A CANONICAL AND/EXOR FORM THAT INCLUDES BOTH THE GENERALIZED REED-MULLER FORMS AND KRONECKER FORMS

Marek Perkowski, Lech Jozwiak †, Rolf Drechsler +

Portland State University,
Department of Electrical Engineering, Portland, Oregon, USA,
† Faculty of Electrical Engineering,
Eindhoven University of Technology, Eindhoven, The Netherlands,
+ Institute of Computer Science, Albert-Ludwigs-University,
Freiburg in Breisgau, Germany.

USE OF CANONICAL AND/EXOR REPRESENTATIONS IN SYNTHESIS

- Canonical families of AND/EXOR representations can be derived from **binary AND/EXOR decision diagrams.**
- Canonical AND/EXOR representations are the highest testable circuits.
- AND/EXOR circuit is created in the first phase of synthesis.
- Then, it is mapped to an arbitrary technology using AND, OR, EXOR and other gates.
- Circuits obtained this way are **highly testable** (Tsai/Sadowska).
- The trade-offs between the implementation cost and the testability are investigated and utilised to satisfy **various requirements** for a given technology.

OPEN PROBLEM

- An open problem was posed by Tsutomu Sasao - *"whether there exists some family, or families, of canonical forms that is a superset of both KRO and GRM families?"*.
- Such family would be interesting because presumably it would be close in cost to ESOP and still have good testability properties similar to GRM.
- In this paper we will answer positively to this question and we will develop a new representation.

MAIN IDEAS

- We call the new forms the **Generalized Kronecker Forms.**
- We present the **corresponding decision diagrams**.
- They can be used for representation of large functions and also for optimal synthesis of highly testable two-level and multilevel circuits in several technologies, including Field Programmable Gate Arrays (FPGAs).
- The new representations introduced here can be used for synthesis with **a wider category of gates**.
- This new representation can be used in the first stage of the proposed logic synthesis process - the *"technology independent, EXOR synthesis"* phase, followed by the *"EXOR-related technology mapping"* (Marek-Sadowska, Sasao).

MAIN IDEAS. I

- Because the class of circuits covered by the new representation includes all the AND/EXOR circuits obtained from previous canonical representations, the new representation should be superior in terms of:
  - the two-level realization areas for forms,
  - the diagrams' complexity,
  - the complexity of multilevel circuits corresponding to the diagrams after technology mapping.

PLAN OF PRESENTATION

- Preliminaries on EXOR type decision trees and corresponding canonical forms.
- Canonical Expansions for many variables,
- GRM Universal Logic Module (node)
- Computation of the expansion data functions for such multi-variable nodes in the process of tree creation.
- Generalized Kronecker Tree Decision Diagrams and Forms.
- Quality of the new forms
- Applications to exact ESOP minimization.
- Conclusions and future work.

AND/EXOR CANONICAL TREES WITH SINGLE-VARIABLE EXPANSION NODES.

The first fundamental idea of generalizing **the functions** realised by the nodes of the tree comes from Davio, who introduced the following three fundamental expansions realised by nodes:

$f(x_1, x_2, ..., x_n) \;=\; \overline{x_1} f_0(x_2, ..., x_n) \oplus x_1 f_1(x_2, ..., x_n)$

in short $\quad f \;=\; \overline{x_1} f_0 \;\oplus x_1 f_1 \qquad (2.1)$

$f(x_1, x_2, ..., x_n) \;=\; 1 \cdot f_0(x_2, ..., x_n) \oplus x_1 f_2(x_2, ..., x_n)$

in short $\quad f \;=\; f_0 \;\oplus x_1 f_2 \qquad (2.2)$

$f(x_1, x_2, ..., x_n) \;=\; 1 \cdot f_1(x_2, ..., x_n) \oplus \overline{x_1} f_2(x_2, ..., x_n)$ in short $\quad f \;=\; f_1 \;\oplus \overline{x_1} f_2 \qquad (2.3)$

where f_0 is f with x_1 replaced by 0 (negative cofactor of variable x_1), f_1 is f with x_1 replaced by 1 (positive cofactor of variable x_1), and $f_2 \;=\; f_0 \;\oplus\; f_1$.

OBSERVATIONS

- All these expansions generalize the Shannon expansion.
- There exist no other canonical expansions for **single** variable.
- All gates realizing these expansions exist in Fine Grain FPGAs (Atmel, Motorola, Xilinx) and in modern VLSI libraries, the libraries allow for even more complex expansion gates.
- Full potential of existing technology is not used, because of the weakness of the expansion theory (like, OR gate is not used).
- Creating more powerful expansions, we make better use of existing technology.

CANONICAL EXPANSIONS FOR MANY VARIABLES

- The **single-variable** expansion tree models that corresponds to the Kronecker family of families of canonical expansions (Green, Sasao) are **insufficient**.
- Observe, that there exists a peculiar canonical EXOR-based expansion, called the Generalized Reed-Muller Expansion (GRM), that **cannot** be obtained by a composition of canonical expansions for single variables.
- It is obtained by certain **simultaneous** expansion for **all** variables.

CANONICAL EXPANSIONS FOR MANY VARIABLES (cont)

- GRM expansions are **less complex** on average and on the worst case than the Kronecker type of expansions (Debnath/Sasao)
- They are therefore of high practical interest in recent years.
- However, until now, they have not been linked to the general theory of tree expansions, their respective diagrams, and the flattened (two-level) expressions obtained from them.

CANONICAL EXPANSIONS FOR MANY VARIABLES

- Below, the generalized canonical expansions will be presented, that will include these Generalized Reed-Muller Expansions as their special cases.
- Also, the new kinds of expansions applied in nodes of the decision trees will be introduced.
- These new **multi-variable node expansions** will generalize the three expansions (2.1) - (2.3) that have been used in the classical representations and in the based on them synthesis methods.
- The generalization of tree representation will be achieved by a simple conceptual extension to the known binary trees: in addition to single-variable nodes in the trees, the nodes with GRM expansions for several variables will be also allowed, see Figure 1.

EXAMPLE OF GENERALIZED KRONECKER TREE

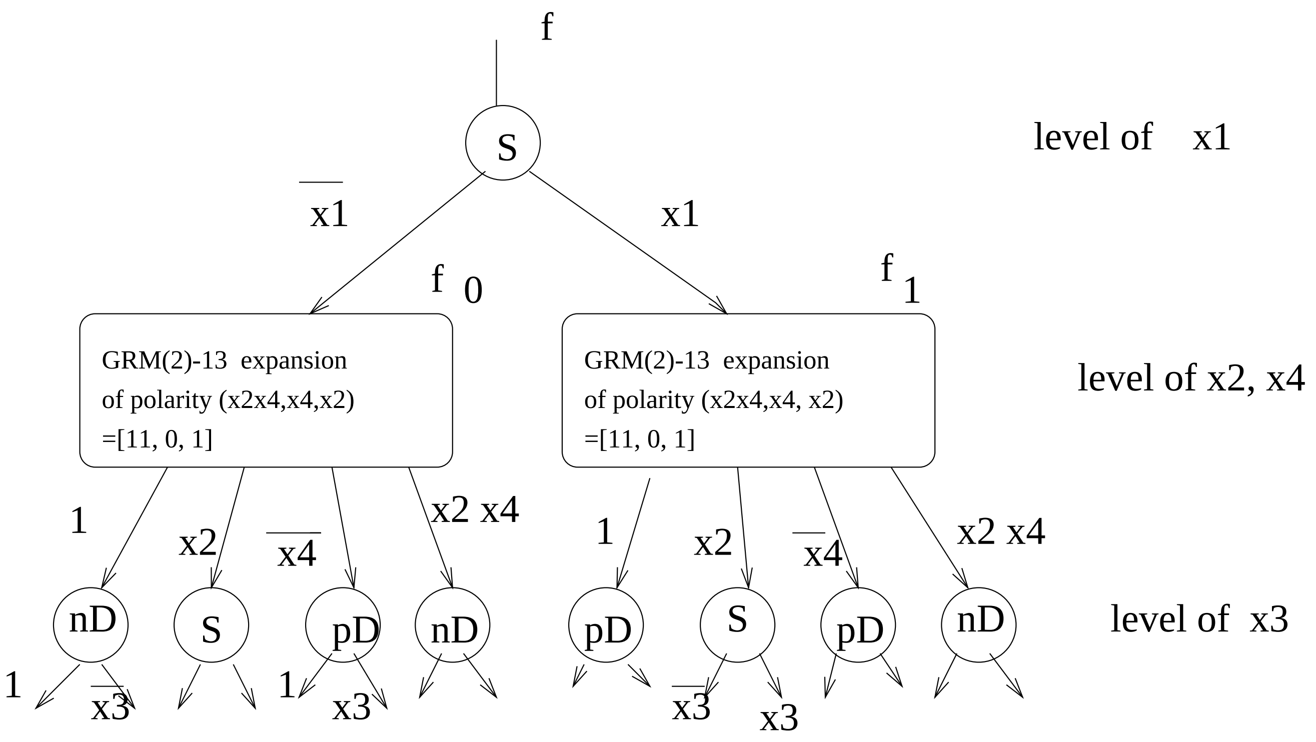


Figure 1:

POSITIVE POLARITY AND GENERALIZED REED-MULLER EXPANSIONS

- An arbitrary n-variable function $f(x_1, x_2, ..., x_n)$ can be represented as

$$f(x_1, ..., x_n) = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus ... \oplus a_n x_n \oplus a_{12} x_1 x_2 \oplus a_{13} x_1 x_3 \oplus ... \oplus a_{n-1,n} x_{n-1} x_n \oplus .... \oplus a_{12...n} x_1 x_2 x_3 ... x_n \tag{3.1}$$

  where a_i's are either 0 or 1.

- Formula (3.1) is a Positive Polarity Reed-Muller Expansion (PPRM).
- Fixed Polarity Reed-Muller Expansion (FPRM).

GENERALIZED REED-MULLER EXPANSION

- *Definition 1.* The **Generalized Reed-Muller Expansion** (GRM) is obtained by freely choosing in (3.1) the polarity for each individual literal.
- In GRM, contrary to FPRMs, the same variable can stand in both positive and negative polarities.
- There are $n2^{n-1}$ literals in (3.1), so there are $2^{n2^{n-1}}$ polarities for n-variable function.
- Each of the polarities determines a unique set of coefficients, and thus a canonical representation of a logic function.

GENERALIZED REED-MULLER EXPANSION (cont)

- There are many more GRMs than FPRMs for a function.
- Thus the minimal GRM (among the GRMs for all the possible polarities) is not worse than the minimal FPRM (among all the respective polarities).
- Usually it is much better.
- **How to compute** these multi-variable GRM expansions, assuming however that coefficients a_i in (3.1) are no longer binary constants, but functions of some variables?

COMPUTATION OF THE DATA FUNCTIONS FOR A GIVEN GRM POLARITY

The *GRM Universal module* computes the output function $f(x_1, ..., x_m, ..., x_n)$ from the values of expansion variables and subfunctions SF_i in the following GRM expansion formula with functional coefficients.

$$f(x_1, x_2, ..., x_m, ..., x_n) \;=\; SF_0(x_{m+1}, ..., x_n) \oplus \hat{x_1} SF_1(x_{m+1}, ...x_n)$$
$$\oplus \hat{x_2} SF_2(x_{m+1}, ..., x_n) \oplus ...$$
$$\oplus\; \hat{x_m} SF_m(x_{m+1}, ..., x_n) \oplus \hat{x_1}\hat{x_2} SF_{12}(x_{m+1}, ..., x_n) \oplus$$
$$\hat{x_1}\hat{x_3} SF_{13}(x_{m+1}, ..., x_n) \oplus\; ...\hat{x_{m-1}}\hat{x_m} \oplus SF_{m-1,m}(x_{m+1}, ..., x_n)$$
$$\oplus .... \oplus\; \hat{x_1}\hat{x_2}\hat{x_3}...\hat{x_m} SF_{12...n}(x_{m+1}, ..., x_n) \tag{4.1}$$

where $\hat{x_i}$ means variable x_i with negation or not (i.e. variable x_i in arbitrary polarity).

OBSERVATIONS

- Observe that formula (4.1) for function $f(x_1, x_2, ..., x_m, ..., x_n)$ is similar to (3.1) with the difference that now:
  - (1) the expansion is with respect to a **subset** of variables $x_1, ..., x_m$,
  - (2) polarities of all literals in product terms are **arbitrary**,
  - (3) coefficients a_i are now replaced with **subfunctions** SF_i of remaining variables $x_{m+1}, ...x_n$.
- Observe also that functions $SF_i(x_{m+1}, ..., x_m)$ are **not** the respective cofactors of function f, as it would be in case of variable-decomposable expansions, FPRM or KRM, but are **computed together** for every GRM polarity.

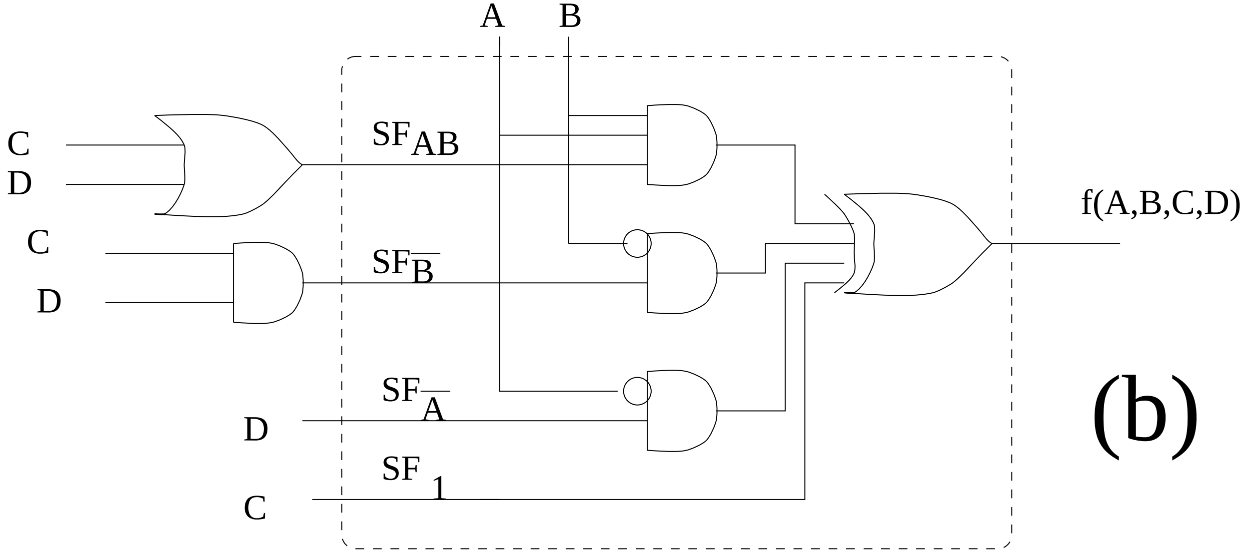
EXPANDING A FUNCTION WITH RESPECT TO GRM FUNCTIONAL COEFFICIENTS

- In this section we will explain how GRM expansion is calculated from given function $f(x_1,...,x_m,...,x_n)$, for only a subset of its variables $(x_1,...,x_m)$.
- This calculation is crucial for the method introduced below, to expand efficiently any function $f(x_1,...,x_m,...,x_n)$ with respect to given product terms $\hat{x_1},...,\hat{x_m}$ of arbitrary GRM on the subset of the function's variables.

EXAMPLE 4.1: THE IDEA OF GRM-EXPANSION USING GRM-UNIVERSAL MODULE.

AB \ CD	00	01	11	10
00	0	1	1	1
01	0	1	0	1
11	0	1	0	0
10	0	0	0	1

(a)



(b)

Figure 2:

USE LINEARLY INDEPENDENT LOGIC

- Use methods of Linearly Independent logic (Perkowski, Falkowski, Sarabi) to uniquely derive functions SF_i from the original function $f(x_1, x_2, ..., x_n)$, assuming given sets of products of literals $\hat{x_1}, ..., \hat{x_m}$.
- These methods are very powerful and generic: they can be applied to **every** field extension.
- Create a $2^m \times 2^m$ matrix M with rows corresponding to minterms (for a function with m variables we have 2^m columns).
- The columns correspond then to some Boolean functions of m variables.

EXAMPLE 4.1. (cont)

- Let us assume that for $f(A,B)$ the LI functions are AB, $\overline{B}$, $\overline{A}$, and 1.
- Because m=2, the minterms are in 2^m-dimensional space, $2^m = 4$.
- The rows of matrix M from top to bottom correspond to the following minterms in $m = 2$-dimensional space:
  $\overline{A}\ \overline{B}$, $\overline{A}\ B$, $A\ \overline{B}$, and $A\ B$, respectively.
- Thus, the $2^m \times 2^m$ matrix M is as follows:

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

EXAMPLE 4.1. (cont)

- The first column corresponds to function AB, the second column to function $\overline{B}$, the third column to function $\overline{A}$, and the last column to function 1.
- It is easy to check that M is nonsingular with respect to EXOR-ing operation, and that it has exactly one inverse matrix M^{-1}.

FUNDAMENTAL THEOREM

Theorem 1. Given is a function F$(x_1, ..., x_m, ..., x_n)$ such that the set of input variables $\{x_1, ..., x_n\}$ includes properly the set $\{x_1, ..., x_m\}$. There exists an unique expansion (called an LI expansion):

$$F(x_1, ..., x_n) = f_0(x_1, ..., x_m)SF_0(x_{m+1}, ..., x_n)$$
$$\oplus f_1(x_1, ..., x_m)SF_1(x_{m+1}, ..., x_n) \oplus ....$$
$$... \oplus f_{2^n-1}(x_1, ..., x_m)SF_{2^n-1}(x_{m+1}, ..., x_n), \tag{4.2}$$

where functions f_i are the given LI functions of m variables, and the coefficient functions (called also the **"data input functions"**) SF_i of remaining variables are determined from the coefficient vector CV $=$ $M^{-1} \times$ FV, where $FV(x_{m+1}, ..., x_n)$ is a vector of all 2^m cofactors of F with respect to variables from the set $\{x_1, ..., x_m\}$.

FUNDAMENTAL THEOREM. (cont)

In general, M is the matrix of 2^m cofactors of F with respect to variables from the set $\{x_1, ..., x_m\}$. Thus, when $m = n$, the cofactors with respect to variables $x_1, ..., x_m$ become minterms on these variables, as in Example 4.1 above.

UNIQUENESS OF THE EXPANSION

- It can be observed that, because the set of product terms $\hat{x_1}\hat{x_2}\hat{x_3}...\hat{x_m}$ is a linearly independent family of functions, the GRM expansion with functional coefficients (4.1) is a **special case** of the LI expansion with functional coefficients (4.2).
- This is an **unique expansion** for the set of variables $x_1, ..., x_m$ and the linearly independent functions on variables $x_1, ..., x_m$.
- Thus, the data functions SF_i are also unique.

EXAMPLE 4.1, (cont)

- Given is function $f(A,B,C,D)$, as before.
- Assume the GRM expansion with respect to variables $\{A,B\}$.
- To calculate functions $SF_i(C,D)$, expand function **twice** $f(A,B,C,D)$ for these variables.
- The **first expansion** will use the computable from function $f(A,B,C,D)$ **standard cofactors**: $f_{\overline{A}\,\overline{B}}(C,D)$, $f_{\overline{A}\,B}(C,D)$, $f_{A\,\overline{B}}(C,D)$, $f_{AB}(C,D)$.
- The **second expansion** will use **unknown functions** $SF_i(C,D)$.
- This will lead to a set of linear logic equations, which after solving will give the values to the unknown functions $SF_i(C,D)$.
- The basis of LI functions of a GRM expansion is here arbitrarily selected as: $f_{AB} = AB$, $f_{\overline{B}} = \overline{B}$, and $f_{\overline{A}} = \overline{A}$, and $f_1 = 1$.

CALCULATION OF EXPANSIONS

$$\overline{A}\ \overline{B}\ f_{\overline{A}\ \overline{B}}(C,D) \ \oplus\ \overline{A}\ B f_{\overline{A}\ B}(C,D) \ \oplus\ A\ \overline{B} f_{A\ \overline{B}}(C,D) \ \oplus\ A\ B\ f_{AB}(C,D)$$

$$= \overline{A}\ \overline{B}\ f(A,B,C,D) \mid_{\ A=0,B=0} \ \oplus\ \overline{A}\ B\ f(A,B,C,D) \mid_{\ A=0,B=1} \ \oplus\ A\ \overline{B}\ f(A,B,C,D) \mid_{\ A=1,B=0} \ \oplus\ A\ B\ f(A,B,C,D) \mid_{\ A=1,B=1} = \overline{A}\ \overline{B}\ (C+D) \ \oplus\ \overline{A}\ B(D \oplus C) \ \oplus\ A\ \overline{B}(C\overline{D}) \ \oplus\ A\ B\ (D\overline{C})$$

$$= AB\ SF_{AB}(C,D) \ \oplus\ \overline{B}\ SF_{\overline{B}}(C,D) \ \oplus\ \overline{A}\ SF_{\overline{A}}(C,D) \ \oplus\ SF_1(C,D)$$

SUBSTITUTION OF ALL POSSIBLE VALUES TO GET LINEAR EQUATIONS.

- By substituting in the above equation A = 0, B = 0, we get $SF_{\overline{B}} \oplus SF_{\overline{A}} \oplus SF_1 = f_{\overline{A}\ \overline{B}}(C, D) = (C + D)$.
- By substituting A = 0, B = 1, we get $SF_{\overline{A}} \oplus SF_1 = f_{\overline{A}\ B}(C, D) = (C \oplus D)$.
- By substituting A = 1, B = 0, we get $SF_{\overline{B}} \oplus SF_1 = f_{A\ \overline{B}}(C, D) = (C\overline{D})$.
- By substituting A = 1, B = 1, we get $SF_{AB} \oplus SF_1 = f_{A\ B}(C, D) = (\overline{C}D)$.

TRANSFORM SET OF EQUATIONS TO MATRIX FORM

Hence we obtain the following equation for cofactors $f_{A^i\ B^j}(C,D)$:
$FV = M \times CV =$

$$= \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} SF_{AB}(C,D) \\ SF_{\overline{B}}(C,D) \\ SF_{\overline{A}}(C,D) \\ SF_{1}(C,D) \end{bmatrix} = \begin{bmatrix} f_{\overline{A}\ \overline{B}}(C,D) \\ f_{\overline{A}\ B}(C,D) \\ f_{A\ \overline{B}}(C,D) \\ f_{A\ B}(C,D) \end{bmatrix}$$

SOLVING SET OF EQUATIONS

CV $= M^{-1} \times FV =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} f_{\overline{A}\ \overline{B}}(C,D) \\ f_{\overline{A}\ B}(C,D) \\ f_{A\ \overline{B}}(C,D) \\ f_{A\ B}(C,D) \end{bmatrix} = \begin{bmatrix} SF_{AB}(C,D) \\ SF_{\overline{B}}(C,D) \\ SF_{\overline{A}}(C,D) \\ SF_{1}(C,D) \end{bmatrix}$$

GRM EXPANSION FOR UNIVERSAL LOGIC MODULE

From Theorem 1, the corresponding GRM expansion for ULM is:

$$f = AB\ SF_{AB}(C,D) \oplus \overline{B}\ SF_{\overline{B}}(C,D) \oplus \overline{A}\ SF_{\overline{A}}(C,D) \oplus SF_1(C,D) \qquad (4.3)$$

and the coefficients $SF_i(C,D)$ are taken from the above vector CV.

Let function F be represented by

vector $FV^T = [(C+D) \quad (C \oplus D) \quad (C\overline{D}) \quad (\overline{C}D)\ ]$.

$$\text{CV} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} (C+D) \\ (C \oplus D) \\ (C\overline{D}) \\ (\overline{C}D) \end{bmatrix} = \begin{bmatrix} S_{AB} \\ S_{\overline{B}} \\ S_{\overline{A}} \\ S_1 \end{bmatrix} =$$

$$\begin{bmatrix} (C+D) \\ (C+D) \oplus (C \oplus D) \\ (C+D) \oplus (C \oplus D) \oplus (\overline{C}D) \\ (C+D) \oplus (C \oplus D) \oplus (C\overline{D}) \end{bmatrix} = \begin{bmatrix} (C+D) \\ (CD) \\ (D) \\ (C) \end{bmatrix}$$

OBSERVATIONS

- Then $f(A,B,C,D) = (AB)(C+D) \oplus \overline{B}(CD) \oplus \overline{A}(D) \oplus (C)$ (4.4)
- Observe, that to calculate new GRM expansion with functional coefficients from M^{-1} and FV^T, one has always to execute certain EXOR-ing on a subset of the set of all cofactors from FV^T.
- These cofactors can be functions of many variables, so it is important to execute this operation efficiently.
- In a circuit the GRM expansion with functional coefficients is realized using universal logic module for GRM expansion with respect to variables A, B (illustrated in Fig. 2b).
- This way, for the set of LI functions (products of literals in this case), $\{\overline{A}, \overline{B}, AB, 1\}$, there exists **only one** GRM expansion specified by its matrix M^{-1}.

OBSERVATIONS TO THE EXAMPLE.

- From this expansion the following ESOP can be found:
  $AB[C \oplus \overline{C}D] \oplus \overline{B}CD \oplus \overline{A}D \oplus C = ABC \oplus AB\overline{C}D \oplus \overline{B}CD \oplus \overline{A}D \oplus C$
- Formula (4.3) describes **only one** of 16 GRM expansions for variables A, B, when 16 possible GRM universal modules exist for two variables.
- Any set of two or three variables out of set $\{A, B, C, D\}$ can be selected for the first level GRM expansion.
- So, there are **many different** trees representing successive expansions.
- Matrices are used here only for dydactic purpose.
- Expansions can be executed on any function representation because only **cofactoring,** and **exoring** operations are needed, plus **solving linear equations**.

GENERALIZED KRONECKER TREES

- The Kronecker Tree has levels that correspond to single input variables and only one of three types of binary expansions is used in every level (Sasao).
- However, such trees cannot generate the Generalized Reed-Muller expressions, and thus both the two-level and multi-level circuits corresponding to Kronecker Trees can be far from the minimum.
- Similarly, the decision diagrams that are created by applying reductions to nodes of such trees have also non-minimum numbers of nodes.
- By allowing to have nodes in the trees for **sets** of variables, instead for single variables only, the concept of the tree is now generalized, and the tree is no longer a binary tree.
- We call this new tree the **Generalized Kronecker Tree** (GKT), to reflect its properties.

GENERALIZED KRONECKER TREES

Definition 2. The **Generalized Kronecker Tree** is a multi-variable expansion node tree created as follows:

1) The set of all n input variables is partitioned into disjoint and nonempty subsets S_j, called *blocks*, such that the union of all these subsets forms the initial set.
   - The blocks are disjoint and non-empty.
   - If each block includes just a single variable, the tree reduces to the special case of a KRO Tree.
   - If there is only one block that includes all variables, the tree reduces to the special case of a GRM.
2) The blocks are ordered, each corresponds to a tree level.
3) For every level, if the block involves a single variable, S, nD, or pD is selected for its nodes. If the block is multi-variable, one GRM expansion polarity is selected for nodes.

POLARITIES OF BLOCKS

- For every block with a single variable, the corresponding expansions are S, pD and nD.
- For a set with two variables there are $2^{n2^{n-1}}$ GRM expansions, i.e. 16 expansions.
- Therefore, for the two- variable nodes there are 16 types of nodes, called GRM(2) nodes (expansion types).
- They will be denoted by GRM(2)-0, GRM(2)-1,...,GRM(2)-15, or as their polarities: [00,0,0], [00,0,1], ...,[11, 1, 1], respectively.
- NOTATION: for variables $\overline{x_2}$, $\overline{x_4}$, the part of the polarity 00 represents the term $\overline{x_2}\ \overline{x_4}$.

NOTATION FOR BLOCK POLARITIES

- In this way, the polarity GRM(2)-0 = [00,0,0] represents LI functions $\overline{x_2}\ \overline{x_4}, \overline{x_4}, \overline{x_2}$ and **1** of the polarity matrix M (the LI function **1** has no polarity since it has no literals).
- Polarity GRM(2)-1 = [00,0,1] represents LI functions $\overline{x_2}\ \overline{x_4}, \overline{x_4}$, x_2 and **1** of the polarity matrix M. And so on. It can be shown in a similar way that there are 2^{12} types of GRM(3) nodes for 3 variables.

GENERALIZED KRONECKER FORMS AND DECISION DIAGRAMS

Definition 3. **Generalized Kronecker Forms** (GK Forms) are obtained by **flattening** the GK Trees, i.e. finding all ordered product terms obtained by multiplication of paranthesized expressions corresponding to AND/EXOR trees, using recursively the rule $a(b \oplus c) = ab \oplus ac$.

Definition 4. **Generalized Kronecker Decision Diagrams** (GK DDs) are created by:

(1) **combining isomorphic nodes** of any kind,

(2) performing **standard Ordered Kronecker Functional Decision Diagram (OKFDD) transformations** (Drechsler) on S, pD and nD nodes,

(3) performing generalizations of standard Ordered Kronecker Functional Decision Diagram (OKFDD) transformations (Drechsler) on multi-variable nodes.

These diagram transformations remove the node that evaluates to its single arguments.

GENERALIZED KRONECKER TREE

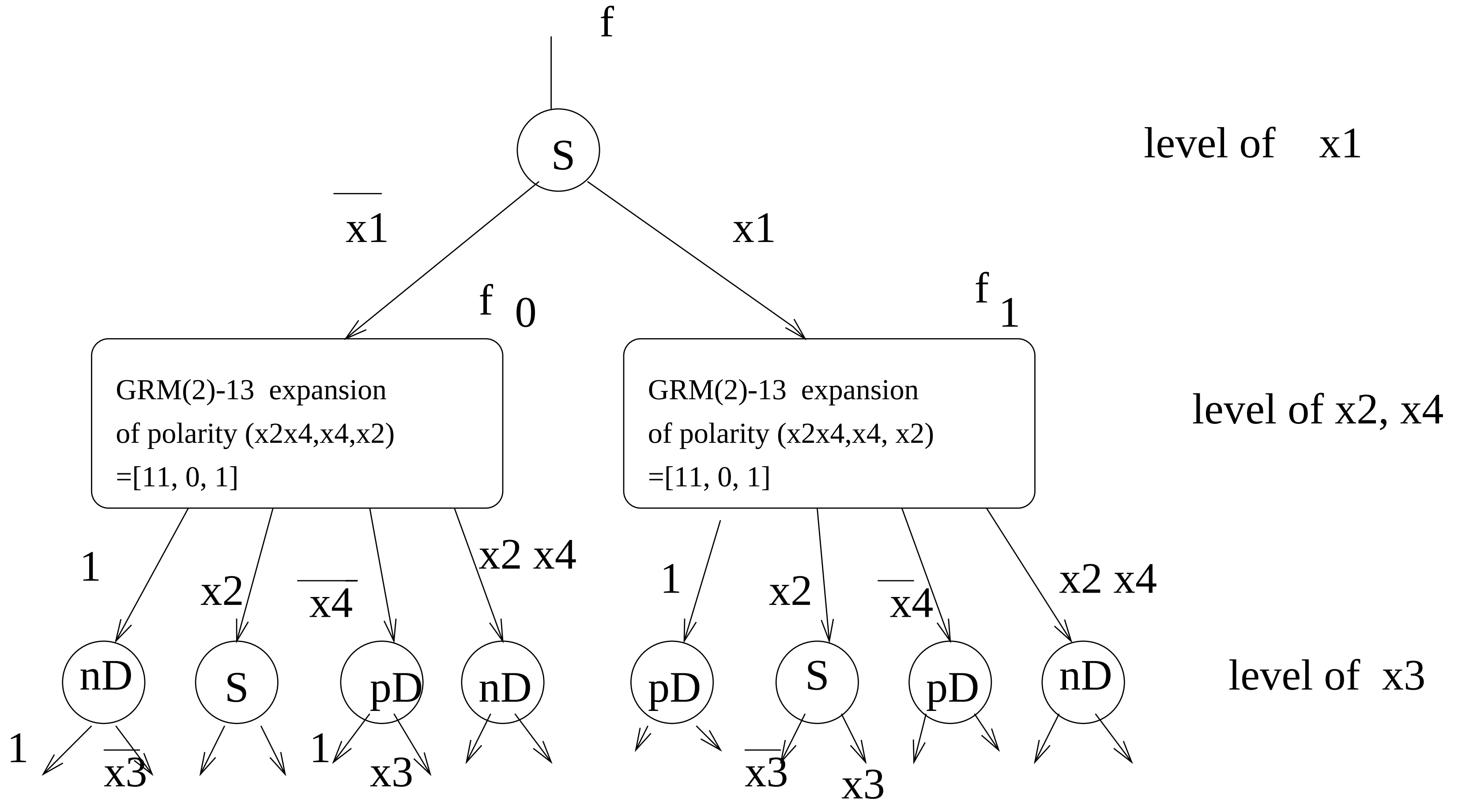

f
S
level of x1
x1
x1
f 0
f 1
GRM(2)-13 expansion
of polarity (x2x4,x4,x2)
=[11, 0, 1]
GRM(2)-13 expansion
of polarity (x2x4,x4, x2)
=[11, 0, 1]
level of x2, x4
1
x2
x4
x2 x4
1
x2
x4
x2 x4
nD
S
pD
nD
pD
S
pD
nD
level of x3
1
x3
1
x3
x3
x3

Figure 3:

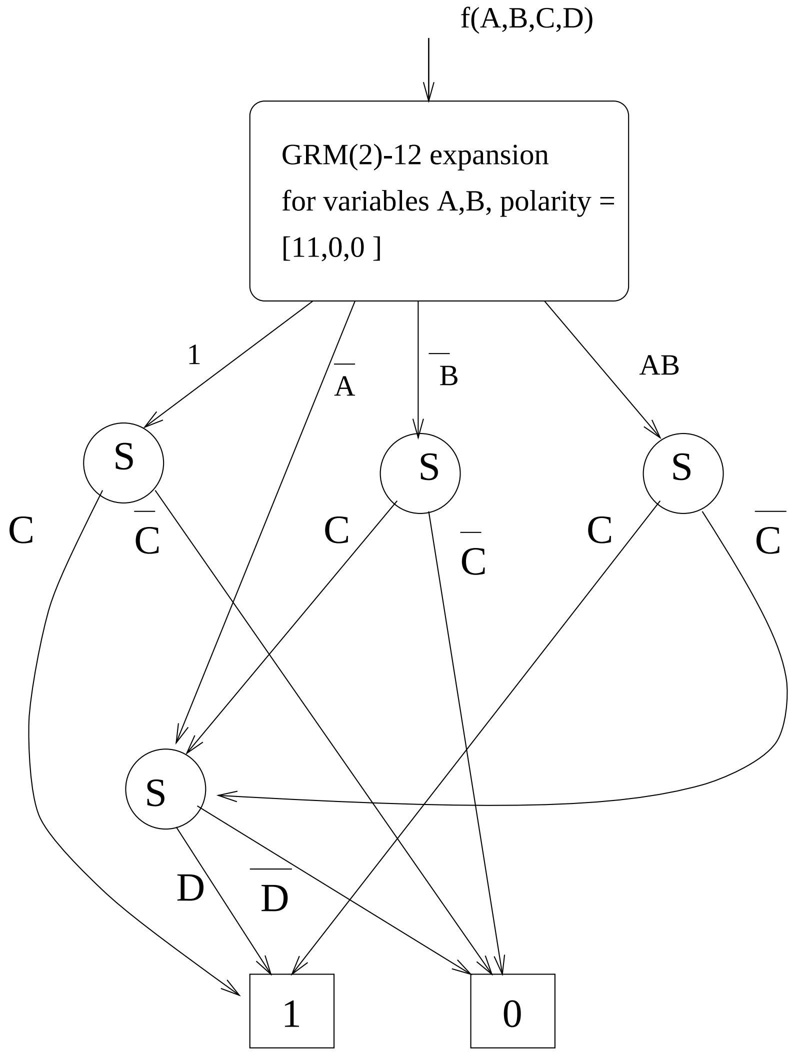


Figure 4: *Generalized Kronecker Diagram for function from Example 4.1.*

GENERALIZED KRONECKER DIAGRAM

- The expansion of the node GRM(2)-13 is described by the following formula:

  $f_0(x_2, x_3, x_4) = SF(f_0)_1(x_3) \oplus$
  $x_2 SF(f_0)_{x_2}(x_3) \oplus \overline{x_4} SF(f_0)_{\overline{x_4}}(x_3) \oplus x_2 x_4 SF(f_0)_{x_2 x_4}(x_3)$

  where notation $SF(f)_i(X)$ denotes function SF_i, with arguments from the set X of variables, applied to argument function f.

- This formula is the specialization of GRM expansion (4.1) for $f_0(x_2, x_3, x_4)$ with expansion variables x_2, x_4, where subfunctions SFi of the remaining variables are all calculated on the cofactor function f_0 (these are functions: $SF(f_0)_1$,$SF(f_0)_{x_2}(x_3)$, $SF(f_0)_{\overline{x_4}}(x_3)$,$SF(f_0)_{x_2 x_4}(x_3)$ in this particular GRM(2)-13 expansion).

EVALUATION OF NODES

- Node evaluates to a single argument if after substituting constants and a single variable value H_i to the function realized by this node, after propagation of constants, the node function evaluates to H_i.
- Observe that only multivariable nodes that have only logic constants and the same signal H_i as arguments should be evaluated.
- For instance:
  - Formula $\overline{a}\,\overline{b}H_1 \oplus \overline{a}\,bH_1 \oplus a\,\overline{b}H_1 \oplus a\,bH_1$ evaluates to H_1.
  - Formula $\overline{a}\,b0 \oplus a\,b0 \oplus \overline{b}H_2 \oplus bH_2$ evaluates to H_2.
  - Formula $ab0 \oplus a0 \oplus b0 \oplus H_3$ evaluates to H_3.
- This method is a generalization of simplifications of S, pD and nD nodes applied in OKFDDs.

IS THERE ANY ADVANTAGE OF THE NEW FORMS?

Example 5.3 An example of a GK form that is not a GRM and not a KRO is the following

$$g(x,y,z) = \overline{x}(\overline{y}\ \overline{z}) \oplus x(yz \oplus \overline{y} \oplus \overline{z} \oplus 1) = \overline{x}\ \overline{y}\ \overline{z} \oplus xyz \oplus x\overline{y} \oplus x\ \overline{z} \oplus x$$

This is not a GRM form, because there are two terms that include the same set of variables $\{x, y, z\}$. The above form can be transformed to one of GRM forms as follows:

$$g(x,y,z) = \overline{x}\ \overline{y}\ \overline{z} \oplus xyz \oplus x\overline{y} \oplus x\ \overline{z} \oplus x = (1 \oplus x)(1 \oplus y)(1 \oplus z) \oplus xyz \oplus x\overline{y} \oplus x\ \overline{z} \oplus x = 1 \oplus x \oplus y \oplus z \oplus xy \oplus xz \oplus yz \oplus xyz \oplus xyz \oplus x\overline{y} \oplus x\ \overline{z} \oplus x = 1 \oplus y \oplus z \oplus x \oplus yz \oplus x = 1 \oplus y \oplus z \oplus yz = 1 \oplus y \oplus z\overline{y}\ g1(x,y,z)$$

As we see, in this case the GRM form $g1(x,y,z)$ is better than the GK form $g(x,y,z)$.

GK QUALITY THEOREMS

- Form $g(x, y, z) =$ is also not a KRO, because $g(x, y, z) \mid_{x=1} = yz \oplus \overline{y} \oplus \overline{z} \oplus 1$ is not a KRO.

  The above form can be transformed to one of KRO forms as follows:

  $\overline{x}(\overline{y}\ \overline{z}) \oplus x(yz \oplus \overline{y} \oplus \overline{z} \oplus 1) = \overline{x}(\overline{y}\ \overline{z}) \oplus x(yz \oplus y \oplus \overline{z}) = \overline{x}(\overline{y}\ \overline{z}) \oplus x(\overline{y}\ \overline{z}) = \overline{y}\ \overline{z} = g2(x, y, z)$

- As we see, in this case the KRO form $g2(x, y, z)$ is better than the GK form $g(x, y, z)$, and better than the GRM form $g1(x, y, z)$.

- We demonstrated thus an example of a GK form that is not included in KRO or in GRM families, therefore we proved that GK forms include **properly** the KRO and GRM families.

GK QUALITY THEOREMS

The question remains, however, *"is the GK form potentially more minimal than GRM and KRO forms?"*.

Theorem 2. Every AND/EXOR canonical form f can be obtained from any other canonical form g by **substituting** terms of f by EXOR operation performed on some subsets of product terms from g.

Example 5.4. By EXOR-ing all terms of $g(x, y, z)$ we obtain $\overline{y}\ \overline{z} = g2(x, y, z)$

Theorem 3. Every AND/EXOR form that can be described by the following formula:

$$f(a, b, c, ..., k; x_1, x_2, ..., x_m) =$$

$$f_1(a, b, c, ...k)x_1 \oplus f_2(a, b, c, ...k)x_2 \oplus ...f_m(a, b, c, ...k)x_m \qquad (5.1)$$

where:

- sets of variables $\{a, b, c, ..., k\}$ and $\{\ x_1, x_2, ..., x_m\}$ are disjoint,
- functions f_i are linearly independent,
- all variables x_i are different.

has the minimum number of terms.

GK QUALITY THEOREMS (cont)

$f(a, b, c, ..., k; x_1, x_2, ..., x_m) =$
$f_1(a, b, c, ...k)x_1 \oplus f_2(a, b, c, ...k)x_2 \oplus ...f_m(a, b, c, ...k)x_m 2$

$= h_1(a, b, c, ..., k; x_1, x_2, ..., x_m) \oplus h_2(a, b, c, ..., k; x_1, x_2, ..., x_m) \oplus$
$h_3(a, b, c, ..., k; x_1, x_2, ..., x_m) =$

$[f_1(a, b, c, ...k)x_1 \oplus f_3(a, b, c, ...k)x_3] \oplus$
$[f_2(a, b, c, ...k)x_2 \oplus f_4(a, b, c, ...k)x_4] \oplus$

$[f_5(a, b, c, ...k)x_1 \oplus f_6(a, b, c, ...k) \oplus f_7(a, b, c, ...k)x_7].$

GK QUALITY THEOREMS (cont)

- However, none of the expressions in brackets can be simplified to less terms, because they have *different variables* x_i.
- In the best case, when all functions f_i are the same, the factorization of f_i can be performed, but it also does not decrease the number of terms.
- So, any EXORs of terms will not decrease the number of terms.
- Thus, the form $f(a, b, c, ..., k; x_1, x_2, ..., x_m)$ is the one with the minimum number of terms among all canonical forms representing this function.
- *Theorem 4.* In every minimal ESOP all product terms are linearly independent.

GK QUALITY THEOREMS (cont)

- *Definition 5.* By **ESOP-simplifiable expression** in AND/EXOR form we will understand one that applying any combination of rules from EXORCISM-MV-2, an ESOP with smaller number of terms were created.
- Expression that is not ESOP-simplifiable, will be called **ESOP-nonsimplifiable**.

GK QUALITY THEOREMS (cont)

Theorem 5. Every AND/EXOR expression of the following form:

$$f(a, b, c, ..., k; X_1, X_2, ..., X_m) = f_1(a, b, c, ...k) f_{h1}(X_1) \oplus$$
$$f_2(a, b, c, ...k) f_{h2}(X_2) \oplus ... \oplus f_m(a, b, c, ...k) f_{hm}(X_m) \qquad (5.2)$$

where:

- set of variables $\{a, b, c, ..., k\}$ and any set of variables $X_1, X_2, ..., X_m$ are disjoint,
- functions f_i are linearly independent or they are EXOR combinations of linearly independent functions that are ESOP-nonsimplifiable,
- variables in formula (5.2) x_i are all different,
- all functions f_{hi} are on disjoint sets X_i of variables.

has the minimum number of terms.

THEOREMS FOR QUALITY PROOF AND EXACT ESOP MINIMIZATION

- *Theorem 6.* The canonical form of the form (5.2) is the minimal ESOP.
- *Theorem 7.* The minimal GK Form is better than the minimal KRO and the minimal GRM forms.
- Theorem 7 demonstrates the usefulness of the concept of GK forms in practical minimization algorithms for high-performance canonical forms (because their expected superior testability) and ESOPs.
- The form shown in Theorem 7 is also the exact minimum ESOP.
- All above notions and methods can be generalized to functions with many outputs.

POLARITY GRM EXPANSIONS

Definition 6. A **Single-Polarity GRM expansion** for a multi-output function is a vector of GRM expansions for its component single-output functions, **all of them of the same polarity.**

Definition 7. A **Multi-Polarity GRM expansion** for a multi-output function is a vector of GRM expansions for its component single-output functions, each of them can have **different** polarity.

- Thus, for a three-output function, the polarity vector of a Single-Polarity GRM expansion has 4 bits, and for Multi-Polarity LI expansion has $3 * 4 = 12$ bits.
- Observe, that Definition 6 of multi-output GRM expansions is in accordance with the definition from (Zeng),
- while Definition 7 is in accordance with the definition from (Debnath, Sasao).

POLARITY GRM EXPANSIONS. (cont)

- Obviously, the minimal DD obtained from Definition 7 is smaller than one obtained from Definition 6.
- There are, however, some advantages of considering representations created according to Definition 6; mostly simpler and more efficient algorithms, and better testability.

APPLICATIONS TO EXACT ESOP MINIMIZATION

- ESOPs are AND/EXOR circuits with no constraints imposed on the products of literals - thus the best AND/EXOR circuits, but not canonical.
- Much research on synthesis of minimal ESOPs.
- Exact ESOPs can be obtained only for few variables.
- The quality of AND/EXOR circuits obtained from the expansions proposed here should be better than those corresponding to GRMs or KROs because the search space of GKTs is much larger than the combined search space of GRM or KRO expansions.

APPLICATIONS TO EXACT ESOP MINIMIZATION (cont)

- Therefore, GK forms will be on average very good approximations of exact ESOPs.
- It is easy to improve every ESOP minimizer by using Theorem 4.
- It is done by adding a final check to it.
- For every solution it has to be checked if the terms are linearly independent.
- If not, the exoring of the terms that are not linearly independent should be executed to obtain a better expression.
- Some top ESOP minimizers have already the rules that satisfy the condition that all terms are linearly independent.
- Therefore, GK forms will be on average very good approximations of exact ESOPs.

APPLICATIONS TO EXACT ESOP MINIMIZATION (cont)

- Theorems 5 and 6 can be used to create better rule-based searching ESOP minimizers (such as EXORCISM-MV-2, and EXORCISM-MV-3).
- This can be achieved by tree-structuring the transformation rules that search the solution space, so that the space is represented by a tree.
- New nodes are created by applying rules to parent nodes, but the rules are selected in such a way that the costs are non-decreasing along the branches, which allows to apply the branch-and-bound principle for cutting off branches with higher costs than some of the previously found solutions.
- Special rules and methods are used to nodes of expressions like (5.1) or like (5.1) but with repeated literals x_i.
- An expression with repeated literals is transformed to expression (5.2) in which ESOP-nonsimplifiable expressions stand by x_i.

- (For instance, the expressions such as $(abc \oplus \overline{a}\ \overline{b}\ \overline{c})x_i$ are ESOP-nonsimplifiable, even they have the same factorized variable x_i.)
- The class of generalized expressions of form (5.2) is a generalization of **sparse functions** introduced by us in previous papers on exact ESOP minimization.
- Whenever a form like (5.2) is found, the branch of the tree is cutted-off, because no better form can be found by exoring subsets of terms. This way, the EXORCISM-3-MV search can be improved.

THEOREMS RELATED TO EXACT ESOP MINIMIZATION

- It was now known if every minimal ESOP can be found from some canonical family of forms.
- *Theorem 8.* For every minimum ESOP one can create a canonical form for which this ESOP is one of the solutions.
- *Theorem 9.* The family of all canonical AND/EXOR forms includes the set of minimal ESOPs.
- Theorem 9 can lead to a new ESOP minimization algorithm.
  - Algorithms are known (Zhang, Zeng), that enumerate all AND/EXOR forms of some type in certain order, for instance, the Grey order (this approach was used for FPRMs or GRMs (Zeng et al).
  - The search is for the form with the smallest cost.
  - If one were able to parametricize all canonical AND/EXOR forms, then, according to Theorem 9, the same approach would be possible to find the exact minimum ESOP.

CURRENT AND FUTURE RESEARCH

- Calculating expansions **not with matrices.** They can be executed on **any** representation of completely or incompletely specified single- or multi- output function.
- We developed **efficient** algorithms that do not use matrices.
- **Exact ESOP** minimization.
- **Counting** the new forms and **comparing** them with the known forms.
- The new graphs may be thus useful to **represent large functions**, larger than those possible to represent by the Ordered Kronecker DDs (which themselves allow to represent larger functions than the BDDs).

CURRENT AND FUTURE RESEARCH (cont)

- Calculating expansions **not with matrices.** They can be executed on **any** representation of completely or incompletely
- Adaptation of the proposed approaches to **multiple-valued, multiple-output functions,** both completely and incompletely specified, as well as **Boolean relations**.
- Find exact minimum solutions for **symmetric functions** represented in the new forms.
- **Testability** should be good, as for all other canonical representations.

CONCLUSIONS

- Positive answer to open problem.
- New AND/EXOR representation.
- Essential expansion to Green/Sasao Hierarchy of Canonical Families.
- Better than all the known canonical AND/EXOR representations with respect to the circuit complexity.
- New trees, diagrams, and expansions will find broad applications, especially in multi-level logic synthesis for FPGAs, gate arrays and standard cell technologies and in the design for high testability.
- Use for design of highly optimized ESOP circuits for small number of variables.
- New theorems allow to create improved exact ESOP algorithms for small number of variables, and prove exactness for some special classes of ESOP expressions.