

# Regular Symmetric Arrays For Non-Symmetric Functions<sup>1</sup>

Malgorzata Chrzanowska-Jeske

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

## Abstract

A new class of non-totally symmetric functions which can be represented as simple regular symmetric arrays without redundancy is identified. Regular circuits are becoming crucially important for deep submicron technologies, which needs new layout approaches to fully utilized their potential. The new class of functions contains the classes of totally symmetric and totally symmetric with mixed polarities functions. Two circuit implementations are proposed, one especially aimed on low-power applications. Generating such functions with a simple method is also described.

## I. INTRODUCTION

Representing a function as a regular array has always been an attractive solution especially when predictability of the area and delay was important. The usual drawbacks of regular arrays are large layout sizes and delays compare to non-regular representations. Newly developed regular arrays, Pseudo-Symmetric Binary Decision Diagrams (PSBDDs) [1], have very compact layouts and their delays are optimized for deep-submicron technologies. They can be directly mapped to a two-dimensional layout without any routing. These diagrams are based on OBDDs [2, 3] for totally symmetric functions which are regular two-dimensional structures. The area and delay are directly proportional to the number of levels in the diagrams, which for totally symmetric functions is equal to the number of function's variables. In general, any arbitrary Boolean function can be represented as a PSBDD, however, for non-totally symmetric functions some variables have to be used multiple times as decomposition variables. This increases a number of levels in a PSBDD and, therefore, also increases the area and delay. In this paper we show that there exists a larger class of functions, which includes a class of totally symmetric functions, which can be represented as PSBDDs without any repeated variables.

The remainder of this paper is organized as follows. Basic terminology and definitions associated with decision diagrams, symmetric functions and symmetric variables are

discussed in section 2. Section 3 discusses simple regular arrays and Pseudo-Symmetric Binary Decision Diagrams are presented in Section 4. The class of with pseudo-symmetric functions associated as well as the method for their generation are presented in Section 5. Conclusion is given in section 6.

## II. PRELIMINARIES

Our approach is based on symmetric networks [5] and Binary Decision Diagrams, BDDs. Let us recall some background information which will be used to explain our approach. Binary Decision Diagrams are created by iterative application of Shannon decomposition theorem to arbitrary Boolean functions.

Any switching function  $f(x_1, x_2, \dots, x_n)$  can be expressed as

$$f(x_1, x_2, \dots, x_n) = x_i \cdot f(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) + x_i' \cdot f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$$

or

$$f = x_i \cdot f_{x_i} + x_i' \cdot f_{x_i'}$$

where  $f_{x_i}$  and  $f_{x_i'}$  are called positive and negative cofactors of function  $f$  in respect to variable  $x_i$ , respectively.

**Definition 1.** A switching function of  $n$  variables  $f(x_1, x_2, x_3, \dots, x_n)$  is called symmetric or (totally symmetric) if and only if it is invariant under any permutation of its variables. In symmetric functions we assume that variables of symmetry can be of both polarities.

It means that function  $f(x_1, x_2, x_3, x_4)$  can be totally symmetric in variables  $(x_1, x_2, x_3, x_4)$  or in any possible combination of the variable polarities like for example  $(x_1', x_2, x_3', x_4)$  or  $(x_1, x_2', x_3', x_4)$ .

Symmetry relation between variables can also be determine by comparing cofactors for each par of variables in a function. For any pair of variables  $x_i$  and  $x_j$ , there are four cofactors, namely,  $f_{x_i x_j}$ ,  $f_{x_i x_j'}$ ,  $f_{x_i' x_j}$ ,  $f_{x_i' x_j'}$ . A function is symmetric in these two variables if any two of the four cofactors, with the choice of negating one of them, are equivalent. Two types of symmetry which are used in defining symmetric functions are given below [4].

<sup>1</sup> This work was supported in part by the NSF grant MIP-9629419.

**Definition 2.** A uncton  $f$  exhibits a non-equivalent symmetry (NE-symmetry) in variables  $x_i$  and  $x_j$ , denoted as  $x_i N E x_j$  or  $\{x_i, x_j\}(\{x_i', x_j'\})$ , if  $f_{x_i x_j} = f_{x_i' x_j'}$ .

**Definition 3.** A function  $f$  exhibits an equivalent symmetry (E-symmetry) in variables  $x_i$  and  $x_j$ , denoted as  $x_i E x_j$  or  $\{x_i, x_j\}(\{x_i', x_j'\})$ , if  $f_{x_i x_j} = f_{x_i x_j}$ .

### III. SIMPLE REGULAR ARRAYS

If two variables exhibit non-equivalent symmetry, the function's two cofactors,  $f_{x_i x_j}$  and  $f_{x_i' x_j'}$ , (in respect to these two variables) are equal and can be represented as one node in a planar drawing of the function OBDD, as shown in Fig. 1.

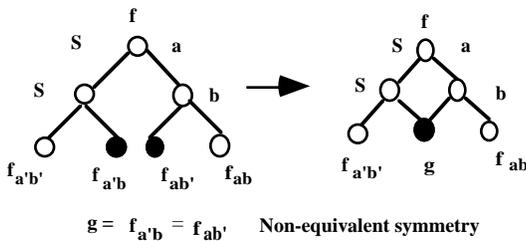


Figure 1. Use of isomorphic cofactors in Shannon expansion.

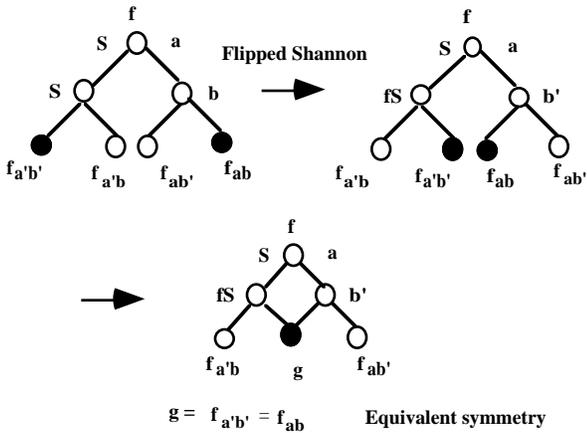
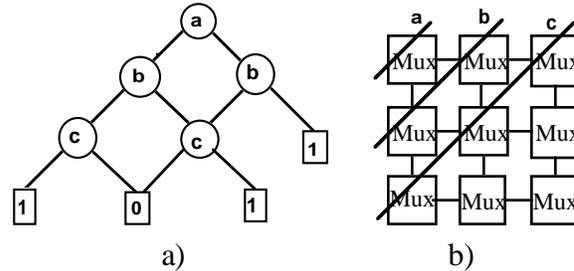


Figure 2. Use of isomorphic cofactors in Flipped Shannon expansion.

In Fig. 1. and Fig 2. black nodes represent the equivalent nodes. We assume that nodes in OBDDs, representing the Shannon decomposition, are always drawn such that the positive (true) cofactor is drawn as the right child and the negative (false) cofactor is drawn as the left child of the parent node. Therefore, in order to take advantage of the equivalent symmetry a flipped Shannon expansion, which is shown in Fig. 2., where cofactors are drawn in opposite direction compared to the Shannon node, is introduced. If a Boolean function is totally symmetric with only non-equivalent symmetries holding between its variables, then its OBDD can be drawn as shown in Fig.3a.

The OBDD there has a desired structure, regular and with only neighbor-to-neighbor connections. Decomposition variables are assigned to long range busses. Such structure can be directly mapped to a two-dimensional array of multiplexers, shown in Figure 3b, with no placement or routing necessary.



$f = a'b'c' + a'bc + ab'c + ab$   
Figure 3. OBDDs for a totally symmetric function with only positive polarities.

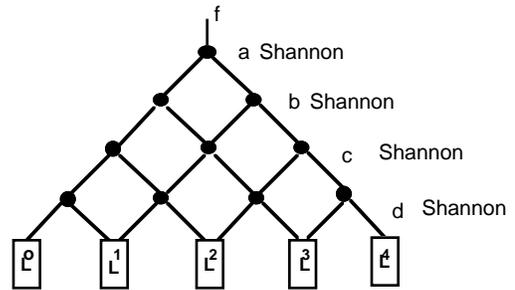


Figure 4. Regular array for symmetric functions with four symmetric variables of only positive polarities.

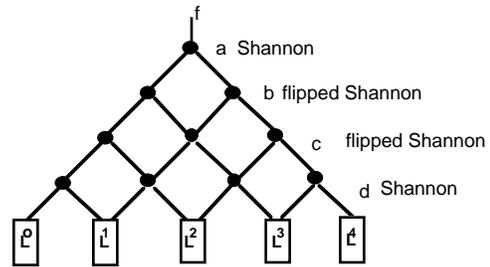


Figure 5. Regular array for symmetric functions with four symmetric variables of mixed polarities.

All nodes in Fig. 4. represent Shannon decomposition. Leave nodes indicated by  $L^i$  represent planes of symmetries and can assume values 0 or 1. In general, if only non-equivalent symmetries are considered, the totally symmetric function will have all symmetric variables in positive polarities and regular structures can be represented with only Shannon (S) decomposition nodes as shown in Fig.4. In the case of totally symmetric

functions with mixed symmetries, where both polarities are allowed, regular diagrams for these functions can be represented with some levels with only Shannon (S) nodes for positive polarity variables, and other levels with flipped Shannon nodes for negative polarity symmetric variables, as shown in Fig 5. In both cases, we assume one variable per level and one type of the decomposition per level. It means that on the same level we can have only  $S$  type nodes or only  $fS$  type nodes. Such array can be realized with two types of muxes with inverse connections to two inputs.

#### IV. PSBDDs

The regular OBDD for a totally symmetric structure is a result of merging together isomorphic adjacent nodes. Unfortunately, not all functions are totally symmetric. Therefore, the merging idea has been extended for the case of non-isomorphic nodes through the *Join-Vertex* operation introduced in [1]. The underlined idea of the *Join-Vertex* operation using the BDD and the PSBDD representations is shown in Fig. 6.

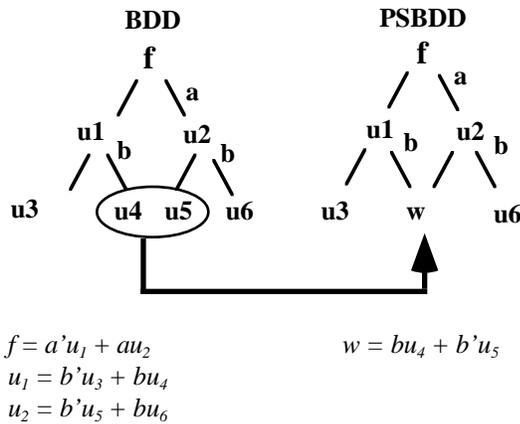


Figure 6. Generating a PSBDD with Join-Vertex operation

The penalty to be paid is the reintroduction of the expansion variables back into the function. The reintroduction of a variable makes it necessary to use the same expansion variable more than once which increases the number of levels. Such diagrams are called PSBDDs with repeated variables and are discussed in details in [6]. The delay of such structures is proportional to a number of levels in a diagram, which is always larger than a number of function variables. A number of variables which have to be repeated depends very strongly on an order of decomposition variables and various order optimization algorithms were developed [6].

**Definition 4.** Pseudo Symmetric Binary Decision Diagram is a Binary Decision Diagram where any two geometrically-adjacent nodes on the same level of

decomposition are represented as one node, and therefore the number of nodes on each level is equal or smaller than the decomposition variable number associated with that level. A node in a PSBDD represents a multiplexer operation on its two inputs.

**Theorem 1.** Any totally symmetric function can be represented as a Pseudo-Symmetric Binary Decision Diagram without repeated variables.

**Theorem 2.** An arbitrary Boolean function can be represented as a Pseudo-Symmetric Binary Decision Diagram with repeated variables.

**Definition 5.** A Pseudo-Symmetric Decision Diagram without Join-Vertex operations (no repeated variables) is called Simple.

**Lemma 1.** A totally symmetric Boolean function  $f$  with only non-skew non-equivalent symmetry can be represented as a Simple Pseudo-Symmetric Binary Decision Diagram with all nodes representing Shannon expansion.

**Lemma 2.** A totally symmetric Boolean function  $f$  with mixed polarity, non-skew non-equivalent or equivalent symmetries (N1 or N2), can be represented as a Simple Pseudo-Symmetric Binary Decision Diagram with any permutation of levels of nodes representing Shannon expansion, and levels of nodes representing flipped Shannon expansion.

#### V. PSEUDO-SYMMETRIC FUNCTIONS

Now, we show that some functions which had to be represented with repeated-variable PSBDDs can be represented as PSBDDs without repeated variables by relaxing one of the restrictions in the PSBDD representation and implementation. We allow both Shannon and flipped Shannon decompositions to be present on the same level as shown in Fig. 7. Based on that we define a larger class of functions which can be represented by regular diagrams, Pseudo-Symmetric Binary Decision Diagram with no repeated variables, Simple PSBDDs. This reduces delay and area of the function representation. Such regular arrays can be represented as arrays of multiplexers or as a pseudo-symmetric network of pass transistors.

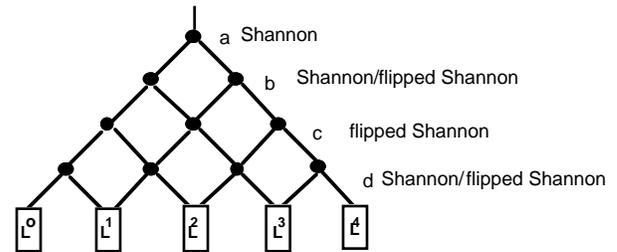


Figure 7. Simple Pseudo-Symmetric Binary Decision Diagram for pseudo-symmetric functions.

Generating functions which belong to these new classes can be done using their representation as simple regular arrays. All classes can be generated systematically, by creating a diagram which represent a specific class of functions like for example pseudo-symmetric functions, shown in Fig. 7. We assign a value of 0 or 1 to each of the pseudo-symmetric planes,  $L^k$ , and for all possible permutaions we generate  $2^n-2$  functions (totally symmetric functions), by including all possible permutations of variables polarities  $2^n$ , totally symmetric functions with mixed polarities are generated. Next, we allow all permutation of feasible expansions (Shannon and flipped Shannon)  $2^n 2^{n-1} \dots 2^2$ , to generate all pseudo-symmetric functions.

Table 1.

var	Symm. Fun.	Symm. Fun. Mixed	Pseudo-Symm. Fun.	Total Fun
2	$2^{2+1}-2$	$(2^{2+1}-2) 2^2 > 16$	$(2^{2+1}-2) 2^1 > 16$	$2^4$
3	$2^{3+1}-2$	$(2^{3+1}-2) 2^3 \approx 2^7$	$(2^{3+1}-2) 2^3 2^5 \approx 2^{12}$	$2^8$
4	$2^{4+1}-2$	$(2^{4+1}-2) 2^4 \approx 2^9$	$(2^{4+1}-2) 2^4 2^9 \approx 2^{18}$	$2^{16}$
5	$2^{5+1}-2$	$(2^{5+1}-2) 2^5 \approx 2^{11}$	$(2^{5+1}-2) 2^5 2^{14} \approx 2^{25}$	$2^{32}$
6	$2^{6+1}-2$	$(2^{6+1}-2) 2^6 \approx 2^{15}$	$(2^{5+1}-2) 2^5 2^{19} \approx 2^{30}$	$2^{64}$

In Table 1. we present estimated numbers of functions which belong to the different classes of functions discussed in this paper. The number of variables is given in column one and the number of functions which belong to the class of totally symmetric functions is given in column two. Column three gives the number of totally symmetric functions with mixed polarities, and column four the number of newly defined pseudo-symmetric functions. The total number of functions for different number of variables is given in column five. As can be observe all functions on up to 4 variables belong to some of the classes of symmetric or pseudo-symmetric functions. For more than five variables a number of pseudo-symmetric functions, defined in this paper, is smaller than the total number of completely specified functions, but still relatively large. Furthermore, we can extend the class of pseudo-symmetric functions by adding single-variable symmetries and constant cofactors [7].

## VI. CONCLUSION

Identification of the extended classes of functions which can be represented as regular simple diagrams, without repeated variables, has been the major step in making these diagrams practical for real life functions. In addition, a function decomposer can be used, as a preprocessor, to decompose large functions into smaller functions with the regularity property. The delay of a circuit designed as a PSBDD without repeated variables cannot be larger than a

unit delay of a cell used for its implementation (for example a multiplexer) multiplied by a number of function's variables. So, such calculated delay can be used as the upper limit for the circuit delay. To decrease the delay we can cover these regular diagrams with cells from the designated cell library while maintaining local connectivity. Such procedure allows us to predict very accurately the upper limits on area, power and delay during a logic synthesis step, which is much before the layout of the design is created.

In addition, for future three-dimensional IC structus the importance of local connections cannot be underestimated. A layout design, relative positions of gates and wires, is known before the layout is completed. The main advantages of these various simple regular diagrams are localized connections, predictable delay, and no placement or routing required. In addition, a known, in a pre-layout phase, interconnection structure can be used to predict power dissipation associated with wires. Methods for generating such simples regular pseudo-symmetric diagrams for unknown functions by utilizing all the symmetry relations are currently being developed. Our experiments on MCNC benchmark functions are very promising.

## References:

1. M.Chrzanowska-Jeske, Z.Wang, "Mapping of Symmetric and Partially Symmetric Functions to the CA-Type FPGAs," *Proc. 38th MWSCS'95*, pp. 290-293, August, 1995.
2. R.Bryant, "Graph Based Algorithms for Boolean Function Manipulation", *IEEE Trans. Comp.*, vol. C-35, pp 677-691, 1986.
3. S.Panda, F. Somenzi, and B.F. Plessier, "Symmetry Detection and Dynamic Variable Ordering of Decision Diagrams," *Proc. of DAC*, pp. 628-631, 1994.
4. C.C. Tsai, and M. Marek-Sadowska, "Generalized Reed-Muller Forms as a Tool to Detect Symmetries," *IEEE Trans.on Computers.*, vol. 45, no.1, pp.33-40, 1996.
5. Zvi Kohavi, "Switching and Finite Automata Theory", McGraw-Hill Inc. 1978.
6. M. Chrzanowska-Jeske, Z. Wang, Y. Xu, "A Regular Representation for Mapping to Fine-Grain, Locally-Connected FPGAs," *ISCAS'97*, pp.2749-2752, 1997.
7. W. Wang, M. Chrzanowska-Jeske, "Optimizing Pseudo-Symmetric Binary Decision Diagrams using Multiple Symmetries," *Proc. IWLS'98* pp. 334-340, June 1998.