CONSTRUCTION OF LINEARLY TRANSFORMED BINARY DECISION DIAGRAMS BY TOTAL AUTOCORRELATION FUNCTIONS

Mark G. Karpovsky*
Dept. of Electrical and Computer Engineering
8 Saint Marry's Street
Boston University
Boston, Ma 02215, USA

Radomir S. Stanković
Dept. of Computer Science
Beogradska 14
Faculty of Electronics
18 000 Niš
Yugoslavia

Jaakko T. Astola
Tampere Int. Center for Signal Processing
Tampere University of Technology
Tampere
Finland

ABSTRACT

This paper discusses optimization of decision diagram (DD) representations of switching functions by total autocorrelation functions. We present an efficient algorithm for construction of Linearly Transformed Binary Decision Diagrams (LT-BDDs) based on linearization of the corresponding Boolean functions by their logical total autocorrelation functions.

1 Introduction

Decision diagrams (DDs) are a data structure permitting efficient representation of discrete functions defined on groups of large orders [15]. Different DDs are defined for representation of different classes of discrete functions by using different decomposition rules to assign a given function f to a DD, [15], [17], [19]. In this paper, the considerations are restricted to two basic DDs for functions on finite dyadic groups $C_2^n = (\{0,1\}, \oplus)$, where $\oplus$ denotes the addition modulo 2, EXOR. Binary DDs (BDDs) are the basic concept used to represent single output switching functions [1]. A given function f is assigned to a BDD through the recursive application of the Shannon decomposition rule $f = \overline{x}_i f_0 \oplus x_i f_1$, where f_0 and f_1 are co-factors of f for $x_1 = 0$, and $x_i = 1$, respectively. Multiple-output switching functions are represented by Shared BDDs [11]. Multi-terminal binary DDs (MTBDDs) [2] are used to represent functions in the space $C(C_2^n)$ of functions $f: C_2^n \rightarrow C$, where C is the field of complex numbers. MTBDDs can represent systems of Boolean functions described by the corresponding integer equivalent functions $f(x)$ [6]. However, extensions and generalizations of the considerations presented to DDs for functions on arbitrary not-necessarily Abelian groups are straightforward. The multiple-valued logic functions are included as an example of functions on p-adic groups C_p into $GF(p)$, $p \in N$ [6].

DDs are derived by the reduction of decision trees (DTs). The reduction is performed by sharing the isomorphic subtrees and deleting the redundant information from the DT. The reduction procedure is formalized through the reduction rules [15] adapted to the range of functions represented and the used decomposition rules.

In many applications, the efficiency of DD representations is determined by the size of the DD defined as the number of nodes in the DD for a given f. The width of the DD is defined as the maximal number of nodes at a level, where a level consist of nodes to which the same variable is assigned. The size and the width determine the area of the DD, which is also an important parameter in applications and comparisons of different DDs [18].

Linearly transformed BDDs (LT-BDDs) are a generalization of BDDs derived by using the Shannon expansion with respect to a linear combination of a subset of variables. It is obvious that the same generalization applies to MTBDDs, since in BDDs and MTBDDs the same underlying group of binary vectors is assumed as the domain for the functions represented. Moreover, extensions to Spectral transform DDs (STDDs), are straightforward.

The simplest example showing the efficiency of the method for some classes of functions, is to compare BDD and LT-BDD for the two-variable EXOR function $f(x_1, x_2) = x_1 \oplus x_2$. Since the truth-vector of f is $\mathbf{F} = [0110]^T$, BDD(f) consists of three non-terminal nodes, two nodes to which x_2 is assigned, and the root node for x_1. Thus, EXOR requires the complete BDD. However, this function can be represented by a LT-BDD with a single non-terminal node to which the linear combination of variables $x_1 \oplus x_2$ is assigned. Construction of LT-BDDs is an interesting and important task, since LT-BDDs permit for some functions the exponential reduction of the size compared to the BDDs.

We note that a price for the reduced size of LT-BDDs is

1. A hardware required to implement a linear transformation of variables.

2. Difficulty to determine an optimal transformation of variables.

*This work was partially supported by the BSF Grant 1998154. The work of the second author was supported by TICSP, Tampere, Finland.

A linear transformation over the variables can be represented by an $(n \times n)$ matrix, and the required space can be neglected compared to the space required to store a BDD or a LT-BDD. However, the other requirement can be considered as a bottleneck for applications of LT-BDDs, although there are heuristic algorithms to determine a suitable linear combination of variables. The algorithm proposed in [10] splits the set of variables into subsets of adjacent variables and combines variables within a subset. A similar algorithm implemented as a windowing procedure is proposed in [3]. The algorithm for construction of LT-BDDs presented in [5] is an application of evolutionary computation techniques to this problem. In [10], the algorithm presented in [9] is combined with sifting method used in variable ordering in DDs with special attention paid to the integration of the method into the existing CAD systems. Algorithms for efficient manipulations with LT-BDDs, prepared as an extension of CUDD package [16] further support the applications of LT-BDDs [4].

In this paper, which is a continuation of the research reported in [8], we discuss applications of total autocorrelation functions to reduction of sizes of SBDDs. We show that the method for linearization of switching functions introduced in [7], and further discussed, elaborated, and extended to multiple-valued functions in [6] provides for a deterministic algorithm to determine the linear transformation of variables in LT-BDDs by the total autocorrelation functions and the inertia groups of the systems [6].

This paper is organized as follows. In Section 2, we briefly review basic definitions of autocorrelation functions and total autocorrelation functions. In Section 3, we review the method for linearization of systems of Boolean functions based on autocorrelation functions [6]. Section 4 is devoted to applications of this method to construction of LT-BDDs. In Section 5, we present some experimental results to verify the method for constructing LT-BDDs and LT-MTBDDs. In Section 6, we provide some closing remarks showing further extensions of the methods presented.

2 Autocorrelation Function

Autocorrelation is very useful in spectral methods for analysis and synthesis of networks realizing logic functions.

For a given n-variable switching function f, the autocorrelation function B_f is defined as

$$B_f(\tau) = \sum_{x=0}^{2^n-1} f(x)f(x \oplus \tau), \quad \tau \in \{0, \dots, 2^n - 1\},$$

2.1 Total autocorrelation function

For a system of k switching functions $f^{(i)}(x_1, \dots, x_n)$, $i = 0, \dots, k-1$, the total autocorrelation function is defined as the sum of autocorrelation functions of each function in the system. Thus,

$$B_f(\tau) = \sum_{i=0}^{k-1} B_{f^{(i)}}(\tau).$$

Note that for any $\tau \neq 0$, $B_f(\tau) \leq B_f(0)$. The set $G_I(f)$ of all values for τ such that $B_f(\tau) = B_f(0) = \sum_{i=0}^{k-1} \sum_{x=0}^{2^m-1} f^{(i)}(x)$ is a group with respect to the EXOR as the group operation which is denoted as the inertia group of the system f.

A generalization of autocorrelation to systems of p-valued m-variable functions is straightforward. For a system of p-valued functions $f(x) = \{f^{(i)}(x_1, \dots, x_n)\}$, $i = 0, \dots, k-1$, $x_i \in \{0, \dots, p-1\}$, a system of characteristic functions is defined as

$$f_r^{(i)}(x) = \begin{cases} 1, & f^{(i)}(x) = r, \\ 0, & f^{(i)}(x) \neq r. \end{cases}$$

Then, the total autocorrelation function is defined as [6]

$$B_f(\tau) = \sum_{r=0}^{p-1} \sum_{i=0}^{k-1} B_r^{(i)}(\tau).$$

The same definition applies to integer valued-functions on p-adic groups, thus it can be used for integer-valued equivalent functions for a system of Boolean functions, since this is a particular example for $p = 2$. Further generalizations to functions on arbitrary finite Abelian groups are also straightforward, see for example, [6]. In this case, the Winer-Khinchin theorem is formulated with respect to the Fourier transforms defined in terms of group characters. It should be noted that this theorem is correct for the autocorrelation functions of discrete functions and characteristic functions associated to them, however, it does not apply to the related total autocorrelation functions.

2.2 Autocorrelation through DDs by using the Wiener-Khinchin Theorem

For applications of the autocorrelation functions considered in this paper, it is convenient to consider determination of the autocorrelation functions though MTB-DDs. Then, the total autocorrelation function is determined by the addition of MTBDDs for the autocorrelation functions for characteristic functions of the integer equivalent function $f(x)$ for the given multiple-output function $(f^{(0)}, \dots, f^{(k-1)})$, where $f(x) = \sum_{i=0}^{k-1} 2^i f^{(i)}$.

The Wiener-Khinchin theorem states a relationship between the autocorrelation function and Walsh (Fourier) coefficients [6]

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

where W denotes the Walsh transform operator. The Walsh transform 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.

The Walsh spectrum for f is determined by performing at each node of the MTBDD(f) the calculations determined by $\mathbf{W}(1)$. For simplicity, we say the nodes in MTBDD(f) are processed by $\mathbf{W}(1)$. In this way, MTBDD(f) is converted into the MTBDD(S_f), where S_f denotes the Walsh spectrum for f. We perform the multiplication of S_f by itself by using the standard procedure for multiplication of functions represented by DDs [15]. Then, the MTBDD(B_f) is determined by performing the calculations determined by $\mathbf{W}(1)$ at each node 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. Fig. 1 illustrate this procedure.

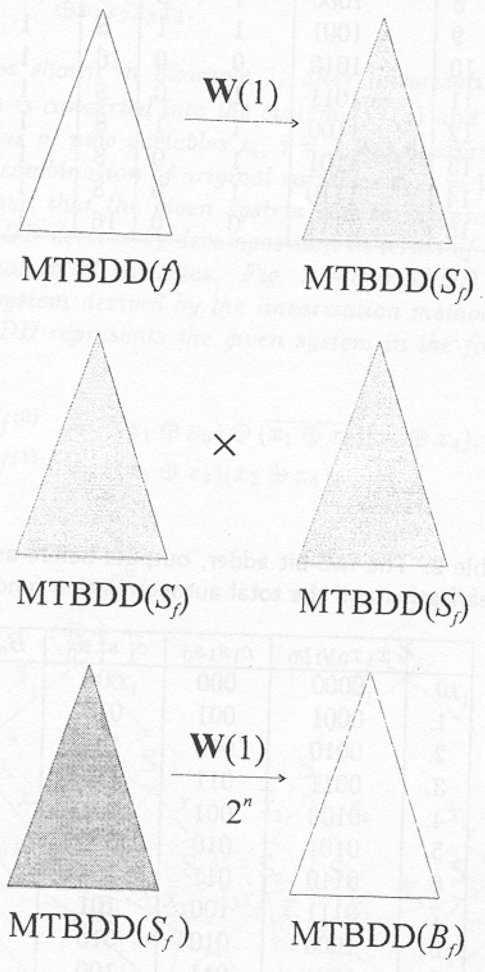

Figure 1: Wiener-Khinchin theorem.

3 Linearization of Switching Functions

Linear transformation of variables is a classical method for optimization of different representations of switching functions [6], [7], [14]. In spectral techniques, this method is studied in the context of spectral invariant operations and as a mean to reduce the number of non-zero coefficients in spectral expressions for discrete functions [6]. In [6], [12], and [13], the extensions to multiple-valued logic functions are discussed.

The linearization of switching functions (LSF) is defined as the assignment of a function $f_\sigma(z) = f(\sigma^{-1}x)$ to a given switching function f, where σ is a linear operator conveniently represented by an $(n \times n)$ non-singular matrix over $GF(2)$. It is assumed that f_σ is simpler or more convenient for a realization or calculation purpose than f with respect to some criteria of optimality. In [7], a method for solution of LSF-problem is presented. In this method, σ is determined such that minimizes the number of two-input circuits for realization of f_σ. The criterion to select σ is determined from the values of the total autocorrelation function $B_f(\tau)$.

The values $\tau = \tau_{max} \neq 0$ where $B_f(\tau)$ takes the maximum value determine the inertia group G_I for f [6]. Any basis of G_I provides a solution of the LSF-problem for f. The binary representations for τ_{max} determine some elements of a basis for G_I for f in the following way. Other elements can be taken arbitrarily providing the linear independence of elements in the basis. These elements are often taken as the corresponding rows of the trivial basis described by rows of the identity matrices. Vectors from the basis are written as columns of a matrix σ, whose inverse over $GF(2)$ determine a linear transform of variables for f which is a solution of LSF-problem. This method can be implemented by using the following algorithm [6].

Algorithm for LSF through B_f

1. Given an n-variable k-output function $f = (f^1, \ldots, f^k)$.
2. Represent f by the integer-valued equivalent function $f(x) = \sum_{i=1}^{k} 2^i f_i$.
3. Calculate the total autocorrelation function B_f and determine $G_I(f)$, where G_I is the inertia group for f.
4. Determine a basis for $G_I(f)$ and write it as columns of an $(n \times n)$ matrix σ.
5. Calculate for σ, the inverse over $GF(2)$, σ^{-1}.
6. Perform the mapping $z : x \to \sigma^{-1}x$, and determine $f_\sigma(z)$. *End of Algorithm.*

Example 1 *Table 1 shows a system of two four-variable Boolean functions $f^{(0)}$ and $f^{(1)}$, and the total autocorrelation function B of this system. The maximum value of $B(\tau) = B(0) = 16$ for the inputs $5 = (0101)$, $10 = (1010)$, and $15 = (1111)$. Thus, the inertia group for this system is $G_I =$*

$\{(0000), (0101), (1010), (1111)\}$. As a basis for G_I we take (0101) and (1010), and determine

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Then,

$$\sigma = \mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The matrix $\mathbf{T}$ determines a reordering of the truth-vector $\mathbf{F}$ of a four-variable function f as

$$\mathbf{F} = [f(0), f(10), f(5), f(15), f(4), f(14), f(1), f(11), f(8), f(2), f(13), f(7), f(12), f(6), f(9), f(3)]^T.$$

Table 1 shows functions $f_\sigma^{(0)}$ and $f_\sigma^{(1)}$ produced by this reordering from $f^{(0)}$ and $f^{(1)}$. Thus, these functions satisfy the relation

$$f_\sigma(x) = f(\sigma^{-1} \odot x),$$

where $\odot$ denotes the multiplication over $GF(2)$. Therefore, we determine the intermediate values z_1, z_2, z_3, z_4 in Fig. 2 from $\sigma = \mathbf{T}^{-1}$, as

$$\begin{aligned} z_1 &= x_1 \oplus x_3, \\ z_2 &= x_2 \oplus x_4 \\ z_3 &= x_4 \\ z_4 &= x_3. \end{aligned}$$

From there,

$$\begin{aligned} f^{(0)} &= z_1 \vee z_2 = (x_1 \oplus x_4) \vee (x_2 \oplus x_3) \\ f^{(1)} &= z_1 z_2 = (x_1 \oplus x_4)(x_2 \oplus x_3), \end{aligned}$$

where $\vee$ denotes logical OR. It should be noted that both $f^{(0)}$ and $f^{(1)}$ do not essentially depend on z_3 and z_4.

Example 2 *Table 2 shows the outputs of a two bit adder (c_1, s_1, s_0) and the total autocorrelation function of this adder $\mathbf{B}_{add2}(\tau)$, $\tau \in \{0, \ldots, 2^n - 1\}$. The maximum of B_{add2} is for $\tau_1 = 5 = (0101)$ and $\tau_2 = 10 = (1010)$. We write the matrix σ whose first two columns are equal to the first two columns in the identity matrix of order four, and the other two columns are binary representations for τ_1 and τ_2. Thus,*

$$\sigma = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

This is a self inverse matrix over $GF(2)$, and thus, $\sigma^{-1} = \sigma$. This matrix determines the linear transform of variables $(x_1, x_0, y_1, y_0) \to (x_1 \oplus y_1, x_0 \oplus y_0, y_1, y_0)$. Table 2 shows the outputs (c_1^T, s_1^T, s_0^T) of add2 transformed by using the matrix σ^{-1}.

Table 1: System of Boolean functions.

	$x_1x_2x_3x_4$	$f^{(0)}$	$f^{(1)}$	B	$f_\sigma^{(0)}$	$f_\sigma^{(1)}$
0	0000	0	0	16	0	0
1	0001	1	0	8	0	0
2	0010	1	0	8	0	0
3	0011	1	1	8	0	0
4	0100	1	0	8	1	0
5	0101	0	0	16	1	0
6	0110	1	1	8	1	0
7	0111	1	0	8	1	0
8	1000	1	0	8	1	0
9	1001	1	1	8	1	0
10	1010	0	0	16	1	0
11	1011	1	0	8	1	0
12	1100	1	1	8	1	1
13	1101	1	0	8	1	1
14	1110	1	0	8	1	1
15	1111	0	0	16	1	1

Table 2: The two-bit adder, outputs before and after the linearization and the total autocorrelation function B_{add2}.

	$x_1x_0y_1y_0$	$c_1s_1s_0$	$c_1^T s_1^T s_0^T$	$B_{add2}(\tau)$
0.	0000	000	000	16
1.	0001	001	010	0
2.	0010	010	100	0
3.	0011	011	110	0
4.	0100	001	001	0
5.	0101	010	001	8
6.	0110	011	101	0
7.	0111	100	101	4
8.	1000	010	010	0
9.	1001	011	100	0
10.	1010	100	010	8
11.	1011	101	100	0
12.	1100	011	011	0
13.	1101	100	011	4
14.	1110	101	011	0
15.	1111	110	011	4

4 Linearization of Boolean Functions and LT-BDDs

The linearization method for Boolean functions presented in Section 3 provides the efficient algorithm for linear transformation of BDDs and Shared BDDs for systems of Boolean functions [15]. This statement will be explained and illustrated by the following example taken from [8].

Example 3 *Fig. 2 shows SBDD for the system of Boolean functions $f^{(0)}(x)$ and $f^{(1)}(x)$ defined in Table 1. This SBDD represents the given system in the form of expressions*

$$
\begin{aligned}
f^{(0)} &= \overline{x}_1\overline{x}_2\overline{x}_3x_4 \oplus \overline{x}_1\overline{x}_2x_3 \oplus \overline{x}_1x_2x_3 \oplus \overline{x}_1x_2\overline{x}_3\overline{x}_4 \oplus \\
&\oplus x_1\overline{x}_2\overline{x}_3x_4 \oplus x_1x_2x_3\overline{x}_4 \oplus x_1\overline{x}_2\overline{x}_3 \oplus x_1x_2\overline{x}_3, \\
f^{(1)} &= \overline{x}_1\overline{x}_2x_3x_4 \oplus \overline{x}_1x_2x_3\overline{x}_4 \oplus x_1\overline{x}_2\overline{x}_3x_4 \oplus \\
&\oplus x_1x_2\overline{x}_3\overline{x}_4.
\end{aligned}
$$

As is shown in Example 1, after linearization, this system is converted into the system $f_\sigma^{(0)}(z)$ and $f_\sigma^{(1)}(z)$, in terms of new variables z_i, $i = 1, 2, 3, 4$ expressed as linear combination of original variables x_i, $i = 1, 2, 3, 4$. It follows that the given system can be represented by the SBDD derived by decomposition in terms of a linear combination of variables. Fig. 3 shows SBDD for the given system derived by the linearization method. This LT-SBDD represents the given system in the following form

$$
\begin{aligned}
f^{(0)} &= (x_1 \oplus x_3) \oplus (\overline{x_1 \oplus x_3})(x_2 \oplus x_4), \\
f^{(1)} &= (x_1 \oplus x_3)(x_2 \oplus x_4).
\end{aligned}
$$

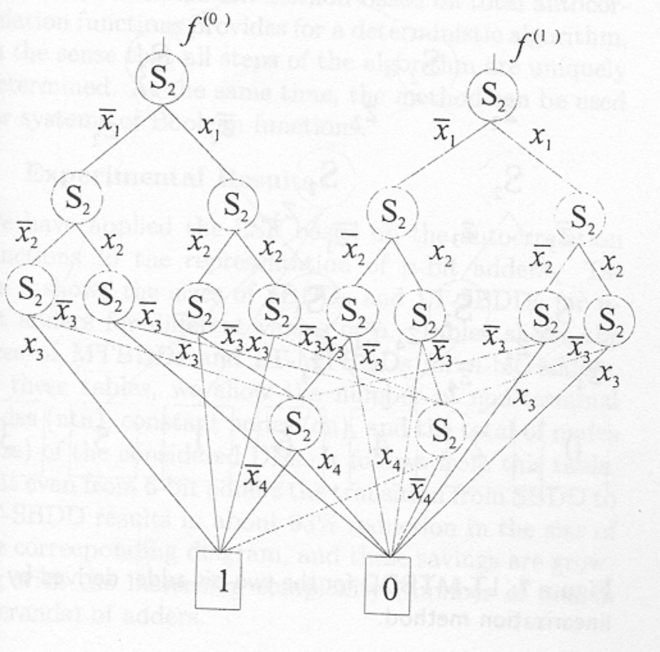

Figure 2: STBDD for the system of functions.

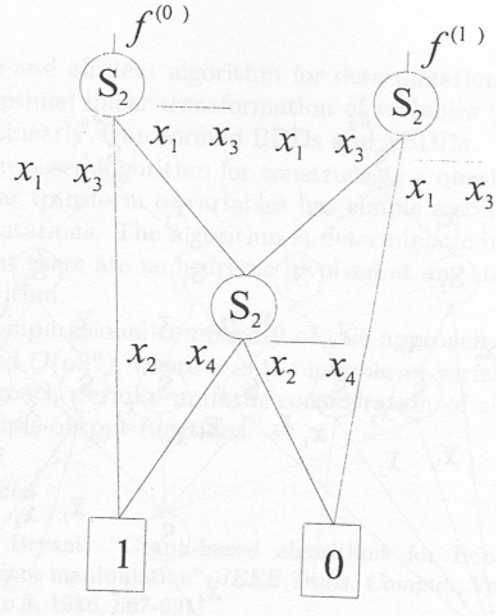

Figure 3: Shared LT-BDD for the system of functions derived by the linearization method.

The following example illustrates application of the LSF-method to representations of adders.

Example 4 *Fig. 4 shows the SBDD for the two-bit adder. Fig. 5 shows the corresponding LT-SBDD determined by using the matrix σ^{-1} in Example 2. If the outputs of the adder are considered as binary representations of the corresponding integers, then, they can be represented by the vector*

$$\mathbf{F} = [0, 1, 2, 3, 1, 2, 3, 4, 2, 3, 4, 5, 3, 4, 5, 6]^T$$

The reordering by using the matrix σ^{-1} converts this vector into the vector

$$\mathbf{F}_\sigma = [0, 2, 4, 6, 1, 1, 5, 5, 2, 4, 2, 4, 3, 3, 3, 3]^T$$

Fig. 6 and Fig. 7 show the MTBDD and LT-MTBDD for the two-bit adder, derived from $\mathbf{F}$ and $\mathbf{F}_\sigma$, respectively.

The following procedure for determination of a linear transformation of variables in LT-BDD for a given function f can be formulated.

Procedure for generation of LT-BDD

1. Given an n-variable k-output switching function $f = (f^{(0)}, \ldots, f^{(k-1)})$.
2. Represent f by the integer-valued equivalent function $f(x)$.
3. Perform the linearization procedure and assign to $f(x)$ a function $f_\sigma(z)$, where $z = \sigma \odot x$.

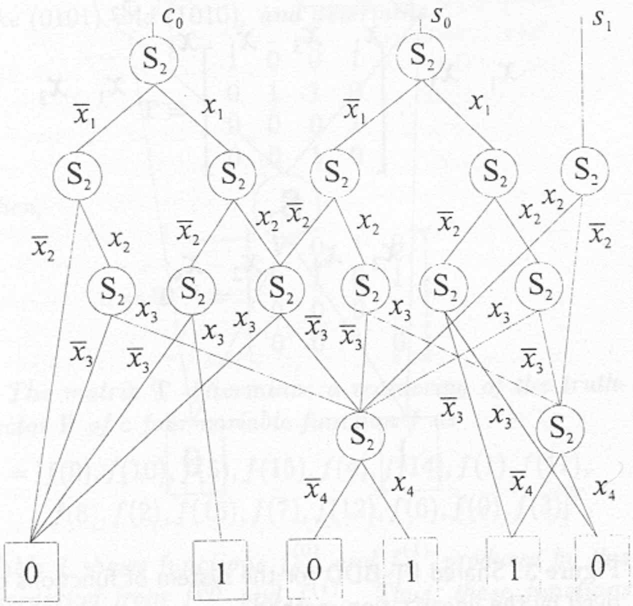

Figure 4: SBDD for the two-bit adder.

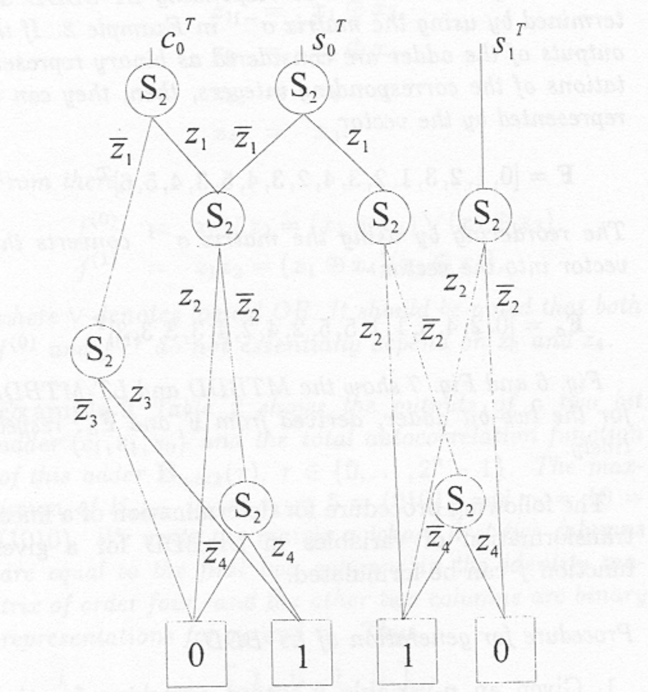

Figure 5: Shared LT-BDD for the two-bit adder derived by the linearization method.

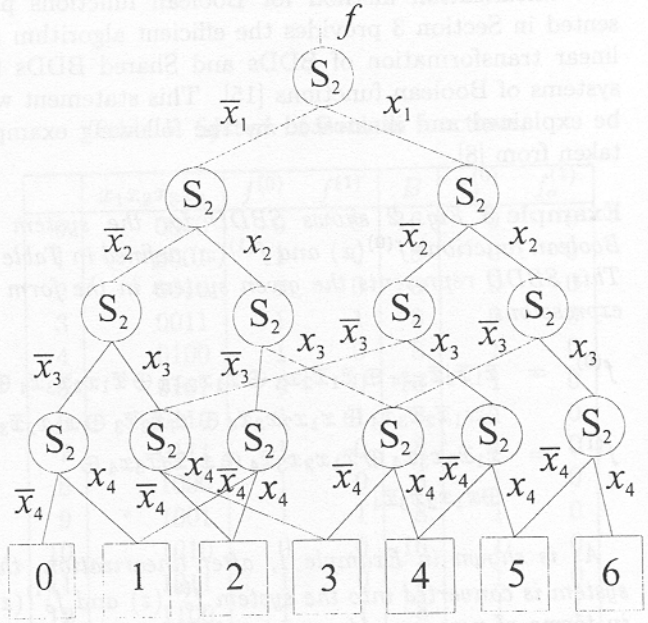

Figure 6: MTBDD for the two-bit adder.

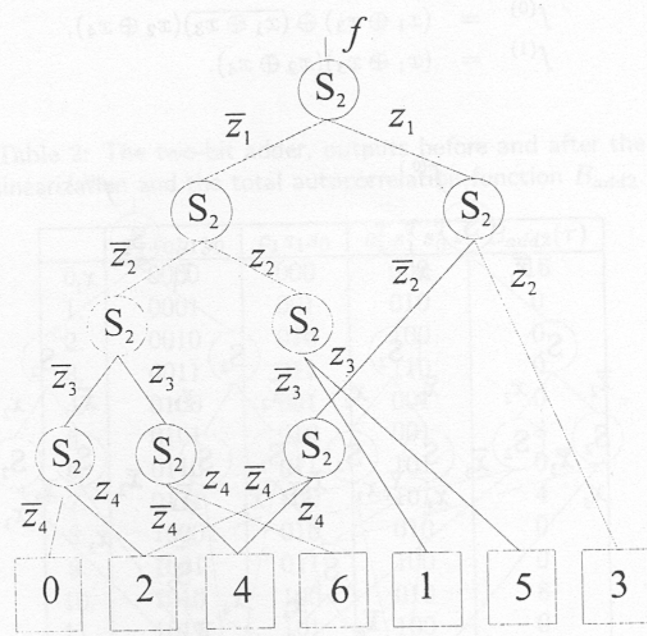

Figure 7: LT-MTBDD for the two-bit adder derived by the linearization method.

Table 3: Sizes of SBDDs and LT-BDDs for adders.

f	SBDD			LT-SBDD		
	ntn	cn	size	ntn	cn	size
add2	15	2	17	8	2	10
add3	42	2	44	13	2	15
add4	101	2	103	18	2	20
add5	224	2	226	23	2	25
add6	475	2	477	28	2	30

Table 4: Sizes of MTBDDs and LT-MTBDDs for adders.

f	SBDD			LT-SBDD		
	ntn	cn	size	ntn	cn	size
add2	13	7	20	8	7	15
add3	41	15	56	24	15	39
add4	113	31	144	64	31	95
add5	289	63	352	160	63	223
add6	705	129	832	384	127	511
add7	1665	255	1920	896	255	1151

4. Determine SBDD for $f_\sigma(z)$.
5. Relabel edges in SBDD($f_\sigma(z)$) by replacing each z_i with the corresponding linear combination of initial variables x_i.

End of Procedure.

Compared to the present methods for linear transformation of DDs, the advantage of the proposed algorithm is that the linearization method based on total autocorrelation functions provides for a deterministic algorithm, in the sense that all steps of the algorithm are uniquely determined. At the same time, the method can be used for systems of Boolean functions.

5 Experimental Results

We have applied the LSF based on the autocrrelation functions to the representation of n-bit adders. Table 3 shows the sizes of SBDDs and LT-SBDDs for n-bit adders for different values of n. Table4 shows the sizes of MTBDDs and LT-MTBDDs for n-bit adders. In these tables, we show the number of non-terminal nodes (ntn), constant nodes (cn), and the total of nodes (size) of the considered DDs. It follows from this table, that even from 6-bit adders the transition from SBDD to LT-SBDD results in about 93% reduction in the size of the correcponding diagram, and these savings are growing with the increasing complexity (number of bits in operands) of adders.

6 Closing Remarks

We have shown that the method for linearizaton of Boolean functions presented in this paper provides for a simple and efficient algorithm for determination of a nearly optimal linear transformation of variables in design of Linearly Transformed BDDs and SBDDs.

The proposed algorithm for constructing a quasioptimal linear transform of variables has simple a software implementations. The algorithm is deterministic in the sense that there are no heuristic involved at any step of the algorithm.

The computational complexity of this approach does not exceed $O(n2^n)$, where n is the number of variables. The approach permits uniform consideration of single and multiple-output functions.

References

[1] R.E. Bryant, "Graph-based algorithms for Boolean functions manipulation", *IEEE Trans. Comput.*, Vol.C-35, No.8, 1986, 667-691.

[2] E.M. Clarke, K.L. Mc Millan, X. Zhao, M. Fujita, "Spectral transforms for extremely large Boolean functions", in Kebschull, 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.

[3] W. Günther, R. Drechsler, "BDD minimization by linear transforms", in *Advanced Computer Systems*, 1998, 525-532.

[4] W. Günther, R. Drechsler, "Efficient manipupation algorithms for linearly transformed BDDs", *Proc. 4th Int. Workshop on Applications of Reed-Muller Expansion in Circuit Design*, Victoria, Canada, May 20-21, 1999, 225-232.

[5] W. Günther, R. Drechsler, "Minimization of BDDs using linear transformations based on evolutionary techniques", *Proc. Int. Symp. Circuit and Systems*, 1999.

[6] M.G. Karpovsky, *Finite Orthogonal Series in the Design of Digital Devices*, John Wiley, 1976.

[7] M.G. Karpovsky, E.S. Moskalev, "Utilization of autocorrelation characteristics for the realization of systems of logical functions", *Avtomatika i Telmekhanika*, No. 2, 1970, 83-90, English translation *Automatic and Remote Control*, Vol. 31, 1970, 342-350.

[8] Karpovsky, M.G., Stanković, R.S., Astola, J.T., "Spectral techniques for design and testing of computer hardware", *Proc. Int. Worksop on Spectral Techniques in Logic Design, SPECLOG-2000*, Tampere, Finland, June 2-3, 2000, 1-34.

[9] C. Meinel, F. Somenzi, T. Theobald, "Linear sifting of decision diagrams", *Proc. Design Automation Conference*, 1997, 202-207.

[10] Ch. Meinel, F. Somenzi, T. Tehobald, "Linear sifting of decision diagrams and its application in synthesis", *IEEE Trans. CAD*, Vol. 19, No. 5, 2000, 521-533.

[11] S. Minato, "Graph-based representations of discrete functions", in [15], 1-28.

[12] C. Moraga, "Introducing disjoint spectral translation in spectral multiple-valued logic design", *IEE Electronics Letters*, 1978, Vol. 14, No. 8, 1978, 248-243.

[13] C. Moraga, "On some applications of the Chrestenson functions in logic design and data processing", *Mathematic and Computers in Simulation*, Vol. 27, 1985, 431-439.

[14] E.I. Nechiporuk, "On the synthesis of networks using linear transformations of variables", *Dokl. AN SSSR.* Vol. 123, No. 4, December 1958, 610-612.

[15] T. Sasao, M. Fujita, (eds.), *Representations of Discrete Functions*, Kluwer, 1996.

[16] F. Somenzi, *CUDD - Colorado University Decision Diagram Package*, 1996.

[17] R.S. Stanković, *Spectral Transform Decision Diagrams in Simple Questions and Simple Answers*, Nauka, Belgrade, 1998.

[18] R.S. Stanković, "Some remarks on basic characteristics of decision diagrams", *Proc. 4th Int. Workshop on Applications of Reed-Muller Expansion in Circuit Design*, August 20 - 21, 1999, 139-146.

[19] R.S. Stanković, T. Sasao, "Decision diagrams for representation of discrete functions: uniform interpretation and classification", *Proc. ASP-DAC'98*, Yokohama, Japan, February 13-17, 1998.