Regular Paper

Double Fixed-Polarity Reed-Muller Expressions: A New Class of AND-EXOR Expressions for Compact and Testable Realization

Takashi Hirayama,† Kazuyuki Nagasawa,† Yasuaki Nishitani†† and Kensuke Shimizu†††

As classes of AND-EXOR expressions, PPRMs, FPRMs, and ESOPs are well-known. In this paper, a new class of AND-EXOR expressions, Double Fixed-Polarity Reed-Muller Expressions (DFPRMs), is proposed. DFPRMs are generalized expressions of FPRMs, and can be the smallest PLA among all the classes of AND-EXOR expressions. We discuss their properties: the relation to other classes, a compact realization with (AND/OR)-EXOR PLAs, and the easy testability of the PLA. We show that all the stuck-at faults in DFPRM PLAs are detected by $(2n+4)$ tests, which are independent of the functions realized by the PLAs. And we demonstrate the compactness of DFPRMs by giving a table of the number of products of the minimum DFPRMs for all 4-variable functions. The table is obtained by a minimization algorithm presented in this paper.

1. Introduction

Logic circuits including exclusive-or (EXOR) gates have some advantages over traditional circuits with only AND and OR gates. EXOR-based realization can reduce the circuit area [4),6),12)] and improve the testability [2),8),10),13),20),22)].

AND-EXOR expressions have been studied as the fundamentals of the EXOR-based realization. An AND-EXOR expression with n variables can be realized in an AND-EXOR programmable logic array (PLA) with $2n$ literal lines as shown in **Fig. 1** [16),17)], which is the AND-EXOR counterpart of the AND-OR PLA. The line c in the PLA is the constant input line. The width t of the PLA corresponds to the number of product terms of the expression. Hence the size of the PLA mainly depends on $2n \times t$.

There are several classes of AND-EXOR expressions [18)] such as PPRMs (positive-polarity Reed-Muller expressions), FPRMs (fixed-polarity Reed-Muller expressions), and ESOPs (exclusive-or sum-of-porducts). Each class has the different properties of size and testability. An expression such that arbitrary product terms are combined by EXORs is called an ESOP, which is realized in the PLA of Fig. 1. In terms of the testability, all the single stuck-at faults of the ESOP PLAs with n variables are detected by $(n+6)$ tests although some additional gates to make the PLA easily testable are required [11)].

An FPRM is an AND-EXOR expression in which either positive or negative literals appear for every variable. The PLA realization is shown in **Fig. 2** [18),21)]. The lines v_i $(1 \leq i \leq n)$ represent the polarity of literals. Since the PLA has only n literal lines, the size depends on $n \times t$. The FPRM PLA is known as an easily testable realization, in which all the single stuck-at faults are detected by $(n+4)$ tests with only one additional gate [8)]. From the compactness and the testability, many minimization or optimization algorithms for FPRMs have been proposed [3),7),15),19),23)].

A PPRM is an AND-EXOR expression without any negative literals. As shown in **Fig. 3**, the structure is the simplest among the AND-EXOR PLAs; it does not have the polarity input lines and the number of the literal lines is n. As well as FPRM PLAs, the size depends on $n \times t$ and the single stuck-at faults are detected by $(n+4)$ tests with one additional gate [14)]. However, the number of products of PPRMs, namely t, tends to be large by the strict restriction of literals.

In this paper, a new class of AND-EXOR expressions, double fixed-polarity Reed-Muller expressions (DFPRMs), is proposed. The DFPRM is an expression such that the EXOR

† Ashikaga Institute of Technology
†† Faculty of Engineering, Iwate University
††† Faculty of Engineering, Gunma University

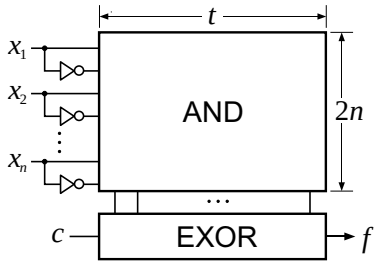


Fig. 1 PLA for ESOPs.

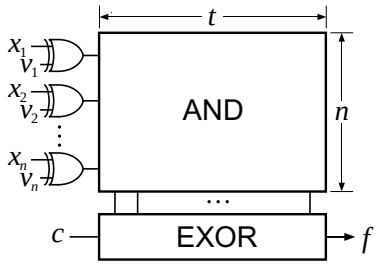


Fig. 2 PLA for FPRMs.

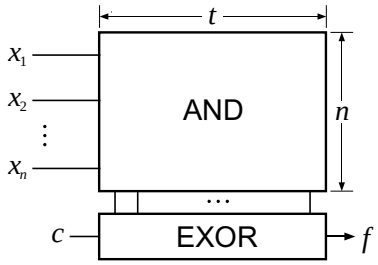


Fig. 3 PLA for PPRMs.

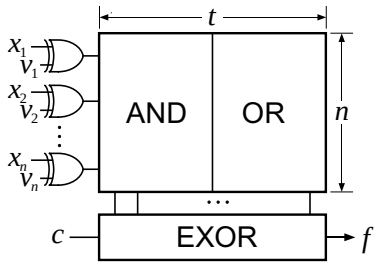


Fig. 4 PLA for DFPRMs.

combination of two FPRMs whose polarities are opposite each other. A DFPRM can be realized in an (AND/OR)-EXOR PLA of **Fig. 4**, which has only n literal lines and then the size depends on $n \times t$. The number of literal lines is the same as the FPRM PLA and is half of the ESOP PLA. The average of t for four-variable functions is 4.13, which is smaller than 5.50 of FPRMs. The results can be obtained from the minimization algorithm which is presented in this paper. We also discuss the easy testability of DFPRM PLAs for the single stuck-at faults; all the faults in DFPRM PLAs are detected by $(2n+4)$ tests with one additional gate and the tests are independent of the functions realized by the PLAs. Although the DFPRM PLA requires more tests than the other kind of PLAs, the number of tests is within two times of that of the others. Consequently, DFPRM PLAs hold easy testability and are smaller than FPRM PLAs.

2. Preliminaries

An arbitrary logic function $f(x_1, x_2, \ldots, x_n)$ can be expanded with EXORs as follows [5),18),21)]:

$$f = \bar{x}_n f_{\{0\}} \oplus x_n f_{\{1\}} \quad (1)$$

$$f = f_{\{0\}} \oplus x_n f_{\{0,1\}} \quad (2)$$

$$f = f_{\{1\}} \oplus \bar{x}_n f_{\{0,1\}}, \quad (3)$$

where $f_{\{0\}} = f(x_1, x_2, \ldots, x_{n-1}, 0)$, $f_{\{1\}} = f(x_1, x_2, \ldots, x_{n-1}, 1)$, and $f_{\{0,1\}} = f_{\{0\}} \oplus f_{\{1\}}$. Equation (1) is called *the Shannon expansion*, and Eqs. (2) and (3) are called *the positive Davio* and *the negative Davio expansions*, respectively.

By applying the above expansions recursively, AND-EXOR expressions are obtained, which are classified according to the choice of expansions [18)]. By expanding the function f with the positive Davio expansion, an expression without negative literals is obtained, which is called *the Positive-Polarity Reed-Muller expression* (PPRM). *A Fixed-Polarity Reed-Muller expression* (FPRM) is an expression obtained by using either the positive Davio or the negative Davio expansion for each variable. FPRMs are generalized PPRMs. An expression such that arbitrary product terms are combined by EXORs is called *an EXOR Sum-Of-Products expression* (ESOP). ESOPs are the most general AND-EXOR expressions.

Definition 1 The number of product terms of an ESOP F is denoted by $\tau(F)$. □

We represent the choice of expansion to obtain FPRMs as the n-bit vector $V_n = [v_1, v_2, \ldots, v_n]$ $(v_i \in \{0, 1\})$, which is called *the polarity vector*. Each v_i $(1 \leq i \leq n)$ denotes the polarity of the variable x_i; $v_i = 0$ means the positive Davio expansion is used for the variable x_i, and $v_i = 1$ means the negative Davio expansion is used instead. The FPRM of f is uniquely obtained if the polarity vector V_n is given.

Definition 2 The polarity vector of a given FPRM F is denoted by $\nu(F)$[☆]. $\bar{\nu}(F)$ is defined as the bitwise complement of $\nu(F)$. □

Example 1 If FPRM $F = x_1x_2 \oplus x_1x_2\bar{x}_3 \oplus x_1x_2\bar{x}_4$ is given, $\nu(F) = [0, 0, 1, 1]$ and $\bar{\nu}(F) = [1, 1, 0, 0]$. □

Definition 3 Let F_a and F_b be FPRMs such that $\nu(F_a) = \bar{\nu}(F_b)$, where we assume that the n-th bit of $\nu(F_a)$ is 0 without loss of generality. The EXOR combination of the two FPRMs, $F_a \oplus F_b$, is called *a Double Fixed-Polarity Read-Muller expression* (DFPRM) *with the polarity vector* $\nu(F_a)$. □

Example 2 Both $F_a = x_1 \oplus \bar{x}_2\bar{x}_3x_4$ and $F_b = \bar{x}_1x_2x_3\bar{x}_4 \oplus \bar{x}_1\bar{x}_4$ are FPRMs. $F = F_a \oplus F_b = x_1 \oplus \bar{x}_2\bar{x}_3x_4 \oplus \bar{x}_1x_2x_3\bar{x}_4 \oplus \bar{x}_1\bar{x}_4$ is a DFPRM. The polarity vector of F is $[0, 1, 1, 0]$. □

Since an FPRM F can be written as $F \oplus 0$ and the constant 0 is considered to be an FPRM with $\bar{\nu}(F)$, the FPRM F is also a DFPRM, i.e., DRPRMs are generalized FPRMs. The FPRM of f is uniquely obtained if the polarity vector V_n is given. However, the DFPRM of f with V_n is not unique. This means that the minimization of DFPRMs is a more difficult problem than that of FPRMs. The minimization of DFPRMs is described in Appendix. In the following, we give the definition of the minimum DFPRMs.

Definition 4 Among all the DFPRMs with a polarity vector V_n which represent an n-variable function f, those with a minimum number of product terms are called *minimum DFPRMs of f with V_n*. The number of product terms in a minimum DFPRM of f with V_n is denoted by $\tau_D[V_n](f)$. Furthermore, $\tau_D(f)$ is defined as follows (note that the n-th bit of V_n is 0):

$$\tau_D(f) = \min_{V_n \in \{0,1\}^{n-1} \times \{0\}} \{\tau_D[V_n](f)\}. \quad (4)$$

□

Since DFPRMs are generalized FPRMs, the DFPRM of f does not require more products than the FPRM. The relation between DFPRMs and other AND-EXOR expressions is shown below.

Property 1 $\mathcal{PPRM}$, $\mathcal{FPRM}$, $\mathcal{DFPRM}$, and $\mathcal{ESOP}$ represent the class of all PPRMs, FPRMs, DFPRMs, and ESOPs, respectively. Then, the relation $\mathcal{PPRM} \subset \mathcal{FPRM} \subset \mathcal{DFPRM} \subset \mathcal{ESOP}$ holds (**Fig. 5**). □

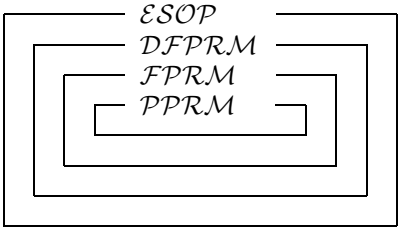


Fig. 5 Relation among classes.

Easy testability is a useful property of EXOR-based circuits [2),11),13),14),20),22)]. We refer to the testing of the FPRM PLA [8),14)] in order to explain the testing of the DFPRM PLA. The FPRM PLA requires only $(n + 4)$ tests to detect all the single stuck-at faults, and the tests are independent of the function realized by the PLA. The single stuck-at fault model is briefly defined as follows [1)].

Definition 5 A stuck-at fault is the logical fault such that a signal line is stuck at a fixed logical value $c \in \{0, 1\}$. According to the value c, it is called *the stuck-at* 0 or *the stuck-at* 1 *fault*. At most one stuck-at fault is assumed to occur in a circuit. The input patterns to test whether the circuit is faulty are called *the test patterns* (or *tests* shortly). □

The test set for the FPRM PLA consists of two test sets: for the AND part and for the EXOR part [8),14)]. The internal faults in the AND part can be detected by the following test set S_a. The faults on the primary input lines and the literal lines are detected by one additional gate, which is described in the testing of the DFPRM PLA in Section 4.

$$S_a = \begin{array}{c} \begin{array}{ccccccccccc} c & x_1 & x_2 & \dots & x_{n-1} & x_n & v_1 & v_2 & \dots & v_n \end{array} \\ \left[\begin{array}{ccccccccccc} d & 1 & 1 & \dots & 1 & 1 & 0 & 0 & \dots & 0 \\ d & 0 & 1 & \dots & 1 & 1 & 0 & 0 & \dots & 0 \\ d & 1 & 0 & \dots & 1 & 1 & 0 & 0 & \dots & 0 \\ & & & \vdots & & & & & \vdots & \\ d & 1 & 1 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ d & 1 & 1 & \dots & 1 & 0 & 0 & 0 & \dots & 0 \end{array}\right] \end{array}$$

In S_a, d represents a don't-care value. The number of tests in S_a is $|S_a| = (n + 1)$; the first test detects the stuck-at 0 faults, and the rest detect the stuck-at 1 faults.

The test set for the EXOR part is as follows.

[☆] Precisely the polarity vector of an FPRM is not uniquely determined. For example, when there is no literals of a variable x_i in an FPRM F, the i-th bit of the polarity vector may be 0 or 1. In such a case we choose an appropriate one.

$$S_b = \begin{array}{c} \begin{matrix} c & x_1 & x_2 & \dots & x_{n-1} & x_n & v_1 & v_2 & \dots & v_n \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & \dots & 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 & 1 & \dots & 1 \end{bmatrix} \end{array}$$

The number of tests is $|S_b| = 4$. Let $S[i]$ be the i-th test in the test set S. The total number of tests is $|S_a \cup S_b| = (n+4)$ because $S_b[2]$ can be merged with $S_a[1]$.

3. Fixed-Polarity OR-EXOR Expressions

The DFPRM PLA of Fig. 4 is a combination of an AND-EXOR PLA and an OR-EXOR PLA. Before discussing the DFPRM PLA, let us consider the OR-EXOR PLA.

Definition 6 An EXOR combination of arbitrary sum terms is called *an OR-EXOR expression. A fixed-polarity OR-EXOR expression* (FPOE) with polarity vector $V_n = [v_1, v_2, \dots, v_n]$ is an OR-EXOR expression in which either positive or negative literals appears for each variable according to V_n, that is, if $v_i = 0$, the positive literal x_i is used in the FPOE, and otherwise the negative literal $\bar{x}_i$. $\nu(F)$ denotes the polarity vector of an FPOE F. □

Example 3 The OR-EXOR expression $F = (\bar{x}_1 \vee \bar{x}_2) \oplus (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \oplus (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \oplus 1$ is an FPOE with the polarity vector $\nu(F) = [1, 1, 0, 0]$. □

From de Morgan's theorem, the product $x_1 x_2 \cdots x_n$ can be converted into $(\bar{x}_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_n) \oplus 1$. To discuss arbitrary products, we define the following notation.

Definition 7 The literals $\bar{x}$ and x of a variable x may be written as $x^{\{0\}}$ and $x^{\{1\}}$, respectively. The special literals $x^{\{\}}$ and $x^{\{0,1\}}$ mean the constant 0 and 1, respectively. □

Example 4 The product $x_1 \bar{x}_2 x_4$ can be written as $x_1^{\{1\}} x_2^{\{0\}} x_3^{\{0,1\}} x_4^{\{1\}}$. □

By using the above notation, an arbitrary product can be written in the form of $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ ($\alpha_i \subseteq \{0,1\}$). Then, we have the following property.

Property 2 Let α_i be a subset of $\{0,1\}$ and β_i be $\{0,1\} - \alpha_i$ for $i = 1, 2, \dots, n$, then the following equation holds:

$$x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} = (x_1^{\beta_1} \vee x_2^{\beta_2} \vee \cdots \vee x_n^{\beta_n}) \oplus 1$$

□

Lemma 1 Let F be an FPRM. $\tilde{F}$ denotes the OR-EXOR expression such that every product of F is converted into the EXOR combination of the sum term and the constant 1 by the equation of Property 2. Then $\tilde{F}$ is an FPOE with $\bar{\nu}(F)$. □

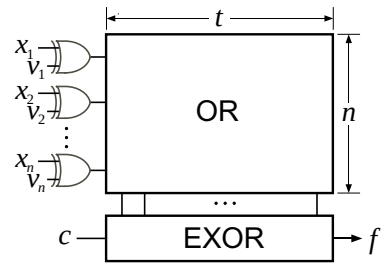


Fig. 6 PLA for FPOEs.

FPOEs have fixed polarities of literals as well as FPRMs. Therefore they can be realized in the OR-EXOR PLA with n literal lines as shown in **Fig. 6**. The FPOE PLA is an alternative realization of FPRMs because an arbitrary FPRM F can be converted into the FPOE $\tilde{F}$.

The testing of FPOE PLAs is similar to FPRM PLAs. From the duality of AND and OR gates, the single stuck-at faults in the OR part are detected by the test set S_c below, which are obtained from the test set S_a for FPRM PLAs by holding $(v_1, v_2, \dots, v_n) = (1, 1, \dots, 1)$.

$$S_c = \begin{array}{c} \begin{matrix} c & x_1 & x_2 & \dots & x_{n-1} & x_n & v_1 & v_2 & \dots & v_n \end{matrix} \\ \begin{bmatrix} d & 1 & 1 & \dots & 1 & 1 & 1 & 1 & \dots & 1 \\ d & 0 & 1 & \dots & 1 & 1 & 1 & 1 & \dots & 1 \\ d & 1 & 0 & \dots & 1 & 1 & 1 & 1 & \dots & 1 \\ & & & \vdots & & & & & \vdots & \\ d & 1 & 1 & \dots & 0 & 1 & 1 & 1 & \dots & 1 \\ d & 1 & 1 & \dots & 1 & 0 & 1 & 1 & \dots & 1 \end{bmatrix} \end{array}$$

The number of tests, $|S_c|$, is $(n+1)$. For the EXOR part, S_b in Sect. 2 can be used as the test set. Then the total number of tests for FPOE PLAs is $|S_c \cup S_b| = n + 4$ because $S_b[3]$ can be merged with $S_c[1]$.

The properties of FPOEs and their PLAs are summarized as follows.

Property 3

(1) FPOEs have the fixed polarity of literals for every variable.
(2) FPOEs can be realized in the OR-EXOR PLA (FPOE PLA) with n literal lines.
(3) The single stuck-at faults of FPOE PLAs can be detected by applying $(n+4)$ tests.
(4) Let F be an FPRM and $\tilde{F}$ be the FPOE. F and $\tilde{F}$ represent the same function.
(5) The polarities of literals of an FPRM F

and the FPOE $\tilde{F}$ are opposite each other, $\nu(F) = \bar{\nu}(\tilde{F})$. □

4. PLA Realization and Testability

In this section, we show that DFPRMs can be realized in the (AND/OR)-EXOR PLA with n literal lines such as Fig. 4. We define a conversion of DFPRMs into the (AND/OR)-EXOR expressions, which correspond to the (AND/OR)-EXOR PLA.

Definition 8 Let F be a DFPRM such that $F = F_a \oplus F_b$ $(\nu(F_a) = \bar{\nu}(F_b))$. $\hat{F}$ is defined as $F_a \oplus \tilde{F}_b$. □

Example 5 The DFPRM F of Example 2 is converted into $\hat{F} = F_a \oplus \tilde{F}_b = x_1 \oplus \bar{x}_2\bar{x}_3x_4 \oplus ((x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4) \oplus 1) \oplus ((x_1 \vee x_4) \oplus 1) = x_1 \oplus \bar{x}_2\bar{x}_3x_4 \oplus (x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4) \oplus (x_1 \vee x_4)$. □

$\hat{F}$ is the EXOR combination of the FPRM F_a and the FPOE $\tilde{F}_b$. Note that F_b and $\tilde{F}_b$ represent the same function as mentioned in Property 3, that is, the DFPRM F and its $\hat{F}$ represent the same function. Since $\nu(F_a) = \bar{\nu}(F_b)$, $\hat{F}$ has the fixed polarity of literals for every variable. The properties of DFPRMs are summarized as follows.

Property 4 Let F be a DFPRM such that $F = F_a \oplus F_b$ $(\nu(F_a) = \bar{\nu}(F_b))$.

(1) F and $\hat{F}$ represent the same function.
(2) The FPRM F_a and the FPOE $\tilde{F}_b$ have the same polarity vector.
(3) $\hat{F}$ has the fixed polarity of literals for every variable. □

From Property 4, $\hat{F}$ can be realized in the PLA of Fig. 4.

Theorem 1 For an arbitrary DFPRM F, $\hat{F}$ represents the same function as F, and can be realized in the (AND/OR)-EXOR PLA with n literal lines. □

When $\hat{F}$ is obtained from F, some $\hat{F}$ have the constant term 1 and the others not. The difference does not affect the PLA realization of Fig. 4 because the constant is assigned to the input line c. If $\hat{F}$ has the constant 1, c is set at 1. Otherwise, c is set at 0.

We discuss the testing for DFPRM PLAs. As described in Sects. 2 and 3, both PLAs of FPRMs and FPOEs require only $(n + 4)$ tests to detect all the single stuck-at faults, and the tests are independent of the functions realized by the PLAs. Since the DFPRM PLA is a pair of an FPRM PLA and an FPOE PLA, the faults are detected by using their test sets.

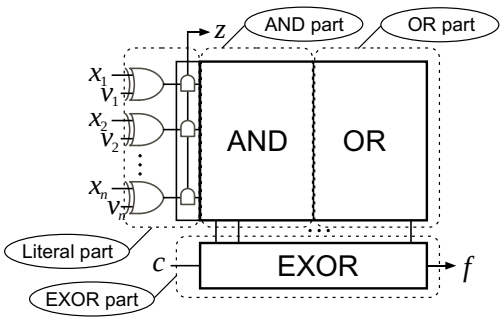


Fig. 7 DFPRM PLA with extra n-input AND gate.

We consider the tests in each of the AND, OR, EXOR, and literal parts in the DFPRM PLA as shown in **Fig. 7**. From Section 2, the faults in the AND part are detected by S_a. And from Section 3, the faults in the OR part are detected by S_c. S_b can be used as the test set for the EXOR part.

A fault signal in the literal part usually fans out to some gates. To detect the fault, some extra gates are often used [3),13),14),20),22)]. If one n-input AND gate is added to the DFPRM PLA such as Fig. 7, the fault detection in the literal part is made by observing z, which is the output of the extra AND gate. The faults on the literal lines and the line z are detected by S_a. The faults on the primary input lines x_i and v_i are detected by the following S_d. Then $|S_d| = 2$.

$$S_d = \begin{array}{c} \begin{matrix} c & x_1 & x_2 & \dots & x_{n-1} & x_n & v_1 & v_2 & \dots & v_n \end{matrix} \\ \begin{bmatrix} d & 0 & 0 & \dots & 0 & 0 & 1 & 1 & \dots & 1 \\ d & 1 & 1 & \dots & 1 & 1 & 0 & 0 & \dots & 0 \end{bmatrix} \end{array}$$

Hence the total test set S for DFPRMs is written as $S = S_a \cup S_b \cup S_c \cup S_d$. The number of tests is $|S| = (2n+4)$ because of the following reasons.

- $S_b[2]$ can be merged with $S_a[1]$.
- $S_b[3]$ can be merged with $S_c[1]$.
- $S_b[2]$ and $S_b[4]$ can be merged with S_d.
- $S_d[2]$ is equal to $S_a[1]$.

From the above discussion, we have the following theorem.

Theorem 2 All the single stuck-at faults in the DFPRM PLA with one extra n-input AND gate are detected by $(2n+4)$ tests and the tests are independent of the functions realized by the PLAs. □

The number of tests of DFPRM PLAs is $(2n + 4)$ while that of FPRM PLAs is $(n + 4)$. DFPRM PLAs require more tests than FPRM PLAs. The number of tests, however, remains $O(n)$.

Table 1 Number of four-variable functions that require t products.

t	PPRM	FPRM	DFPRM	ESOP	SOP
0	1	1	1	1	1
1	16	81	81	81	81
2	120	836	1660	2268	1804
3	560	3496	11520	21744	13472
4	1820	8878	29426	37530	28904
5	4368	17884	21840	3888	17032
6	8008	20152	1008	24	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	4.13	3.66	4.13

av: average

5. Size of DFPRM PLAs

To evaluate the size of DFPRM PLAs, we consider the width t of the PLA, which corresponds to the number of products of the DFPRM. From Property 1, DFPRMs require less products than FPRMs and more products than ESOPs. To confirm the property, we obtain $\tau_D(f)$ of all the four-variable functions by the minimization algorithm presented in Appendix.

We implemented the algorithm in Lisp on SUN Ultra Sparc 60 Model 1450 (19.7 SPECint95). The program computes a minimum DFPRM of a given function f with four or less variables within 0.1 seconds. By using the program, we obtained a table of minimum DFPRMs of all the four-variable functions. A five-variable function can be minimized in 50 seconds if the table is stored in the memory in order to reduce the number of recursive calls.

The results and the comparison with other classes of expressions are shown in **Table 1**, where these columns except DFPRM are quoted from Refs. 18), 21). The average of t for four-variable functions is 4.13, which is smaller than 5.50 of FPRMs. In the worst case, an FPRM requires 10 products while a DFPRM and an ESOP require only 6 products. SOPs (sum-of-products expressions or AND-OR expressions) require at most 8 products.

6. Conclusions and Comments

We proposed DFPRMs, which are a new class of AND-EXOR expressions. They are generalization of FPRMs, and are represented by fewer product terms than FPRMs. DFPRMs can be realized in the (AND/OR)-EXOR PLA that has only n literal lines as well as the FPRM PLA, where n is the number of input variables. We presented a minimization algorithm of DFPRMs with $O(n2^n2^{2^n})$ and computed the minimum DFPRMs for all the four-variable functions. From the results, it was found that the average number of products of DFPRMs for four-variable functions is 4.13. This is smaller than 5.50 of FPRMs and larger than 3.66 of ESOPs. Since the DFPRM PLA has only n literal lines, DFPRMs can be the smallest PLA among all the classes of the AND-EXOR expressions.

The testability of DFPRM PLAs was also discussed. All the stuck-at faults in DFPRM PLAs are detected by $(2n + 4)$ tests. DFPRM PLAs require more tests than FPRM PLAs. The number of tests, however, within two times of that of the other kind of PLAs. This is a trade-off between the testing time and the circuit area. As described above, the main advantage of DFPRMs is that they can be realized in the small PLA. Holding the similar testability to FPRMs is the second advantage of DFPRMs.

Our minimization program can compute minimum DFPRMs for up to five-variable functions. It is difficult to compute minimum DFPRMs for functions with six or more variables. The development of a faster heuristic algorithm to obtain DFPRMs is a future work.

References

1) Abramovici, M., Breuer, M.A. and Friedman, A.D.: *Digital Systems Testing and Testable Design*, revised printing, IEEE Press (1990).
2) Chatterjee, M., Pradhan, D.K. and Kunz, W.: LOT: Logic Optimization with Testability—New Transformations for Logic Synthesis, *IEEE Trans. CAD*, Vol.17, No.5, pp.386–399 (1998).
3) Chattopadhyay, S., Roy, S. and Chaudhuri, P.P.: Synthesis of Highly Testable Fixed-Polarity AND-EXOR Canonical Networks – A Genetic Algorithm-Based Approach, *IEEE Trans. Comput.*, Vol.45, No.4, pp.487–490 (1996).
4) Chattopadhyay, S., Roy, S. and Chaudhuri, P.P.: KGPMIN: an Efficient Multilevel Multioutput AND-OR-XOR Minimizer, *IEEE Trans. CAD*, Vol.16, No.3, pp.257–265 (1997).
5) Davio, M., Deschamps, J.P. and Thayse, A.: *Discrete and Switching Functions*, McGraw-

Hill International (1978).
6) Debnath, D. and Sasao, T.: Minimization of AND-OR-EXOR Three-Level Networks with AND Gate Sharing, *IEICE Trans. Inf. & Syst.*, Vol.E80-D, No.10, pp.1001–1008 (1997).
7) Drechsler, R., Theobald, M. and Becker, B.: Fast OFDD Based Minimization of Fixed Polarity Reed-Muller Expressions, *Proc. Euro. DAC with EURO-VHDL '94*, pp.2–7 (1994).
8) Fujiwara, H.: *Logic Testing and Design for Testability*, MIT Press (1985).
9) Hirayama, T. and Nishitani, Y.: A Simplification Algorithm of AND-EXOR Expressions Guaranteeing Minimality for Some Subclass of Logic Functions, *IEICE Trans.*, Vol.J78-D-I, No.4, pp.409–415 (1995).
10) Hirayama, T., Koda, G., Nishitani, Y. and Shimizu, K.: Easily Testable Realization Based on Single-Rail-Input OR-AND-EXOR Expressions, *IEICE Trans. Inf. & Syst.*, Vol.E82-D, No.9, pp.1278–1286 (1999).
11) Kalay, U., Perkowski, M.A. and Hall, D.V.: A Minimal Universal Test Set for Self-Test of EXOR-Sum-of-Products Circuits, *IEEE Trans. Comput.*, Vol.49, No.3, pp.267–276 (2000).
12) Luccio, F. and Pagli, L.: On a New Boolean Function with Applications, *IEEE Trans. Comput.*, Vol.48, No.3, pp.296–310 (1999).
13) Pradhan, D.K.: Universal Test Sets for Multiple Fault Detection in AND-EXOR Arrays, *IEEE Trans. Comput.*, Vol.C-27, No.2. pp.181–187 (1978).
14) Reddy, S.M.: Easily Testable Realization for Logic Functions, *IEEE Trans. Comput.*, Vol.C-21, No.11. pp.1183–1188 (1972).
15) Sarabi, A. and Perkowski, M.A.: Fast Exact and Quasi-Minimal Minimization of Highly Testable Fixed-Polarity AND/XOR Canonical Networks, *Proc. 29th ACM/IEEE DAC*, pp.30–35 (1992).
16) Sasao, T. and Besslich, P.: On the Complexity of Mod-2 Sum PLA's, *IEEE Trans. Comput.*, Vol.39, No.2, pp.262–266 (1990).
17) Sasao, T.: EXMIN2: A Simplification Algorithm for Exclusive-OR-Sum-of-Products Expressions for Multiple-Valued Input Two-Valued Output Functions, *IEEE Trans. CAD*, Vol.12, No.5, pp.621–632 (1993).
18) Sasao, T.: Representations of Logic Functions Using EXOR Operators, Sasao, T. and Fujita, M. (Eds.), *Representations of Discrete Functions* pp.29–54, Kluwer Academic Publishers (1996).
19) Sasao, T. and Izuhara, F.: Exact Minimization of FPRMs Using Multi-Terminal EXOR TDDs, Sasao, T. and Fujita, M. (Eds.), *Representations of Discrete Functions*, pp.191–210, Kluwer Academic Publishers (1996).
20) Sasao, T.: Easily Testable Realizations for Generalized Reed-Muller Expansions, *IEEE Trans. Comput.*, Vol.46, No.6, pp.709–716 (1997).
21) Sasao, T.: *Switching Theory For Logic Synthesis*, Kluwer Academic Publishers (1999).
22) Yamada, T.: Easily Testable AND-EOR Combinational Logic Circuits, *IECE Trans.*, Vol.J66-D, No.1, pp.105–110 (1983).
23) Zhang, Y.Z. and Rayner, P.J.W.: Minimisation of Reed-Muller Polynomials with Fixed Polarity, *IEE Proc.*, Vol.131.Pt.E, No.5, pp.177–186 (1984).

Appendix: Minimization Algorithm

For an n-variable function f and a polarity vector V_n, let $\tau_F[V_n](f)$ denote the number of products of the FPRM of f with polarity vector V_n. Since a DFPRM of an n-variable function f is $F_a \oplus F_b$ such that $\nu(F_a) = \bar{\nu}(F_b)$ and the n-th bit of $\nu(F_a)$ is 0, the minimum number of products of DFPRMs is represented by the following equations, where $\bar{V}_n$ denotes the bitwise complement of V_n, and $\mathcal{F}^n$ denotes the set of all the n-variable functions:

$$\begin{aligned}
&\tau_D[V_n](f) \\
&= \min_{f=f_a\oplus f_b} \{\tau_F[V_n](f_a) + \tau_F[\bar{V}_n](f_b)\} \\
&= \min_{f_b\in\mathcal{F}^n} \{\tau_F[V_n](f \oplus f_b) + \tau_F[\bar{V}_n](f_b)\} \\
&\tau_D(f) = \min_{V_{n-1}\in\{0,1\}^{n-1}\times\{0\}} \{\tau_D[V_n](f)\}
\end{aligned}$$

From the above equations, we have a naive minimization algorithm, and its time complexity is $2^{n-1} \cdot 2^{2^n} \cdot 2T_F(n) = O(2^n 2^{2^n}) T_F(n)$, where n is the number of variables of f and $T_F(n)$ is the computational time for calculating the FPRM of f with V_n. Since $T_F(n) = O(2^n)$ is known, the time complexity of this naive algorithm is written as $O(2^{2n+2^n})$. Then we propose a faster minimization algorithm based on the following theorem.

Theorem 3 For an n-variable function f and an n-bit polarity vector $V_n \in \{0,1\}^{n-1} \times \{0\}$, the following recurrence equation holds, where V_{n-1} is the $(n-1)$-bit polarity sub-vector of V_n deleting n-th bit from V_n, and $\dot{V}_{n-1}$ denotes V_{n-1} if the $(n-1)$-st bit of V_{n-1} is 0 and $\bar{V}_{n-1}$ otherwise.

$$\tau_D[V_n](f) = \min_{g\in\mathcal{F}^{n-1}} \{\tau_F[\bar{V}_{n-1}](f_{\{0\}} \oplus g)$$

$$+\tau_F[V_{n-1}](f_{\{1\}} \oplus g)$$
$$+\tau_D[\dot{V}_{n-1}](g)\}$$

Proof: $f = \bar{x}_n(f_{\{0\}} \oplus g) \oplus x_n(f_{\{1\}} \oplus g) \oplus g$ holds for an arbitrary $(n-1)$-variable function g [9]. For a polarity vector $V_n \in \{0,1\}^{n-1} \times \{0\}$ and an $(n-1)$-variable function g, let G_0 be the FPRM of $f_{\{0\}} \oplus g$ with $\bar{V}_{n-1}$, G_1 be the FPRM of $f_{\{1\}} \oplus g$ with V_{n-1}, and G_2 be a minimum DFPRM of g with $\dot{V}_{n-1}$. Then it is obvious that the expression $\bar{x}_n G_0 \oplus x_n G_1 \oplus G_2$ is a DFPRM of f with polarity vector V_n. Hence we have

$$\begin{aligned}&\tau_D[V_n](f)\\ &\le \min_{g \in \mathcal{F}^{n-1}} \{\tau_F[\bar{V}_{n-1}](f_{\{0\}} \oplus g)\\ &\quad +\tau_F[V_{n-1}](f_{\{1\}} \oplus g) + \tau_D[\dot{V}_{n-1}](g)\}.\end{aligned}$$

In the rest, we show the converse of the above inequality. Let $F = F_a \oplus F_b$ be a minimum DFPRM of f with polarity vector $V_n = \nu(F_a)$. Since F_a and F_b are FPRMs with polarity vectors V_n and $\bar{V}_n$, respectively, the DFPRM $F = F_a \oplus F_b$ can be written in the forms $F_{a,\{0\}} \oplus x_n F_{a,\{0,1\}} \oplus F_{b,\{1\}} \oplus \bar{x}_n F_{b,\{0,1\}}$, where $F_{a,\{0\}}$, $F_{a,\{0,1\}}$, $F_{b,\{1\}}$, and $F_{b,\{0,1\}}$ have no literals of the variable x_n. Furthermore, since $F_{a,\{0\}}$ and $F_{a,\{0,1\}}$ are FPRMs with V_{n-1}, and $F_{b,\{1\}}$ and $F_{b,\{0,1\}}$ are FPRMs with $\bar{V}_{n-1}$, we have that $F_{a,\{0,1\}}$ and $F_{b,\{0,1\}}$ are FPRMs with V_{n-1} and $\bar{V}_{n-1}$, respectively, and $F_{a,\{0\}} \oplus F_{b,\{1\}}$ is a DFPRM with $\dot{V}_{n-1}$.

Let g be the function represented by $F_{a,\{0\}} \oplus F_{b,\{1\}}$. Then $F_{a,\{0,1\}}$ and $F_{b,\{0,1\}}$ represent $f_{\{1\}} \oplus g$ and $f_{\{0\}} \oplus g$, respectively. Hence we have the following inequalities.

$$\tau(F_{a,\{0,1\}}) = \tau_F[V_{n-1}](f_{\{1\}} \oplus g)$$
$$\tau(F_{b,\{0,1\}}) = \tau_F[\bar{V}_{n-1}](f_{\{0\}} \oplus g)$$
$$\tau(F_{a,\{0\}} \oplus F_{b,\{1\}}) \ge \tau_D[\dot{V}_{n-1}](g)$$

From the above inequalities, we have

$$\begin{aligned}&\tau_D[V_n](f)\\ &= \tau(F)\\ &= \tau(F_{a,\{0,1\}}) + \tau(F_{b,\{0,1\}})\\ &\quad +\tau(F_{a,\{0\}} \oplus F_{b,\{1\}})\\ &\ge \tau_F[V_{n-1}](f_{\{1\}} \oplus g)\\ &\quad +\tau_F[\bar{V}_{n-1}](f_{\{0\}} \oplus g) + \tau_D[\dot{V}_{n-1}](g)\\ &\ge \min_{g \in \mathcal{F}^{n-1}} \{\tau_F[\bar{V}_{n-1}](f_{\{0\}} \oplus g)\\ &\quad +\tau_F[V_{n-1}](f_{\{1\}} \oplus g) + \tau_D[\dot{V}_{n-1}](g)\}.\end{aligned}$$

□

From the above theorem, we can construct an algorithm $dfprm_V(f, V_n)$ to calculate a minimum DFPRM of f with a polarity vector V_n, which is shown in **Fig. 8**. In the algorithm, $fprm(f, V_n)$ is the procedure to obtain the FPRM of f with V_n. The computational time, $T_{DV}(n)$, of $dfprm_V(f, V_n)$ is represented by the recurrence relation $T_{DV}(n) = (2T_F(n-1) + T_{DV}(n-1)) \cdot 2^{2^{n-1}}$, where n is the number of variables and $T_F(n)$ is the computational time for calculating an FPRM. Since $T_F(n) = O(2^n)$, we have $T_{DV}(n) = O(n2^{2^n})$.

The algorithm $dfprm(f)$ to calculate a minimum DFPRM of f calls $dfprm_V(f, V_n)$ for all the polarity vectors $V_n \in \{0,1\}^{n-1} \times \{0\}$. Its time complexity is $O(n2^{2^n})2^{n-1} = O(n2^{n+2^n})$, which is smaller than that of the naive algorithm.

$dfprm_V(f, V_n)$:
(1) $F_{\min} \leftarrow$ (a large dummy).
(2) Do the following for every $g \in \mathcal{F}^{n-1}$.
  (a) $G_0 \leftarrow fprm(f_{\{0\}} \oplus g, \bar{V}_{n-1})$.
  (b) $G_1 \leftarrow fprm(f_{\{1\}} \oplus g, V_{n-1})$.
  (c) $G_2 \leftarrow dfprm_V(g, \dot{V}_{n-1})$.
  (d) $F \leftarrow \bar{x}_n G_0 \oplus x_n G_1 \oplus G_2$.
  (e) If $\tau(F) \le \tau(F_{\min})$ then $F_{\min} \leftarrow F$.
(3) Return $F_{\min}$.
$dfprm(f)$:
(1) $F_{\min} \leftarrow$ (a large dummy).
(2) Do the following for every polarity vector $V_n \in \{0,1\}^{n-1} \times \{0\}$.
  (a) $F \leftarrow dfprm_V(f, V_n)$.
  (b) If $\tau(F) \le \tau(F_{\min})$ then $F_{\min} \leftarrow F$.
(3) Return $F_{\min}$.

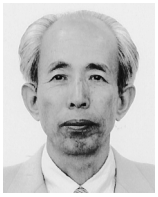
Fig. 8 Minimization algorithm.

(Received September 14, 2000)
(Accepted December 1, 2000)

Takashi Hirayama received his BE, ME, and PhD degrees in computer science from Gunma University, Kiryu, Japan, in 1994, 1996, and 1999, respectively. He is currently a lecturer in the Department of Electrical and Electronics Engineering, Ashikaga Institute of Technology, Ashikaga, Japan. His research interests include high level and logic synthesis and design for testability.

Kazuyuki Nagasawa received his ME and PhD degrees in computer science from Tohoku University, Sendai, Japan, in 1962 and 1967, respectively. He is a professor in the Department of Electrical and Electronics Engineering, Ashikaga Institute of Technology, Ashikaga, Japan. In addition, he holds the post of the director of the Center for Computer Science at the university. His research interests include logic synthesis of digital circuits and application of neural networks to the design of character recognition systems.

Yasuaki Nishitani received his BE degree in electrical engineering, his ME and PhD degrees in computer science from Tohoku University, Sendai, Japan, in 1975, 1977, and 1984, respectively. In 1981 he joined the Software Product Engineering Laboratory at the NEC corporation. From 1987 to 2000 he was an associate professor in the Department of Computer Science, Gunma University. Since 2000 he has been a professor in the Department of Computer and Information Science, Faculty of Engineering, Iwate University. His current research interests include switching theory, software engineering, and distributed algorithms.



Kensuke Shimizu received his BE, ME, and PhD degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1962, 1964, and 1967, respectively. He was a lecturer from 1967 to 1968 and an associate professor from 1968 to 1976 in the Department of Electronic Engineering, Faculty of Engineering, Gunma University. Since 1976 he has been a professor in the Department of Computer Science, Faculty of Engineering, Gunma University, engaged in research and education in logic circuits, switching theory, and expert systems.