ISSN 0005-1179, Automation and Remote Control, 2011, Vol. 72, No. 3, pp. 615–625. © Pleiades Publishing, Ltd., 2011.
Original Russian Text © O. Keren, I. Levin, R. Stankovič, 2011, published in Avtomatika i Telemekhanika, 2011, No. 3, pp. 161–172.

LOGIC CONTROL

Linearization of Logical Functions Defined by a Set of Orthogonal Terms. I. Theoretical Aspects

O. Keren,* I. Levin, and R. Stankovič*****

* *Bar Ilan University, Ramat Gan, Israel*
** *Tel Aviv University, Tel Aviv, Israel*
*** *Niš University, Niš, Serbia*

Received May 14, 2009

Abstract—Consideration was given to the linearization of logical functions defined by a set of pairwise orthogonal terms. The linearization is carried out by computing the autocorrelation functions. Proposed was a method consisting of (i) calculation of the autocorrelation function in the space of orthogonal terms, (ii) generation of the corresponding matrix of linear transformation, and (iii) the linear transformation proper of the variables in the space of orthogonal terms. Complexity of the proposed method and its effectiveness were estimated. Effectiveness was verified by a series of experiments with standard benchmarks. The distinctions of the proposed method from other existing methods of linearization were examined.

DOI: 10.1134/S0005117911030118

1. INTRODUCTION

More than 35 years passed since the publication of the book of M.G. Karpovskii and E.S. Moskalev [1] which by right is regarded as one of the pioneering works that opened up the way for using the spectral methods in the design of digital circuits. The spectral method attracted attention of both the academic and engineering communities by its originality and unexpectedness. This approach by right is distinguished for its mathematical profoundness and strictness which shows it to the best advantage in comparison with many methods that are mostly based on the heuristic optimization algorithms. The recent publication of M. Karpovsky, R. Stankovič, and J. Astola [2] is an encyclopedia of the spectral approach to logical design, a summary of a sort of the entire line of research during the recent decades. The spectral methods were developed both theoretically and practically. As for their practical application, it deserves mentioning the minimization of the decision diagrams, design of circuits with predefined characteristics, design of the functional transformers, design of testable circuits, test generation, improvement of computer system reliability, and so on. Without asserting that the spectral methods for design and analysis of the digital circuits "defeated convincingly" the traditional approaches, we note the essence of their "success." The spectral methods provided a fundamental theoretical platform for the future theoretical and practical applications and formed an independent informative field of research which attracts attention of the designers and researchers, as well as the undergraduates and postgraduates majoring in the computer sciences.

The present work is one more step in the study of the spectral methods. It is devoted to one of the central components of the spectral theory, the so-called linearization of logical functions which lies in decomposing the given logical function into a linear and nonlinear components. At that, the value of the linear component which has a low complexity is maximized, thus making the logical function minimized in general.

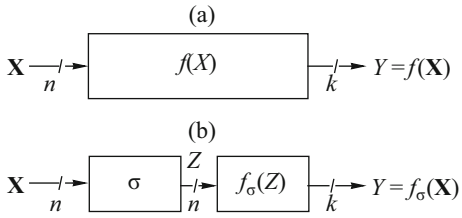

Fig. 1. Linear decomposition of a logical function.

The problem of linear transformation of the logical functions—more precisely, the variables of these functions—is well known and studied intensively beginning from [3] followed by [4, 5]. Its topicality is corroborated by the recent publications [6–8], as well as other papers, where the linearization of Boolean functions is used to advantage in diverse applications for optimization by various criteria [9–11].

The present paper considers a function describing the n-input k-output logical circuit as a function mapping the finite field $GF(2^n)$ into the finite field $GF(2^k)$. Therefore, by the function f is meant a collection of k Boolean functions of n variables, $f : GF(2^n) \to GF(2^k)$.

The totality of the binary values of the variables $(x_{n-1}, \ldots, x_0)$ of the function f is considered as a vector of coefficients. An element x of the function $GF(2^n)$ is representable as a linear combination of n basic vectors δ_i, $x = \sum\limits_{i=0}^{n-1} x_i \delta_i$ where each δ_i is a binary vector of length n corresponding to an integer 2^i. The collection of vectors $\{\delta_i\}_{i=0}^{n-1}$ is called the original basis of the function $GF(2^n)$. Any collection of n linearly independent vectors generates a basis.

The present paper relies on the concept stating that the complexity of the function mapping the elements $GF(2^n)$ into $GF(2^k)$ depends on the selected basis. Therefore, selection of the basis enabling minimization of this transformation function is a natural aim of the method of optimization described in Section 3.

The existing linearization methods represent the logical function f as a superposition of the linear (function σ) and nonlinear (functions f_σ) parts (see Fig. 1). The function f_σ is determined from f by the linear transformation σ of the input variables. By the linearization is meant the replacement of the original basis by another basis (collection of the basic vectors). This linearization is equivalent to the replacement of the initial collection of variables in the given function by another collection of variables obtained by the linear transformation σ of the initial variables. The linear transformation should be selected so that, except for the case where σ is an identical transformation, the corresponding function f_σ has complexity smaller than that of the function f.

The most popular criteria for complexity of realization of the logical functions are as follows: the number of two-input logical elements of conjunction and disjunction that are required to realize the given Boolean function [6, 12], the number of vertices in its binary decision diagram (BDD) [2, 6, 9] or the number of terms in the logical expressions [7, 11]. The present paper makes use of the first criterion [1]. This generally accepted criterion enables one to compare adequately the complexities of various realizations of f in different bases [10, 12]. It is invariant to the order of the basic vectors. At the same time, linearization is useful in the cases where the realizations are sensitive to the order of the input variables. For example, the procedure of optimization of the decision diagrams may use the linearization procedure to determine the initial collection of the basic vectors. In such cases the linear transformation represents a permutation matrix.

The methods of linearization using the autocorrelation functions generate σ with the use of the basis of the inertia group for f [2]. If the basis elements are represented as the matrix columns, its inverse matrix defines the desired linear transformation σ of the variables.

The methods of linearization based on the autocorrelation function consist of (i) calculation of the autocorrelation function, (ii) construction of the linear transformation of the input variables with the use of the maximal value of the autocorrelation coefficients, and (iii) determination of the corresponding linearized function f.

The general complexity of the linearization procedure is defined by the complexity of executing the aforementioned stages. The existing methods may prove to be actually useless for functions with many input variables (see Section 2). For such functions, the present paper solves the problem of complexity under the assumption that the function is defined by a collection of orthogonal terms and proposes a procedure to realize such functions in the space of the orthogonal terms.

Effectiveness of the methods for calculation of the autocorrelation function R_f depends on the method of description of the function f which in turn often depends on the particular field of function application. The present paper considers those applications of the logical functions where their representation with by the orthogonal terms proves to be convenient.

The existing methods for calculation of the autocorrelation function for the logical functions defined by arbitrary terms require that each term be represented as a collection of minterms, which results in an excessive waste of memory and time. In the proposed method, the autocorrelation function is calculated on the orthogonal terms and the autocorrelation function is represented as the so-called arithmetic sum of the terms.

The paper demonstrates that this method of calculation is effective from the following standpoints: (i) execution time which is a function of the number of terms, (ii) need for the memory resources in the course of calculations, and (iii) memory space to store the values of the autocorrelation function. This method is effective if the number of orthogonal terms is smaller than $\sqrt{2^n/n^3}$.

In principle, the generation of the basis for an inertia group requires verification of linear independence of each added basis vector from the existing basis vectors. In the proposed algorithm, executed are the local linear transformations, and as the result there is no need for such checks which enables one to facilitate determination of the corresponding linearized function.

The following section discusses the works in this field, that is, the existing methods for calculation of the autocorrelation function and the linearization procedures. It also formulates the problem to be solved. The mathematical apparatus used in the knowledge domain under consideration is considered in Section 3. Section 4 describes the procedure for linearization in the space of orthogonal terms. The following Sections 5–7 describe in detail the calculation of the autocorrelation function, generation of the transformation matrix and the linear transformation on the orthogonal terms. Section 8 compares different linearization procedures in terms of complexity and the number of vertices in the decision diagrams corresponding to the functions before and after the linear transformation of the variables. Complexity of the algorithms and the advantages of linearization in the space of orthogonal terms are discussed in Section 9. The results and conclusions are given in Section 10.

2. REVIEW OF THE SPECIAL LITERATURE

2.1. Calculation of the Autocorrelation Function

The value of the autocorrelation function R_f of the logical function $f : GF(2^n) \to GF(2)$ at the point $\tau \in GF(2^n)$ is defined as

$$R_f(\tau) = \sum_{x \in GF(2^n)} f(x) f(x + \tau).$$

There are two methods of calculation of the autocorrelation function:

(1) directly on the basis of the definition;

(2) using the Wiener–Khinchin theorem [10, 13]:

$$R_f(\tau) = 2^n W^{-1}(Wf)^2,$$

where W is the normalized operator of the Walsh transformation.

Complexity of calculation of the values of the autocorrelation function depends on the method of its representation. Calculation of the autocorrelation functions with the use of the Wiener–Khinchin theorem usually takes less time ($O(n2^n)$) than the calculations based on determination of the autocorrelation function ($O(2^{2n})$). This is true in the majority of cases where the calculations are carried out with the decision diagrams and not the truth table [14]. For a relatively great number of variables, however, for the given resources of memory or/and time the calculations on vectors may turn out to be nonrealizable owing to their exponential length 2^n. Similarly, for some functions the decision diagrams representing the given function or Walsh spectrum may have an exponential complexity expressed in terms of the number of vertices. Contrary to this, at representation of a function by orthogonal terms the calculations may be carried out independently for each terms or a pair of terms. Therefore, the calculations can be carried out concurrently which opens the way to working with complicated functions.

Many studies focused on the optimization of calculations of the spectral transformations. In particular, the algorithm of Walsh transformation for the switching functions defined by the orthogonal terms was considered in [15, 16]. However, this approach may be inefficient for calculation of the autocorrelation function R_f with the use of the Wiener–Khinchin theorem because complexity of the method depends on the number of terms. If their number is relatively small, the direct transformation of the function f into its Walsh spectrum may be effective. The problem is created by the complexity of the inverse transformation of $(Wf)^2$ into R_f. The number of various values that may be taken by the Walsh coefficients usually is high, and therefore, sometimes the quadratic components of the spectrum cannot be grouped and effectively described by orthogonal terms. For example, the Walsh spectrum of the function $f(x_3, x_2, x_1, x_0) = \overline{x}_3\overline{x}_2 + x_3\overline{x}_2x_0 + \overline{x}_3x_2\overline{x}_1x_0$ requires nine orthogonal terms.

Another approach to calculation of R_f on the orthogonal terms is represented by the tabular method developed in [17] on the basis of transformation of the representation of the logical function from the DNF form into that of Reed-Muller. This method relies of operations with minterms and may be used for any number of input variables of complexity $O(2^n)$ [11]. This tabular method was improved in [18] where it was suggested to process the minterms in parallel which required to store the index table of size 2^n. Despite the fact that this approach is efficient for calculation of the autocorrelation function for greater n, it operates with minterms, rather than terms, and therefore, its complexity is a function of the number of minterms and not terms.

In [5] the values of the autocorrelation function are calculated directly on the collection of N orthogonal terms without representation of the function in PDNF. Each value of the autocorrelation function is calculated separately with the complexity $O(nN^2)$. Obviously, this approach is efficient for calculation of a small number of the values of autocorrelation.

The domains of application of the methods using calculation of the autocorrelation functions include the design and optimization of the combinatorial circuits [5, 19], BDD optimization [6, 20–22], estimation of the complexity of logical functions [2], and classification of the logical functions [23].

The present paper proposes and examines a new method of determination of the set of basic vectors of the finite field $GF(2^2)$. The method is based on calculation of the autocorrelation function and search of the maximal value of this function.

2.2. Linearization Algorithms and Minimization of the Logical Functions

The algorithm of linearization with the aim of minimizing the logical functions defined by a union of orthogonal terms was described in [5] whose authors proposed to calculate $R_f(\tau)$ directly by determining the autocorrelation function. Complexity of calculation of one particular value of R_f depends on the number of terms and not the minterms covered by them. At that, a single value of R_f is calculated at each individual step of the algorithm.

The linearization algorithm is based on enumeration of terms, and for each term it determines heuristically a candidate τ among those where the autocorrelation function $R_f(\tau)$ must be maximal. If τ under consideration is beyond the space "covered" by the preceding values of τ, a new value of $R_f(\tau)$ is calculated and included, if greater than the preceding calculated values of autocorrelation, in the basis. Therefore, the complexity of calculating $R_f(\tau)$ for all values of $\tau \in \{0, 1, \ldots, 2^n - 1\}$ is equivalent to the complexity of determining a suitable basis.

The main disadvantage of this method lies in that the final collection of the values of τ depends on the order of terms and the subspace defined by the values of τ of the previously determined terms. To provide uniformity of the approach to the data structures used to represent the processed functions, the transformed function f_σ using a linear transformation σ on terms was determined in [5]. Since in the general case the linear transformation of variables decomposes the term into several smaller terms, this procedure may require a larger memory space.

2.3. Linearization Algorithms and the Decision Diagrams

An algorithm for linearization of the Boolean functions through calculation of the autocorrelation values was proposed in [6] and used to minimize the binary decision diagrams (BDD). The algorithm which was named the K-procedure minimizes the diagram's size using a linear transformation of the input variables. This linear transformation is determined as a linear superposition of the linear transformations minimizing the number of vertices at each level beginning from the diagram apex. The linearization is carried out at each level though selecting the basis of the inertia group for the function represented by the upper-level vertices. Then, at each step of the algorithm the diagram is convoluted. (We recall that the inertia group is defined as the set of assignments $\tau = (\tau_0, \ldots, \tau_{n-1})$ for the variables $x = (x_0, \ldots, x_{n-1})$ where the autocorrelation function $R_f(\tau)$ for $f(x)$ takes on the maximal value [2].) The main difficulty of the algorithm lies in the need for calculating the autocorrelation function at each level, which is its weak spot, especially for functions with many variables. For that reason, consideration was given in [14] to the methods of enhancing effectiveness of the autocorrelation functions. It deserves noting that in many practical cases the values of the autocorrelation functions can be calculated analytically without any laborious calculations.

The operation of convolution, that is, of removal of the "lower" level of the binary decision tree by increasing the cardinality of the tree leaf alphabet, is an important stage of the K-procedure. The convolution can be carried out rather readily straight from the truth table or the decision diagram. In the case of a function defined by the set of orthogonal terms, however, convolution requires substantial calculations because this operation does not retain orthogonality and, therefore, the terms must be decomposed into many new terms. In the proposed method, this problem is resolved by a careful selection of the basic vectors.

2.4. Algorithm to Linearize the Logical Functions Defined by Orthogonal Terms

The proposed method stems from the analysis of experimental studies which demonstrated that it is possible to confine the space of search at determining a suitable basis of the linear transformation. The suggested algorithm has no need for convolution and makes the basis elements, that is, τ, linearly independent by constraining their values (see Section 3). The constraint is

expressed in terms of the decimal values of τ understood as the binary representation of an integer that may be taken on by τ.

The experimental results demonstrated that by assuming that τ has the Hamming weight smaller than or equal to three one can obtain results that are almost as good as those obtained under the assumption that τ has arbitrary values, that is, without constraining the Hamming weight. For sufficiently effective linearization, therefore, in practice it often suffices to consider approximately n^3, instead of 2^n, values of the parameter R_f. Moreover, according to the definition, for functions with a relatively low number of terms the complexity of calculation of R_f may prove to be lower than the complexity of the algorithms based on the Wiener–Khinchin theorem or the tabular method. The proposed algorithm has complexity of the order of $n^{w+1}N^2$, where N is the number of terms and w is the parameter bounding the maximal Hamming weight for τ.

As for using the linearization procedure, we notice that to simplify calculations the methods handling the truth vectors [10] or the decision diagrams [6] calculate the autocorrelation function using the Wiener–Khinchin theorem. With this approach, the linearly transformed function f_σ is determined in several steps by factorizing the linear transition matrix σ as the product of sparse matrices. However, if the functions are represented by orthogonal terms, then it is more convenient to calculate the autocorrelation function straight according to the definition. This approach was used in [5], but the algorithm proposed there calculates separately each autocorrelation coefficient.

Here we propose a new algorithm to calculate $R_f(\tau)$ simultaneously for different values of τ. The calculated values of the autocorrelation coefficients are defined on terms, which results in a compact—in terms of the memory space—representation of the complete autocorrelation function. Complexity of the proposed algorithm is comparable with that of calculating $R_f(\tau)$ for a unit τ by the method of [5].

3. MATHEMATICAL FUNDAMENTALS OF THE METHOD

3.1. Definitions

Let the function $f : GF(2^n) \to GF(2^k)$ describe an n-input, k-output logical device. We assume that f is a completely specified function. Let $\Im = \{0, 1, \phi\}$, where ϕ stands for the don't-care values. The representation of the function f by the orthogonal terms includes a collection of N pairs $F = \{(P_i, Y_i)\}_{i=1}^N$, where $P_i \in \Im^n$ is a term and $Y_i \in GF(2^k)$ is the corresponding output vector of the logical device.

We confine ourselves to the representation of the functions defined by N orthogonal terms. Stated differently, each pair of terms $P_i = (a_{n-1}, \ldots, a_1, a_0)$ and $P_j = (b_{n-1}, \ldots, b_1, b_0)$ has $d(P_i, P_j) > 1$, where

$$d(P_i, P_j) = \{k | a_k, b_k \in GF(2),\ a_k \neq b_k\}.$$

We notice that, despite the fact that $d(P_i, P_j)$ is a symmetrical and nonnegative function, it is not a measure because the "triangle inequality" is not satisfied for the vectors from $\Im$.

The set of term describing a function may be decomposed into subsets with identical output values. We refer to them as the *characteristic* subsets. The characteristic subset $F_u(u \in GF(2^k))$ is defined as follows:

$$F_u = \{(P_i, Y_I) | (P_i, Y_i) \in F, Y_i = u\}. \quad (1)$$

The logical function f_u defined through the characteristic terms F_u is called the characteristic function for u.

In this paper we evaluate the realization complexity of a function in terms of the two-input logical conjunction and disjunction elements that are required to realize it. As was demonstrated

in [12], this complexity may be evaluated by the value of the functional $\mu(f)$ defined as the number of neighboring minterms where the function takes on identical values. Formally, this functional is defined as follows:

$$\mu(f) = \left| \left\{ (x_1, x_2) \;\middle|\; \begin{array}{l} x_1, x_2 \in GF(2^n),\ d(x_1, x_2) = 1, \\ f(x_1) = f(x_2) \end{array} \right\} \right|.$$

It is common practice to refer to this functional as the function's "*complexity*," though it would be more appropriate to call it the function's "*simplicity*" because it decreases with increased realization complexity. In what follows, we traditionally call $\mu(f)$ the function's complexity.

The relation between the complexity $\mu(f)$ and the autocorrelation function

$$\mu(f) = \sum_{u \in GF(2^k)} \mu(f_u), \quad \text{where} \quad \mu(f_u) = \sum_{\|\tau\|=1} \sum_{x \in GF(2^n)} f_u(x) f_u(x+\tau) = \sum_{\|\tau\|=1} R_u(\tau),$$

where R_u is the autocorrelation of the characteristic function f_u, was shown in [1]. Additionally, let us assume that $R(\tau) = \sum_u R_u(\tau)$ and T is the matrix whose columns are the vectors τ_i. We denote $\hat{R}(T) = \sum_i R(\tau_i)$, then $\mu(f) = \hat{R}(I)$, where I is the identity matrix.

3.2. Optimization of the Logical Function by Linearization

The linear transformation with respect to the variables enables one to realize the logical function as a superposition of the linear transformation of the function σ and the nonlinear part f_σ, $f(x) = f_\sigma(\sigma x)$. If σ is not the matrix of identical transformation, then f_σ is less complicated by the criterion $\mu(f)$.

The element $GF(2^n)$ is representable either as a linear combination of the elements $\{\delta_i\}$, $i = 0, \ldots, n-1$, in the original basis with the coefficient vector $x = (x_{n-1}, \ldots, x_1, x_0)$ or as a collection of the linearly independent basic vectors $\{\tau_i\}$, $i = 0, \ldots, n-1$, with the coefficient vector z. The basic vectors are the columns of the nonsingular matrix T, that is, $\tau_i = T\delta_i$. The corresponding coefficient vectors $z = (z_{n-1}, \ldots, z_1, z_0)$ are defined by the matrix of the linear transformation $\sigma = T^{-1}$ as $z = \sigma x$. For example,

$$\sigma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad T = (\tau_2, \tau_1, \tau_0) = \sigma^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

The element $(101) \in GF(2^3)$ is representable as $(101)^T = \delta_2 + \delta_0 = \tau_2 + \tau_1$, where $x = (101)$ and $z = (110)$ are the coefficient vectors. The corresponding transformed function f_σ satisfies $f_\sigma(110) = f(101)$.

The autocorrelation functions $f(x)$ and $f_\sigma(x)$ have identical values but in different positions. Namely,

$$R_{f_\sigma}(\tau) = R_f(\sigma^{-1}\tau). \tag{2}$$

For the given function f, therefore, the problem of optimization lies in determining a nonsingular $(n \times n)$ linearization matrix σ_{opt} such that $\mu(f_{\sigma_{\text{opt}}})$ is maximal. Obviously,

$$\mu_{\max} = \max_\sigma \mu(f_\sigma) = \max_\sigma \hat{R}_{f_\sigma}(I) = \max_\sigma R_f(\sigma^{-1}) = \hat{R}_f(T), \tag{3}$$

where $T = \sigma_{\text{opt}}^{-1}$ is a nonsingular matrix $(n \times n)$ where the columns $(\tau_{n-1}, \ldots, \tau_1, \tau_0)$, $\tau_i \in GF(2^n)$, generate the basis and the sum $\sum_i R_f(\tau_i)$ is maximal. We notice that, as the following example demonstrates, there may be more than one collection of the basic vectors of the same $\hat{R}_f(T)$.

Table 1

$x = (x_2, x_1, x_0)$	$f(x)$	$R(x)$
000	0	8
001	0	2
010	1	0
011	2	2
100	2	2
101	1	0
110	1	2
111	0	6

Table 2a

$x_1x_2x_3$	00	01	11	10
0	0	1	1	2
1	0	2	0	1

Table 2b

$z_1z_2z_3$	00	01	11	10
0	0	0	1	0
1	2	1	1	2

Example 1. Let us consider a system of two functions with three inputs $f : GF(2^3) \to GF(2^2)$ as defined by Table 1. The element $GF(2^2)$ is a binary vector of length two. We denote it by its decimal equivalent. For example, $(10) = 2$. The autocorrelation function $R_f(\tau)$ is shown in the right column of the table.

The value of the complexity criterion μ of the original function is as follows:

$$\hat{R}_f(I) = R_f(001) + R_f(010) + R_f(100) = 2 + 0 + 2,$$

whereas for the basis

$$T_1 = \{(111)^T, (001)^T, (011)^T\} \quad \text{or} \quad T_2 = \{(111)^T, (110)^T, (001)^T\} :$$
$$\hat{R}_f(T_1) = \hat{R}_f(T_2) = 6 + 2 + 2.$$

Tables 2a and 2b show the functions corresponding to the original basis and the basis T_1.

The function shown in Table 2a is representable by a smaller collection of the orthogonal terms than that represented in Table 2b.

The complexity criterion μ is independent of the order of variables. For example, the matrix T_1 of Example 1 may be put down as $\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ or $\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$.

Wherever it is obvious from the context, we simplify expressions by using the decimal value corresponding to the binary vector τ. For example, the original basic vectors of Example 1 have values 1, 2, and 4 (or $I = (4, 2, 1)$) and the ordered representation $T_1 - T_1 = (7, 3, 1)$. One may readily see that the following property is satisfied.

Property 1. If the ordered collection K of the vectors $\{\tau_k\}_{k=0}^{K-1}$ satisfies $2^k \le \tau_k < 2^{k+1}$, then the vectors are linearly independent.

We notice that (7, 6, 1) is the ordered collection of basic vectors defined by T_2 of Example 1. This collection does not satisfy Property 1 because 6 is greater than 2^2. However, T_2 corresponds to the basis of greater μ. Stated differently, Property 1 as such may turn out to be overstrict as a criterion for generating a collection of the basic vectors with greater μ.

Complexity of linearization may be reduced by calculating the autocorrelation coefficients for the values of τ having the Hamming weight smaller than or equal to a certain value w, and this is

instead of calculating all autocorrelation coefficients, that is, for all 2^n possible values of τ. The number of all possible values of τ over the interval $[2^k; 2^n - 1]$ of the Hamming weight which is smaller than or equal to w obeys

$$W = \sum_{j=1}^{w} \left(\begin{pmatrix} n \\ j \end{pmatrix} - \begin{pmatrix} k \\ j \end{pmatrix} \right). \tag{4}$$

For comparison, the procedure K [6] for minimization of the size of the decision diagram performs convolution of the binary tree after each step and, consequently, τ lies within the interval $[1; 2^{n-k} - 1]$. The number of different τ bounded by w is expressed as $\sum_{j=1}^{w} \begin{pmatrix} n-k \\ j \end{pmatrix}$, which is smaller than or equal to W. In the next section we return to the same problem by carrying out linear transformations on the collection of terms.

3.3. Construction of the Basis and Linear Transformation of the Orthogonal Terms

The linearization procedure described here is classified with the so-called *greedy* procedures. It constructs in n steps a collection of n basic vectors each time taking τ such that it does not belong to a subspace bounded by the preceding vectors and has the maximal value of the autocorrelation function. Since consideration is not given to every possible variant of τ having the maximal value of autocorrelation, $\hat{R}(T)$ needs not to be optimal.

To avoid highly complex verification that the candidate for the basic vectors is linearly independent of the vectors included previously in the basis, the algorithm constrains the scatter of possible values of τ as follows. At the end of each step, the local linear transformation σ_i is performed on the current collection of the orthogonal terms $F_i = \sigma_i F_{i-1}$, $i = 1, \ldots, n$, where $F_0 = F$ and F_n is the transformed collection of terms corresponding to f_σ. The linear transformation of term is defined formally in Section 7.

We define as R_i the autocorrelation function of the function f_i corresponding to the collection F_i. The linearization matrix σ_i is constructed so that the ith basis element is determined at the ith step. Therefore, at the ith step the autocorrelation R_i has a value associated with τ selected in the position 2^i (see Theorem 2 in Section 7). Since the procedure is recursive, the i first values of autocorrelation of the collection F_i situated in the positions 2^k $(k = 0, \ldots, i-1)$ are equal to the maximal R established at the previous stages.

Stated differently, at the ith stage the $i-1$ first vectors in the basis have the decimal value smaller than 2^{i-1} and, therefore, all τ having values greater than or equal to 2^{i-1} are independent and satisfy Property 1. Therefore, the calculation of the autocorrelation function of the transformed (and not

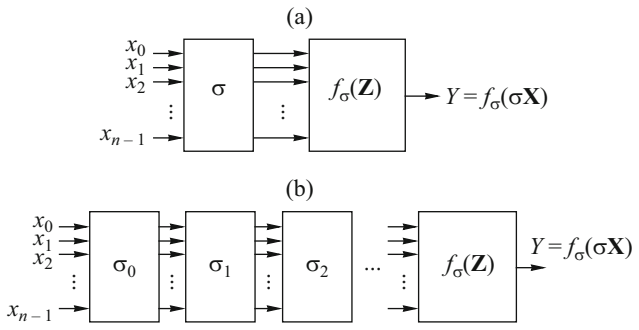

Fig. 2. Decomposition.

the original) collection of terms and the requirement that τ of the ith stage have decimal value greater than or equal to 2^{i-1} together make the extended collection of vectors linearly independent.

According to the aforementioned, the linear transformation matrix σ (Fig. 2a) is generated as a product of n' matrices where $n' \le n$, that is, $\sigma = \sigma_{n'-1} \dots \sigma_1 \sigma_0$ (Fig. 2b).

Decomposition of σ in a matrix product resolves another problem of complexity which is a unique feature of the calculations on the orthogonal terms. Since x is a binary vector and σx is calculated on $GF(2)$, it is rather easy to perform the linear transformation of the function represented by the truth table. Yet in the system of orthogonal terms σ "multiplies" the term (which is not necessarily a minterm). As is shown, for instance, by the following example, in some cases the result cannot be expressed by a single term.

Example 2. Let us assume that $\sigma = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$ and $x = (0, 1, \phi, \phi) = \begin{Bmatrix} (0,1,0,0) \\ (0,1,0,1) \\ (0,1,1,0) \\ (0,1,1,1) \end{Bmatrix}$. Then, $\sigma x^T = \begin{Bmatrix} (1,1,0,1) \\ (0,1,1,0) \\ (0,1,0,0) \\ (1,1,1,1) \end{Bmatrix}$.

This result cannot be described by a single term.

In the general case, it is difficult to determine the collection of transformed terms straight from the linear transformation σ without representing each term as a collection of the corresponding minterms. The proposed algorithm enables one to avoid this by using local linearization. The local linearization matrix σ_i is the product of two matrices, the permutation matrix and the linearization matrix (see Section 7).

REFERENCES

1. Karpovskii, M.G. and Moskalev, E.S., *Spektral'nye metody analiza i sinteza diskretnykh ustroistv* (Spectral Methods of Analysis and Design of Discrete Devices), Leningrad: Energiya, 1973.
2. Karpovsky, M., Stankovic, R., and Astola, J., *Spectral Logic and Its Applications for the Design of Digital Devices*, New York: Wiley, 2009.
3. Nechiporuk, E.I., On the Synthesis of Networks Using Linear Transformations of Variables, *Dokl. Akad. Nauk SSSR*, 1958, vol. 123, no. 4, pp. 610–612.
4. Trachtenberg, E.A., Applications of Fourier Analysis on Groups in Engineering Practices, in *Recent Developments in Abstract Harmonic Analysis with Applications in Signal Processing*, Stankovic, R.S., Stojic, M.R., and Stankovic, M.S., Eds., Belgrade: Nauka; Niš: Elektronski fakultet, 1996, pp. 331–403.
5. Varma, D. and Trachtenberg, E.A., Design Automation Tools for Efficient Implementation of Logic Functions by Decomposition, *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, 1989, vol. 8, no. 8, pp. 901–916.
6. Karpovsky, M.G., Stankovic, R.S., and Astola, J.T., Reduction of Sizes of Decision Diagrams by Autocorrelation Functions, *IEEE Trans. Comput.*, 2003, vol. 52, no. 5, pp. 592–606.
7. Meinel, C., Somenzi, F., and Theobald, T., Linear Sifting of Decision Diagrams and Its Application in Synthesis, *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, 2000, vol. 19, no. 5, pp. 521–533.
8. Stankovic, R.S. and Astola, J.T., *Spectral Interpretation of Decision Diagrams*, New York: Springer, 2003.
9. Gunther, W. and Drechsler, R., Efficient Manipulation Algorithms for Linearly Transformed BDDs, in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, 1999, pp. 50–54.

10. Karpovsky, M.G., *Finite Orthogonal Series in the Design of Digital Devices*, New York: Wiley, 1976.

11. Miller, J.F., Luchian, H., Bradbeer, P.V.G., et al., Using a Genetic Algorithm for Optimizing Fixed Polarity Reed-Muller Expansions of Boolean Functions, *Int. J. Electron.*, 1994, vol. 4, no. 76, pp. 601–609.

12. Shannon, C.E., The Synthesis of Two-Terminal Switching Circuits, *Bell Syst. Technic. J.*, 1949, vol. 28, pp. 59–98.

13. Pichler, F., Walsh Functions and Linear System Theory, in *Proc. Appl. Walsh Functions*, 1970, pp. 175–182.

14. Stankovic, R.S. and Karpovsky, M.G., Remarks on Calculation of Autocorrelation on Finite Dyadic Groups by Local Transformations of Decision Diagrams, in *Proc. EUROCAST 2005, Lecture Notes Comput. Sci.*, Berlin: Springer, 2005, vol. 3643, pp. 301–310.

15. Falkowski, B.J. and Kannurao, S., Calculation of Sign Walsh Spectra of Boolean Functions from Disjoint Cubes, in *Proc. IEEE Int. Symp. Circuits Syst.*, 2001, vol. 5, pp. 61–64.

16. Falkowski, B.J., Schafer, I., and Perkowski, M.A., Calculation of the Rademacher-Walsh Spectrum from a Reduced Representation of Boolean Functions, in *Proc. Conf. Eur. Design Automat.*, 1992, pp. 181–186.

17. Almaini, A.E.A., Thomson, P., and Hanson, D., Tabular Techniques for Reed-Muller Logic, *Int. J. Electron.*, 1991, vol. 70, no. 1, pp. 23–34.

18. Tan, E.C. and Yang, H., Fast Tabular Technique for Fixed-polarity Reed-Muller Logic with Inherent Parallel Processes, *Int. J. Electron.*, 1998, vol. 85, no. 4, pp. 511–520.

19. Tomczuk, R., Autocorrelation and Decomposition Methods in Combinational Logic Design, *PhD Dissertation*, Univ. of Victoria, 1996.

20. Jain, J., Moundanos, D., Bitner, J., et al., Efficient Variable Ordering and Partial Representation Algorithm, in *Proc. 8th Int. Conf. on VLSI Design*, 1995, pp. 81–86.

21. Kolpakov, A. and Latypov, R.Kh., Approximate Algorithms for Minimization of Binary Decision Diagrams on the Basis of Linear Transformations of Variables, *Autom. Remote Control*, 2004, no. 6, pp. 938–954.

22. Keren, O. and Levin, I., Linearization of Multi-Output Logic Functions by Ordering of the Autocorrelation Values, *Facta Univ., Ser. Elec. Energ.*, 2007, vol. 20, no. 3, pp. 479–498.

23. Rice, J.E. and Jansen, R., Symmetrical, Dual and Linear Functions and Their Autocorrelation Coefficients, in *Proc. Int. Workshop Logic Synthesis*, 2005, pp. 30–35.

This paper was recommended for publication by O.P. Kuznetsov, a member of the Editorial Board