Generalized Inclusive Forms — New Canonical Reed-Muller Forms Including Minimum ESOPs

Marek Perkowski
Portland State University, marek.perkowski@pdx.edu

Alan Mishchenko
Portland State University

Malgorzata Chzanowka-Jeske
Portland State University

VLSI Design, 2002 Vol. 14 (1), pp. 13–21

Generalized Inclusive Forms—New Canonical Reed-Muller Forms Including Minimum ESOPs

MALGORZATA CHRZANOWSKA-JESKE*, ALAN MISHCHENKO and MAREK PERKOWSKI

Department of Electrical and Computer Engineering, Portland State University, 1800, 6th Avenue, Portland, OR 97207-0751, USA

(Received 20 January 2000; In final form 4 October 2000)

This paper describes two families of canonical Reed-Muller forms, called inclusive forms (IFs) and their generalization, the generalized inclusive forms (GIFs), which include minimum ESOPs for any Boolean function. We outline the hierarchy of known canonical forms, in particular, pseudo-generalized Kronecker forms (PGKs), which led us to the discovery of the new families. Next, we introduce special binary trees, called the S/D trees, which underlie IFs and permit their enumeration. We show how to generate IFs and GIFs and prove that GIFs include minimum ESOPs. Finally, we present the results of computer experiments, which show that GIFs reduce the search space for minimum ESOP by several orders of magnitude, and this reduction grows exponentially with the number of variables.

Keywords: Reed-Muller expansions; Canonical forms; Decision trees; A minimum ESOP; Generalized Davio expansion; S/D trees

INTRODUCTION

Reed-Muller (AND/EXOR) expansions play an important role in logic synthesis and circuit design by producing economical and highly-testable implementations of Boolean functions [3–6]. The range of Reed-Muller expansions include canonical forms, i.e. expansions that create unique representations of a Boolean function. Several large families of canonical forms: fixed polarity Reed-Muller forms (FPRMs), generalized Reed-Muller forms (GRMs), Kronecker forms (KROs), and pseudo-Kronecker forms (PKROs), referred to as the Green/Sasao hierarchy, have been described [7–9]. (See Fig. 1 for a set-theoretic relationship between these families.)

Research in the field of canonical forms is motivated to a large extent by the need to improve the algorithms currently used for ESOP minimization. Efficient exact algorithms exist only for certain families of Reed-Muller expansions belonging to the Green/Sasao hierarchy, for instance [10–12]. These families, however, do not exhaust all ESOPs. This is why state-of-the-art ESOP minimizers [13–15] are based on heuristics and give the exact solution only for functions with a small number of variables. The well-known formulation for finding the exact ESOP was given in Ref. [16], but all known exact algorithms can deliver solutions only for some of the functions on less than 10 variables.

Recently, new general families of canonical forms have been proposed [1,2], that include the above-mentioned well-known families, in particular GRMs and PKROs. The discovery of these forms suggests future advances in exact ESOP minimization. Still none of these families has been proven powerful enough to include minimum ESOPs for every given function.

In this paper, we propose two still more general families of canonical Reed-Muller forms, called inclusive forms (IFs) and generalized inclusive forms (GIFs). The second family is the first ever discovered to include minimum ESOPs.

The remainder of this paper is organized as follows. The basic definitions of the families of forms belonging to the Green/Sasao hierarchy and their recent generalizations [1,2] are given in second section. The concept of S/D trees, which is essential for creation and enumeration of IFs, is presented in third section. Properties of IFs and the formula to calculate their quantity is given in fourth section and illustrated by comprehensive enumeration of IFs for two variables. Fifth section is devoted to generalizations of IFs, called the GIFs. The application of the GIFs to exact logic

[1]This work was supported in part by the NSF grant MIP-9629419
*Corresponding author. Tel.: +1-503-725-5415. Fax: +1-503-725-3807. E-mail: jeske@ee.pdx.edu

ISSN 1065-514X print/ISSN 1563-5171 online © 2002 Taylor & Francis Ltd
DOI: 10.1080/10655140290009774

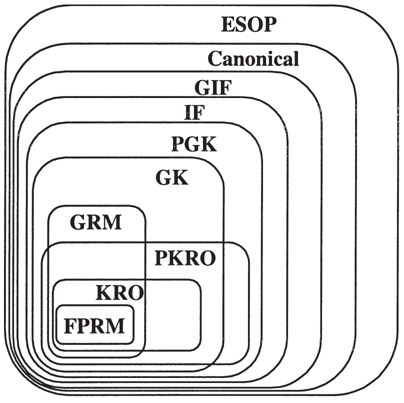

FIGURE 1 Set-theoretic relationship between families of canonical forms.

minimization is discussed in sixth section. Experimental results are presented in the seventh section, followed by conclusions in the eigth section.

GREEN/SASAO HIERARCHY OF CANONICAL FORMS AND THEIR GENERALIZATIONS

The Green/Sasao hierarchy of families of canonical forms and corresponding decision diagrams is based on three generic expansions

$$f(x_1, x_2, \ldots, x_n) = x_1 f_0(x_2, \ldots, x_n) \oplus \bar{x}_1 f_1(x_2, \ldots, x_n) \quad \text{(Shannon–S)} \qquad (1)$$

$$f(x_1, x_2, \ldots, x_n) = f_0(x_2, \ldots, x_n) \oplus x_1 f_2(x_2, \ldots, x_n) \quad \text{(Positive Davio–pD)} \qquad (2)$$

$$f(x_1, x_2, \ldots, x_n) = f_0(x_2, \ldots, x_n) \oplus \bar{x}_1 f_2(x_2, \ldots, x_n) \quad \text{(Negative Davio–nD)} \qquad (3)$$

Here f_0 is $f(0,x_2,\ldots,x_n)$ with x_1 replaced by 0 (negative cofactor of variable x_1), f_1 is $f(1,x_2,\ldots,x_n)$ with x_1 replaced by 1 (positive cofactor of variable x_1), f_2 is $f_0 \oplus f_1$, and symbol $\oplus$ means Exclusive OR.

An arbitrary n-variable function $f(x_1,x_2,\ldots,x_n)$ can be represented using the positive polarity Reed-Muller form (PPRM)

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

For each function f, the coefficients a_i are determined uniquely, so PPRM is a canonical form. If we use either only the positive literal (x_i) or only the negative literal ($x_{\bar{i}}$) for each variable in Eq. (4), we get the FPRM. There are 2^n possible combinations of polarities and as many FPRMs for any given logic function.

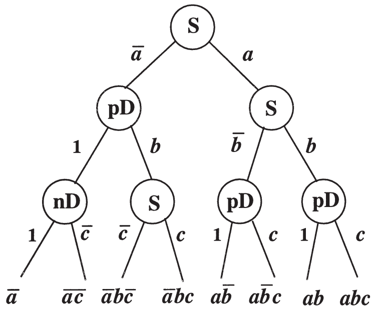

FIGURE 2 A pseudo-Kronecker tree and canonical expansion it produces.

If we freely choose the polarity of each literal in Eq. (4), we get a GRM. In GRMs, contrary to FPRMs, the same variable can appear in both positive and negative polarities. There are $n2^{n-1}$ literals in Eq. (4), so there are $2^{n2^{n-1}}$ polarities for an n-variable function and as many GRMs. Each of the polarities determines a unique set of coefficients, and thus each GRM is a canonical representation of a function.

Two other types of expansions result from flattening [1] of certain binary trees. To create these trees, the following procedure has been proposed. Let us create a binary tree in such a way that each k-th level ($0 \leq k < n$), starting from the root node on top of the tree, contains 2^k nodes. There are $1 + 2 + \cdots + 2^{n-1} = 2^n - 1$ nodes in this tree. Suppose we select an ordering of n variables and use one of the elementary expansions (1)–(3) in each node.

If throughout each level of the tree only one elementary expansion (S, pD, or nD) is used, the resulting canonical form is the KRO. If an arbitrary expansion is allowed in each node, the result is the PKRO. There are 3^n and at most $3^{2^n} - 1$ different KROs and PKROs [3], respectively. These families intersect with GRMs but do not contain them (Fig. 1). An example of a pseudo-Kronecker tree and the resulting canonical form are given in Fig. 2.

In Refs. [1,2], three more families of canonical expansions were given. These forms are generated by flattening certain type of trees. The following procedure for building the tree was proposed. First, partition all n variables into disjoint non-empty sets S_j [1] such that the union of these sets is equal to the initial set of variables. Next, order these blocks and put them in correspondence with levels of the tree. For every level, if the variable block consists of a single variable, one of the generic expansions (S, pD, or nD) is selected for its nodes. If the block contains more than one variable, one GRM polarity is selected for its nodes.

DEFINITION 1 The family of forms created by flattening this tree is called generalized Kronecker forms (GKs) [1].

DEFINITION 2 If we allow any of the generic expansions (1)–(3) to be used with single variable blocks and any of the GRM polarities to be selected for many-variable nodes on the same level, it is called pseudo-generalized Kronecker forms (PGKs) [1].

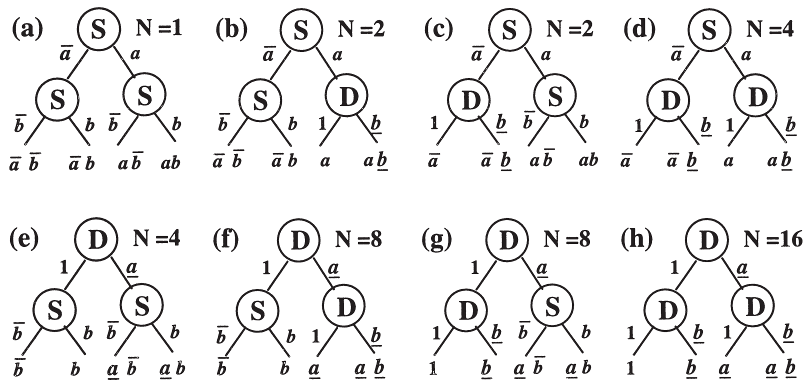

FIGURE 3 All S/D trees and GE for two variables.

Let us consider two extreme cases. If each block includes only one variable, the tree reduces to a special case of a PKRO tree. If there is only one block containing all variables, the tree reduces to one of GRMs. Thus, we may conclude that PGKs subsume PKROs and GRMs.

S/D TREES AND INCLUSIVE FORMS

In this section, we introduce the concept of S/D trees, which is important to define the family of IFs.

First, we present a generalization of the Positive Davio (2) and Negative Davio (3) expansions introduced in the previous section. We call this new expansion the *generalized Davio expansion*

$$f(x_1, x_2, \ldots, x_n) = f_0(x_2, \ldots, x_n) \oplus \underline{x_1} f_2(x_2, \ldots, x_n) \quad \text{(Generalized Davio}-\text{gD)} \qquad (5)$$

Here the underlined literal $\underline{x_1}$ is a *generalized literal*. It stands for any polarity of variable x_1, positive or negative. In a sense, generalized Davio expansion is a compact notation for both Positive and Negative Davio expansions. It is helpful to note at the outset that the generalized Davio expansion is not used in this paper to build decision diagrams for functions, but only to describe expressions, which produce a family of canonical forms. A literal which cannot change its polarity is called an ordinary literal and is created by Shannon node.

Let us now create a binary tree in the same way we created trees for Kronecker and pseudo-Kronecker expressions. Each of the nodes of the tree is selected to have either Shannon expansion (1) or generalized Davio expansion (5).

Definition 3 The tree created in this way is called the *S/D trees* for the given ordering of n variables.

As it was already pointed out, an S/D tree for n variables has $2^n - 1$ nodes and so there are 2^{2^n-1} such distinct trees for each variable order. Figure 3 shows all S/D trees for two variables.

Definition 4 A generalized expansion (GE) is the expansion containing both ordinary and generalized literals produced by the S/D tree.

In particular, a GE may have no generalized literals (when S nodes are used throughout the tree) or consist of $n2^{n-1}$ generalized literals only (when gD nodes are used throughout the tree). It is easy to see that in the latter case, the GE produces all GRMs for the given number of variables.

Definition 5 IFs for a given variable ordering is a set of expansions created by flattening the S/D tree of this variable order and allowing generalized literals presented in the GE to have all possible combinations of polarities.

It is easy to see that a GE with m generalized literals produces as many ordinary forms as there are distinct polarity assignments of generalized literals, namely 2^m.

Example 1 Figure 3 shows derivation of IFs for two variables, when the variable ordering is fixed (a,b). The number N positioned over each tree shows how many expansions can be created from this tree. For example, tree (b) and its corresponding GE $\{\bar{a}\bar{b}, \bar{a}b, a, a\underline{b}\}$ produces two ordinary expansions $\{\bar{a}\bar{b}, \bar{a}b, a, a\bar{b}\}$ and $\{\bar{a}\bar{b}, \bar{a}b, a, ab\}$. By adding numbers N for each tree, we get the total number of IFs for $n = 2$.

$$N_{\text{IF}} = (1 + 2 + 2 + 4) + (4 + 8 + 8 + 16) = 45.$$

In the next section, we derive an exact formula for N_{IF} for an arbitrary number of variables.

PROPERTIES OF INCLUSIVE FORMS

In this section, we prove that all IFs for the given variable ordering are canonical and unique.

Theorem 1 *Each IF $\{t_i\}$, $1 \le i \le n$, is canonical, i.e. for any function F of the same number of variables, there exists one and only one set of coefficients $\{a_i\}$, such that this function can be represented as $F = a_1 t_1 \oplus \cdots \oplus a_n t_n$.*

Proof In Ref. [5], it was shown that an expansion is canonical iff its terms are linearly independent, that is, none of the terms is equal to a linear combination of other terms.

Let us therefore prove by induction on the number of variables that terms in IFS are linearly independent. For $n = 1$, there are only three IF forms, which coincide with the generic Shannon and Davio expansions, introduced in "S/D trees and inclusive forms section". These forms are linearly independent and canonical.

Let us now assume that the theorem is true for the number of variables $n = k$ and prove that it is true for $n = k + 1$. Suppose that it is not true, i.e. there exists an S/D tree for $n = k + 1$ variables (a_0,a_1,...,a_k) such that, although all the forms for $n = k$ are linearly independent, there is the form f_i generated by this tree such that one of its terms t_j is a linear combination of other terms.

Suppose a_k is the variable on top of the tree. Then, all the terms of f_i are split into two equal groups G_1 and G_2. In case of Shannon expansion, exactly one half of the terms (group G_1) has variable a_k complemented while the other half (group G_2) has a_k uncomplemented. In case of generalized Davio expansion, exactly one half of the terms (group G_1) does not have variable a_k at all, while the other half of them (group G_2) have it present in any polarity. It is easy to see that the term t_j and all the terms that constitute the linear combination equal to t_j belong to only one of the groups, either G_1 or G_2. In case of Shannon expansion, we factor a_k from both t_j and the linear combination and get the equality, that depends only on variables $a_0, a_1, \ldots a_{k-1}$, meaning that the terms are not linearly independent for $n = k$, which is a contradiction. In case of generalized Davio expansion, if the term t_j and all the terms that constitute the linear combination belong to group G_1, it is a contradiction. If they belong to group G_2, again all of them can belong to either those terms which have a_k complemented, or to those terms that have a_k uncomplemented. We repeat our previous argument for Shannon expansion and arrive at a contradiction. □

Theorem 2 *The IFs are unique.*

Proof The forms are unique, which means that if a form is produced by an S/D tree, there is no other S/D tree for the given variable ordering, which will produce the same form.

Let us prove by induction on the number of variables. For $n = 1$, there are only three possible forms and they are unique. Suppose it is true for $n = k$. Let us prove that it is true for $n = k + 1$.

Suppose it is not true, i.e. there are two different S/D trees for the given variable ordering, which produce the same expansion. Since the theorem is true for $n = k$, these expansions may differ only in the variable a_k, which is found on top of the S/D tree. But there are only two distinct S/D trees produced by the variable a_k, in one of them the root node has Shannon expansion, in another the root node has generalized Davio expansion. Obviously, these two trees cannot create identical forms. This proves the second part of the theorem, the uniqueness of IFs. □

Theorem 3 *For the given ordering of n variables, there are*

$$\prod_{k=0}^{n-1} (1 + 2^{2^{n-k-1}})^{2^k}$$

unique IFs.

Proof To derive the formula, let us enumerate the levels of the tree starting from the root node with 0-based integers. Let us consider a node on the k-th level of an S/D tree. If it is a Shannon node, it does not contribute generalized literals to the GE produced by the tree and does not produce more than one resulting canonical expansion. If it is a generalized Davio node, it contributes 2^{n-k-1} generalized literals to the GE, which, in turn, produce $2^{2^{n-k-1}}$ resulting canonical expansions.

Now we observe that the k-th level consists of 2^k nodes, each of which can be either Shannon or generalized Davio. It is possible to evaluate the contribution to the quantity of resulting canonical expansions of the entire k-th level of nodes for all S/D trees, which differ only in polarity assignments. This contribution is $(1 + 2^{2^{n-k-1}})^{2^k}$. The only thing left to do after this, is to create the product of these contributions, since each level adds to the total sum of expansions independently of all others. □

Example 2 For $n = 3$, there are

$$\binom{3^n}{2^n} = \binom{27}{8} = 2,220,075$$

possible expansions containing 2^n cubes. Among them, only 527,121 are linearly independent, or canonical. According to the formula (1), there are $N_{\mathrm{IF}} = (1 + 16)1(1 + 4)2(1 + 2)4 = 34,425$ IFs for each ordering of variables. We have verified these results using a program, which systematically generates all linearly independent forms for three variables and counts only those which can be created by S/D tree for one given variable order.

GENERALIZED INCLUSIVE FORMS AND THEIR PROPERTIES

It is easy to see that, for different variable orderings, some forms are not repeated while other forms are, for example, KROs and GRMs. Therefore, the union of sets of IFs for all variable orders contains more forms than any of the IF set taken separately and less forms than the total number of forms in all IF sets.

Definition 6 The family of forms, which is created as a union of sets of IFs for all variable orders, is called the GIFs.

TABLE I The number of IFs as a function of the number of repetitions of these forms for six possible variable orders ($n = 3$)

#Repetitions	#IFs
1	45,696
2	44,880
3	13,872
6	4913
Total GIFs	109,361

THEOREM 4 *GIFs are canonical in respect to any particular variable order.*

Proof It follows from Theorem 2 and Definition 6.

If in Definition 6 we relax the requirement of fixed variable ordering, and allow any ordering of variables in the branches of the tree but do not allow repetitions of variables in the branches, we generate a still more general family of canonical forms.

DEFINITION 7 The family of forms, generated by the S/D tree with no fixed ordering of variables, provided that variables are not repeated along the same branches, is called free generalized inclusive forms (FGIFs).

Example 3 It is easy to calculate the number of GIFs for $n = 2$, if we notice that four out of eight S/D trees in Fig. 3 generate forms, which are repeated when the variable ordering is changed from (a, b) to (b, a). These are trees (a), (d), (e) and (h). So to calculate the number of GIFs we have the following calculation:

$$N_{\text{GIFs}} = 2 \times 45 - (1 + 4 + 4 + 16) = 65.$$

For $n = 2$, the number of FGIFs is the same as the number of GIFs.

The studies show that it is difficult to trace the relationship between the number of forms that are repeated for $n > 2$ and the number of forms that are not, similarly to PKROs. In Table I, we give the result of a computer experiment, which shows that for $n = 3$ this relationship becomes rather complicated. The number of IFs for a given variable order is 34,425 and a number of different variables orders is 6 (3!). If we multiply the number of IFs for a variable order by a number of orders we get 206,550 but in such calculations we included some forms multiple times. Some of the repeated forms are GRMs and KROs which are easy to count. Unfortunately, in addition, there are others which cannot be counted so easily. In Table I, based on computer calculations, we show how many times each of the GIFs is repeated in our simplified calculations. The total number of GIFs for $n = 3$ is given in the last row of the table.

Similarly, it can be shown that for FGIFs for $n = 3$ there are $2 \times 3 + 6 = 12$ variable orders and at most $2 \times 109,361$ FGIF forms. As for any Reed-Muller forms and decision diagrams, the size of a S/D tree and the size of a GIF form, for a given function, depend on a variable order. Search for a good variable order is computationally expensive.

GIFS AND A MINIMUM ESOP

In this section, we will explore the relation between GIFs and a minimum ESOP.

DEFINITION 8 An ESOP is called a minimum ESOP if the number of terms is the minimum among all possible ESOPs and the number of literals is also minimum among all solutions with the minimum number of terms.

In general any function can be represented (decomposed, expanded) as:

$$F = f_1(x_1, \ldots\ldots, x_n) \oplus x_k f_2(x_1, \ldots\ldots, x_n) \oplus \bar{x}_k f_3(x_1, \ldots\ldots, x_n) \quad (6)$$

where f_1, f_2, f_3 are sets of product terms (called terms) grouped together according to the presence and polarity of an arbitrary decomposition variable x_k. Let us now assume that function F is a minimum ESOP. In such case the following properties hold:

1. f_1 has no repeated terms
2. f_2 has no repeated terms
3. f_3 has no repeated terms
4. there are no identical terms in any of the pairs $\{f_1,f_2\}$, $\{f_1,f_3\}$, $\{f_2,f_3\}$

Also observe that if all three sets are non-empty, F cannot be an EXOR of terms on the same set of variables because S expansion would be applicable to it. It cannot be a GRM as well. It will be now our goal to consider all possible cases of F and determine that for each of them F is realizable as a certain GIF. It means that, for any minimum ESOP expression, we can always find the order of variables to create an S/D tree that would generate a GIF form corresponding to this minimum ESOP expression. In order to do this, we have first to explain the S/D tree building procedure.

S/D Tree Building Procedure

Assume that we build a S/D tree, for function F, by selecting one variable at a time and choosing one of two (S or gD) expansions. To choose the feasible expansion for the selected variable we divide all terms of a function (as shown in expression 6) into three sets: terms that do not contain the given variable, 1-Set, those that contain it as a complemented literal, complemented-set (CS), and those that contain it as a non-complemented literal, non-complemented-set (NCS).

- If the 1-Set is empty, assume Shannon expansion on the given variable, decompose it using the two remaining sets, choose the next variable from the variable set and continue using CS set and NCS set from the previous step as starting sub-functions at the current level. Such variable is called an ordinary variable.
- If the 1-Set is not empty assume the generalized Davio

expansion (1-Set used as one sub-function and CS set combined with NCS set used as the second sub-function) and go to the next level.

- Continue until all variables are decomposed and all sub-functions on level n are equal to 0 or 1.

During the decomposition process several sub-function sub-trees are created. It can happen that due to choosing a wrong variable (for example using a predetermined variable order) we obtain two identical terms in a sub-function and we cannot thus generate the next level of a S/D tree. Such identical terms can only appear after at least three generalized Davio expansions were performed.

In the following example, we show that if the given order of variables is not good, the S/D tree cannot be generated because two identical cubes can appear in a sub-function.

Example 4 Let us create the S/D tree with variable ordering (abcdef) for the function

$$\bar{a}\oplus \mathrm{abef}\oplus \mathrm{abcde}\oplus \bar{a}\bar{b}\bar{c}\mathrm{de}\oplus \bar{b}\bar{e}\mathrm{f}.$$

It is easy to see that this is the minimum ESOP, because the exorlink-distance [13] between any pair of cubes is three or more. First, we perform generalized Davio expansions (decompositions) on variables a, b and c:

$$\bar{b}\bar{e}\mathrm{f}\oplus \underline{a}(1\oplus \mathrm{bef}\oplus \mathrm{bcde}\oplus \bar{b}\bar{c}\mathrm{de}).$$

$$\bar{b}\bar{e}\mathrm{f}\oplus \underline{a}(1\oplus \underline{b}(\mathrm{ef}\oplus \mathrm{cde}\oplus \bar{c}\mathrm{de})).$$

$$\bar{b}\bar{e}\mathrm{f}\oplus \underline{a}(1\oplus \underline{b}(\mathrm{ef}\oplus \underline{c}(\mathrm{de}\oplus \mathrm{de}))).$$

Thus, because of repeated term *de*, an IF that represents the given minimum ESOP cannot be found for *abcdef* order and thus for an arbitrary order of variables. However if we choose variable ordering (abdcef) the S/D can be generated.

$$\bar{b}\bar{e}\mathrm{f}\oplus \underline{a}(1\oplus \underline{b}(\mathrm{ef}\oplus \underline{d}(\mathrm{ce}\oplus \bar{c}\mathrm{e}))).$$

$$\bar{b}\bar{e}\mathrm{f}\oplus \underline{a}(1\oplus \underline{b}(\mathrm{ef}\oplus \underline{d}(\mathrm{c(e)}\oplus \bar{c}\mathrm{(e)}))).$$

The remaining part of building the S/D tree is obvious.

Definition 9 Terms that are defined on the same set of variables and include the same set of literals, of cardinality at least one, are called *equal-variable terms* or (*ev terms*).

For instance, terms *abcde* and $\bar{a}\bar{b}\bar{c}de$ in Example 4 are ev-terms with variables $\{a, b, c, d, e\}$ and the same literal set $\{de\}$

Definition 10 The *distance* of two ev-terms is the number of variables for which the corresponding literals of these terms have different polarities.

Lemma 1 In a (single-output) minimum ESOP, a distance between any ev-terms terms has to be at least three.

Proof If a distance between any ev-terms is smaller than three, ev-terms can be substituted with two terms (not ev-terms any more) with a number of literals smaller by two. □

Example 5

$$\bar{a}\mathrm{bcd}\oplus \mathrm{a}\bar{b}\mathrm{cd} = \mathrm{acd}\oplus \mathrm{bcd}$$

Ev-terms of distance 2 were replaced with two terms that are no longer ev-terms.

Lemma 2 A number of identical terms in a sub-function on any level of the S/D tree, before the expansion process becomes infeasible, cannot be larger than two regardless of a chosen variable order.

Proof If a number of identical terms on a level is larger than two it means that if we move back one level (add one more variable), we will still have at least two identical terms which should have been noticed on the previous level. □

Based on Lemma 2 we conclude that Example 4 exhausts all possible cases for a single set of literals, which leads to Lemma 3.

Lemma 3 Situation such as in Example 4 cannot happen when the order of variables used in the subsequent expansions is not predetermined, but appropriately selected.

Now we are able to formulate the main theorem.

Theorem 5 *GIF family of forms includes a minimum ESOP for an arbitrary Boolean function*.

We have thus to prove that if function F is a minimum ESOP, there exists a S/D tree that generates F. Consequently, we have to prove that starting from a minimum ESOP we can always find a variable order such that a S/D tree, generating this minimum ESOP, can be build.

The proof will be based on Lemma 4 and Lemma 5.

Lemma 4 S/D tree, for an ESOP expression, can be build if no identical terms are created at any stage of the decomposition process (S/D Building Procedure).

Proof For simplification of the proof we will discuss only one of the sub-functions, represented as in expression (6), created during the decomposition process. It can be shown that the same reasoning applies to all of them but the proof would become more complicated.

Let us assume that after several levels of decomposition one of sub-functions on level k is as given below.

$$1\oplus \mathrm{c(de\ldots)}\oplus \bar{c}\mathrm{(de\ldots)} \tag{7}$$

If we choose c as the next decomposition variable, we need to use generalized Davio expansion because all three sets are non-empty and we will create two identical terms as shown in Eq. (8).

$$1\oplus\underline{c}((\mathrm{de}\ldots)\oplus(\mathrm{de}\ldots)) \tag{8}$$

Thus, two identical terms exist in a sub-function. This is infeasible, because no tree expansion can generate two identical cubes in one sub-function. However, if instead of variable *c*, for decomposition, we first choose one of the variables that belong to the set of identical literals, the S/D tree can be generated as shown below.

$$1\oplus\underline{d}(\mathrm{c}(\mathrm{e}\ldots)\oplus\bar{c}(\mathrm{e}\ldots.)) \tag{9}$$

$$1\oplus\underline{d}(\mathrm{c}((\mathrm{e}\ldots))\oplus\bar{c}((\mathrm{e}\ldots))) \tag{10}$$

In general, the number of literals in the subset of identical literals (no two literals can be generated from the same variable), in two ev-terms under consideration, can be arbitrary. The existence of the situation as in Eq. (7) means that at the previous decomposition level we made a bad choice (variable *c*) for the decomposition variable. A variable present in two different polarities in the twins needs to be decomposed as the last one of the group. We can also say that the variables that belong to the common set of literals have to be decomposed first in the order. If we extrapolate expression (7) to the previous level $k-1$, and assume that the variable *b* on that level is the one which distinguishes the ev-terms, the expression (7) will have a form as given below:

$$\underline{b}\oplus\mathrm{bcde}\oplus\bar{b}\bar{c}\mathrm{de} \tag{11}$$

So a variable order {*d* and *e* in any order} and next {*b* and $\bar{c}$ in any order} is feasible (Lemma 4). □

In general case we can have many groups of two ev-terms that are defined on different sets of variables. If these sets of variables do not overlap we deal with each set separately. If they overlap, we will show in Lemma 5, that there is always a way to choose a decomposition variable to avoid the problem.

LEMMA 5 S/D tree, for an ESOP expression, can be build if no identical terms are created at any stage of the decomposition process (S/D Building Procedure)

Proof Let us assume the worst case; a variable on level $k-1$ was chosen such that on level *k* the remaining set of variables to be used for decomposition is such that if any one, from the set (cyclic set), is chosen then two identical terms in one branch are created. It means that the distance between any terms at that level is only one. It means that for these ev-terms to belong to the minimum solution at least two additional variables (the most difficult case that covers all cases with more than two variables) are needed to distinguish them. It is obvious that these variables had been used already on two of the previous levels. Let us also assume that these levels were $k-1$ and $k-2$, and the variables are x_{k-1} and x_{k-2}.

So, on level $k-2$ all the terms differ in three variable positions (distance three), which property does not allow for a reduction (a minimum ESOP). Therefore, at this level any two terms, which have the same set of literals are distinguished by the polarity of variables and x_{k-k} and x_{k-2}. So, if one of variables x_{k-1} or x_{k-2} is placed behind all variables from the cyclic set, no identical terms will appear and the next level of the S/D tree can be created (Lemma 5). □

Example 6 At some level *k* of the tree the sub-function contains many pairs of terms that differ only in one variable (distance 1). It is also the case that none of the variables in the set is a good choice. For example:

$$1\oplus(\bar{c}\mathrm{de}\oplus\mathrm{cde})\oplus(\mathrm{bcd}\oplus\bar{b}\mathrm{cd})\oplus(\mathrm{bc}\bar{e}\oplus\mathrm{bce})\oplus(\mathrm{b}\bar{d}\mathrm{e}\oplus\mathrm{bde})$$

In this sub-function for any of the variables only the generalized Davio expansion can be used. Regardless which of the four variables (b, c, d, e) (cyclic set) is selected, we end up with two identical terms in the sub-function on the next, $k+1$, level.

However, the terms separated by the distance of one on level *k* need to have at least two additional variables to separate them by distance of three so they belong to the minimum ESOP. They have to be the same variables for both terms because they appear in the same sub-function. Let us assume that the variable *a*, used on level $k-1$, separates all these terms to the distance 2.

So in our example:

$$\mathrm{term}(b,c,d,e)\ \oplus\underline{a}\oplus(\mathrm{a}\bar{c}\mathrm{de}\oplus\bar{a}\mathrm{cde})\oplus(\mathrm{abcd}\oplus\bar{a}\bar{b}\mathrm{cd})$$
$$\oplus(\mathrm{abc}\bar{e}\oplus\bar{a}\mathrm{bce})\oplus(\mathrm{ab}\bar{d}\mathrm{e}\oplus\bar{a}\mathrm{bde})$$

where term (*b*,*c*,*d*,*e*) contains all the terms from the 1-Set on level $k-1$. Obviously, there are no identical terms in this expression, as they should have been noticed on the previous level. For the same reason there are no distance-one pairs in the entire sub-function on level $k-1$. Now, we select decomposition variables in such order that variable *a* is the last one and we can create the S/D tree for that expression.

Let us now assume that, for a Boolean function on *n* variables, it is possible to create a minimum ESOP expression with the number of ev-terms such that no variable order exists for which a S/D tree can be created. Let us count a number of terms that needs to exist in such minimum ESOP expression. We need to recall here that a distance between any ev-terms needs to be at least three. For such a expression to exist it needs to contain at least *n* ev-terms, *n* single variable cubes and the combinations of all possible two-variable cubes to assure that on all levels of decomposition the generalized Davio expansion is used. Only generalized Davio expansion can produce identical terms. So, the number of terms in such expression is proportional to *n*!, which is much larger than 2^{n-1} that is larger than an upper bound on the number

TABLE II The number of canonical forms and IFs depending on the number of variables

Vars	#all	#can	#if	#gif	all/can	all/if	can/if	all/gif	can/gif
1	3*	3	3	3	1	1	1	1	1
2	126*	81	45	65	1.56	2.80	1.80	1.94	1.25
3	2,220,075*	527,121	34,425	109,361	4.21	64.5	15.3	20.3	4.8
4	100,000,000	1,037,459	175	1583	96.4	5.7×10^5	5.9×10^3	6.3×10^4	6.6×10^2
5	100,000,000	108,044	0	1	925	$>1.0 \times 10^8$	$>1.0 \times 10^5$	$>1.0 \times 10^8$	$>1.0 \times 10^5$

Notation used in the table: Vars is the number of variables in the expansions. #all is the number of generated expansions (asterisk* means that for this number of variables the program exhaustively generated all expansions). #can, #if and #gif are numbers of canonical, IF and GIFs, respectively, among these (randomly) generated by the program. In the other columns, the ratios all/can, all/if, can/if, all/gif, can/gif are given.

of terms in the minimum ESOP. Therefore, such expression is not a minimum ESOP.

So, we showed, that the order of variables can be always found for an expression that does not lead to identical terms, if this expression is a minimum ESOP. Therefore, the GIFs include minimum ESOPs, and this completes the proof of Theorem 5. □

EXPERIMENTAL RESULTS

Theorem 5 proved in the previous section facilitates creating algorithms for exact ESOP minimization by substantially reducing the search space for the exact solution. To study this property, we conducted a computer experiment. In the course of this experiment, we generated random expansions for each number of variables, checked whether this expansion is linearly independent (canonical), and next checked whether it is possible to create the S/D tree for the first variable ordering $(a_1, a_2, \ldots a_n)$. The results are given in Table II. Please observe that in column "#all" the number of functions reported is equal to all possible functions on the given number of variables only if asterisk appears next to the number. For all others the given numbers of functions were generated randomly. We did not generate results for GIFs with more than three variables.

This table allows us to observe two properties of canonical expansions. As the number of variables grows, the percentage of linearly independent (canonical) forms significantly decreases. Still more dramatic change is observed in the percentage of all possible (and canonical) expansions with respect to GIF and IFs; while for two variables there is only 1.8 more canonical forms than IFs, for five variables it is at least 10^5 more canonical forms than IFs. The experiment proves a remarkable property of GIFs. They allow us to restrict the search space for minimum ESOPs.

CONCLUSIONS

In this paper we reviewed the hierarchy of known families of canonical forms described in Refs. [7–9] and introduced two new families of forms. We presented a number of properties of IFs, as well as proved their canonicity and uniqueness. We proposed a generalization of IFs, called GIFs, created as a union of IFs for all orders of the given number of variables. We derived the formula for the exact number of IFs as a function over the number of variables and showed that the ratio of the quantity of IFs to the quantity of all canonical forms decreases exponentially over the number of variables. Most importantly, we proved that a minimum ESOP is included in the GIF family. We believe that GIFs will find application in the exact ESOP minimization because they will allow to reduce significantly the search space.

Another important result of our paper is that any search for larger canonical families of Green/Sasao hierarchy loses its potential importance since it will not help to find a minimum ESOP more efficiently. Further research should therefore concentrate on investigating GIFs properties that would help create structured search algorithms for a minimum ESOP.

Acknowledgements

This work was supported in part by the NSF grant MIP-9629419.

References

[1] Perkowski, M., Jozwiak, L. and Drechsler, R. (1997) "A canonical AND/EXOR form that includes both the generalized Reed-Muller forms and Kronecker Reed-Muller forms", Proc. RM'97 (Oxford University, UK), Sept. 1997, pp. 219–233.
[2] Perkowski, M., Jozwiak, L. and Drechsler, R. (1997) "Two hierarchies of generalized Kronecker trees, forms, decision diagrams, and regular layouts", Proc. RM'97 (Oxford University, UK), Sept. 1997, pp. 115–132.
[3] Reddy, S.M. (1972) "Easily testable realizations of logic functions", *IEEE Trans. Comp.* **C-11**, 1083–1088.
[4] Fujiwara, H. (1985) Logic Testing and Design for Testability (MIT Press, Cambridge, MA).
[5] Perkowski, M. (1995) "A fundamental theorem for Exor circuits," *Proc. Reed-Muller'95*, pp. 52–60.
[6] Perkowski, M., Sarabi, A. and Beyl, F. (1995) "Fundamental theorems and families of forms for binary and multiple-valued linearly independent logic," *Proc. Reed-Muller'95*, pp. 288–299.
[7] Green, D.H. (1991) "Families of Reed-Muller canonical forms", *Int. J. Electr.* **2**, 259–280.
[8] Sasao, T. (1995) "Representation of logic functions using EXOR operators." *Proc. Reed-Muller'95*, pp. 11–20.
[9] Perkowski, M.A. (1992) "The generalised orthonormal expansion of functions with multiple-valued inputs and some of its applications." *Proc. ISMVL'92*, pp. 442–450.
[10] Sasao, T. (1993) "An exact minimization of AND–EXOR expressions using BDDs." *Proc. Reed-Muller'93*, pp. 91–98.

[11] Sasao, T. and Izuhara, F. (1996) “Exact minimization of FPRMs using multi-terminal EXOR TDDs”, In: Sasao, T., ed, Representations of Discrete Functions (Kluwer, Dordrecht), pp 191–210.

[12] Escobar, M. and Somenzi, F. (1995) “Synthesis of AND/EXOR expressions via satisfiability.” *Proc. of Reed-Muller’95* pp. 80–87.

[13] Song, N. and Perkowski, M. (1993) “;EXORCISM-MV-2: minimization of exclusive sum of products expressions for multiple-valued input incompletely specified functions”, *Proc. ISMVL’93* **May**, 132–137.

[14] Zeng, X., Perkowski, M., Dill, K. and Sarabi, A. (1995) “Approximate minimization of generalized Reed-Muller forms,” *Proc. Reed-Muller’95*, pp. 221–230.

[15] Sasao, T. (1993) “EXMIN2: a simplification algorithm for exclusive-OR-sum-of-products expressions for multiple-valued input two-valued output functions”, *IEEE Trans. CAD Int. Circuits Syst.* **12**(5), 621–632.

[16] Perkowski, M. and Chrzanowska-Jeske, M. (1990) “An exact algorithm to minimize mixed-radix exclusive sums of products for incompletely specified Boolean functions,” *Proc. IEEE ISCAS’90*, International Symposium on Circuits and Systems, pp. 1652–1655.

Authors’ Biographies

Malgorzata Chrzanowska-Jeske received her M.S. in Electrical Engineering degree from the Technical University of Warsaw, Warsaw, Poland, and Ph.D degree in Electrical Engineering from Auburn University, Auburn, Alabama. She has served on the faculty of the Technical University of Warsaw, Poland, and as a design automation specialist at the Research and Production Center of Semiconductor Devices, Warsaw. Currently, she is a Professor of Electrical and Computer Engineering at Portland State University in Portland, Oregon. Her research interests include logic and layout synthesis of VLSI circuits and systems, FPGA synthesis and architecture, and design automation and testing for deep sub-micron technology. She is a member of IEEE Circuits and System Society VLSI Systems and Applications Technical Committee, a senior member of the IEEE, and a member of Eta Kappa Nu.

Alan Mishchenko received M.S. (1993) in Applied Mathematics from Moscow Institute of Physics and Technology, Moscow, Russia, and Ph.D. (1997) in Computer Science from Glushkov Institute of Cybernetics, Kiev, Ukraine. He served as a junior research worker at Glushkov Institute of Cybernetics. Currently, he is a visiting scientist at Portland State University, Portland, Oregon. His research area is hardware design and CAD tools. His research interests include applying Binary Decision Diagrams (BDDs) and BDD-based methods to computationally hard problems in logic synthesis and verification. He is a member of IEEE and IEEE Computer Society.

Marek Perkowski has his Ph.D. in Automatic Control from the Technical University of Warsaw, Warsaw, Poland. He served on the faculty of Technical University of Warsaw and University of Minnesota. Currently he is a Professor of Electrical and Computer Engineering at Portland State University, Portland, Oregon. He has been a visiting Professor at the University of Montpellier, France, and Technical University of Eindhoven, The Netherlands. He was also a summer Professor and consultant at Wright Laboratories of U.S. Air Force, GTE, Intel, Cypress, Sharp, and other high technology and EDA companies. Dr Perkowski was the general chair of International Symposium on Multiple-Valued Logic, 2000, the Vice-Chair for Technical Activities of IEEE Technical Committee on MVL, and the Chair of Fourth Oregon Symposium on Logic, Design and Learning, 2001. His recent research interests include spectral decision diagrams, functional decomposition, intelligent robotics, walking robots, robot theatre, axiomatic morality, and all applications of logic synthesis outside circuit design.