

# Remarks on Calculation of Autocorrelation on Finite Dyadic Groups by Local Transformations of Decision Diagrams

Radomir S. Stanković<sup>1</sup> and Mark G. Karpovsky<sup>2</sup>

<sup>1</sup> Dept. of Computer Science, Faculty of Electronics  
Beogradska 14, 18 000 Niš, Serbia

<sup>2</sup> Dept. of Electrical and Computer Engineering, Boston University  
8 Saint Marry's Street, Boston Ma 02215, USA

**Abstract.** The paper considers calculation of autocorrelation functions on finite dyadic groups over decision diagrams. The methods exploit recursive structure of both autocorrelation matrices and decision diagrams. First, it is discussed calculation of the autocorrelation through the Wiener-Khinchin theorem implemented over decision diagrams. Then, it is proposed a method for calculation of separate autocorrelation coefficients over decision diagrams with permuted labels at the edges. For the case of restricted memory resources, a procedure with in-place calculations over the decision diagram for the function processed has been defined.

## 1 Introduction

Autocorrelation is an important operation in signal processing and systems theory [1], [2]. In particular, the autocorrelation on finite dyadic groups, denoted as dyadic autocorrelation  $B_f$ , (see Definition 1) is useful in switching theory and design of systems whose inputs and outputs are represented by functions defined in  $2^n$ ,  $n \in N$  points, including switching functions as an example [6], [8], [7], [11], [14], [16], [17], [19], [20], [25]. Recently, some new applications of dyadic autocorrelation in spectral methods for switching functions, [5], testing of logic networks [9], and optimization of decision diagrams (DDs) for representation of discrete functions have been reported [21].

In this paper, we define and discuss a method for calculation of the dyadic autocorrelation through decision diagrams, the use of which permits processing of functions of a large number of variables. Then, we discussed calculation of separate autocorrelation coefficients over decision diagrams with permuted labels of the edges. In case of restricted memory resources, these calculations can be performed by traversing in a suitable way the decision diagram for the function whose autocorrelation coefficients are required.

## 2 Background Theory

Denote by  $C_2^n$  the finite dyadic group, where  $C_2^n = \times_{i=1}^n C_2$ , and  $C_2 = (\{0, 1\}, \oplus)$ , where  $\oplus$  denotes multiplication modulo 2 (EXOR).

**Definition 1** For a function  $f : C_2^n \rightarrow R$ , where  $R$  is the field of real numbers, the autocorrelation  $B_f$  is defined by  $B_f(\tau) = \sum_{x=0}^{2^n-1} f(x)f(x \oplus \tau)$ , where  $\tau = 0, \dots, 2^n - 1$ . In binary notation,  $x = (x_1, \dots, x_n)$  and  $\tau = (\tau_1, \dots, \tau_n)$ , where  $x_i, \tau_i \in \{0, 1\}$ .

In matrix notation, if a given function  $f$  and the corresponding autocorrelation function  $B_f$  for  $f$  are represented by vectors  $\mathbf{F} = [f(0), \dots, f(2^n - 1)]^T$  and  $\mathbf{B}_f = [B_f(0), \dots, B_f(2^n - 1)]^T$ , respectively, then,

$$\mathbf{B}_f = \mathbf{B}(n)\mathbf{F},$$

where  $\mathbf{B}(n)$  is the dyadic autocorrelation matrix for  $f$ . The recursive structure of the autocorrelation matrix will be exploited in calculation of the autocorrelation coefficients.

The Walsh transform for functions on  $C_2^n$  is defined by the Walsh matrix

$$\mathbf{W}(n) = \bigotimes_{i=1}^n \mathbf{W}(1),$$

where  $\otimes$  denotes the Kronecker product, and  $\mathbf{W}(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$  is the basic Walsh matrix [6].

The relationship between the autocorrelation function and Walsh coefficients can be expressed as [6]

$$B_f = 2^n W^{-1}(Wf)^2,$$

where  $W$  denotes the Walsh transform operator.

This theorem we are using is called the Wiener-Khinchin theorem in classical Fourier analysis, and by this analogy the same term is used also in this paper. However, it seems that this theorem for the Walsh transform, was formulated for the first time by Franz Pichler in the paper [12], and also, in a mathematically more satisfying paper based on *sal* and *cal* functions in [13].

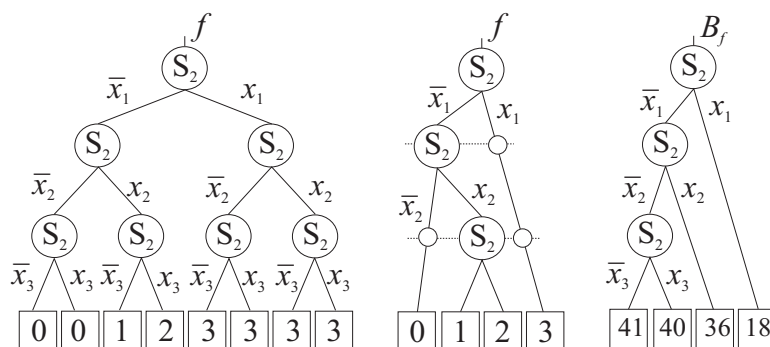
## 3 Decision Diagrams

Decision diagrams are data structures providing compact representations of discrete functions defined in a large number of points [22]. In this paper, we assume that a given function  $f$  with binary-valued variables is represented by a Multi-terminal Binary DD (MTBDD( $f$ )) [4], [22]. A MTBDD is a directed acyclic graph consisting of non-terminal nodes and constant nodes connected by edges.

Each node has two outgoing edges labeled by the negative and positive literals  $\bar{x}_i$  and  $x_i$  of the decision variable assigned to the node. Nodes to which the same variable is assigned form a level in the MTBDD.

If  $f$  is a switching binary-valued function, instead of MTBDDs [4], Binary decision diagrams (BDDs) [3] are used, since there are two possible values for constant nodes. MTBDDs and BDDs are derived by the reduction of the Multi-terminal binary decision trees (MTBDTs) and Binary decision trees (BDTs), respectively. The reduction is performed by deleting the redundant information and sharing isomorphic subtrees in the MTBDT, respectively BDT, for a given function  $f$  [22]. Notice that in calculations over decision diagrams, the impact of the deleted nodes should be taken into account through the cross points defined as points of intersections of paths from the root node to the constant nodes with the imaginary lines showing levels in decision diagrams, which means lines connecting nodes to which the same decision variable is assigned [24]. Complexity of a decision diagram is usually expressed in terms of the number of non-terminal and constant nodes, called the size of the decision diagram. In this paper, the notion of MTBDTs and MTBDDs will be introduced by the following example.

**Example 1** Fig. 1 shows a MTBDT, the corresponding MTBDD, and the MTBDD of the autocorrelation function  $B_f(\tau)$  for the function  $f$  of  $n = 3$  variables, which is given by the vector  $\mathbf{F} = [0, 0, 1, 2, 3, 3, 3, 3]^T$ . In this figure, we also show the cross points in the MTBDD for  $f$ .



**Fig. 1.** MTBDT, MTBDD, and the MTBDD for the autocorrelation function for  $f$  in Example 1.

## 4 Wiener-Khinchin Theorem over Decision Diagrams

The Walsh spectrum  $S_f$  of a given function  $f$  represented by a MTBDD is determined by performing at each node and the cross point of the MTBDD( $f$ ) the calculations determined by  $\mathbf{W}(1)$ . For simplicity, we say the nodes and cross points

in  $MTBDD(f)$  are processed by  $\mathbf{W}(1)$ . In this way,  $MTBDD(f)$  is converted into the  $MTBDD(S_f)$ . We perform the multiplication of  $S_f$  by itself by replacing the values of constant nodes  $S_f(i)$  with  $S_f^2(i)$  [22]. Then, the  $MTBDD(B_f)$  is determined by performing the calculations determined by  $\mathbf{W}(1)$  at each node and the cross point of the resulting  $MTBDD(S_f)$  followed by the normalization with  $2^n$ , since the Walsh matrix is self-inverse up to the constant  $2^{-n}$ .

#### 4.1 Complexity of the method

Since in calculation of the Walsh spectrum, we perform an addition and a subtraction at each node and the cross point distributed over  $n$  levels, the complexity is  $O(2n \cdot size(MTBDD(f)))$ . Notice that the number of cross points in a  $MTBDD$  is on the average at about 30% of the number of non-terminal nodes [22]. The result of these calculations is the  $MTBDD(S_f)$ . Then, we perform squaring of the values of constant nodes and perform the inverse transform. Thus, since the Walsh transform is self inverse, the complexity of these calculations is  $O(2n \cdot size(MTBDD(S_f)))$ . After multiplication with the scaling factor  $2^n$ , the  $MTBDD(B_f)$  is derived.

Notice that the size of the  $MTBDD$  for the Walsh spectrum is usually greater than that of the  $MTBDD$  for functions with a limited number of different values. Since in calculation of the autocorrelation function,  $MTBDD(f)$  is converted into a  $MTBDD(S_f)$ , which is in many cases larger in terms of size than the  $MTBDD(f)$ , the space complexity of the method is  $O(size(MTBDD(S_f)))$ .

For an illustration, Table 1 shows the sizes of  $MTBDD$ s and Walsh transform decision diagrams (WDDs) [24] for few standard *mcnc* benchmark functions used in logic design. Notice that, due to spectral interpretation of decision diagrams [23], WDDs are actually  $MTBDD$ s for the Walsh spectrum, and thus, this table provides a relevant information for these considerations. This table shows the number of inputs (In) of benchmark functions, number of non-terminal nodes (ntn), constant nodes (cn), size (s), and number of paths (paths) in the  $MTBDD$ s and WDDs.

**Example 2** For the function  $f$  represented by the  $MTBDD$  in Fig. 1, the Walsh spectrum is calculated as follows.

We first process the cross points and the node at the level for  $x_3$ . For the left cross point, calculation is trivial since the constant node shows the values 0, the result will be the zero valued vector of order 2. For the completeness of presentation, we show also these calculations

$$\mathbf{W}(1) \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 + 0 \\ 0 - 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

For the node for  $x_3$

$$\mathbf{W}(1) \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 + 2 \\ 1 - 2 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}.$$

**Table 1.** Characteristics of MTBDDs and WDDs for some benchmark functions.

$f$	In	MTBDD				WDD			
		ntn	cn	s	paths	ntn	cn	s	paths
5xp1	7	127	128	255	128	41	14	55	128
9sym	9	43	3	46	125	101	30	131	224
add4	8	147	31	178	256	36	11	47	37
add5	10	387	63	450	1024	55	13	68	56
apex4	9	446	319	765	450	511	512	1023	512
bw	5	29	24	53	30	31	32	63	32
clip	9	339	33	372	498	449	170	619	464
con1	7	46	5	51	83	83	26	109	96
ex1010	10	899	178	1077	1887	1023	972	1995	1024
mul2	4	13	7	20	14	12	8	20	13
mul3	6	59	26	85	59	30	16	46	31
rd53	5	21	6	27	24	30	13	43	32
rd73	7	57	8	25	96	64	24	88	98
rd84	8	85	9	94	192	118	40	158	193
sao2	10	96	11	107	237	295	70	365	508
sqrt8	8	64	17	81	65	127	54	181	176
xor5	5	15	3	18	22	9	6	15	10
av.	7.55	163.44	49.05	210.27	297.72	167.50	111.72	279.22	201.33

For the right cross point

$$\mathbf{W}(1) \begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 3+3 \\ 3-3 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \end{bmatrix}.$$

Then, we process the node for  $x_2$

$$\mathbf{W}(1) \circ \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 3 \\ -1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 3 \\ -1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ -1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \\ -3 \\ 1 \end{bmatrix}$$

where  $\circ$  symbolically denotes multiplication of a matrix by a vector consisting of subvectors.

For the cross point at the level for  $x_2$

$$\mathbf{W}(1) \circ \begin{bmatrix} \begin{bmatrix} 6 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 6 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 6 \\ 0 \end{bmatrix} + \begin{bmatrix} 6 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 6 \\ 0 \end{bmatrix} - \begin{bmatrix} 6 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 12 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

For  $x_1$ ,

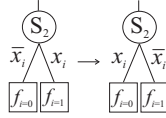
$$\mathbf{W}_1 \circ \begin{bmatrix} \begin{bmatrix} 3 \\ -1 \\ -3 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 12 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 3 \\ -1 \\ -3 \\ 1 \end{bmatrix} + \begin{bmatrix} 12 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 3 \\ -1 \\ -3 \\ 1 \end{bmatrix} - \begin{bmatrix} 12 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 15 \\ -1 \\ -3 \\ 1 \\ -9 \\ -1 \\ -3 \\ 1 \end{bmatrix}.$$

Thus determined vector is multiplied by  $1/8$  to get the Walsh spectrum for  $f$ .

Notice that matrix calculations are used for the explanations of the method. In practice, each step of the calculation is represented by a decision diagram which is a subdiagram in a decision diagram representing the Walsh spectrum for the function  $f$ .

## 5 In-place Calculation of Autocorrelation Coefficients

We define a transformation of nodes in MTBDDs that consists of permutation of labels at the outgoing edges as shown in Fig. 2 The  $i$ -th row of the autocor-

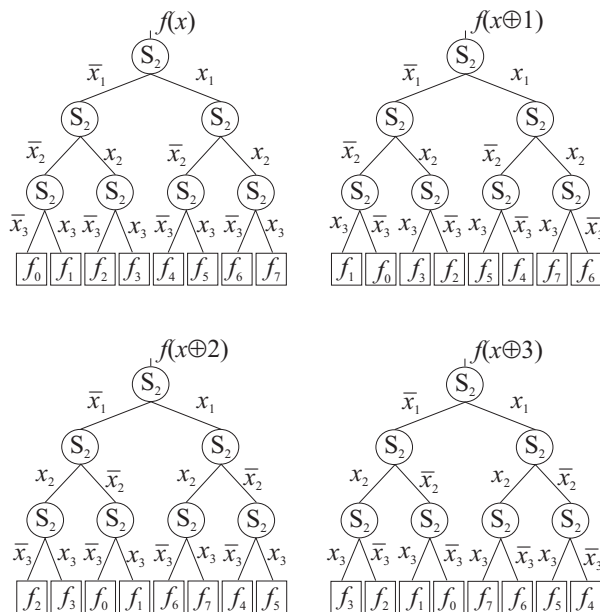


**Fig. 2.** Transformation of nodes.

relation matrix is the vector of function values  $f(x \oplus i)$ , where  $\oplus$  denotes the componentwise EXOR over the binary representations for  $x = (x_1, \dots, x_n)$ , and  $i = (i_1, \dots, i_n)$ . In decision diagrams, this shift of the argument for  $f$  implies permutation of labels at the edges of some nodes in the decision diagram for  $f$ . Nodes whose edges should be permuted are situated at the levels whose position within the decision diagram corresponds to the position of 1-bits in the binary representation for the row index  $i$ .

**Example 3** Fig. 3 shows MTBDDs for the first four rows of the autocorrelation matrix  $\mathbf{B}_f$  for a function of  $n = 3$  binary-valued variables.

The  $i$ -th autocorrelation coefficient is calculated by the multiplication of the  $i$ -th row of the autocorrelation matrix  $\mathbf{B}_f$  by the vector  $\mathbf{F}$  of function values for  $f$ . When  $f$  and rows of  $\mathbf{B}_f$  are represented by decision diagrams, it follows



**Fig. 3.** MTBDTs for the first four rows of the autocorrelation matrix for  $n = 3$ .

that the  $i$ -th autocorrelation coefficient is calculated by the multiplication of the decision diagrams for  $f$  and  $f(x \oplus i)$ . This can be performed by the classical procedure for multiplication of decision diagrams. However, since decision diagrams for  $f(x)$  and  $f(x \oplus i)$  differ in labels at the edges, in practical programming implementations, calculations can be organized over a single diagram similar as calculations of FFT can be organized in-place [1]. The complexity of calculation is proportional to the number of nodes in the decision diagram for  $f$ .

Fig. 4 shows a procedure for in-place calculation of the autocorrelation function through decision diagrams with permuted labels at the edges. In this procedure,  $f$  is represented by a MTBDD which is then traversed in such a way to multiply values of constant nodes in the MTBDD for  $f$  with the values of constant nodes in the MTBDD for  $f(x \oplus \tau)$  and perform the addition of these values. The way of traversing is determined by the binary components  $\tau_i$ ,  $i = 0, 1, \dots, n-1$  of  $\tau$ . A flag is associate to each non-terminal node, to show if the node was already traversed. In this manner, the coefficient  $B_f(\tau)$  is calculated. The procedure has to be repeated for each coefficient.

### 5.1 Complexity of in-place calculations

In-place calculations are performed over the  $MTBDD(f)$  and, therefore, the space complexity is  $O(size(MTBDD(f)))$ . Since for each coefficient we perform a multiplication at each constant node and an addition at each non-terminal

```

int AUTOCORREL(*node1, *node2, level)
{
    r = level - node → level
    if (node NOT TERMINAL)
    {
        if (node → flag = 0)
        {
            if(τi = 1)
            {
                pom1 = node2 → right
                pom2 = node2 → left
            }
            else
            {
                pom1 = node2 → left
                pom2 = node2 → right
            }
            i = i + 1
            a = AUTOCORREL(node1 → left, pom1)
              + AUTOCORREL(node1 → right, pom2)
            node → sub - value
            node → flag = 1
            return (2r-1 · a)
        }
        else
        {
            return node → sub - value
        }
    }
    else
    a = node1 → value · node2 → value
    return (2r-1 · a)
}
End of pseudocode.

```

**Fig. 4.** Calculation of the autocorrelation coefficient  $B_f(\tau)$ .



node, the number of multiplications is  $O(cn)$ , and the number of additions is  $O(n tn)$ , where  $cn$  and  $n tn$  are the number of constant and non-terminal nodes, respectively. Therefore, the total complexity of in-place calculation of an autocorrelation coefficient is  $O(\text{size}(\text{MTBDD}(f)))$ . Table 1 shows number of constant nodes and non-terminal nodes in the considered set of benchmark functions. The procedure is performed for each coefficient. Thus, it is suitable for calculation of a single coefficient or a subset of coefficients.

## 6 Closing Remarks

In this paper, we discussed calculation of autocorrelation functions over decision diagram representations of functions with binary-valued variables. Two approaches are considered, calculation of the autocorrelation functions by performing Wiener-Khinchin theorem over decision diagrams, and in-place calculations by decision diagrams with permuted labels at the edges. In the implementation of the Wiener-Khinchin theorem, the complete autocorrelation function is determined and represented by a decision diagram. The time complexity of calculations is  $O(2n \cdot \text{size}(\text{MTBDD}(f)))$ , and since the interim calculations involve determination of the Walsh spectrum, which is also represented by a decision diagram, the space complexity is maximum of  $O(\text{size}(\text{MTBDD}(f)))$  and  $O(\text{size}(\text{MTBDD}(S_f)))$ .

Calculation over decision diagrams with permuted edges permits determination of a single coefficient with both space and time complexity proportional to the size of the diagram for a given function  $f$ .

## Acknowledgments

This work was supported by the Academy of Finland, Finnish Center of Excellence Programme, Grant No. 44876.

## References

1. Aghaian, S., Astola, J., Egiazarian, K., *Binary Polynomial Transforms and Nonlinear Digital Filters*, Marcel Dekker, 1995.
2. Bracewell, R., "The Autocorrelation Function", in *The Fourier Transform and Its Applications*, 3rd ed., McGraw-Hill, New York, 1999, 40-45.
3. Bryant, R.E., "Graph-based algorithms for Boolean functions manipulation," *IEEE Trans. Comput.*, Vol. C-35, No. 8, 1986, 667-691.
4. Clarke, E. M., M.C., Millan, K.L., Zhao, X., Fujita, M., "Spectral transforms for extremely large Boolean functions", in Keschull, U., Schubert, E., Rosenstiel, W., Eds., *Proc. IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expression in circuit Design*, Hamburg, Germany, September 16-17, 1993, 86-90. Workshop Reed-Muller'93, 86-90.
5. Falkowski, B.J., Chang, C.H., "Properties and methods of calculating generalized arithmetic and adding transforms", *IEE Proc. Circuits, Devices and Systems*, Vol. 144, No. 5, 1997, 249-258.

6. Karpovsky, M.G., *Finite Orthogonal Series in the Design of Digital Devices*, John Wiley, 1976.
7. Karpovsky, M.G., (ed.), *Spectral Techniques and Fault Detection*, Academic Press, 1985, 35-90.
8. Karpovsky, M.G., Moskalev, E.S., "Utilization of autocorrelation characteristics for the realization of systems of logical functions", *Avtomatika i Telemekhanika*, No. 2, 1970, 83-90, English translation *Automatic and Remote Control*, Vol. 31, 1970, 342-350.
9. Karpovsky, M.G., Stanković, R.S., Astola, J.T., "Spectral techniques for design and testing of computer hardware", *Proc. Int. Workshop on Spectral Techniques in Logic Design, SPECLOG-2000*, Tampere, Finland, June 2-3, 2000, 1-34.
10. Meinel, Ch., Somenzi, F., Tehobald, T., "Linear shifting of decision diagrams and its application in synthesis", *IEEE Trans. CAD*, Vol. 19, No. 5, 2000, 521-533.
11. Pichler, F., "Walsh functions and linear system theory", *Proc. Applic. Walsh Functions*, Washington, D.C., 1970, 175-182.
12. Pichler, F., *Some Aspects of a Theory of Correlation with respect to Walsh Harmonic Analysis*, Techn. Report. Dept. of Electrical Engineering, University of Maryland, Washington, DC, Report R-70-11, August 1970.
13. Franz Pichler, "Walsh-Fourier Synthese optimaler Filter", *AEÜ*, Band 24, 1970, Heft 7/8, 350-360.
14. Pichler, F., "Realizations of Prigogine's  $A$ -transform by dyadic convolution", in Trapp, R., Horn, W., (eds.), *Austrian Society for Cybernetic Studies*, ISBN 385206127X, 1992.
15. Rice, J.E., *Autocorrelation Coefficients in Representation and Classification of Switching Functions*, Ph.D. Dissertation, University of Victoria, Victoria, B.C., Canada, 2003.
16. Rice, J.E., Muzio, J.C., "Methods for calculating autocorrelation coefficients", in *Proc. 4th Int. Workshop on Boolean Problems, (IWBP2000)*, 2000, 6976.
17. Rice, J.E., Muzio, J.C., "Use of autocorrelation function in the classification of switching functions", *Euromicro Symposium on Digital System Design*, 2002, 244-251.
18. Rice, J.E., Muzio, J.C., "Properties of autocorrelation coefficients", *Proc. IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing*, 2003.
19. Rice, J.E., Muzio, J.C., "On the use of autocorrelation coefficients in the identification of three-level decompositions", *Proc. PAC 2003*, 2003, 577-580.
20. Rice, J.E., Muzio, J.C., "On the use of autocorrelation coefficients in the identification of three-level decompositions", *Proc. Int. Workshop on Logic Synthesis, (IWLS 2003)*, 2003.
21. Rice, J., Serra, M., Muzio, J.C., "The use of autocorrelation coefficients for variable ordering for ROBDDs", *Proc. 4th Int. Workshop on Applications of Reed-Muller Expansion in Circuit Design*, Victoria, Canada, August 20-21, 1999, 185-196.
22. Sasao, T., Fujita, M., (eds.), *Representations of Discrete Functions*, Kluwer, 1996.
23. Stanković, R.S., Astola, J.T., *Spectral Interpretation of Decision Diagrams*, Springer, 2003.
24. Stanković, R.S., Sasao, T., Moraga, C., "Spectral transform decision diagrams", in: T. Sasao, M. Fujita, Ed., *Representations of Discrete Functions*, Kluwer Academic Publishers, 1996, 55-92.
25. Tomczuk, R., *Autocorrelation and Decomposition Methods in Combinational Logic Design*, PhD thesis, University of Victoria, 1996.