PAPER *Special Section on VLSI Design and CAD Algorithms*

Generalized Reed-Muller Expressions: Complexity and an Exact Minimization Algorithm*

Tsutomu SASAO†, ***Member and*** **Debatosh DEBNATH**†, ***Nonmember***

SUMMARY A generalized Reed-Muller expression (GRM) is obtained by negating some of the literals in a positive polarity Reed-Muller expression (PPRM). There are at most $2^{n2^{n-1}}$ different GRMs for an n-variable function. A minimum GRM is one with the fewest products. This paper presents certain properties and an exact minimization algorithm for GRMs. The minimization algorithm uses binary decision diagrams. Up to five variables, all the representative functions of NP-equivalence classes were generated and minimized. Tables compare the number of products necessary to represent four- and five-variable functions for four classes of expressions: PPRMs, FPRMs, GRMs and SOPs. GRMs require, on the average, fewer products than sum-of-products expressions (SOPs), and have easily testable realizations.

key words: *AND-EXOR, Reed-Muller expression, complexity of logic networks, logic minimization, binary decision diagrams, easily testable networks*

1. Introduction

Conventional logic design is based on AND and OR gates. However, exclusive-OR (EXOR) based designs have certain advantages. The first is that arithmetic and telecommunication circuits are efficiently realized with EXOR gates [21]. Examples of such circuits are adders and parity checkers. The second advantage is that the circuits can be made easily testable by using EXOR gates. Various classes exist in AND-EXOR expressions [9], [13], [20]. Among them, positive polarity Reed-Muller expressions (PPRMs) are well known: a PPRM, an exclusive-OR sum-of-products with positive literals, uniquely represents an arbitrary logic function of n variables. Networks based on PPRMs are easily testable [15], [16], but they require more products than ones based on other expressions. Generalized Reed-Muller expressions (GRMs) [4] are generalization of PPRMs. They were studied many years ago [2], but no practical applications have been shown. Recently, we have developed easily testable realizations for GRMs [23]. Because GRMs require many fewer products than PPRMs and have very good testability, the optimization of GRMs have practical importance. As for the optimization of GRMs, only a few papers have been published [3], [5], [14]. This paper presents some properties and an exact minimization algorithm for GRMs. GRM based design is useful in field programmable gate arrays (FPGAs), where ORs and EXORs have the same costs.

Manuscript received March 21, 1996.

†The authors are with the Department of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka-shi, 820 Japan.

*This paper is based on [24].

2. Definitions and Basic Properties

2.1 PPRM, FPRM, and GRM

Definition 1: An expression for f is said to be *minimum* if it represents f and has the least number of product terms.

The following Lemma is the basis of the EXOR-based expansion:

Lemma 1: [25] An arbitrary logic function $f(x_1, x_2, \ldots, x_n)$ can be expanded as

$$f = \bar{x}_1 f_0 \oplus x_1 f_1, \qquad (1)$$

$$f = f_0 \oplus x_1 f_2, \qquad (2)$$

$$f = f_1 \oplus \bar{x}_1 f_2, \qquad (3)$$

where $f_0 = f(0, x_2, \ldots, x_n)$, $f_1 = f(1, x_2, \ldots, x_n)$, and $f_2 = f_0 \oplus f_1$.

Equations (1), (2) and (3) are called the *Shannon expansion*, the *positive Davio expansion*, and the *negative Davio expansion*, respectively. If we use Eq. (2) recursively to a function f, then we have the following:

Lemma 2: [25] An arbitrary n-variable logic function $f(x_1, x_2, \ldots, x_n)$ can be represented as

$$\begin{aligned} f = {} & a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_n x_n \\ & \oplus a_{12} x_1 x_2 \oplus a_{13} x_1 x_3 \oplus \cdots \oplus a_{n-1\,n} x_{n-1} x_n \\ & \oplus \cdots\cdots\cdots \oplus a_{12\cdots n} x_1 x_2 x_3 \cdots x_n, \end{aligned} \qquad (4)$$

where a's are either 0 or 1.

Equation (4) is called a *positive polarity Reed-Muller expression* (PPRM). For a given function f, the coefficients $a_0, a_1, a_2, \ldots, a_{12\cdots n}$ are uniquely determined. Thus, the PPRM is a canonical representation. This unique representation is also the minimum. The number of products in Eq. (4) is at most 2^n and all the literals are positive (uncomplemented).

In Eq. (4), for each variable x_i $(i = 1, 2, \ldots, n)$, if we use either the positive literal (x_i) throughout or the negative literal $(\bar{x}_i)$ throughout, then we have a *fixed polarity Reed-Muller expression* (FPRM). For each variable x_i, there are two ways of choosing the polarities:

positive (x_i) or negative ($\bar{x}_i$). Thus, 2^n different set of polarities exist for an n-variable function. For a given function and a given set of polarities, a unique set of coefficients $(a_0, a_1, \ldots, a_{12\cdots n})$ exists. Thus, an FPRM is a canonical representation.

In Eq. (4), if we can freely choose the polarity for each literal, then we have a *generalized Reed-Muller expression* (GRM). Unlike FPRMs, both x_i and $\bar{x}_i$ can appear in a GRM. There are $n2^{n-1}$ literals in Eq. (4), so $2^{n2^{n-1}}$ different set of polarities exist for an n-variable function. For a given set of polarities, a unique set of coefficients $(a_0, a_1, \ldots, a_{12\cdots n})$ exists. Thus, a GRM is a canonical representation for a logic function. Properties were analyzed in [3] for GRMs and an exact minimization algorithm was shown. However, this algorithm can simplify functions with only a few input variables. In the next section, we will develop a more efficient minimization algorithm for GRMs.

Example 1:

1. $x_1x_2x_3 \oplus x_1x_2$ is a PPRM.
2. $x_1x_2\bar{x}_3 \oplus x_2\bar{x}_3$ is an FPRM, but not a PPRM (x_3 has negative literals).
3. $x_1 \oplus x_2 \oplus \bar{x}_1\bar{x}_2$ is a GRM, but not an FPRM (x_1 and x_2 both have positive and negative literals).

From the above arguments, we have the following:

Theorem 1: Suppose that $\mathcal{PPRM}$, $\mathcal{FPRM}$ and $\mathcal{GRM}$ denote the corresponding set of expressions. Then, the following relations hold:

$$\mathcal{PPRM} \subset \mathcal{FPRM} \subset \mathcal{GRM}.$$

Tables 1 and 2 show the number of 4- and 5-variable functions requiring t products for different classes of minimum expressions, where SOP denotes *sum-of-products expressions*. In the case of five-variable functions, on the average, GRMs require 6.230 products while SOPs require 7.463 products.

Table 1 Number of 4-variable functions requiring t products in minimum expressions.

t	PPRM	FPRM	GRM	SOP
0	1	1	1	1
1	16	81	81	81
2	120	836	2212	1804
3	560	3496	20856	13472
4	1820	8878	37818	28904
5	4368	17884	4512	17032
6	8008	20152	56	3704
7	11440	11600		512
8	12870	2336		26
9	11440	240		
10	8008	32		
11	4368			
12	1820			
13	560			
14	120			
15	16			
16	1			
av	8.00	5.50	3.68	4.13

av : average

Definition 2: Let $\eta(PPRM : n)$, $\eta(GRM : n)$, and $\eta(SOP : n)$ denote the average numbers of products needed in the minimal representation for n-variable functions by PPRMs, GRMs, and SOPs, respectively.

Theorem 2: $\eta(PPRM : n) = 2^{n-1}$.

Proof: An arbitrary function of n variables can be written as Eq. (4). The average is

$$\begin{aligned}\eta(PPRM : n) &= \frac{1}{2^{2^n}} \sum_{t=0}^{2^n} t \cdot (\text{number of functions} \\ &\qquad \text{requiring } t \text{ products}) \\ &= \frac{1}{2^{2^n}} \sum_{t=0}^{2^n} t \binom{2^n}{t} \\ &= \frac{1}{2^{2^n}} 2^n 2^{2^n - 1} \\ &= 2^{n-1}. \qquad \square\end{aligned}$$

Definition 3: Let $\tau(GRM : f)$ denote the number of products in a minimum GRM for f. Let $\tau(GRM : n)$ denote the maximum number of products to realize an n-variable function by minimum GRMs.

Lemma 3: $\tau(GRM : n) \leqq 2\tau(GRM : n-1)$.

Proof: An arbitrary n-variable function can be expanded as $f = f_0 \oplus x_n f_2$, where f_0 and f_2 are functions of variables $x_1, x_2, \ldots,$ and x_{n-1}. Let G_0 and G_2 be minimum GRMs for f_0 and f_2, respectively. Note that if G_0 and G_2 are GRMs, then $G_0 \oplus x_n G_2$ is also a GRM. Thus, we have $\tau(GRM : f) \leqq \tau(GRM : f_0) + \tau(GRM : f_2)$. Because $\tau(GRM : f_0)$ and $\tau(GRM : f_2)$ are at most $\tau(GRM : n-1)$, f can be represented by a GRM with at most $2\tau(GRM : n-1)$ products. $\square$

Table 2 Number of 5-variable functions requiring t products in minimum expressions.

t	FPRM	GRM	SOP
0	1	1	1
1	243	243	243
2	6932	24452	20676
3	79820	1283820	818080
4	575930	36127630	16049780
5	3228162	489868278	154729080
6	14327120	2243146768	698983656
7	49694224	1494589544	1397400512
8	138496600	29183904	1254064246
9	319912340	677056	571481516
10	587707228	65600	160200992
11	877839192		34140992
12	955078352		6160176
13	803257168		827120
14	393502216		84800
15	130238200		5312
16	19114960		114
17	1816640		
18	88032		
19	3680		
20	208		
21	48		
av	11.566	6.230	7.463

av : average

Lemma 4: $\eta(GRM : n) \leq 2\eta(GRM : n-1)$.

Proof: From the proof of Lemma 3, $\tau(GRM : f) \leq \tau(GRM : f_0) + \tau(GRM : f_2)$. Let F_n be the set of all the n-variable functions.

$$\begin{aligned}\eta(GRM : n) &= \frac{1}{2^{2^n}} \sum_{f \in F_n} \tau(GRM : f) \\ &\leq \frac{1}{2^{2^n}} \sum_{f \in F_n} \{\tau(GRM : f_0) \\ &\quad + \tau(GRM : f_2)\} \\ &= \frac{1}{2^{2^n}} \cdot 2^{2^n} \{\eta(GRM : n-1) \\ &\quad + \eta(GRM : n-1)\} \\ &= 2\eta(GRM : n-1). \qquad \square\end{aligned}$$

Theorem 3: $\eta(GRM : n) \leq (6.230) \cdot 2^{n-5}$, when $n \geq 5$.

Proof: From Lemma 4, we have $\eta(GRM : n) \leq 2^{n-5}\eta(GRM : 5)$. From Table 2, we have $\eta(GRM : 5) = 6.230$. Hence, we have the theorem. $\square$

Theorems 2 and 3 show that GRMs require, on the average, less than a half of the products for PPRMs.

Similarly, we have the following theorem.

Theorem 4: $\eta(SOP : n) \leq 7.463 \cdot 2^{n-5} \quad (n \geq 5)$.

Table 2 also shows that GRMs require fewer products than SOPs. Thus, we have the following:

Conjecture 1: $\eta(GRM : n) \leq \eta(SOP : n)$.

The above conjecture considers the number of products on the average. However, there are exceptions. There exists functions whose minimum GRMs require more products than their minimum SOPs. For example, the n-variable function $x_1x_2\cdots x_n \vee \bar{x}_1\bar{x}_2\cdots\bar{x}_n$ requires n products in a minimum GRM and two products in a minimum SOP.

3. Some Properties of GRMs

Definition 4: Let p be a product. The set of variables in p is denoted by $V(p) = \{x_i \mid x_i \text{ or } \bar{x}_i \text{ appears in } p\}$.

Example 2: $V(x_1\bar{x}_2\bar{x}_4) = \{x_1, x_2, x_4\}$.

Definition 5: Let G be a GRM. A product p is said to have a *maximal variable set* if $V(p) \not\subset V(p')$, for all other products p' in G.

Example 3: Let a GRM be $G = x_1\bar{x}_2 \oplus \bar{x}_1x_3 \oplus x_1\bar{x}_2x_3 \oplus \bar{x}_4$. Then, $V(x_1\bar{x}_2) = \{x_1, x_2\}$, $V(\bar{x}_1x_3) = \{x_1, x_3\}$, $V(x_1\bar{x}_2x_3) = \{x_1, x_2, x_3\}$, and $V(\bar{x}_4) = \{x_4\}$. Thus, $x_1\bar{x}_2x_3$ and $\bar{x}_4$ have maximal variable sets.

Definition 6: Let x be a variable and $\alpha \in \{0,1,2\}$. x^α is a *literal* of x such that

$$x^\alpha = \begin{cases} \bar{x} & \text{if } \alpha = 0, \\ x & \text{if } \alpha = 1, \\ 1 & \text{if } \alpha = 2. \end{cases}$$

From the definition of PPRM, we have the following lemma:

Lemma 5: An arbitrary PPRM can be represented by an expression

$$F = \bigoplus\!\!\!\!\!\sum h(\boldsymbol{\beta}) x_1^{\beta_1} x_2^{\beta_2} \cdots x_n^{\beta_n},$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)$, $\beta_i \in \{1,2\}$ $(i = 1, 2, \ldots, n)$, and $h(\boldsymbol{\beta}) \in \{0,1\}$.

Example 4: Consider a PPRM $F = x_1 \oplus x_1x_2 \oplus x_3$. It can be represented as $F = x_1^1x_2^1x_3^2 \oplus x_1^1x_2^2x_3^2 \oplus x_1^2x_2^2x_3^1$.

From the definition of GRM, we have the following lemma:

Lemma 6: An arbitrary GRM can be represented by an expression

$$G = \bigoplus\!\!\!\!\!\sum g(\boldsymbol{\alpha}) x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n},$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\alpha_i \in \{0,1,2\}$ $(i = 1, 2, \ldots, n)$, and $g(\boldsymbol{\alpha}) \in \{0,1\}$.

Example 5: Consider a GRM $G = \bar{x}_1 \oplus x_1\bar{x}_2 \oplus \bar{x}_3$. It can be represented as $G = x_1^0x_2^2x_3^2 \oplus x_1^1x_2^0x_3^2 \oplus x_1^2x_2^2x_3^0$.

Lemma 7: Let the PPRM for f be

$$F = \bigoplus\!\!\!\!\!\sum h(\boldsymbol{\beta}) x_1^{\beta_1} x_2^{\beta_2} \cdots x_n^{\beta_n},$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)$, $\beta_i \in \{1,2\}$ $(i = 1, 2, \ldots, n)$, and $h(\boldsymbol{\beta}) \in \{0,1\}$. Also let a GRM for f be

$$G = \bigoplus\!\!\!\!\!\sum g(\boldsymbol{\alpha}) x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n},$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n), \alpha_i \in \{0,1,2\}$ $(i = 1, 2, \ldots, n)$, and $g(\boldsymbol{\alpha}) \in \{0,1\}$. If F has a product $p = x_1^{\beta_1}x_2^{\beta_2}\cdots x_n^{\beta_n}$ with a maximal variable set, then G has a product $q = x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_n^{\alpha_n}$, where

$$\alpha_i = \begin{cases} 0 \text{ or } 1 & \text{if } \beta_i = 1, \\ 2 & \text{if } \beta_i = 2, \end{cases}$$

and q has the maximal variable set in G.

Proof: Without loss of generality, we can assume that the PPRM for f contains a product $p_1 = x_1x_2\cdots x_t$ with a maximal variable set.

1) Suppose that a GRM for f contains a product

$$q_1 = x_1^{c_1}x_2^{c_2}\cdots x_s^{c_s},$$

where $s > t$, $c_j \in \{0,1\}$ $(j = 1, 2, \ldots, s)$, and q_1 has a maximal variable set. Then, q_1 can be written as

$$q_1 = (x_1 \oplus \bar{c}_1)(x_2 \oplus \bar{c}_2)\cdots(x_s \oplus \bar{c}_s). \qquad (5)$$

By expanding Eq. (5), it can be shown that the PPRM for f must contain the product $x_1x_2\cdots x_s$. Also, because q_1 has a maximal variable set, the product will not disappear. However, this contradicts the assumption that p_1 has a maximal variable set of size t. Thus, the GRM does not contain the product q_1.

2) Suppose that a GRM for f contains a product

$$q_2 = x_1^{c_1}x_2^{c_2}\cdots x_s^{c_s},$$

where $s < t$, $c_j \in \{0,1\}$ $(j = 1, 2, \ldots, s)$, and q_2 has a maximal variable set. Because q_2 is maximal and can be written as

$$q_2 = (x_1 \oplus \bar{c}_1)(x_2 \oplus \bar{c}_2) \cdots (x_s \oplus \bar{c}_s),$$

the PPRM must contain the product $x_1 x_2 \cdots x_s$. Also, the product will not disappear since q_2 has a maximal variable set. However, this contradicts the assumption that the product p_1 has a maximal variable set.

From 1) and 2), the GRM contains a product with a form

$$q = x_1^{c_1} x_2^{c_2} \cdots x_t^{c_t},$$

where $c_j \in \{0,1\}$ $(j = 1, 2, \ldots, t)$, and q has a maximal variable set. □

Corollary 1: If all the products in the PPRM of a function f have a maximal variable set, then a minimum GRM for f contains the same number of products as the PPRM.

Corollary 2: The PPRM in Corollary 1 is also a minimum GRM for f.

Example 6: Let the PPRM for a function f be $F = x_1 \oplus x_2 x_3$. Because both of the products have a maximal variable set, a minimum GRM has two products. Thus, F is also a minimum GRM for f.

Corollary 3: Let p_1 be a product in the PPRM for f which has a maximal variable set. Then,

a) any GRM for f contains a product p_2 such that $V(p_2) = V(p_1)$, and

b) any GRM for f does not contain a product p_3 such that $V(p_3) \supset V(p_1)$ and $V(p_3) \neq V(p_1)$.

Example 7: Let the PPRM for f be $F = x_1 \oplus x_2 x_3$. GRMs for f are $G_1 = x_1 \oplus x_2 x_3$, $G_2 = x_1 \oplus x_3 \oplus \bar{x}_2 x_3$, $G_3 = x_1 \oplus x_2 \oplus x_2 \bar{x}_3$, $G_4 = \bar{x}_1 \oplus x_2 \oplus x_3 \oplus \bar{x}_2 \bar{x}_3$, etc. Note that, in the PPRM for f, the products x_1 and $x_2 x_3$ have maximal variable sets. Thus, all the GRMs for f contain the products with the form $x_1^{b_1}$ and $x_2^{b_2} x_3^{b_3}$. GRMs for f do not contain the products with the form $x_1^{b_1} x_2^{b_2}$, $x_1^{b_1} x_3^{b_3}$, nor $x_1^{b_1} x_2^{b_2} x_3^{b_3}$, where b's are binary constants.

4. Basic Idea for Minimization

Definition 7: x^a is called a literal of x, where $a \in \{0,1\}$.

$$x^a = \begin{cases} \bar{x} & \text{if } a = 0, \\ x & \text{if } a = 1. \end{cases}$$

Lemma 8: Let $a \in \{0,1\}$, then $x^a = x \oplus a \oplus 1 = \bar{x} \oplus a$, and

$$x^a = \begin{cases} \bar{a} & \text{if } x = 0, \\ a & \text{if } x = 1. \end{cases}$$

Definition 8: Let $f(x_1, x_2, \ldots, x_n)$ be a function of n variables. The *Boolean difference* of f with respect to x_i is

$$\frac{df}{dx_i} = f(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \oplus f(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n).$$

Lemma 9: [7] For an arbitrary function $f(x_1, x_2, \ldots, x_n)$:

$$\frac{df}{dx_i} = \frac{df}{d\bar{x}_i}, \quad \frac{d^2 f}{dx_i dx_j} = \frac{d^2 f}{dx_j dx_i},$$

and if g does not depend on x_i, then

$$\frac{dg}{dx_i} = 0, \quad \frac{d(x_i g)}{dx_i} = g.$$

In order to obtain the minimum GRM of a given function, we have to solve a system of logic equations. Such a system is given by

$$f_i(y_1, y_2, \ldots, y_t) = g_i(y_1, y_2, \ldots, y_t),$$

where, $i = 1, 2, \ldots, k$.

However, these equations are converted into one equation as follows:

Lemma 10: $f_i = g_i$ holds for all i $(i = 0, \ldots, k)$ iff $GR(f) = 1$, where

$$GR(f) = \bigwedge_{i=0}^{k} (f_i \oplus g_i \oplus 1).$$

4.1 A Naive Method for Optimization

An arbitrary two-variable function can be represented by a GRM:

$$f(x_1, x_2) = a_{00} \oplus a_{01} x_2^{b_1} \oplus a_{10} x_1^{b_2} \oplus a_{11} x_1^{b_3} x_2^{b_4}, \quad (6)$$

where the a's and b's are binary constants. By setting (x_1, x_2) to $(0,0), (0,1), (1,0)$ and $(1,1)$ in Eq. (6), we have

$$\begin{aligned} f(0,0) &= a_{00} \oplus a_{01}\bar{b}_1 \oplus a_{10}\bar{b}_2 \oplus a_{11}\bar{b}_3\bar{b}_4, & (7) \\ f(0,1) &= a_{00} \oplus a_{01} b_1 \oplus a_{10}\bar{b}_2 \oplus a_{11}\bar{b}_3 b_4, & (8) \\ f(1,0) &= a_{00} \oplus a_{01}\bar{b}_1 \oplus a_{10} b_2 \oplus a_{11} b_3\bar{b}_4, & (9) \\ f(1,1) &= a_{00} \oplus a_{01} b_1 \oplus a_{10} b_2 \oplus a_{11} b_3 b_4. & (10) \end{aligned}$$

From Eqs. (7)–(10) and by Lemma 10, we have

$$\begin{aligned} GR(f) &= \psi(0,0) \cdot \psi(0,1) \cdot \psi(1,0) \cdot \psi(1,1) \\ &= 1, \end{aligned} \quad (11)$$

where

$$\begin{aligned} \psi(0,0) &= f(0,0) \oplus a_{00} \oplus a_{01}\bar{b}_1 \oplus a_{10}\bar{b}_2 \\ &\quad \oplus a_{11}\bar{b}_3\bar{b}_4 \oplus 1, \\ \psi(0,1) &= f(0,1) \oplus a_{00} \oplus a_{01} b_1 \oplus a_{10}\bar{b}_2 \end{aligned}$$

$$\begin{aligned}
&\oplus a_{11}\bar{b}_3 b_4 \oplus 1,\\
\psi(1,0) &= f(1,0) \oplus a_{00} \oplus a_{01}\bar{b}_1 \oplus a_{10} b_2\\
&\oplus a_{11} b_3 \bar{b}_4 \oplus 1,\\
\psi(1,1) &= f(1,1) \oplus a_{00} \oplus a_{01} b_1 \oplus a_{10} b_2\\
&\oplus a_{11} b_3 b_4 \oplus 1.
\end{aligned}$$

Thus, the assignment of a's and b's that satisfy $GR(f)$ in Eq. (11) also satisfies Eq. (6). The minimum GRM is one that has the fewest products, i.e., a GRM with the sum of a's minimum. The number of b's in Eq. (6) is four. Thus, a minimum GRM can be found out of $2^4 (= 16)$ different GRMs. However, the expression in Eq. (11) is very complex, and it is not easy to obtain the minimum solution.

4.2 An Efficient Method of Optimization

This method is more complex than the previous method, but it is more efficient. In Eq. (6), by obtaining the Boolean difference, and by setting $(x_1, x_2) = (0, 0)$, we have

$$\frac{d(df)}{dx_1 dx_2} = a_{11}, \quad \frac{df}{dx_1} = a_{10} \oplus a_{11}\bar{b}_4, \tag{12}$$

$$\frac{df}{dx_2} = a_{01} \oplus a_{11}\bar{b}_3. \tag{13}$$

On the other hand, consider the PPRM for the function:

$$f(x_1, x_2) = c_{00} \oplus c_{01} x_2 \oplus c_{10} x_1 \oplus c_{11} x_1 x_2. \tag{14}$$

By obtaining the Boolean difference of Eq. (14), and by setting $(x_1, x_2) = (0, 0)$, we have

$$\frac{d(df)}{dx_1 dx_2} = c_{11}, \quad \frac{df}{dx_1} = c_{10}, \quad \frac{df}{dx_2} = c_{01}. \tag{15}$$

From Eqs. (12), (13) and (15), we have

$$c_{11} = a_{11}, \quad c_{10} = a_{10} \oplus a_{11}\bar{b}_4, \tag{16}$$

$$c_{01} = a_{01} \oplus a_{11}\bar{b}_3. \tag{17}$$

In Eqs. (6) and (14), by setting $(x_1, x_2) = (0, 0)$, we have

$$c_{00} = a_{00} \oplus a_{01}\bar{b}_1 \oplus a_{10}\bar{b}_2 \oplus a_{11}\bar{b}_3\bar{b}_4 \tag{18}$$

From Eqs. (16)–(18) and Lemma 10, we have

$$GR(f) = \phi(0,0) \cdot \phi(0,1) \cdot \phi(1,0) \cdot \phi(1,1) = 1, \tag{19}$$

where

$$\begin{aligned}
\phi(0,0) &= c_{00} \oplus a_{00} \oplus a_{01}\bar{b}_1 \oplus a_{10}\bar{b}_2 \oplus a_{11}\bar{b}_3\bar{b}_4 \oplus 1,\\
\phi(0,1) &= c_{01} \oplus a_{01} \oplus a_{11}\bar{b}_3 \oplus 1,\\
\phi(1,0) &= c_{10} \oplus a_{10} \oplus a_{11}\bar{b}_4 \oplus 1,\\
\phi(1,1) &= c_{11} \oplus a_{11} \oplus 1.
\end{aligned}$$

Note that ϕ's of Eq. (19) is simpler than ψ's of Eq. (11): ϕ's contain fewer EXOR and AND operators than ψ's. In Sect. 5, we will formulate a method to solve $GR(f)$ by generating its binary decision diagrams (BDDs) [1]. In that case, at first it would be necessary to compute the BDDs of all the ψ's of Eq. (11) or all the ϕ's of Eq. (19). Because ϕ's are simpler than ψ's, computation of BDDs for $GR(f)$ using Eq. (19) is more efficient than using Eq. (11).

4.3 Three-Variable Case

An arbitrary 3-variable function f can be represented by a GRM:

$$\begin{aligned}
f(x_1, x_2, x_3) &= a_{000} \oplus a_{001} x_3^{b_1} \oplus a_{010} x_2^{b_2}\\
&\oplus a_{011} x_2^{b_3} x_3^{b_4} \oplus a_{100} x_1^{b_5}\\
&\oplus a_{101} x_1^{b_6} x_3^{b_7} \oplus a_{110} x_1^{b_8} x_2^{b_9}\\
&\oplus a_{111} x_1^{b_{10}} x_2^{b_{11}} x_3^{b_{12}},
\end{aligned} \tag{20}$$

where a's and b's are binary constants.

On the other hand, the PPRM for the function f is:

$$\begin{aligned}
f(x_1, x_2, x_3) &= c_{000} \oplus c_{001} x_3 \oplus c_{010} x_2 \oplus c_{100} x_1\\
&\oplus c_{011} x_2 x_3 \oplus c_{101} x_1 x_3\\
&\oplus c_{110} x_1 x_2 \oplus c_{111} x_1 x_2 x_3,
\end{aligned} \tag{21}$$

where c's are binary constants.

Similarly to the two-variable case, we have

$$\begin{aligned}
GR(f) &= \phi(1,1,1) \cdot \phi(1,1,0) \cdot \phi(1,0,1)\\
&\cdot \phi(0,1,1) \cdot \phi(1,0,0) \cdot \phi(0,1,0)\\
&\cdot \phi(0,0,1) \cdot \phi(0,0,0)\\
&= 1,
\end{aligned} \tag{22}$$

where

$$\begin{aligned}
\phi(1,1,1) &= c_{111} \oplus a_{111} \oplus 1,\\
\phi(1,1,0) &= c_{110} \oplus a_{110} \oplus a_{111}\bar{b}_{12} \oplus 1,\\
\phi(1,0,1) &= c_{101} \oplus a_{101} \oplus a_{111}\bar{b}_{11} \oplus 1,\\
\phi(0,1,1) &= c_{011} \oplus a_{011} \oplus a_{111}\bar{b}_{10} \oplus 1,\\
\phi(1,0,0) &= c_{100} \oplus a_{100} \oplus a_{101}\bar{b}_7 \oplus a_{110}\bar{b}_9\\
&\oplus a_{111}\bar{b}_{11}\bar{b}_{12} \oplus 1,\\
\phi(0,1,0) &= c_{010} \oplus a_{010} \oplus a_{110}\bar{b}_4 \oplus a_{011}\bar{b}_9\\
&\oplus a_{111}\bar{b}_{10}\bar{b}_{12} \oplus 1,\\
\phi(0,0,1) &= c_{001} \oplus a_{001} \oplus a_{011}\bar{b}_5 \oplus a_{101}\bar{b}_7\\
&\oplus a_{111}\bar{b}_{10}\bar{b}_{11} \oplus 1,\\
\phi(0,0,0) &= c_{000} \oplus a_{000} \oplus a_{001}\bar{b}_1 \oplus a_{010}\bar{b}_2\\
&\oplus a_{011}\bar{b}_3\bar{b}_4 \oplus a_{100}\bar{b}_5 \oplus a_{101}\bar{b}_6\bar{b}_7\\
&\oplus a_{110}\bar{b}_8\bar{b}_9 \oplus a_{111}\bar{b}_{10}\bar{b}_{11}\bar{b}_{12} \oplus 1.
\end{aligned}$$

4.4 n-Variable Case

Similar to the two and three-variable cases, we can make 2^n different equations, and can get the expression for $GR(f)$ for an n-variable function. An assignment of a's and b's that satisfies $GR(f)$ corresponds to a GRM for the given function f. For n-variable case, a GRM similar to Eq. (6) contain 2^n a's and $n2^{n-1}$ b's, thus, the total number of variables in $GR(f)$ is $2^n + n2^{n-1} = (n+2)2^{n-1}$. The minimum GRM corresponds to the assignment of a's and b's that makes the sum of a's minimum.

5. An Algorithm Using BDDs

5.1 Minimization Using BDDs

Consider the binary decision diagram (BDD) for $GR(f)$, where the edges for uncomplemented a's have distance one, and other edges (i.e., edges for $\bar{a}$'s, b's and $\bar{b}$'s) have distance zero. Then, each path in the BDD from the root node to the terminal 1 corresponds to an assignment of a's and b's satisfying Eq. (22). And the shortest path from the root node to the terminal 1 corresponds to a minimum GRM. Theoretically, it is possible to obtain a minimum GRM by using the BDD for $GR(f)$. However, a naive method using the BDD often requires excessive memory and computation time. To reduce the size of the BDDs and the computation time, we use various techniques, which will be shown in Sects. 5.2–5.5.

5.2 Threshold Function

$GR(f)$ represents all possible GRMs for a given function. However, we need only one minimum GRM. Suppose that we have a near minimal GRM for f, and let t_0 be the number of products in it. Then, we only need to find a GRM for f that has less than t_0 products. If such a GRM does not exist, then the near minimal GRM is also an exact minimum GRM for f.

Definition 9: Let $a_i \in \{0,1\}$ for $i = 0, 1, \ldots, 2^n - 1$, and t be a positive integer. A function

$$TH(a_0, a_1, \ldots, a_{2^n-1} : t) = \begin{cases} 1 & \text{if } \sum_{i=0}^{2^n-1} a_i < t, \\ 0 & \text{otherwise.} \end{cases}$$

$TH(a_0, a_1, \ldots, a_{2^n-1} : t)$ is used to represent the set of GRMs with less than t products.

5.3 Computation of $GR(f)$

A naive method for computing $GR(f)$ requires excessive memory and computation time. In the case of three-variable functions, we use the following method:

$$\begin{aligned}
\phi(1,1,1) &\leftarrow \phi(1,1,1) \cdot TH(a_0, a_1, \ldots, a_7 : t), \\
\phi(0,1,1) &\leftarrow \phi(0,1,1) \cdot \phi(1,1,1), \\
\phi(1,0,1) &\leftarrow \phi(1,0,1) \cdot \phi(1,1,1), \\
\phi(1,1,0) &\leftarrow \phi(1,1,0) \cdot \phi(1,1,1), \\
\phi(0,0,1) &\leftarrow \phi(0,0,1) \cdot \phi(0,1,1) \cdot \phi(1,0,1), \\
\phi(0,1,0) &\leftarrow \phi(0,1,0) \cdot \phi(0,1,1) \cdot \phi(1,1,0), \\
\phi(1,0,0) &\leftarrow \phi(1,0,0) \cdot \phi(1,0,1) \cdot \phi(1,1,0), \\
\phi(0,0,0) &\leftarrow \phi(0,0,0) \cdot \phi(0,0,1) \\
&\qquad \cdot \phi(0,1,0) \cdot \phi(1,0,0), \\
GR(f) &\leftarrow \phi(0,0,0).
\end{aligned}$$

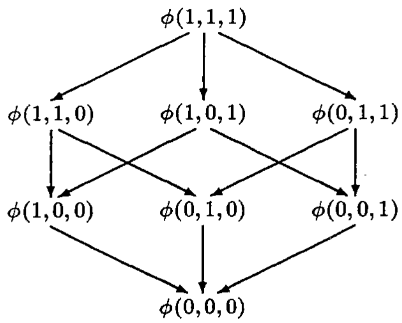


Fig. 1 Computation of $GR(f)$.

This method drastically reduces the computation time as well as the memory requirement for generating the BDD for $GR(f)$. Figure 1 illustrates this multiplication method. Extension to the n-variable case is straightforward.

5.4 Variable Ordering in the BDDs

The ordering of the variables in the BDDs influences the memory requirement as well as computation time. In the case of $GR(f)$ for three-variable functions Eq. (22), we use the following ordering: $a_{111} < b_{10} < b_{11} < b_{12} < a_{110} < b_8 < b_9 < a_{101} < b_6 < b_7 < a_{011} < b_3 < b_4 < a_{100} < b_5 < a_{010} < b_2 < a_{001} < b_1 < a_{000}$, where a_{111} is the nearest to the root node. Extension to the n-variable case is straightforward.

5.5 Maximal Variable Sets

Corollary 3 shows the products that will never appear in the GRMs for a given function. In generating BDDs for $GR(f)$, we do not use the variables (a's and b's) corresponding to such products.

5.6 Minimization Algorithms

Algorithm 1 (Exact Minimum GRM):

1. Obtain a near minimal GRM by Algorithm 2, and let t_0 be the number of products.
2. Construct the BDD for $TH(a_0, a_1, \ldots, a_{2^n-1} : t_0)$.
3. Construct the BDD for $TH(a_0, a_1, \ldots, a_{2^n-1} : t_0) \cdot GR(f)$.

4. Find a shortest path to the terminal one for the BDD computed in step 3.
5. Obtain the corresponding GRM.

Algorithm 2 (Near minimal GRM):

1. Obtain a minimal PSDRM [20] for the function $f(x_1, x_2, \ldots, x_n)$ by the similar algorithm to [20], and let t_1 be the number of products.
2. Decompose f into 2^{n-k} sub-functions as

$$f = \bigoplus x_1^{\beta_1} x_2^{\beta_2} \cdots x_{n-k}^{\beta_{n-k}} g(x_{n-k+1}, x_{n-k+2}, \ldots, x_n : \beta_1, \beta_2, \ldots, \beta_{n-k}), \quad (23)$$

where β's are 1 or 2, and $g(x_{n-k+1}, x_{n-k+2}, \cdots, x_n : \beta_1, \beta_2, \ldots, \beta_{n-k})$ represents a k-variable sub-function. For each sub-function, obtain the GRM by using the table of exact minimum GRMs of k-variables ($k = 3, 4$ or 5). Let t_2 be the number of products in Eq. (23).

3. Obtain the GRM with $\min\{t_1, t_2\}$ products.

6. Experimental Results

We developed a minimization program, which extensively uses BDDs. The computation time of the program depends on the size of the BDDs, and the size of the BDDs depends on the number of inputs and the number of the products in the near minimal GRMs obtained from Algorithm 2. The program can minimize GRMs for all the functions with up to five variables. We minimized many functions of 6 variables with up to 10 products, and some functions of 7 and 8 variables with up to 9 products. We also minimized parity functions with up to 9 variables. The minimization program proved the minimality of the solutions, produced by a heuristic simplification program [5], for some 6 and 7 variable functions with up to 16 and 11 products, respectively. In the above experiments, we used a Sun Ultra1 Model 170 workstation with 256 megabytes main memory. We generated all the 1,228,158 representative functions for NP-equivalence classes of five or fewer variables, and minimized each function. On the average, a five-variable function could be minimized in 25 seconds by an Hewlett Packard Model 715/50 workstation with 64 megabytes main memory. We also developed minimization programs for FPRMs and SOPs. Tables 1 and 2 show the number of four- and five-variable functions requiring t products, respectively. For five-variable functions, on the average, GRMs require 6.230 products while SOPs require 7.463 products. Thus, we verified that Conjecture 1 is correct for $n = 4$ and 5.

7. Conclusion and Comments

In this paper, we presented three classes of AND-EXOR expressions: PPRM, FPRM and GRM. Among these classes, GRMs have easily testable realizations, and a GRM never require more products than the corresponding PPRM or FPRM. Thus, the optimization problem for GRMs is important, especially in FPGAs, where the EXORs have the same costs as ORs. We presented some properties of GRMs, and showed an exact minimization algorithm. The minimization program can minimize GRMs for all the functions up to five variables, and some functions with more inputs. We have completed the table of minimum GRMs with up to five-variable functions. Thus, the minimum GRMs with up to five variables can be found in a table look-up method. The table of minimum GRMs is also useful in a heuristic optimization program for GRMs with six or more inputs. An exact minimization algorithm is useful to test the minimality of the solutions produced by a heuristic minimization algorithm [5]. In addition, we obtained the statistical data for the minimum expressions for other classes of expressions up to five variables. On the average, GRMs require 6.230 products while SOPs require 7.463 products for five-variable functions. We conjecture that GRMs, on the average, require fewer products than SOPs for the functions with more inputs. This result shows that GRM based design is useful not only for arithmetic and telecommunication circuits but also for general circuits. Recently, another GRM minimization algorithm based on BDDs were published [11]. We think, the performance of [11] is comparable to ours.

Acknowledgments

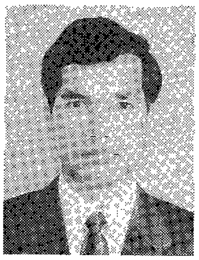
This work was supported in part by a Grant in Aid for Scientific Research of the Ministry of Education, Science, Culture and Sports of Japan. Prof. J.T. Butler carefully reviewed the manuscript. Prof. N. Koda provided the table of representative functions of five variables.

References

[1] R.E. Bryant, "Graph-based algorithms for Boolean function manipulation," IEEE Trans. Comput., vol.C-35, no.8, pp.677–691, Aug. 1986.
[2] M. Cohn, "Inconsistent canonical forms of switching functions," IRE Trans. Electronic Comput., vol.EC-11, pp.284–285, April 1962.
[3] L. Csanky, M.A. Perkowski, and I. Schäfer, "Canonical restricted mixed-polarity exclusive-OR sums of products and the efficient algorithm for their minimization," IEE Proceedings-Computers and Digital Techniques, vol.140, no.1, pp.69–77, Jan. 1993.
[4] M. Davio, J.-P. Deschamps, and A. Thayse, "Discrete and Switching Functions," McGraw-Hill International, New York, 1978.
[5] D. Debnath and T. Sasao, "GRMIN: A heuristic simplification algorithm for generalized Reed-Muller expressions," Proc. IEEE Asia and South Pacific Design Automation Conference, pp.341–347, Aug. 1995.
[6] D. Debnath and T. Sasao, "GRMIN2: A heuristic simplifi-

cation algorithm for generalized Reed-Muller expressions," IEE Proceedings-Computers and Digital Techniques, (accepted for publication).
[7] H. Fujiwara, "Logic Testing and Design for Testability," The MIT Press, Cambridge, 1985.
[8] D. Green, "Modern Logic Design," Addison-Wesley Publishing Company, Wokingham, England, 1986.
[9] D. Green, "Families of Reed-Muller canonical forms," Int. Journal of Electronics, vol.70-2, pp.259–280, 1991.
[10] A. Mukhopadhyay and G. Schmitz, "Minimization of exclusive OR and logical equivalence of switching circuits," IEEE Trans. Comput., vol.C-19, pp.132–140, Feb. 1970.
[11] H. Ochi, "A zero-suppressed BDD package with pruning and its application to GRM minimization," Proc. IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design, pp.28–37, Aug. 1995.
[12] G. Papakonstantinou, "Minimization of modulo-2 sum of products," IEEE Trans. Comput., vol.C-28, pp.163–167, Feb. 1979.
[13] M.A. Perkowski, "The generalized orthonormal expansions of functions with multiple-valued input and some of its applications," Proc. Int. Symposium on Multiple-Valued Logic, pp.442–450, May, 1992.
[14] M.A. Perkowski, L. Csanky, A. Sarabi, and I. Schäfer, "Fast minimization of mixed-polarity AND-XOR canonical networks," Proc. Int. Conference on Computer Design, pp.33–36, Oct. 1992.
[15] S.M. Reddy, "Easily testable realizations for logic functions," IEEE Trans. Comput., vol.C-21, no.11, pp.1183–1188, Nov. 1972.
[16] K.K. Saluja and S.M. Reddy, "Fault detecting test sets for Reed-Muller canonic networks," IEEE Trans. Comput., vol.C-24, no.1, pp.995–998, Oct. 1975.
[17] A. Sarabi and M.A. Perkowski, "Fast exact and quasi-minimal minimization of highly testable fixed polarity AND/XOR canonical networks," Proc. Design Automation Conference, pp.30–35, June 1992.
[18] A. Sarabi and M.A. Perkowski, "Design for testability properties of AND/XOR networks," Proc. IFIP 10.5 Workshop on Applications of the Reed-Muller expansion in Circuit Design, Sept. 1993.
[19] T. Sasao and P. Besslich, "On the complexity of MOD-2 sum PLA's," IEEE Trans. Comput., vol.C-39, no.2, pp.262–266, Feb. 1990.
[20] T. Sasao, "AND-EXOR expressions and their optimization," in Logic Synthesis and Optimization, T. Sasao, ed., Kluwer Academic Publishers, 1993.
[21] T. Sasao, "EXMIN2: A simplification algorithm for exclusive-OR sum-of-products expressions for multiple-valued input two-valued output functions," IEEE Trans. Comput. Aided Des. Integrated Circuits & Syst., vol.CAD-12, no.5, pp.621–632, May 1993.
[22] T. Sasao, "An exact minimization of AND-EXOR expressions using BDDs," Proc. IFIP 10.5 Workshop on Applications of the Reed-Muller expansion in Circuit Design, Sept. 1993.
[23] T. Sasao, "Easily testable realizations for generalized Reed-Muller expressions," Proc. IEEE The Third Asian Test Symposium, pp.157–162, Nov. 1994.
[24] T. Sasao and D. Debnath, "An exact minimization algorithm for generalized Reed-Muller expressions," Proc. IEEE Asia-Pacific Conference on Circuits & Systems, pp.460–465, Dec. 1994.
[25] T. Sasao, "Representations of logic functions using EXOR operators," in Representations of Discrete Functions, eds., T. Sasao and M. Fujita, Kluwer Academic Publishers, Boston, 1996.

Tsutomu Sasao received the B.E., M.E., and Ph.D. degrees in Electronic Engineering from Osaka University, Osaka, Japan, in 1972, 1974, and 1977, respectively. He was with Osaka University, IBM T.J. Watson Research Center and Naval Postgraduate School in Monterey, California. Now, he is a Professor of Kyushu Institute of Technology, Iizuka, Japan. He has published six books on switching theory and logical design, including "Logic Synthesis and Optimization," and "Representations of Discrete Functions," Kluwer 1993, and 1996, respectively. He has served Program Chairman for the IEEE International Symposium on Multiple-Valued Logic many times. He received the NIWA Memorial Award in 1979, and Distinctive Contribution Awards from IEEE Computer Society MVL-TC in 1987 and 1996. He is a Fellow of IEEE.



Debatosh Debnath received the B.S. degree in Electrical and Electronic Engineering in 1991 and the M.S. degree in Computer Science in 1993, both from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh. He is currently working towards Ph.D. at the Kyushu Institute of Technology, Iizuka, Japan. He has been a recipient of the Japanese Government Scholarship since October, 1993.