How to Catch the Golden Fish of Minimum ESOP Into the Net of Canonical Forms

Malgorzata Chrzanowska-Jeske, Marek Perkowski, and Alan Mishchenko

Portland State University, Department of Electrical and Computer Engineering
Portland, OR 97207, Tel: 503-725-5411, Fax: 503-725-4882
[jeske, mperkows, alanmi]@ee.pdx.edu

Abstract

This paper describes a family of canonical Reed-Muller forms, called Inclusive Forms, which allows to represent all minimum ESOPs for any boolean function. We outline the hierarchy of known canonical forms, in particular, Pseudo-Generalized Kronecker Forms [1, 2], which led us to the discovery of the new family. Next we introduce special binary trees, called S/D trees, which underlie Inclusive Forms and permit their enumeration. We show how to generate IFs and Generalized IFs. Finally, we present the results of computer experiments, which show that Inclusive Forms reduce the search space for minimum ESOP by several orders of magnitude, and this reduction grows exponentially with the number of variables.

1 Introduction

Reed-Muller (AND/EXOR) expansions play an important role in logic synthesis by producing economical and highly-testable implementations of boolean functions [3, 4, 5, 6]. The range of Reed-Muller expansions includes 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 (PSDKROs), referred to as the Green/Sasao hierarchy, have been described [7, 8, 9]. (See Figure 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, 11, 12, 13]. These families, however, do not exhaust all ESOPs. This is why state-of-the-art ESOP minimizers [14, 15, 16] are based on heuristics and give the exact solution only for functions with a small number of variables.

Recently, new general families of canonical forms have been proposed [1, 2], which include the above-mentioned well-known families, in particular GRMs and PSDKROs. The discovery of these forms suggests future advances in exact ESOP minimization. Still none of these families has been proven powerful enough to include all minimum ESOPs, or a subset of them.

In this paper, we propose two still more general families of canonical Reed-Muller forms, called Inclusive Forms and Generalized Inclusive Forms, which include all minimum ESOPs.

The remainder of this paper is organized as follows. The basic definition of the families of forms belonging to the Green/Sasao hierarchy and their recent generalizations [1, 2] are given in Section 2. The concept of S/D trees, which is essential for creation and enumeration of Inclusive Forms, is presented in Section 3. Properties of Inclusive Forms and the formula to calculate their quantity is given in Section 4 and illustrated by comprehensive enumeration of IFs for two variables. Section 5 is devoted to possible generalizations of IFs. The application of the new forms to exact logic minimization is discussed in Section 6. Experimental results are presented in Section 7, followed by conclusions in Section 8.

2 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)

$$f(x_1, x_2, \ldots, x_n) = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus \ldots \oplus a_nx_n \oplus$$
$$a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus \ldots \oplus a_{n-1,n}x_{n-1}x_n \oplus \ldots \oplus a_{12\ldots n}x_1x_2\ldots x_n. \qquad (4)$$

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 ($\bar{x}_i$) for each variable in (4), we get the *Fixed Polarity Reed-Muller form* (FPRM). There are 2^n possible combinations of polarities and as many FPRMs for any given logic function.

If we freely choose the polarity of each literal in (4), we get a *Generalized Reed-Muller form* (GRM). In GRMs, contrary to FPRMs, the same variable can appear in both positive and negative polarities. There are $n2^{n-1}$ literals in (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 of certain binary trees. To create these trees, the following procedure is proposed. Let us create a binary tree in such a way that each k-th level ($0 \le k < n$), starting from the root node on top of the tree, contains 2^k nodes. There are $1+2+\ldots+ 2^{n-1} = 2^n-1$ 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 *Kronecker form* (KRO). If an arbitrary expansion is allowed in each node, the result is the *Pseudo-Kronecker form* (PKRO). There are $3^{\boldsymbol{n}}$ and $3^{2^{\boldsymbol{n}}-1}$ different KROs and PKROs [3], respectively. These families intersect with GRMs but do not contain them (Figure 1). An example of a Pseudo-Kronecker tree and the resulting canonical form are given in Figure 2.

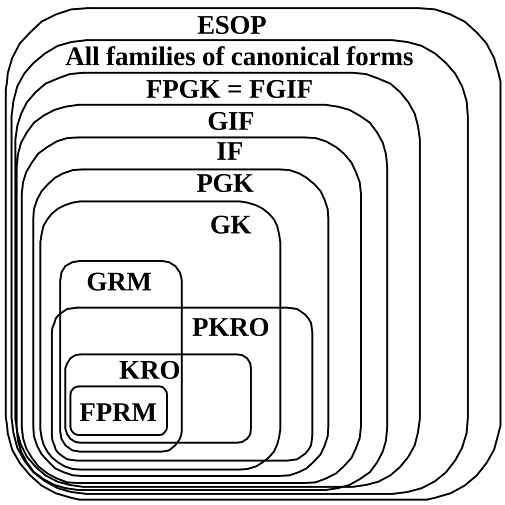

Figure 1. Set-theoretic relationship between families of canonical forms.

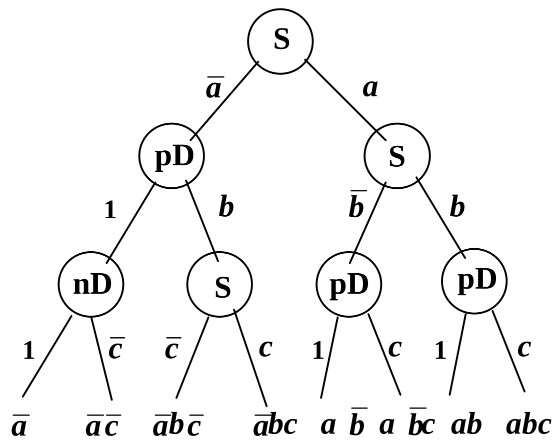

Figure 2. A Pseudo-Kronecker tree and canonical expansion it produces

In [1, 2], three more families of canonical expansions were proposed. These forms are generated by flattening certain type of trees. The following procedure for building the tree is proposed. First, partition all $\boldsymbol{n}$ variables into disjoint non-empty sets $\boldsymbol{S_j}$, 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 Kronocker forms* (GKs) [1].

Definition 2. If we allow any of the generic expansions (1), (2), and (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 *Pseudo-Generalized Kronecker forms* (PGKs) [1].

Definition 3. If we additionally allow any variable ordering to be used along any path in the tree, provided that variables are not repeated, it is *Free Pseudo-Generalized Kronecker forms* (FPGKs) [1].

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.

3 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 Inclusive Forms.

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

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

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 at the same time. 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.

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 4. 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 $\boldsymbol{n}$ variables has $2^{\boldsymbol{n}}$-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 5. *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 $\boldsymbol{n}2^{\boldsymbol{n}-1}$ generalized literals only (when gD nodes are used throughout the tree). It is easy to see that is the latter case, the GE produces all GRMs for the given number of variables.

Definition 6. *Inclusive Forms* (IFs) for a given variable ordering is a set of expansions created by all generalized expansions for this variable order, when generalized literals presented in the generalized expansions are allowed to have all possible combinations of polarities.

It is easy to see that a generalized expansion with $\boldsymbol{m}$ generalized literals produces as many ordinary forms as there are distinct polarity assignments of generalized literals, namely $2^{\boldsymbol{m}}$.

Example 1. Figure 3 illustrates derivation of Inclusive Forms 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 corresponsing 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 $\{\bar{a}\,\bar{b},\ \bar{a}\,b,\ a,\ a\,b\}$. By adding numbers N for each tree, we get the total number of IFs for n = 2.

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

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

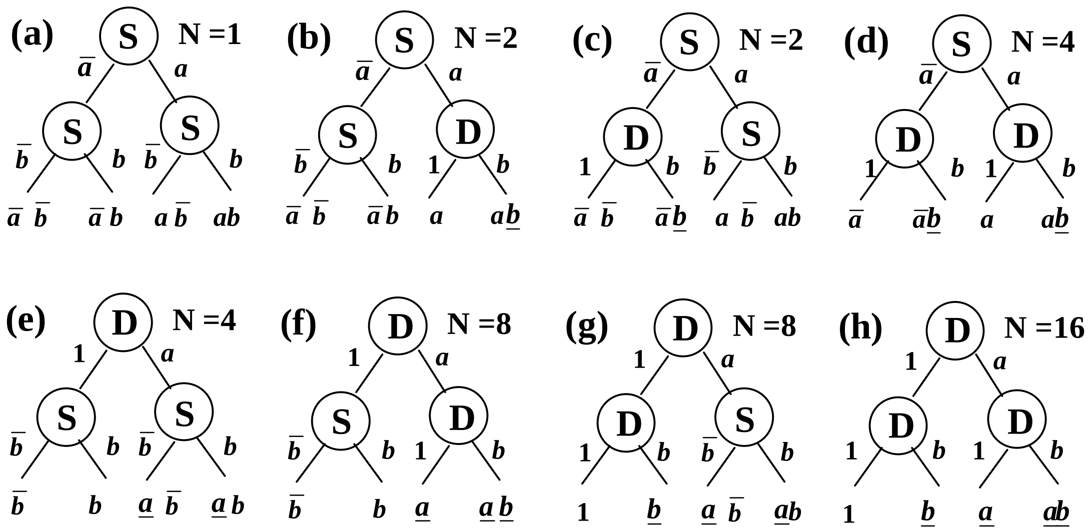

Figure 3. All S/D trees and generalized expansion for two variables.

4 Properties of Inclusive Forms

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

Theorem 1. Each Inclusive Form $\{t_i\}$, $1 \le i \le n$, 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 $F = a_1t_1 \oplus \ldots \oplus a_nt_n$.

Proof: In [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 prove this statement by induction on the number of variables. For n =1, there are only three IF forms, which coincide with the generic Shannon and Davio expansions, introduced in Section 3. 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 ($a_0, a_1, \ldots, a_k$) 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 group, 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, which 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 $\boldsymbol{t_j}$ and all the terms that constitute the linear combination belong to group $\boldsymbol{G_1}$, it is a contradiction. If they belong to group $\boldsymbol{G_2}$, again all of them can belong to either those terms which have $\boldsymbol{a_k}$ complemented, or to those terms that have $\boldsymbol{a_k}$ uncomplemented. We repeat our previous argument for Shannon expansion and arrive at a contradiction. **Q.E.D.**

Theorem 2. For the given ordering of n variables, there are

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

unique Inclusive Forms.

Proof: First, let us prove that the forms are unique, that is 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 $\boldsymbol{n} = 1$, there are only three possible forms and they are unique. Suppose it is true for $\boldsymbol{n = k}$. Let us prove that it is true for $\boldsymbol{n = k+1}$.

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 $\boldsymbol{n = k}$, these expansions may differ only in the variable $\boldsymbol{a_k}$, which is found on top of the S/D tree. But there are only two distinct S/D trees produced by the variable $\boldsymbol{a_k}$, 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 Inclusive Forms.

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 $\boldsymbol{k}$-th level of an S/D tree. If it is a Shannon node, it does not contribute generalized literals to the generalized expansion produced by the tree and does not produce more than one resulting canonical expansions. If it is a Generalized Davio node, it contributes 2^{n-k-1} generalized literals to the generalized expansion, which, in turn, produce $2^{2^{n-k-1}}$ resulting canonical expansions.

Now we observe that there $\boldsymbol{k}$-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 $\boldsymbol{k}$-th level of nodes for all S/D trees, which differ only in polarity assignments. This contribution is $\left(1+2^{2^{n-k-1}}\right)^{2^k}$. The only thing left to do after this, is to create the product of these contributions, since each level adds to the sum total of expansions independently of all others. **Q.E.D.**

Example 2. For n = 3, there are $\binom{3^n}{2^n} = \binom{27}{8}$ = 2,220,075 possible expansions. Among them, only 527,121 are linearly independent, or canonical. According to the formula (1), there are N_{IF} = (1 + 16)1(1 + 4)2(1 + 2)4 = 34,425 Inclusive Forms for each ordering of variables. We have verified these results using a program, which systematically generates all linearly independent forms for three variables and checks whether it is possible to create an S/D tree for a given variable ordering.

5 Generalizations of Inclusive Forms

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

Definition. The family of forms, which is created as a union of sets of IFs for all variable orders, is called *Generalized Inclusive Forms* (GIFs).

If in the later definition 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 get a still more general family of canonical forms.

Definition. The family of forms, generated by the S/D tree with no fixed ordering or variables, provided that variables are not repeated along the same branches, is called *Free Generalized Inclusive Forms* (FGIFs).

It can be shown that FGIFs is the same of FPGK forms, intoduced in Section 2.

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 Figure 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 for the number of GIFs we have the following calculation:

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

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. In Table 1, we give the result of a computer experiment, which shows that for n = 3 this relationship becomes rather complicated. The total number of GIFs for n = 3 is given in the last row of the table.

Table 1. The number of Inclusive Forms 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	4,913
Total GIFs	109,361

6 Applications of Inclusive Forms to exact logic minimization

In this section, we provide an algorithm to represent any minimum ESOP as an Inclusive Form.

Algorithm. Given a variable ordering and a minumum ESOP, build the S/D tree which generates a canonical form representing the minimum ESOP.

Step 1. Assume given variable is the first variable in the ordered list of variables.

Step 2. Divide all terms of the expansion into three sets: those that do not contain the given variable, those that contain it as a complemented literal, and those that contain it as a non-complemented literal.

Step 3. (a) If the first set is empty, assume the Shannon expansion on the given variable, factorize it from the two remaining sets, take the next variable from the list, assume two expansions are the second and third sets of terms (without the factorized variable) and go to Step 2 for both expansions.

(b) If the first set is not empty, but either second or third or both of these sets are empty, assume the Generalized Davio expansion and then proceed as in Step 3 (a).

(c) If neither of the sets is empty, assume Generalized Davio expansion, factorize the given variable from the second and third sets as a generalized literal, take the next variable from the list of ordered variables, and check for identical terms in the set of terms after factorization (see Example 4 for the case when such terms exist). If such terms exist and include the next variable taken from the list, introduce as many new variables instead of this one as needed to make all of the terms different. (There will be ***s*** - 1 such variables, if ***s*** is the number of repeated terms.) Insert these variables into the list after the current one. Assume two expansions are the first set of terms and the set of terms after factorization at the beginning of Step 3 (c). Go to Step 2 for both expansions.

If there are no more variables in the list, exit the algorithm.

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

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

It is easy to see that this is the minimum ESOP, because the Humming distance between any pair of cubes is three or more. First, we perform Generalized Davio expansion on variables a, b and c:

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

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

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

Next we introduce variable d_1 and do Generalized Davio expansion on variable d.

$$\bar{b}\,\bar{e}\,f \oplus \underline{a}\,(1 \oplus \underline{\mathbf{b}}\,(ef \oplus \underline{c}\,(de \oplus d_1e))).$$

$$\bar{b}\,\bar{e}\,f \oplus \underline{a}\,(1 \oplus \underline{\mathbf{b}}\,(ef \oplus \underline{c}\,(d_1e \oplus \underline{\mathbf{d}}\,e))).$$

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

Computer experiments show that, for the majority of functions, it is possible to create the S/D tree for any variable ordering, which generates a canonical form representing the minimum ESOP without introducing additional variables, as it was done in Example 4. Only 5% (???) of randomly generated functions and 2% (???) of MCNC benchmarks require additional variables in order to represent their minimum ESOP as an S/D tree. Statistical results for the functions that require additional variables are given in Table 2.

Table 2. The number of additional variables to be introduced to represent the minimum ESOP by an Inclusive Form

Function	# vars	# terms	# add vars
Random 1			
Random 2			
Random 3			
Benchmark 1			
Benchmark 2			
Benchmark 3			

Notation used in the table: **Function** is the name of the random (benchmark) function. **# vars** is the number of variables in this functions. **# terms** is the number of terms in the minimum ESOP expression. **# add vars** is the number of additional variables needed to represent the function as the S/D tree.

7 Experimental Results

Theorem 3 proved in the previous section facilitates creating algorithms of ESOP minimization by substantially reducing the search space for the exact solution. To study this property, we conducted a computer experiment. In the course 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,… a_n). The results are given in the Table 3.

Table 3. The number of canonical forms and Inclusive Forms depending on the number of variables

# vars	# all	# canon	# if	#all/#canon	#all/#if	#canon/#if
1	3*	3	3	1	1	1
2	126*	81	45	1.56	2.80	1.80
3	2,220,075*	527,121	34,425	4.21	64.5	15.3
4	100,000,000	1,037,459	175	96.4	$5.7*10^5$	$5.9*10^3$
5	100,000,000	108,044	0	925	$>1.0*10^8$	$>1.0*10^5$

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). **# canon** and **# if** are numbers of canonical and inclusive forms, respectively, among these (randomly) generated by the program. In the other columns, the ratios **#all/#canon, #all/#if, #canon/#if** are given.

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 decrease is observed in the percentage of Inclusive Forms with respect to all possible (and canonical) expansions. The experiment proves a remarkable property of IFs. They allow us to restrict the search space for minimum ESOPs. In a sense, IFs are similar to a net into which one may try to catch the golden fish of minimum ESOP.

8 Conclusions

In this paper we review the hierarchy of known families of canonical forms described in [3,4,5] and introduce a new family of forms, which includes all minimum ESOPs. We present a number of properties of Inclusive Forms, as well as prove their canonicity and uniqueness. We propose a generalization of IFs, called Generalized Inclusive Forms and created as a union of IFs for all orders of the given number of variables. We derive the formula for the exact number of IFs as a function over the number of variables and show that the ratio of the quantity of Inclusive Forms to the quantity of all canonical forms decreases exponentially over the number of variables. We believe that Inclusive Forms will find application in exact ESOP minimization.

9 References

1. M. Perkowski, L. Jozwiak, and R. Drechsler, "A Canonical AND/EXOR Form that includes both the Generalized Reed-Muller Forms and Kronecker Reed-Muller Forms," *Proc. RM'97*, Oxford Univ. U.K., Sept. 1997, pp. 219-233.

2. M. Perkowski, L. Jozwiak, and R. Drechsler, "Two Hierarchies of Generalized Kronecker Trees, Forms, Decision Diagrams, and Regular Layouts," *Proc. RM'97*, Oxford Univ. U.K., Sept. 1997, pp. 115-132.

3. S. M. Reddy, "Easily Testable Realizations of Logic Functions", *IEEE Trans. on Comp.*, C-11, pp. 1083-1088, 1972..

4. H. Fujiwara, *Logic Testing and Design for Testability*, The MIT Press, 1985.

5. M. Perkowski, "A Fundamental Theorem for Exor Circuits," *Proc. Reed-Muller'93*, pp. 52-60

6. M. Perkowski, A.Sarabi, and F. Beyl, "Fundamental Theorems and Families of Forms for Binary and Multiple-Valued Linearly Independent Logic," *Proc. Reed-Muller'95*, pp. 288-299.

7. D.H. Green, "Families of Reed-Muller Canonical Forms," *Intern. Journal. of Electr*., pp. 259-280, Febr. 1991, No 2.

8. T. Sasao, "Representation of Logic Functions using EXOR Operators," *Proc. Reed-Muller '95*, pp. 11-20.

9. M.A. Perkowski, "The Generalised Orthonormal Expansion of Functions with Multiple-Valued Inputs and Some of its Applications," *Proc. ISMVL'92*, pp. 442-450.

10. M. Perkowski, and M. Chrzanowska-Jeske, "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, New Orleans, 1-3 May 1990..

11. T. Sasao, "An Exact Minimisation of AND-EXOR Expressions Using BDDs," *Proc. Reed-Muller'93*, pp. 91-98.

12. T.Sasao, F. Izuhara, "Exact Minimization of FPRMs Using Multi-Terminal EXOR TDDs", In *Representations of Discrete Functions*, T. Sasao (editor), Kluwer 1996, pp. 191-210.

13. M.Escobar, F. Somenzi, "Synthesis of AND/EXOR Exressions via Satisfiability ", *Proc. of Reed-Muller '95*, pp. 80-87.

14. N. Song, and M. Perkowski, "EXORCISM-MV-2: Minimisation of Exclusive Sum of Products Expressions for Multiple-Valued Input Incompletely Specified Functions," Proc. ISMVL'93, May 24, 1993, pp. 132-137.

15. X. Zeng, M. Perkowski, K. Dill, and A. Sarabi, "Approximate Minimization of Generalized Reed-Muller Forms," Proc. Reed-Muller'95, pp. 221-230.

16. T.Sasao, 'EXMIN2: A simplification algorithm for exclusive-OR-Sum-of-products expressions for multiple-valued input two-valued output functions", IEEE Trans. on CAD of Int. Circuits and Systems, Vol. 12, No. 5, May 1993, pp. 621-632.