to create better ESOP minimizers, various approaches are being researched. Recently, there has been an interest in Generalized Reed-Muller forms which produce results very close to exact minimum ESOPs [3, 7, 14]. Moreover, they are highly testable [10, 12, 13]. Mathematically, the GRM forms do not exhibit a general structure in the nested hierarchy of families of canonical forms in [5] because in general they are not constructed from Kronecker matrix products [6]. Therefore, it is important to find their sub-families that would have some structure, thus possibly leading to efficient exact algorithms. All sub-families of GRM will inherit their excellent testability, which makes one more argument for their study.

Another argument for investigating forms with well-defined structure is this. When a data-flow structure similar to Fast Fourier Transforms is found, there are many efficient ways of implementing respective algorithms, both in software and in hardware. Such algorithms have been already created for sequential and parallel processors, pipelined processors, Digital Signal Processors, vector processors, cellular automata, systolic arrays and FPGA custom processors. The Fixed Polarity Reed-Muller and Kronecker Reed-Muller [2] expansions have this required structure [17], which makes them good candidates for parallelization and hardware realizations. Moreover, they use only EXOR operations, in contrast to Fast Walsh transform that uses adders and subtractors and Fast Fourier transform that uses adders and multipliers. Therefore, these two forms are good candidates to be implemented in hardware, and especially in FPGA custom engines. However, the quality of solutions based on FPRMs and KRMs is worse than those from GRM. Unfortunately mapping GRM synthesis algorithms [7, 16] to hardware is difficult, which is one of the reasons why we are investigating sub-families of GRM that have a regular structure.

Since the GPMPRM forms can offer $n2^{n-1}2^{2^{n-1}} - (n-1)2^n$ forms including 2^n FPRM forms, the minimal GPMPRM form is much closer to the minimal ESOP than all known forms that have a regular structure. Moreover, GPMPRM forms, being a subset of highly testable GRM forms [10, 12, 13] are much better testable than ESOPs and even the GRM themselves. Sasao [12] introduced a multiple-stack-at testability method for GRMs which requires one additional EXOR gate for every mixed-polarity variable, plus a total of four checking gates for the entire circuit. Using this method for GPMPRMs, only one additional EXOR gate is needed to make the circuit easily testable since only one variable is mixed-polarity in GPMPRMs. This approach makes GPMPRMs practically as easily testable as FPRMs for single-stack-at faults.

The structure of this paper is the following. Section 2 defines GPMPRMs. Section 3 reviews a data-flow algorithm for exact FPRM minimization. Section 4 expands these ideas to exact GPMPRM minimization. An improved algorithm from section 4 is presented in section 5. Section 6 evaluates numerical results, section 7 generalizes GPMPRMs, and section 8 gives the conclusions.

2 GPMPRM expansions

The Generalized Reed-Muller (GRM) forms are created by selecting any combination of literals in the positive polarity (or zero polarity) RM expansion and replacing these with their inverses. They can be expressed as follows:

$$
\begin{aligned}
f(x_1,\ x_2,\ \ldots,\ x_i,\ \ldots,\ x_n)\ =\ & \\
& b_0 \dot{x}_1{}^0 \dot{x}_2{}^0 \cdots \dot{x}_i{}^0 \cdots \dot{x}_n{}^0 \\
& \oplus\ b_1 \dot{x}_1{}^0 \dot{x}_2{}^0 \cdots \dot{x}_i{}^0 \cdots \dot{x}_n{}^1\ \oplus\ \cdots \\
& \oplus\ b_j \dot{x}_1{}^{e_1} \dot{x}_2{}^{e_2} \cdots \dot{x}_i{}^{e_i} \cdots \dot{x}_n{}^{e_n}\ \oplus\ \cdots \\
& \cdots\ \oplus\ b_{2^n-1} \dot{x}_1{}^1 \dot{x}_2{}^1 \cdots \dot{x}_i{}^1 \cdots \dot{x}_n{}^1 \qquad (1)
\end{aligned}
$$

where $\dot{x}_i{}^0 = 1$, $\dot{x}_i{}^1 = \dot{x}_i$, $(i = 1,\ 2,\ \ldots,\ n)$; $b_j \in \{0,1\}$, $(j = 0,\ 1,\ 2,\ \cdots,\ 2^n - 1)$, $e_i \in \{0,1\}$,$(j)_{10} = (e_1 e_2 \cdots e_i \cdots e_n)_2$.
Moreover, for each $\dot{x}_i$ in different terms, $\dot{x}_i = x_i$ or $\dot{x}_i = \bar{x}_i$.

By setting some constraints on the GRM definition, the Generalized Partially-Mixed-Polarity Reed-Muller (GPMPRM) expansion has been defined in [15].

Definition 1 *The* Generalized Partially-Mixed-Polarity Reed-Muller(GPMPRM) *form is created by selecting any combination of the 2^{n-1} literals of one variable in the Fixed-Polarity RM expansion (FPRM) and replacing them with their inverses but keeping all the literals of the other variables under consistent fixed polarities.*

In [15] GPMPRM forms were introduced and an exact algorithm to calculate them was provided. Below, a more efficient approach to the same problem will be presented.

Obviously, from the definition, the GPMPRM forms are included in the GRM forms but include the FPRM forms. Let's consider a 3-variable function. It can be written as the following GRM expression:

$$
\begin{aligned}
f\ =\ & b_0\ \oplus\ b_1(x_3 \oplus \gamma_1)\ \oplus b_2(x_2 \oplus \beta_1) \\
& \oplus b_3(x_3 \oplus \gamma_2)\ \oplus b_4(x_1 \oplus \alpha_1) \\
& \oplus b_5(x_1 \oplus \alpha_2)(x_3 \oplus \gamma_3)\ \oplus b_6(x_1 \oplus \alpha_3)(x_2 \oplus \beta_3) \\
& \oplus\ b_7(x_1 \oplus \alpha_4)(x_2 \oplus \beta_4)(x_3 \oplus \gamma_4) \qquad (2)
\end{aligned}
$$

where:
$\delta(\dot{x}_1) = (\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4)$, $\delta(\dot{x}_2) = (\beta_1,\ \beta_2,\ \beta_3,\ \beta_4)$ and $\delta(\dot{x}_3) = (\gamma_1,\ \gamma_2,\ \gamma_3,\ \gamma_4)$ are the polarity sets of each variable, respectively.
Hence, there are $2^{3 \cdot (2^3\ -\ 1)} = 2^{12}$ GRM forms for a 3-variable function. For the GPMPRM expansion, the literals of one variable can take mixed-polarities, and all the other variables should take fixed-polarities. If x_3 is the mixed-polarity variable, then x_1, x_2 must take fixed polarities. Thus, we have $\gamma_1, \gamma_2, \gamma_3, \gamma_4, \in \{0,1\}$ but $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4\ (= \alpha)\ \in \{0,1\}$, and $\beta_1 = \beta_2 = \beta_3 = \beta_4\ (= \beta)\ \in \{0,1\}$. Since there are 2^4 mixed polarities for the 4 literals of the mixed-polarity variable, 2^2 fixed polarities for the remaining two fixed polarity variables, and 3 choices of selecting the mixed polarity variable, thus there are 176 (i.e. $3 \cdot 2^2 \cdot 2^4 - 2 \cdot 2^3$) GPMPRM forms for a 3-variable function. (The $3 \cdot 2^2 \cdot 2^4$ forms include 3 overlapping 2^3 fixed polarity forms, that is, all the three variables take fixed polarities, so $2 \cdot 2^3$ should be subtracted).

Lemma 1 *For an n-variable function, there are $n2^{n-1}2^{2^{n-1}} - (n-1)2^n$ GPMPRM forms.*

Proof: Each variable has 2^{n-1} literals in 2^n terms of the positive RM canonical form for an n-variable function. When the literals of one variable take the mixed polarities, there are $2^{2^{n-1}}$ mixed polarities for the variable. The other $(n-1)$ variables should take the fixed polarities, and thus there are 2^{n-1} fixed polarities. For an n-variable function, we have n choices of selecting the mixed polarity variable, and therefore, we have $n2^{n-1}2^{2^{n-1}}$ alternative forms including the n-times overlapping 2^n FPRM forms. Hence, the total number of GPMPRM forms is $n2^{n-1}2^{2^{n-1}} - (n-1)2^n$.

The relations among the GPMPRM and the other RM expansion families are shown in Fig. 1.

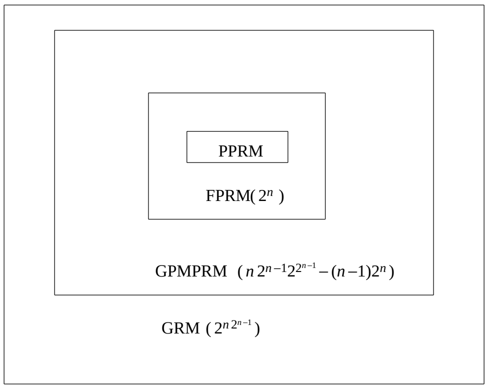

Figure 1: The Relationship among the GRMPRM, PPRM, FPRM and GRM

The GPMPRM class is much larger than the Kronecker Reed-Muller(KRM) expansion, thus generally the minimal form of this expansion should be much closer to the minimal ESOP than the minimal form of the KRM expansion.

3 FPRM Minimization

An algorithm for the minimization of Fixed Polarity Reed-Muller expression has been presented by Zhang and Rayner in [17]. According to this algorithm, only $2^n - 1$ EXOR operations are needed to calculate the coefficients of FPRM expansion from one of its adjacent polarity expansions.

For instance, two FPRM adjacent polarity expansions of a 3-variable function are shown as the following:

$$
\begin{aligned}
f(x_1, x_2, x_3)\ =\ & b_0 \oplus b_1 x_3 \oplus b_2 x_2 \oplus b_3 x_2 x_3 \oplus b_4 x_1 \\
& \oplus b_5 x_1 x_3 \oplus b_6 x_1 x_2 \oplus b_7 x_1 x_2 x_3 \quad (3)
\end{aligned}
$$

$$f(x_1, x_2, x_3) \;=\; b'_0 \oplus b'_1 x_3 \oplus b'_2 \bar{x}_2 \oplus b'_3 \bar{x}_2 x_3 \oplus b'_4 x_1 \oplus b'_5 x_1 x_3 \oplus b'_6 x_1 x_2 \oplus b'_7 x_1 \bar{x}_2 x_3 \quad (4)$$

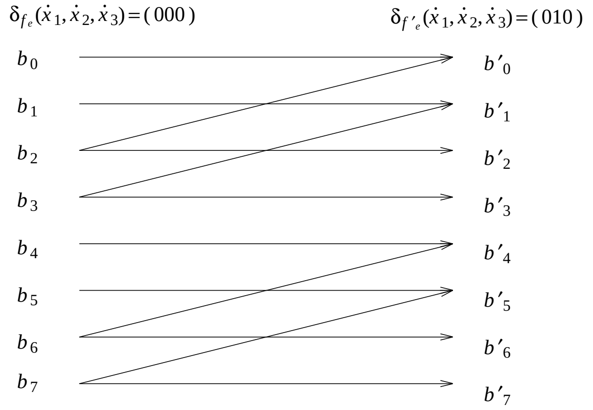

Figure 2: Calculation of f'_e from f_e

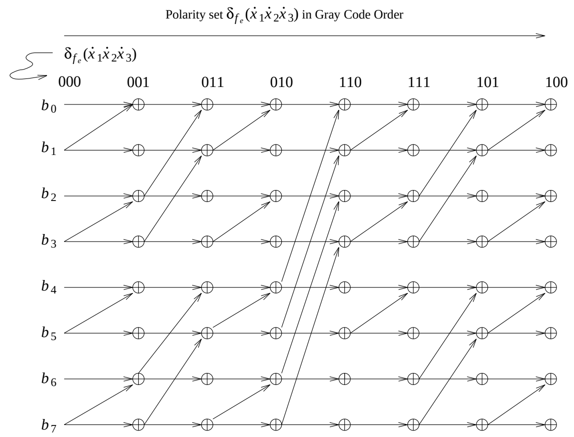

Figure 3: Calculation of FPRM expansions for a 3-variable Function

Notice that between the FPRM expansions (3) and (4), there is only one variable x_2 whose polarity is inverted. To calculate the coefficient of Eqn. (4) from Eqn. (3), we substitute $\bar{x}_2$ with $1 \oplus x_2$ in Eqn. (3) and obtain:

$$f \;=\; b_0 \oplus b_1 x_3 \oplus b_2(\bar{x}_2 \oplus 1) \oplus b_3(\bar{x}_2 \oplus 1)x_3 \oplus b_4 x_1 \oplus b_5 x_1 x_3 \oplus b_6 x_1(\bar{x}_2 \oplus 1) \oplus b_7 x_1(\bar{x}_2 \oplus 1)x_3$$
$$= (b_0 \oplus b_2) \oplus (b_1 \oplus b_3)x_3 \oplus b_2 x_2 \oplus b_3 x_2 x_3 \oplus (b_4 \oplus b_6)x_1 \oplus (b_5 \oplus b_7)x_1 x_3 \oplus b_6 x_1 x_2 \oplus b_7 x_1 x_2 x_3$$
$$= b'_0 \oplus b'_1 x_3 \oplus b'_2 \bar{x}_2 \oplus b'_3 \bar{x}_2 x_3 \oplus b'_4 x_1 \oplus b'_5 x_1 x_3 \oplus b'_6 x_1 \bar{x}_2 \oplus b'_7 x_1 \bar{x}_2 x_3$$

where:

$$b'_0 = b_0 \oplus b_2;\; b'_2 = b_2$$
$$b'_1 = b_1 \oplus b_3;\; b'_3 = b_3$$
$$b'_4 = b_4 \oplus b_6;\; b'_6 = b_6$$
$$b'_5 = b_5 \oplus b_7;\; b'_7 = b_7$$

This can be demonstrated by Fig. 2:

For an n-variable function, if all the polarities of the n variables are arranged according to the Gray code, then each polarity is adjacent to the next one, since the Gray code is a cyclic code, that is, in changing from one value to the next value only one bit is changed. Therefore, all 2^n sets of the RM polynomial coefficients may be arranged as the adjacent polarity coefficients based on Gray code ordering. Then the minimum coefficient can be obtained by exhaustive search through the adjacent polarity data flow diagram.

Fig. 3 is the flow graph representing the algorithm for the calculation of all FPRM expansions for a 3-variable function. Let us observe that in Fig. 3 there are 8 different columns. Each represents one of the 2^3 fixed polarity expansions. Thus, searching through all FPRM expansions leads to the exact minimum FPRM solution.

4 Computation of the GPMPRM Expansion

The main concepts necessary and the algorithm to identify a minimal GPMRPM will be next presented.

Definition 2 *In GPMPRM forms, if a mixed polarity of the 2^{n-1} literals of a variable has only one inverse bit difference with the other mixed polarity of the 2^{n-1} literals of the variable, then these two mixed polarities are defined as the* Adjacent Polarities.

Theorem 1 *When an n-variable GPMPRM form shown in Eqn. (1) is transformed to its adjacent polarity form, that is, a literal $\dot{x}_i$ in the term of $\dot{x}_1^{e_1}\dot{x}_2^{e_2}\cdots\dot{x}_i\cdots\dot{x}_n^{e_n}$ is inverted, only the $b_{j-2^{n-i}}$ is changed into $b_{j-2^{n-i}} \oplus b_j$, and all other coefficients remain unchanged.*

Proof: In the form of Eqn. (1), when the literal $\dot{x}_i$ in the term $\dot{x}_1^{e_1}\dot{x}_2^{e_2}\cdots\dot{x}_i^{e_i}\cdots\dot{x}_n^{e_n}(e_i = 1)$ is inverted, we obtain its adjacent polarity form:

$$f(x_1, ..., x_i, ..., x_n) =$$
$$b'_0\dot{x}_1^0\dot{x}_2^0\cdots\dot{x}_i^0\cdots\dot{x}_n^0 \;\oplus\; b'_1\dot{x}_1^0\dot{x}_2^0\cdots\dot{x}_i^0\cdots\dot{x}_n^1 \;\oplus\; \cdots$$
$$\cdots \oplus\; b'_{j\,-\,2^{n-i}}\dot{x}_1^{e_1}\dot{x}_2^{e_2}\cdots\dot{x}_i^0\cdots\dot{x}_n^{e_n} \;\oplus\; \cdots \;\oplus\; \cdots$$
$$\cdots \;\oplus\; b'_j\dot{x}_1^{e_1}\dot{x}_2^{e_2}\cdots(\dot{x}_i \;\oplus\; 1)\;\cdots\; \dot{x}_n^{e_n} \;\oplus\; \cdots$$
$$\cdots \;\oplus\; b'_{2^n-1}\dot{x}_1^1\dot{x}_2^1\cdots\dot{x}_i^1\cdots\dot{x}_n^1 \qquad (5)$$

From Eqn. (5), we have:

$$f(x_1,\; \cdots,\; x_i, \cdots,\; x_n) = b'_0\dot{x}_1^0\dot{x}_2^0\cdots\dot{x}_i^0\cdots\dot{x}_n^0$$
$$\oplus\; b'_1\dot{x}_1^0\dot{x}_2^0\cdots\dot{x}_i^0\cdots\dot{x}_n^1 \;\oplus\; \cdots$$
$$\oplus\; b'_{j\,-\,2^{n-i}}\dot{x}_1^{e_1}\dot{x}_2^{e_2}\cdots\dot{x}_i^0\cdots\dot{x}_n^{e_n}$$
$$\oplus\; b'_j\dot{x}_1^{e_1}\dot{x}_2^{e_2}\cdots\dot{x}_i^0\cdots\dot{x}_n^{e_n} \;\oplus\; \cdots$$
$$\oplus\; b'_j\dot{x}_1^{e_1}\dot{x}_2^{e_2}\cdots\dot{x}_i\cdots\dot{x}_n^{e_n}$$
$$\oplus\; \cdots \;\oplus\; b'_{2^n-1}\dot{x}_1^1\dot{x}_2^1\cdots\dot{x}_i^1\cdots\dot{x}_n^1 \qquad (6)$$

By comparing Eqn. (6) to Eqn. (1) we have

$$\begin{cases} b_{j-2^{n-i}} = b'_{j-2^{n-i}} \oplus b'_j \\ b_j = b'_j \end{cases} \tag{7}$$

Finally from Eqn. (7), we obtain:

$$\begin{cases} b'_k = b_k; \qquad (k \neq j - 2^{n-i}) \\ b'_{j-2^{n-i}} = b_{j-2^{n-i}} \oplus b_j \end{cases} \tag{8}$$

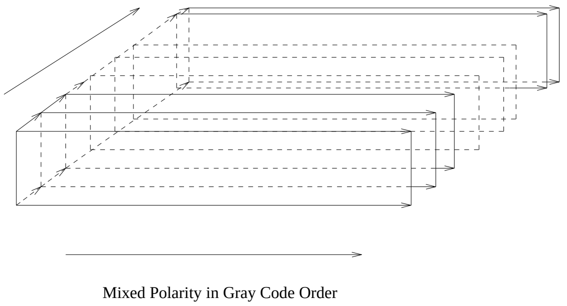

Figure 4: Two-dimensional data flow diagram

From Theorem 1, if the mixed polarities of the 2^{n-1} literals of a variable are arranged in the Gray code order, each GPMPRM form under the mixed polarities can be computed by using only one EXOR operation. The entry vectors are under the fixed polarities of the other variables and the zero polarity of 2^{n-1} literals of this variable. The entry vectors are also computed in Gray code order of the fixed polarities [17]. Actually, this creates a two-dimensional data flow as shown in Fig. 4. One dimension is in Gray code order of the fixed polarities while the other dimension is in Gray code order of the mixed polarities. Thus, according to the Definition 2, all the GPMPRM forms are generated.

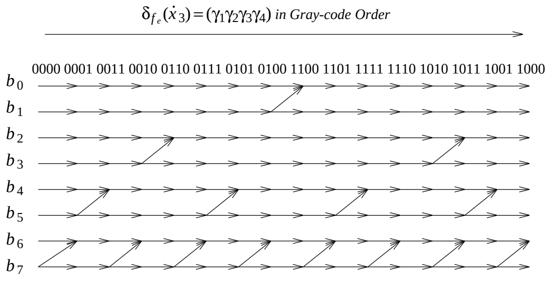

Figure 5: Flow Graph-Calculation of GPMPRM forms when $\dot{x}_3$ is the mixed variable

In the following, our fast computation algorithm will be explained by using a 3-variable function. In Eqn. (2), if the literals of variable x_3 take mixed polarities and x_1, x_2 take fixed polarities, one has $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4\ (= \alpha) \in \{0,1\}$, $\beta_1 = \beta_2 = \beta_3 = \beta_4\ (= \beta) \in \{0,1\}$, and $\gamma_1,\ \gamma_2,\ \gamma_3,\ \gamma_4 \in \{0,1\}$. The expansions under mixed-polarities $\gamma_1,\ \gamma_2,\ \gamma_3,\ \gamma_4$ and fixed-polarities α, β is computed in the flow graph in Fig. 5, where $\gamma_1,\ \gamma_2,\ \gamma_3,\ \gamma_4$ are arranged in Gray code order.

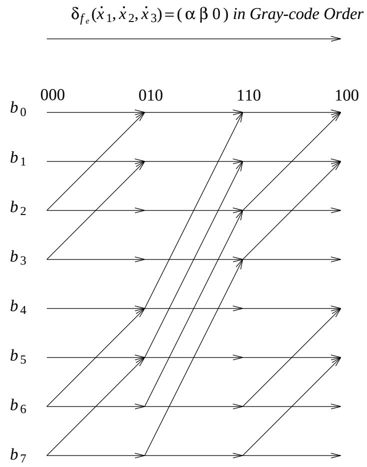

Figure 6: Calculation of FPRM forms: $\dot{x}_1$ and $\dot{x}_2$ are the fixed polarity variables

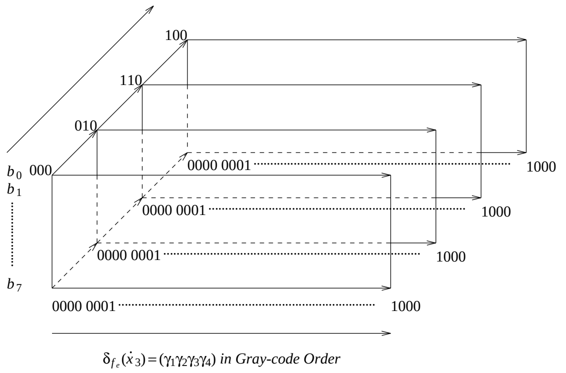

Figure 7: Two dimensional data flow graph with $\dot{x}_3$ being the mixed polarity variable

In Fig. 5 there are 16 columns, each representing one of the 2^4 GPMPRM expansions under the mixed polarity of $\dot{x}_3$ and fixed polarity of $\dot{x}_1$ and $\dot{x}_2$. Throughout the data flow in Fig. 5 α and β, the fixed polarities of $\dot{x}_1$ and $\dot{x}_2$, remain unchanged. The entry vector(first column) is under the zero (positive) polarity of $\dot{x}_3$ ($\delta_{f_e}(\dot{x}_3) = (\gamma_1\gamma_2\gamma_3\gamma_4) = (0000)$) and certain states of α and β, the polarities of $\dot{x}_1$ and $\dot{x}_2$. Since $\alpha,\ \beta \in \{1,0\}$, there are $2^2 = 4$ entry vectors with polarity set $\delta_{f_e}(\dot{x}_1,\ \dot{x}_2,\ \dot{x}_3) = (\alpha\beta 0)$. Each of these entry vectors $[\dot{b}_1,\ \dot{b}_2,\ \cdots,\ \dot{b}_7]^T$ under fixed-polarities α, β in Fig. 5(a) is computed in the flow graph from Fig. 6, where α and β are arranged in Gray code order.

Combining Fig. 5 and Fig. 6 results in a two-dimensional data flow graph, as shown in Fig. 7.

Similarly, if the literals of the variable x_2 take mixed-polarities and x_1 and x_3 take fixed polarities,

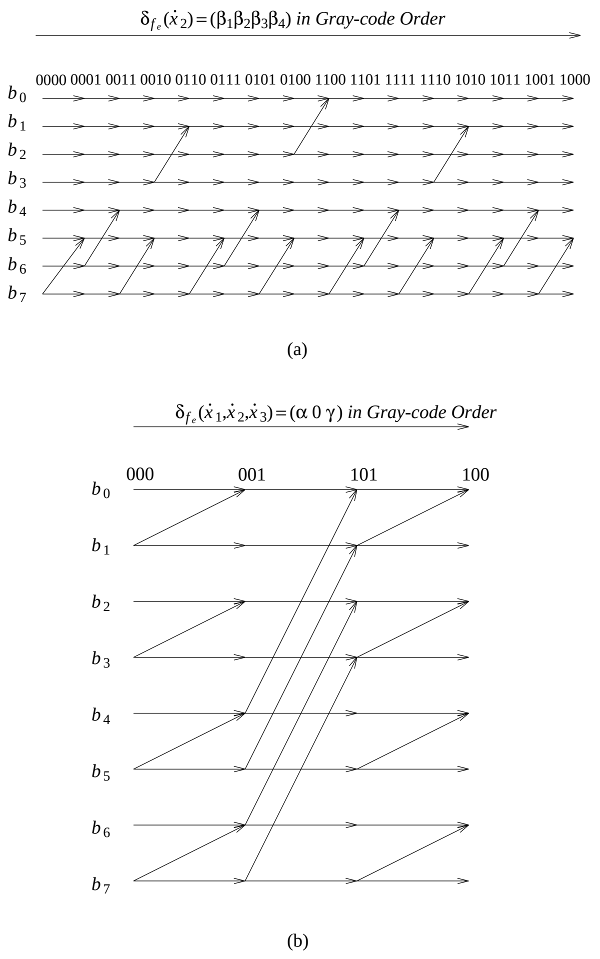

Figure 8: Flow Graph (a) $(\beta_1\beta_2\beta_3\beta_4)$ in Gray code order (b) $(\alpha\gamma)$ in Gray code order

one has $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 \;(= \alpha) \in \{0,1\}$, $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 \in \{0,1\}$, and β_1, β_2, β_3, β_4 $(= \beta) \in \{0,1\}$. The expansions under mixed-polarities β_1, β_2, β_3, β_4 and fixed-polarities α and γ are computed in the flow graph in Fig. 8(a), where β_1, β_2,β_3, β_4 are arranged in Gray code order.

Each entry vector $[\dot{b}_1,\ \dot{b}_2,\ \cdots,\ \dot{b}_7]^T$ under fixed-polarities α, γ in Fig. 8(a) is computed in the flow graph in Fig. 8(b), where α, γ are arranged in Gray code order.

Example 1
If the literals of variable x_1 take mixed-polarities and x_2, x_3 take fixed polarities, one has $\beta_1 = \beta_2 = \beta_3 = \beta_4$ $(= \beta\ ) \in \{0,1\}$, $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 \in \{0,1\}$, and α_1, α_2, α_3, α_4 $(= \alpha) \in \{0,1\}$. The expansions under mixed-polarities α_1, α_2, α_3, α_4 $(= \alpha) \in \{0,1\}$, and fixed polarities β, γ are computed in the flow graph in Fig. 9(a) where α_1, α_2, α_3, α_4 $(= \alpha) \in \{0,1\}$ are arranged in Gray coder order. Each entry vector $[\dot{b}_1,\ \dot{b}_2,\ \cdots,\ \dot{b}_7]^T$ under fixed-polarities β, γ in Fig. 9(a) is computed in the flow graph in Fig. 9(b), where

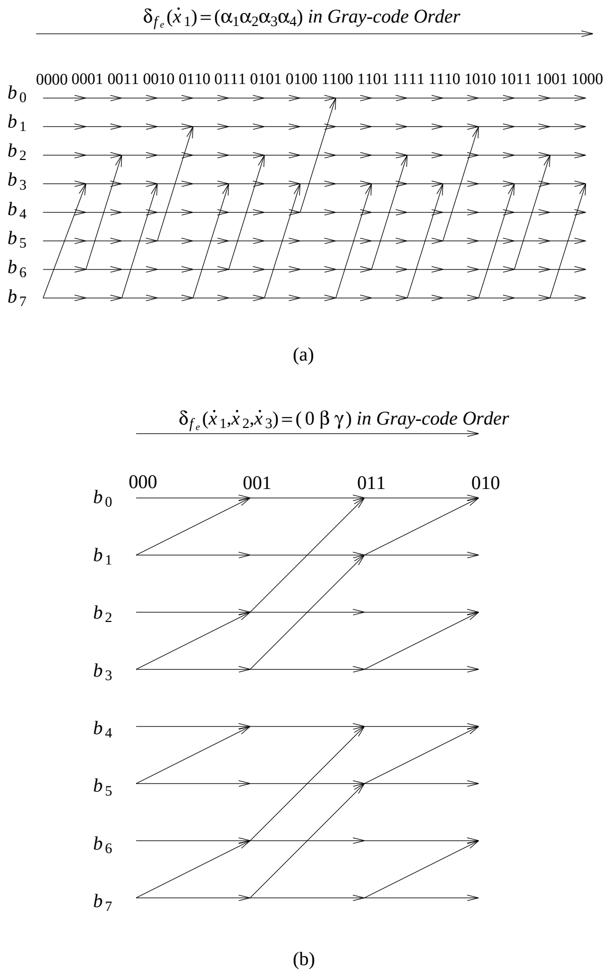

Figure 9: Flow Graph (a) $(\alpha_1\alpha_2\alpha_3\alpha_4)$ in Gray code order (b) $(0\beta\gamma)$ in Gray code order

β, γ are arranged in Gray code order.

As a result, the optimal polarity vector (one with the minimal number of nonzero elements) for the 3-variable function is selected among all the vectors.

Since the encoding of Gray codes is a reflective and cyclic encoding, for any n-variable function this algorithm is a recursive one, and thus it can be readily programmed. Since the algorithm executes only one EXOR operation for each mixed-polarity vector, this algorithm is highly efficient. The computation of GPMPRM expansions for an n-variable function needs a total of $n(2^{n-1}-1)2^{n-1} + n2^{n-1}(2^{2^{n-1}}-1)$ EXOR operations.

The algorithm is formulated as follows.

Algorithm 1 (Minimization of GPMPRM)

1. Start from PPRM, $f_e := PPRM$, $f_{min} := f_e$.
2. Let $i := 1$. Let mp be the polarity set of x_i and fp be the Fixed polarity vector of other variables.
3. $mp := mp + 1$ in Gray code order. Let T_j be the term in which the literal of x_i changes polarity. Now calculate the adjacent polarity GPMPRM expansion

f_e' from f_e by:

$$\begin{cases} b_k' := b_k; & (k \neq j - 2^{n-i}) \\ b_{j-2^{n-i}}' := b_{j-2^{n-i}} \oplus b_j \end{cases}$$

4. If $cost(f_e') < cost(f_{min})$, then $f_{min} := f_e'$
5. $f_e := f_e'$. If $mp < 2^{2^n-1}$, goto step 3. /*search all possible mixed polarity expansions of x_i*/
6. $fp := fp + 1$ in Gray code order. Calculate the adjacent polarity FPRM expansion under fp.
7. If $fp < 2^{n-1} - 1$, go to step 3
   /* search all fixed-polarity expansions */
8. If $i < n$, then $i := i + 1$, goto step 2.

$b_{j-2^{n-i}}$	0	0	1	1
b_j	0	1	0	1
$b'_{j-2^{n-i}} = b_{j-2^{n-i}} \oplus b_j$	0	1	1	0

Figure 10: Table of possible combinations of $b_{j-2^{n-i}}'$

In this algorithm one has to calculate all possible GPMPRM expansions including those that overlap. For functions with more than seven variables it is not feasible because of the large number of forms. To search all GPMPRM expansions n number of searches from Fig. 7 is needed. Thus, the time complexity of this algorithm is $O(n \cdot 2^{n-1} 2^{2^n-1})$. In the next section we are going to give another algorithm which has the same result as Algorithm 1 but with a time complexity of $O(n \cdot 2^n)$.

5 The Speedup Approach

Using Algorithm 1, one has to calculate all the GPMPRM forms in order to find the minimal one. This would not be feasible for functions that have many input variables. In this section, we introduce an algorithm with which we can find the minimum GPMPRM form without necessarily calculating all the forms. Before we present the algorithm formally, let us consider the following example first.

Given is a function in PPRM form:

$f = f_e = 1 \oplus x_2 \oplus x_2x_3 \oplus x_1x_3 \oplus x_1x_2 \oplus x_1x_2x_3 = 1 \oplus 0 \cdot (x_3 \oplus \gamma_1) \oplus x_2 \oplus x_2(x_3 \oplus \gamma_2) \oplus x_1(x_3 \oplus \gamma_3) \oplus x_1x_2 \oplus x_1x_2(x_3 \oplus \gamma_4)$

where $\delta_{f_e} = (\gamma_1\gamma_2\gamma_3\gamma_4) = (0000)$. By inverting all the literals of x_3 in the above expansion, we obtain a FPRM form as follows:

$$f = f_e' = 1 \oplus x_2\bar{x}_3 \oplus x_1 \oplus x_1\bar{x}_3 \oplus x_1x_2\bar{x}_3$$
$$= 1 \oplus 0 \cdot (x_3 \oplus \bar{\gamma}_1) \oplus x_2(x_3 \oplus \bar{\gamma}_2) \oplus x_1 \oplus x_1(x_3 \oplus \bar{\gamma}_3) \oplus x_1x_2(x_3 \oplus \bar{\gamma}_4)$$

where:

$\delta_{f_e'} = (\bar{\gamma}_1\bar{\gamma}_2\bar{\gamma}_3\bar{\gamma}_4) = (1111)$.

Comparing f_e and f_e' we find that the inversion of γ_1 does not cause any change because the coefficient of term $0 \cdot (x_3 \oplus \gamma_1)$ is 0, thus γ_1 can be arbitrary. Inverting γ_2 (x_3 in term x_2x_3) in f_e causes the term x_2 to be reduced in f_e'. Inverting γ_4 (x_3 in term $x_1x_2x_3$) causes term x_1x_2 to be reduced in f_e'. But inverting γ_3 (x_3 in term x_1x_3) creates one more term x_1 in f_e'. Now if we only invert polarities γ_2 and γ_4 but leave the polarities γ_1 and γ_3 unchanged, we have a GPMPRM expansion as follows:

$f = f_e'' = 1 \oplus x_2\bar{x}_3 \oplus x_1x_3 \oplus x_1x_2\bar{x}_3 = 1 \oplus 0 \cdot (x_3 \oplus \gamma_1) \oplus x_2(x_3 \oplus \bar{\gamma}_2) \oplus x_1(x_3 \oplus \gamma_3) \oplus x_1x_2(x_3 \oplus \bar{\gamma}_4)$

This is the minimum GPMPRM form with zero polarity of x_1, x_2 and mixed polarity of x_3.

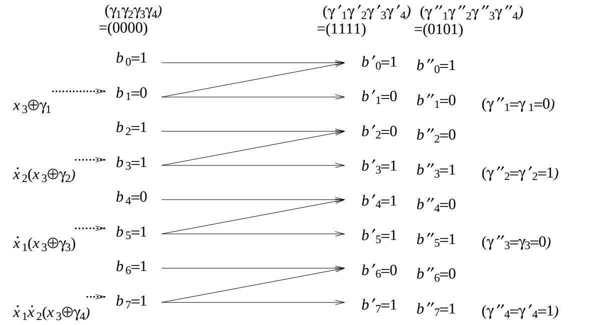

Figure 11: Calculation of the Minimal GPMPRM for a 3-variable Function

From Theorem 1 and Fig. 4, 5, we observe that whenever a literal $\dot{x}_i$ in term T_j is inverted, only the coefficient $b_{j-2^{n-i}}$ is changed to $b_{j-2^{n-i}} \oplus b_j$. This coefficient is only affected by the literal $\dot{x}_j$ in term T_j, no matter how the literals of the same variable in other terms are changed.

For $b_{j-2^{n-i}}' = b_{j-2^{n-i}} \oplus b_j$, the following combinations are possible if $\dot{x}_i$ in term T_j is inverted (Note this inversion can be from x_i to $\bar{x}_i$ or from $\bar{x}_i$ to x_i).

From table in Fig. 10 we see that only when $b_{j-2^{n-i}}$ and b_j are both equal to "1" the cost will be improved.

Owing to the above reasons, the minimum GPMPRM under the mixed-polarity of x_i and certain fixed polarities of other variables can be obtained by attempting to invert any single literal from the $2^n - 1$ literals of x_i. If the inversion of literal $\dot{x}_i$ in term T_j causes $b_{j-2^{n-i}}$ to change from "1" to "0", then the inverted literal is the one that should stand in the minimum GPMPRM. Otherwise the original literal should remain in the minimum GPMPRM. Again we use a 3-variable function to explain our algorithm. In Eqn. (2), if the literals of variable x_3 take mixed polarities and x_1, x_2 take fixed polarities, we first invert the polarities γ_1, γ_2, γ_3, and γ_4 and keep α_1, α_2, α_3, α_4 and β_1, β_2, β_3, β_4 unchanged. The resulting adjacent polarity FPRM expansion has a coefficient vector $f(b_0', b_1', b_2', b_3', b_4', b_5', b_6', b_7')$ and a polarity set $(\gamma_1'\gamma_2'\gamma_3'\gamma_4') = (1111)$, as shown in Fig. 11.

Let $f(b_0'', b_1'', b_2'', b_3'', b_4'', b_5'', b_6'', b_7'')$ be the coefficient set of the minimum GPMPRM, and $(\gamma_1''\gamma_2''\gamma_3''\gamma_4'')$ be the corresponding polarity set, then

$$\text{if } b_j = 1 \text{ and } b_j' = 0 \text{ then } b_j'' = b_j', \quad \gamma_k'' = \gamma_k''$$

$$\text{for others} \qquad b_j'' = b_j, \quad \gamma_k'' = \gamma_k$$

In Example 1,

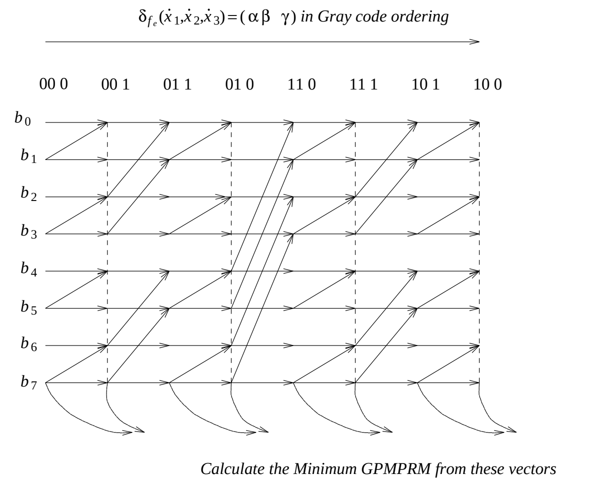

Figure 12: Flow graph, α, β, γ in Gray-code Order

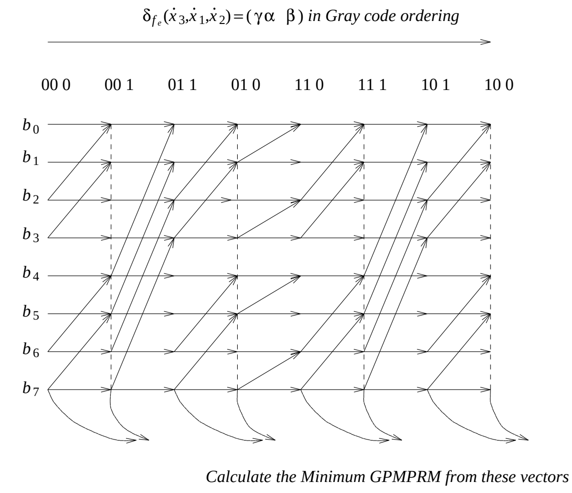

Figure 13: Flow graph, γ, α, β in Gray-code Order

$f(b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7) = f(10110111)$ and $(\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4) = (0000)$.
We have $f(b'_0, b'_1, b'_2, b'_3, b'_4, b'_5, b'_6, b'_7) = f(10011101)$ with $(\gamma'_1 = \gamma'_2 = \gamma'_3 = \gamma'_4) = (1111)$.

Thus the minimum GPMPRM will be $f(b''_0, b''_1, b''_2, b''_3, b''_4, b''_5, b''_6, b''_7) = f(10010101)$ with the polarity set of x_3 $\delta(\dot{x}_3) = (\gamma''_1\gamma''_2\gamma''_3\gamma''_4) = (0101)$.

Fig. 11 is actually the calculation of Adjacent Polarity FPRM Expansion under which x_3 is the variable that is inverted. From Fig. 8 we obtain the same result as in Fig. 5. By combining Fig. 11 and Fig. 6 we obtain Fig. 12 from which we can obtain the minimum GPMPRM with x_3 being the mixed polarity variable.

In Fig. 12, γ is arranged as the least significant bit (1st bit). β and α are arranged as the 2nd and the 3rd bits, respectively. During every increment in Gray-code ordering, if the inverted bit is the 1st bit, then we can calculate the optimal GPMPRM expansion, as stated in the above. If the inverted bit is the 2nd or the 3rd, then we can calculate the FPRM under fixed polarities of α and β, the same result as from Fig. 6. In order to distinguish the mixed polarity variable and the fixed polarity variables, in Fig. 12 we leave a space between the least significant bit and the others.

Similarly, Fig. 8 and Fig. 9 can be transformed into data flow diagrams, as shown in Fig. 13 and Fig. 14, in which the polarity bit of the mixed polarity variable is always arranged as the least significant bit.

The algorithm is presented as the following.

Algorithm 2: (Fast minimization of GPMPRM)

1. Start from PPRM, let $f_{min} := PPRM$. $cost_{min} := cost(PPRM)$.
2. $i := 1$. Let x_i be the mixed polarity variable; $f_e := PPRM$; fp be the fixed polarity vector in which the polarity of x_i is arranged as the least significant bit. Let δ be the zero polarity of x_i. δ' be the polarity set of x_i in the minimal GPMPRM.
3. $fp := fp + 1$ in Gray code order. Calculate the Adjacent Polarity FPRM Expansion f'_e from f_e by:
for($j := 0; \leq 2^n - 1; j++$) {
    if $\overline{(T}_{j+2^{n-i}}$ contains literal x_i)
        then $b'_j := b_j \oplus b_{j+2^{n-i}}$;
    else
        $b'_j := b_j$;
}
4. If $cost(f'_e) < cost_{min}$, then $f_{min} := f'_e$.
5. If the bit changed is the least significant bit, derive the optimal GRMPRM f''_e from f'_e and f_e by:
if $b_j = 1$ and $b'_j = 0$ then $b''_j := b'_j$, $\gamma''_k := \gamma'_k$
for others $b''_j := b_j$, $\gamma''_k := \gamma_k$
6. If $cost(f''_e) < cost_{min}$, then $f_{min} := f''_e$.
7. $f_e := f'_e$. If $fp < 2^n$ then goto step 2.
/* search all mixed polarity expansions of $\dot{x}_i$ */
8. If $i < n$, then $i := i + 1$. Goto step 2.
/* go through all variables */

To find the minimum FPRM expansion one has to calculate 2^n FPRM forms. By calculating $n2^n$ FPRM forms, we can obtain the minimum GPMPRM expansion, which is much closer to the minimum ESOP, than is the minimum FPRM form. The time complexity of Algorithm 2 is $O(n \cdot 2^n)$.

6 Implementation and Evaluation of exGPMPRM

Algorithm 2 presented in the previous section has been implemented in a program called exGPMPRM. It reads in an input file and outputs the exact minimum GPMPRM. The cyclic code is used through out the program for the polarity encoding. Over 110 single-output functions generated from MCNC benchmark have been tested and the results are very encouraging.

6.1 Pseudo-Code for Program exGPMPRM

In our program, counting is implemented in function $k = count()$ where k is the bit that changes from

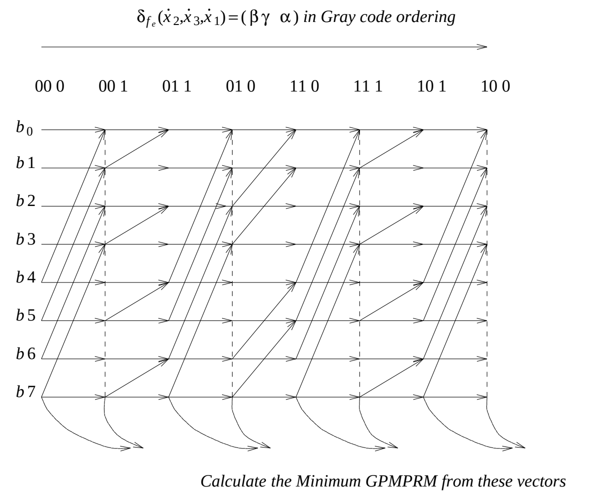

Figure 14: Flow graph, β, γ, α in Gray-code Order

"0" to "1" in this increment.
Following is the pseudo code of exGPMPRM.

```
/* exact minimum GPMPRM */
exGPMPRM()
{
   read in input file f_i in FPRM expression;
    f_min := f_i;
   for(i := 1; i <= n; i++) {
       /* go through all variables */
     f_e = f_i;
     fp := polarity of f_e;
    /* fp is a n-bit polarity vector of FPRM. */
     /* the polarity of x_i is arranged at the */
     /* 1st bit, the least significant bit */
     do {
         k := count();
         inverse the kth-bit of fp;
         for(j := 0; j <= 2^n - 1; j++) {
         /*calculate the adjacent polarity */
         /* expansion* f'_e from f_e */
             if (T_{j + 2^{n-i}} contains x_i)
                    b'_j := b_j XOR b_{j + 2^{n-i}};
             else
                    b'_j := b_j;
         }
         if(cost(f'_e) < cost(f_min))
              f_min := f'_e
         if(k = 1) {
             for(j := 0; j <= 2^n - 1; j++) {
          /*calculate the optimal GPMPRM */
          /* f''_e from f'_e and f_e*/
                if(b_j = 1 and b'_j = 0)
                       b''_j := b'_j;
                else
                       b''_j := b_j;
             }
             if(cost(f''_e) < cost(f_min))
                  f_min := f''_e
         }
```

Function	Var	GPMPRM	ESPRE	CGRMIN	Time
5x3	7	14	18	19	0.22
5x5	7	6	10	7	0.20
5xp11	7	9	7	12	0.22
9sym	9	139	86	173	3.9
con1	7	9	6	12	0.22
f52	8	14	18	19	0.85
f53	8	10	14	11	0.83
misex47	11	13	4	18	72.22
misex54	6	12	11	16	0.05
misex58	6	5	6	7	0.07
misex62	10	40	7	50	16.45
misex63	10	62	7	84	16.48
misex64	10	21	4	28	16.22
rd532	5	5	16	5	0.03
rd732	7	7	64	7	0.22
rd842	8	8	128	8	0.83
sao22	10	37	20	58	16.58
sao23	10	35	22	47	16.52
xor5	5	5	16	5	0.00
z42	7	9	28	9	0.18
z4ml	7	22	7	25	0.20
Total terms		482	499	620	

Figure 15: Table of benchmark results for GPMPRM, SOP, and FPRM

```
             f_e := f'_e;
         }while(k != n + 1);
     }
     output f_min;
}
```

6.2 Experimental Results

As most of the benchmarks are multi-output, the BLIF format of the functions was used to generate single output components of these functions for testing. These functions were first run through the minimizer CGRMIN [9], which gave the minimum solutions of FPRM. Then the minimum FPRMs were used as the input functions to exGPMPRM. The results of this program are shown in the table from Fig. 15. The column headed *Function* denotes the name of the function from the benchmark. *Var* stands for the number of input variables. The columns headed GPMPRM, ESPRE and CGRMIN are the number of terms in the output files of minimizers exGPMPRM, ESPRESSO and CGRMIN [9], respectively. The results of exGPMPRM program are always equal or better than CGRMIN, and are on average 22% better. For instance, function 9sym has 139 terms in exact GPMPRM while it has 173 gates in exact FPRM solution from CGRMIN. Adding 5 gates for testability we get 144 gates versus 173, which is still better, with the same number of tests for single stack-at faults.

7 The k-Variable-Mixed GRMs

The ideas presented in this paper lead to defining new highly testable canonical forms with mathematical structures that are supersets of GPMPRMs and subsets of GRMs. Such form is defined by allowing **a subset of input variables in GRM** to have mixed polarities. This leads to a family of 2^n new families of families of canonical forms. All these families can be represented as a lattice of subsets. FPRM family will be the minimum element, GRM family the maximum element, and all GPMPRM families for every single variable will be elements of the lattice one level higher than the minimum FPRM family. We will call these families the *k-variable-mixed GRMs* ($1 \leq k \leq n$), or *k-VMGRM*, for short. Thus 0-VMGRM is a FPRM, n-VMGRM is a GRM, and 1-VMGRMs are the GPMPRM families of forms, which means that FPRM is a special case of k-VMGRMs with no mixed variables, GPMPRM families for each variable - a case with one mixed variable, and GRM family a case with all variables mixed. Unfortunately, we did not find yet an efficient exact algorithm for $k > 1$, and the straightforward generalization of the algorithm from this paper to flow-diagrams with four and more dimensions leads to many repetitions of forms.

8 Conclusions

A new efficient and exact algorithm for highly testable Generalized Partially Mixed Polarity Reed-Muller Expansions (GPMPRM), which are a subclass of the Generalized Reed-Muller expansions, has been presented. We showed that the results are always better than for FPRM and are of similar testability, which does not leave FPRMs any advantages over GPMPRMs from now on.

Since the new class of GPMPRMs contains many more forms than the KRM family, we expect its minimum form to be generally closer to the minimum ESOP, than is the minimum KRM form. For several examples that we have tested, the GPMPRMs were both smaller and better testable than the KRMs. We plan to perform a systematic comparizon but we have no exact KRM minimizer yet. As another application of GPMPRMs, the exact solution to the minimization of this new form provides an upper-bound for the minimization of the GRM expansion of the same function, [16].

We proved also in this paper that to calculate a GPMPRM expansion from one of its adjacent polarity expansions, only one EXOR operation is needed. By calculating the adjacent polarity expansions one-by-one and searching all the GPMPRM forms, the minimum one can be found. A speedup approach allows thus to find the exact minimum GPMPRM without calculating all forms. Comparing to other **exact** algorithms the efficiency of exGPMPRM is high. Because of its regularity, the proposed by us algorithm can be also realized in hardware, as the only algorithm for forms other than subsets of KRMs.

Future work includes development of GPMPRM minimizers for multi-output functions and incompletely specified functions. We hope also to find better algorithms for k-VMGRMs with $k > 1$, $k \neq n$. The ideas outlined here can also serve to create efficient algorithms for the minimization of forms that are more general than the AND/EXOR forms - namely the *non-singular canonical forms* [1, 13, 8].

References

[1] M. A. Perkowski, A. Sarabi, F. R. Beyl, "XOR Canonical Forms of Switching Functions", *Proc. of the IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, Hamburg, Germany, September 1993.

[2] M. Davio, J. P. Deschamps, and A. Thayse, *Discrete and Switching Functions*, Mc-Graw-Hill, 1978.

[3] M. Cohn, "Inconsistent Canonical Forms of Switching Functions", *IRE Trans. on Electr. Comp.*, Vol. EC-11, pp. 284-285, 1962.

[4] M. Helliwell and M.A. Perkowski, "A Fast Algorithm to Minimize Multi-Output Mixed-Polarity Generalized Reed-Muller Forms", *Proc. of the 25th DAC*, pp. 427-432, 1988.

[5] D. H. Green, "Families of Reed-Muller canonical forms", *Intern. J. of Electr.*, pp. 259-280, Febr. 1991.

[6] D. H. Green, "Reed-Muller Canonical Forms With Mixed Polarity and Their Manipulations", *Proc. of IEE Pt. E*, Vol. 137, part E, pp. 103-113, Jan. 1990.

[7] M. A. Perkowski, L. Csanky, A. Sarabi, and I. Schäfer, "Fast Minimization of Mixed-Polarity AND/XOR Canonical Networks", *Proc. of the ICCD*, pp. 33-36, Cambridge, MA, October 1992.

[8] M. A. Perkowski, A. Sarabi, and F. R. Beyl, "Fundamental Theorems and Families of Forms for Binary and Multiple-Valued Linearly Independent Logics," *in print*, this Workshop, 1995.

[9] A. Sarabi and M. A. Perkowski, "Fast Exact and Quasi-Minimal Minimization of Highly Testable Fixed-Polarity AND/XOR Canonical Networks", *Proc. of the 29th DAC*, pp. 30-35, Anaheim, CA, June 1992.

[10] A. Sarabi and M. A. Perkowski, "Design For Testability Properties of AND/XOR Networks", *Proc. of the IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, Hamburg, Germany, September 1993.

[11] T. Sasao, *Logic Synthesis and Optimization*, Kluwer Academic Publishers, 1993.

[12] T. Sasao, "Easily Testable Realizations for Generalized Reed-Muller Expressions," *Proc. IEEE Third Asian Test Symposium*, pp. 157-162, Nov. 1994.

[13] T. Sasao and M. Perkowski, *EXOR Logic Synthesis and Optimization*, Kluwer Academic Publishers, unpublished.

[14] N. Song, *Minimization of Exclusive Sum of Products Expressions for Multiple-Valued Input Incompletely Specified Functions*, Master thesis, Portland State University, 1994.

[15] H. Wu, M.A. Perkowski, N. Zhuang, and X. Zeng, "Generalized Partially-Mixed-Polarity Reed-Muller Expansion and Its Fast Computation", submitted to *IEEE Trans. on Comp.*, 1994.

[16] X. Zeng, M. A. Perkowski, K. Dill, and A. Sarabi, "Approximate Minimization of Generalized Reed-Muller Forms", *in print*, this Workshop, 1995.

[17] Y.Z. Zhang and P.J.W. Rayner, "Minimization of Reed-Muller Polynomials with Fixed Polarity", *Proc. of IEE Pt. E*, Vol. 131, part E, pp. 177-186, 1984.