ISSN 0005-1179, Automation and Remote Control, 2011, Vol. 72, No. 4, pp. 818–827. © Pleiades Publishing, Ltd., 2011.
Original Russian Text © O. Keren, I. Levin, R. Stankovič, 2011, published in Avtomatika i Telemekhanika, 2011, No. 4, pp. 155–165.

LOGIC CONTROL

Linearization of Logical Functions Defined by a Set of Orthogonal Terms. II. Algorithmic Aspects[1]

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

DOI: 10.1134/S0005117911040126

4. ALGORITHM FOR LINEARIZATION OF ORTHOGONAL TERMS

The proposed algorithm constructs a set of linearly independent vectors of the space $GF(2^n)$. The logic function F defined by the set of orthogonal terms provides the algorithm with initial information. The algorithm determines the matrix σ of linear transformation and a set of transformed orthogonal terms corresponding to this transformation. If the logic function is defined by an arbitrary set of terms, then this set may be orthogonalized using one of the existing algorithms realized, for example, in the ESPRESSO program. For simplicity, the details of the algorithm are omitted here and left for the next section.

Algorithm 1. Linearization of the logic function defined by orthogonal terms. We state that $\sigma = I_{(n\times n)}$ and $i = 1$.

(a) Using the method described in Section 5, calculate the autocorrelation function $R_f(\tau)$ for all $\tau \in (2^n)$, $\|\tau\| \le w$ and $\tau \ge 2^{i-1}$.

(b) Determine τ for which $R_f(\tau)$ is the smallest one. If there is more than one such τ, then select it arbitrarily.

(c) Construct the linear transformation matrix σ_i using the method of Section 4.

(d) Carry out the linear transformation on the set of terms $F = \sigma_i F$. The corresponding procedure is described in Section 7.

(e) Correct the current value of σ; $\sigma = \sigma_i\sigma$.

(f) Take $i = i+1$ and repeat the procedure until $i = n$ of $R_f(\tau) = 0$ for all calculated values of τ.

We notice that the linearization procedure may consist of less than n iterations if the system is described using less than n basic vectors. The procedure can be extended successfully to incompletely defined functions at the expense, true enough, of an increased complexity [1].

In the following sections we describe in detail the linearization procedure and begin with calculation of the autocorrelation function in the space of orthogonal terms. Next, we give a simple method to determine the matrix σ_i directly from τ_i without linear transformation of the terms.

5. CALCULATION OF THE AUTOCORRELATION FUNCTION FOR ORTHOGONAL TERMS

Let N_u be the number of terms $\{P_i\}_{i=1}^{N_u}$ related with the characteristic set F_u, $\sum\limits_{u\in GF(2^k)} N_u = N$. Since F is defined by a set of orthogonal terms, F_u is expressed in the same way, and therefore,

[1] The first part was published in no. 3, 2011.

$f_u(x) = \bigcup\limits_{i=1}^{N_u} P_i(x) = \sum\limits_{i=1}^{N_u} P_i(x)$, where $\bigcup$ is the logic OR (disjunction) and $\sum$ is the arithmetic sum. The autocorrelation function of f_u is as follows:

$$R_u(\tau) = \sum_{i=0}^{N_u}\sum_{j=0}^{N_u}\sum_{x\in GF(2^n)} P_i(x)P_j(x+\tau) = \sum_{i=0}^{N_u}\sum_{j=0}^{N_u} R_{i,j}^{(u)}(\tau).$$

We omit u for simplicity, if this is implied in the context, and use the autocorrelation function $R_{i,j}$ which may be calculated using its definition but without enumerating all x's. Moreover, as is shown below, it is calculated for the set of terms τ and not for a single τ.

Let $P_i = (p_{n-1}^{(i)}, \ldots, p_1^{(i)}, p_0^{(i)}) \in \Im^n$ be a term, and let n_ϕ be the number of "free" variables in this term. It is common knowledge (see, for example, [1]) that the autocorrelation $R_{i,i}(\tau)$ of $P_i(x)$ is equal to 2^n for any τ having the form $(t_{n-1}, \ldots, t_1, t_0)$, where $t_k = \begin{cases} \phi, & p_k^{(i)} = \phi \\ 0, & p_k^{(i)} \neq \phi \end{cases}$ $(k = 0, \ldots, n-1)$. For the rest of τ, $R_{i,i}(\tau) = 0$.

For the two terms P_i and $P_j \in \Im^n$, we denote by $p_k^{(i)}$ and $p_k^{(j)}$ the kth symbol in P_i and P_j, respectively. Since $p_k^{(i)}$ and $p_k^{(j)}$ belong to the set $\Im$, there are nine possible types of pairs $(p_k^{(i)}, p_k^{(j)})$ denoted by T_l, $l = 1, \ldots, 9$:

$$\left(p_k^{(i)}, p_k^{(j)}\right) \in \begin{cases} T_1 = (0,0), & T_2 = (0,1), & T_3 = (0,\phi), \\ T_4 = (1,0), & T_5 = (1,1), & T_6 = (1,\phi), \\ T_7 = (\phi,0), & T_8(\phi,1), & T_9 = (\phi,\phi). \end{cases}$$

Let n_l denote the number of pairs of the type T_l,

$$n_i = \left|\left\{k \;\middle|\; \left(p_k^{(i)}, p_k^{(i)}\right) = T_l, \;\; 0 \le k < n\right\}\right|, \quad l = 1, \ldots, 9.$$

For example, if $P_i = (0\phi 11\phi\phi)$ and $P_j = (00\phi 1\phi\phi)$, then n_1 denotes the number of occurrences of the pair $(0,0)$ and is equal to one. The value of n_2 is zero because there is no position where $p_k^{(i)} = 0$ and $p_k^{(j)} = 1$. Similarly, $n_3 = 0$; $n_4 = 0$; $n_5 = 1$; $n_6 = 1$; $n_7 = 1$; $n_8 = 0$; $n_9 = 2$.

The following theorem establishes that the function of mutual correlation $R_{i,j}$ is calculable directly by comparing the terms. Additionally, the mutual correlation function is calculable through the terms in τ and not from τ themselves. This allows one to represent the mutual correlation by an arithmetic sum and not a vector of length 2^n.

Theorem 1. *Let $P_i = (p_{n-1}^{(i)}, \ldots, p_1^{(i)}, p_0^{(i)}) \in \Im^n$ and $P_j = (p_{n-1}^{(j)}, \ldots, p_1^{(j)}, p_0^{(j)}) \in \Im^n$. Let n_9 denotes the number of pairs $(p_k^{(i)}, p_k^{(j)})$ of the type T_9. We denote by A the set of vectors like $(t_{n-1}, \ldots, t_1, t_0)$, where*

$$t_k = \begin{cases} 0, & (p_k^{(i)}, p_k^{(j)}) \in \{T_1, T_5\} \\ 1, & (p_k^{(i)}, p_k^{(j)}) \in \{T_2, T_4\} \\ \phi, & (p_k^{(i)}, p_k^{(j)}) \in \{T_3, T_6, T_7, T_8, T_9\} \end{cases} \qquad (k = 1, \ldots, n-1).$$

Then, the mutual correlation $R_{i,j}(\tau)$ of $P_i(x)$ and $P_j(x)$ is $R_{i,j}(\tau) = \begin{cases} 2^{n_9}, & \tau \in A \\ 0, & \tau \notin A. \end{cases}$

Proof. Let a_n be a vector composed of the elements $a \in \Im$ of length n. For example, $1_5 = (11111)$. We consider two terms P_i and P_j and assume without loss of generality that they are as follows:

$$P_i: \; (0_{n_1} 0_{n_2} 0_{n_3} 1_{n_4} 1_{n_5} 1_{n_6} \phi_{n_7} \phi_{n_8} \phi_{n_9}),$$
$$P_j: \; (0_{n_1} 1_{n_2} \phi_{n_3} 0_{n_4} 1_{n_5} \phi_{n_6} 0_{n_7} 1_{n_8} \phi_{n_9}),$$

Table 1

$A_{i,j}$, $R_{i,j}$	1	2	3	4
1	$(0,0,\phi,\phi),4$			
2	$(0,1,\phi,\phi),2$	$(0,0,0,\phi),2$		
3	$(1,0,\phi,\phi),2$	$(1,1,\phi,\phi),1$	$(0,0,\phi,0),2$	
4	$(1,1,\phi,\phi),1$	$(1,0,0,\phi),1$	$(0,1,\phi,1),1$	$(0,0,0,0),1$

where $\sum_{i=1}^{9} n_i = n$. Let S_i be the set $x \in GF(2^n)$ corresponding to the term $P_i(x)$ and consisting of x such that for them $P_i(x) \neq 0$. Any $x \in S_i$ has the form

$$x = (0_{n_1}0_{n_2}0_{n_3}1_{n_4}1_{n_5}1_{n_6}x_{n_{789}-1}\ldots,x_1,x_0), \quad \text{where} \quad n_{789} = n_7 + n_8 + n_9.$$

We notice that the set A is definable as follows:

$$(0_{n_1}1_{n_2}\phi_{n_3}1_{n_4}0_{n_5}\phi_{n_6+n_{789}}).$$

Then, for any $\tau \in A$ and $x \in S_i$:

$$x + \tau \in (0_{n_1}1_{n_2}\phi_{n_3}0_{n_4}1_{n_5}\phi_{n_6+n_{789}}).$$

Obviously, $x+\tau \in S_j$ if the bit n_7 in x is equal to the bit n_7 in τ and the bit n_8 in x and τ have opposite values, that is, $x_k = \begin{cases} 1+t_k, & k = n_9,\ldots,n_9+n_8-1 \\ t_k, & k = n_9+n_8,\ldots,n_9+n_8+n_7-1. \end{cases}$

Stated differently, for the given $\tau \in A$ the cardinality of the intersection of the sets $S_i+\tau$ and S_i is equal to 2^{n_9} and, therefore, $R_{i,j} = \begin{cases} 2^{n_9}, & \tau \in A \\ 0, & \tau \notin A, \end{cases}$ which is what we set out to prove.

Example 3. Let the logic function of four variables be given by the following set of the orthogonal terms:

$$P_1 = (0,0,\phi,\phi), \quad P_2 = (0,1,1,\phi), \quad P_3 = (1,0,\phi,0), \quad P_4 = (1,1,1,1).$$

We consider P_3 and P_4. Their corresponding types of pairs are (T_5,T_2,T_8,T_2). Various τ for which the mutual correlation $R_{3,4}(\tau)$ is other than zero are the elements of the term $A_{3,4} = (0,1,\phi,1)$. On these τ, $R_{3,4}(\tau)$ has value $2^{n_9} = 2^0 = 1$. Similarly, the terms P_2 and P_3 have pair types (T_2,T_4,T_6,T_7) and, consequently, $R_{2,3}(\tau) = 1$ for $\tau \in A_{3,4} = (1,1,\phi,\phi)$; otherwise, $R_{2,3}(\tau) = 0$.

The elements of the set $A_{i,j}$ for the function of Example 3 are compiled in Table 1 which contains τ for which the corresponding values of autocorrelation are other than zero.

For example, the value of $R(2) = R(0010)$ is calculated as follows. The pairs of terms with nonzero correlation for $\tau = 0010$ are $(i,j) = \{(1,1),(3,3)\}$ and, consequently, $R(2) = R_{1,1} + R_{3,3} = 4 + 2 = 6$. Another example is given by the autocorrelation function $R(7) = R(0111) = 2R_{1,2} + 2R_{3,4} = 2 \times 2 + 2 \times 1 = 6$ to which the expression

$$\begin{aligned} R(x_3,x_2,x_1,x_0) = {} & 4\overline{x}_3\overline{x}_2 + 4\overline{x}_3x_2 + 2\overline{x}_3\overline{x}_2\overline{x}_1 + 4x_3\overline{x}_2 + 2x_3x_2 \\ & + 2\overline{x}_3\overline{x}_2\overline{x}_0 + 2x_3x_2 + 2x_3\overline{x}_2\overline{x}_1 + 2\overline{x}_3x_2x_0 + \overline{x}_3\overline{x}_2\overline{x}_1\overline{x}_0 \end{aligned}$$

corresponds. Therefore, the autocorrelation function of the logic function described by the set N of the orthogonal terms is representable in a compact form consisting at most only of $N(N+1)/2$ terms. Each term corresponds to the value of $v_{i,j}$ defined as follows: $v_{i,j} = \begin{cases} R_{i,j}, & i = j \\ 2E_{i,j}, & i \neq j. \end{cases}$

The value of the autocorrelation function is calculated like the value of the Boolean function represented as a set of orthogonal terms. At that, the logic OR is replaced by an arithmetic sum.

6. GENERATION OF σ FROM τ

Linearization results in a nonsingular matrix of the linear transformation $\sigma = T^{-1}$ which is the product of n' matrices ($n' \le n$), $\sigma = \sigma_{n'} \dots \sigma_2 \sigma_1$. Each of the matrices $\sigma_i = T_i^{-1}$ is calculated in i steps, $i = 1, \dots, n'$, with the use of τ. The matrices T_i and σ_i are representable as the product of two nonsingular matrices Π_i and L_i: $T_i = \Pi_i L_i$ and $\sigma_i = L_i \Pi_i$, where Π_i is the permutation matrix ($\Pi_i = \Pi_i^T = \Pi_i^{-1}$), L_i possesses the unit value on its diagonal, and each column has a Hamming weight greater than one. It is clear that L_i satisfies $L_i^{-1} = L_i$. We present an example.

Example 4. Consider a linear transformation obeying the matrix

$$\sigma_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

For the element $b = (b_{n-1}, \dots, b_1, b_0) \in GF(2^n)$ and the superscript k, we denote by $\chi = \chi(b, k)$ the index of bit which is equal to 1 and satisfies $\chi \ge k$. We assume without loss of generality that $\chi = \chi(b, k) = \lfloor \log_2(b) \rfloor$ and define $\widetilde{b}^{(k)}$ as the binary assembly

$$\widetilde{b}^{(k)} = \begin{cases} (b_{n-1}, \dots, b_1, b_0), & b_k = 1 \\ (b_{n-1}, \dots, b_{\chi+1}, \overline{b}_\chi, b_{\chi-1}, \dots, b_{k+1}, \overline{b}_k, b_{k-1}, \dots, b_1, b_0), & b_k = 0, \end{cases}$$

where $\overline{b}_i$ stands for the inverted bit b_i. We notice that the $(k+1)$st bit is always 1. For example,

$$b = (001001), \quad \chi = 3 \Rightarrow \widetilde{b}^{(0)} = (001001),$$
$$b = (001010), \quad \chi = 3 \Rightarrow \widetilde{b}^{(2)} = (000110).$$

The corresponding matrices Π_i and L_i are defined as follows.

At the ith step, the binary vector $\tau = (t_{n-1}, \dots, 1, 0)$ has a decimal value greater than or equal to 2^{i-1}. Therefore, $\chi \ge i - 1$. The structure of the permutation matrix Π_i depends at the ith step on t_{i-1}. If $t_{i-1} = 1$, then Π_i is an identity matrix; otherwise, Π_i is the permutation matrix interchanging the ith and χth bits of the vector.

The matrix L_i is defined as follows:

$$L_i = \begin{pmatrix} I_{(n-i)\times(n-i)} & \left|\widetilde{\tau}^{(i-1)}\right| & 0 \\ 0 & & I_{(i-1)\times(i-1)} \end{pmatrix}, \tag{5}$$

where $I_{(k\times k)}$ is the $(k \times k)$ identity matrix.

Example 4 has the parameters $n = 4$ and $i = 2$, and τ is equal to 12, which is greater than $2^{(i-1)}$. The binary representation of τ is (1100); therefore, $\chi = 3$. The second bit is 0. Consequently, $\widetilde{\tau}(1) = (0110)$.

7. LINEAR TRANSFORMATION OF THE ORTHOGONAL TERMS

The matrix of linear transformation σ and a new set of the transformed orthogonal terms $\hat{F} = (\hat{P}_i, Y_i)$, $i = 0, 1, \dots, \hat{N}$, result from the linearization procedure. The original term P is representable as a sum of minterms $P(x) = \sum m_k(x)$, the transformed term, as the sum $\hat{P} = \sum \hat{m}_k(x)$. At that, the relation $P(x) = \hat{P}(\sigma x)$ is retained or $m_k(x) = \hat{m}_k(\sigma x) = m_j(\sigma x)$. Therefore, $j = \sigma k$. Stated differently, the new set of terms results from the action of the operator σ on the original set of terms. We notice that $\hat{N}$ may be other than N. It is desirable that $\hat{N} < N$.

The transformed set $\hat{F} = \sigma F$ is calculated at the first steps of the linearization procedure. Let $F_0 = F$ and $F_{n'} = \hat{F}$. Then, at the ith step of the procedure $F_i = \sigma_i F_{i-1}$. It follows from Eq. (2) that the autocorrelation function $R_i(\tau)$ of the set F_i satisfies the equality

$$R_i(\tau) = R_{i-1}(\sigma_i^{-1}\tau) = R_{i-1}(T_i\tau). \tag{6}$$

Since T_i in the lower right side is an identity matrix, the following lemma is valid.

Lemma 1. *For $\tau = 2^k$, $0 \le k < i-1$,*

$$\begin{gathered} R_i(2^k) = R_{i-1}(T_i 2^k) = R_{i-1}(2^k), \\ R_i(2^k) = R_{i-1}(2^k) = R_{i-2}(2^k) = \ldots = R_k(2^k), \\ R_i(2^k) \ge R_{k-1}(2^k). \end{gathered} \tag{7}$$

At the ith step, the value of the complexity function μ_i of the set F_i is equal to $\mu_i = \hat{R}_i(I) = \hat{R}_{i-1}(T_i)$, T_i replacing one of the previous basic vectors by a vector with greater value of the autocorrelation function. Therefore, the following theorem is valid.

Theorem 2. *If not constraints are imposed on the values of the Hamming weight for τ, then $R_i(2^k) \le R_i(2^{k-1})$ and $\mu_i \ge \mu_{i-1}$ is valid for $k = 0, 1, \ldots, i-1$.*

We describe now a simplified procedure of linear transformation of terms. The operator Π_i only reorders the variables. The operator L_i on the set $F = \{(P_j, Y_j)\}_{j=1}^N$ is defined as $L_i F = \left\{\left(\left(L_i P_j^T\right)^T, Y_j\right)\right\}_{j=1}^N$.

In some cases, a pair of terms results from the multiplication of a single term P by L_i. For example, if $\sigma_1 = L_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and $P = (1, \phi, 0) = \{(1,0,0), (1,1,0)\}$, then the transformed term is equal to $\sigma_1 P^T = P^T$. However, for $P = (1, 1, \phi)$ we get $\sigma_1 P^T = \{(1,1,0), (0,1,1)\}$. Obviously, this function is not representable as a single term.

The following lemma guarantees that as the result of linearization the term either may remain single or divide into two individual terms.

Lemma 2. *Let given be the term $P = (p_{n-1}, \ldots, p_1, p_0) \in \Im^n$, and let L_i be the matrix of linear transformation defined as (5), $L_i \ne I$. If $p_{i-1} = \phi$, then after the transformation $L_i P$ the original term turns into a pair of orthogonal terms. At that, the "distance" between these terms is equal to or greater than 2. Otherwise, the original term turns into a single term.*

Proof. Any term having a free variable at the ith position can be decomposed into two terms $P^{(0)}$ and $P^{(1)}$ of smaller dimensions. These terms have, respectively, 0 and 1 at the ith position.

Let $\hat{P} = (\hat{p}_{n-1}, \ldots, \hat{p}_1, \hat{p}_0)$ be the term where $\hat{p}_{i-1} \in GF(2)$ and $\hat{p}_j \in \Im$ for all $j \ne i-1$. The term $\hat{P}$ corresponds to the subset of $\{0,1\}^n$. Let L_i be the matrix of linear transformation (5), that is, $L_i = I_{n\times n} + (0_{n\times n-i}, \tilde{\tau}^{(i-1)} - \delta_{i-1}, 0_{n\times i-1})$, where $0_{a\times b}$ is a zero $(a \times b)$ matrix. In this case, the transformed term $L_i\hat{P}$ corresponds to the subset $\hat{P} + (\tilde{\tau}^{(i-1)} - \delta_{i-1})\hat{p}_{i-1}$.

It is clear that if L is an identity matrix or if $p_{i-1} = 0$, then the transformed term $L_i\hat{P}$ is equal to the original term P. If $p_{i-1} = 1$, then the transformed term is at least at the Hamming distance 1 from it ($d(L_iP, P) \ge 1$). Additionally, if $L_i \ne I$, then $d(L_iP^{(0)}, L_iP^{(1)}) = \|(\tilde{\tau}^{(i-1)} - \sigma_{i-1})\| + 1 \ge 2$, where $\|a\|$ is the Hamming weight. Therefore, the term P where the ith position corresponds to the "free" variable is decomposed into two terms. And since these two term are at distance two from each other, they cannot make up a single term, which is what we set out to prove.

According to Lemma 2, to calculate L_iP, it suffices to determine the sum of the binary vector $V = (v_{n-1}, \ldots, v_1, v_0)$ and the term $P = (p_{n-1}, \ldots, p_1, p_0) \in \Im^n$, at that $p_{i-1} \neq \phi$. Let $a \in GF(2)$, $b \in \Im$, and let $\Diamond$ denotes the noncommutative operation

$$a \Diamond b = \begin{cases} a \oplus b, & b \neq \phi \\ \phi, & b = \phi. \end{cases} \tag{8}$$

Then, $V + P = W = (w_{n-1}, \ldots, w_1, w_0)$, where $w_j = v_j \Diamond p_j$.

Example 5. Let the system of three logical functions of four input variables be defined by the following set of the orthogonal terms:

$$F = \left\{ \begin{array}{l} (0,0,0,1),(1,1,1) \\ (1,1,j,0),(1,1,1) \\ (0,j,1,j),(1,0,0) \end{array} \right\}.$$

Let $i = 2$ and

$$\sigma = L_2\Pi_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

After the transformation σ, the transformed set $\hat{F}$ is constructed as follows. First, the variables are permuted:

$$\Pi_2 F = \left\{ \begin{array}{l} (0,0,0,1),(1,1,1) \\ (\phi,1,1,0),(1,1,1) \\ (1,\phi,0,\phi),(1,0,0) \end{array} \right\}$$

and then multiplication by L_2 is performed: $\hat{F} = L_2(\Pi_2 F) = \left\{ \begin{array}{l} (0,0,0,1),(1,1,1) \\ (\phi,0,1,0),(1,1,1) \\ (1,\phi,0,\phi),(1,0,0) \end{array} \right\}$. We notice that this example was aimed just to illustrate the method of linearization which here did not reduce the number of terms.

The terms of the transformed set feature orthogonality. This fact enables recursive linearization. Indeed, if $L_i = I$, then after linearization the number of transformed terms remains the same. If $L_i \neq I$, then, as is shown by the following example, after transformation the number of terms may reduce owing to "gluing."

Example 6. Let the function of five variables F be represented as a set of orthogonal terms. It takes on a unit value at the vertices of the Boolean cube corresponding to the numbers 0, 6, 9, 11, 12, 16, 18, 24, and 30. The autocorrelation function of this Boolean function has the greatest value for $\tau = 30 = (11110)$, which defines at the first step the matrix of linear transformation σ_1:

$$\sigma_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

After this linear transformation, the new set of terms $\sigma_1 F$ is defined as follows:

$$L_1\Pi_1 F = \left\{ \begin{array}{l} (0,0,0,0,0),(1) \\ (0,0,0,0,1),(1) \\ (0,0,1,1,0),(1) \\ (0,0,1,1,1),(1) \\ (0,1,1,0,0),(1) \\ (0,1,1,\phi,1),(1) \\ (1,1,0,\phi,0),(1) \end{array} \right\} = \left\{ \begin{array}{l} (0,0,0,0,\phi),(1) \\ (0,0,1,1,\phi),(1) \\ (0,1,1,0,0),(1) \\ (0,1,1,\phi,1),(1) \\ (1,1,0,\phi,0),(1) \end{array} \right\}.$$

As can be seen from the example, the number of terms reduced already after the first step of the linearization procedure.

8. EXPERIMENTAL RESULTS

This section represents the results of experiments with the standard example from the MCNC assembly. Different algorithms were compared in terms of the complexity function and the number of vertices in the corresponding decision diagrams.

Table 2 shows the values of the complexity function before and after linearization. In this table, μ_{orig} is the value of the original function before linearization and μ_k, after the K-procedure without any constraint on the Hamming weight τ, μ_{dc} is the complexity of the linearized system after executing the proposed procedure over the space of orthogonal terms. The maximal Hamming weight τ is $w = 3$, μ_{upb} is the upper bound of the weight function μ defined as the sum of n maximal values of the autocorrelation function R. We notice, however, that the corresponding T may be single and, therefore, the upper bound is not always attainable.

We note that both the K-procedure and the algorithms of linearization of the separated terms are the so-called "greedy" and, therefore, nonoptimal algorithms. However, in all examples $\mu_{dc} \geq \mu_k$, though there is no constraint on τ in the K-procedure, whereas μ_{dc} are constrained by $w = 3$. An improvement in the complexity function is reached owing to the fact that the proposed method considers more τ.

Table 3 shows the values of the complexity function and the greatest values of Hamming weight τ_i for $w = 1, 2, 3, 5, 7$. As was expected, in the case of $w = 1$ the algorithm replaces the basic vectors and orders them according to the value of the autocorrelation function. Therefore, for $w = 1$ the complexity function is equal to the original (prior to linearization) μ. It follows from the

Table 2. Complexity function of the original and linearized function for the assembly of standard examples

Example	n	k	μ_{orig}	μ_k	μ_{dc}	μ_{up}
z4	7	4	320	412	476	588
sqn	7	3	292	310	348	504
rd73	7	3	308	476	568	644
5xp1	7	10	512	520	578	670
inc	7	9	304	304	324	560
misex1	8	7	1472	1536	1664	2048
radd	8	5	824	1112	1304	1556
root	8	5	868	870	942	1712
f51m	8	8	884	1076	1244	1536
adr4	8	5	1040	1212	1340	1492
dc2	8	7	820	820	888	1310
clip	9	5	2170	2562	2792	3164

Table 3. Comparative complexity of realization of the original and linearized functions for different w

	n	k	μ_{orig}	$w=1$	$w=2$	$w=3$	$w=5$	$w=7$
misex1	8	7	1472	1472	1664	1664	1664	1664
dist	8	5	638	638	680	700	700	700
f51m	8	8	884	884	1172	1244	1264	1268
adr4	8	5	1040	1040	1284	1340	1340	1340

Table 4. Number of vertices in the decision diagrams of the original and linearized functions

Example	n	k	v_{orig}	v_k	v_{dc}
sqn	7	3	81	57	54
rd73	7	3	24	31	29
5xp1	7	10	59	47	50
inc	7	9	39	37	37
misex1	8	7	8	7	5
radd	8	5	63	42	31
root	8	5	70	69	71
f51m	8	8	81	78	83
adr4	8	5	68	43	36
dc2	8	7	73	111	114
clip	9	5	135	100	89

Table 5. Parameters of the decision diagrams in the original and linearized functions

	n, k	N, N_{dc}	μ_{orig}, μ_{dc}	v_{orig}, v_{dc}	P_{orig}, P_{dc}	APL_{orig}, Apl_{dc}
dk27	9, 9	10, 41	2192, 2192	79, 44	86, 47	6.31, 4.91
sao2	10, 4	58, 142	8244, 8258	95, 75	237, 92	7.10, 2.90
dk17	10, 11	18, 128	5950, 6006	160, 91	377, 106	8.39, 5.71
apla	10, 12	26, 89	6818, 6836	128, 115	264, 161	7.19, 6.29

experimental results that for the most part it is possible to reduce the linearization complexity by constraining the Hamming weight.

Table 4 shows the number of vertices in the decision diagrams before and after linearization. In the table: v_{orig} is the original number of vertices of the decision diagrams before linearization, v_k is the number of vertices after the K-procedure without constraints on the Hamming weight τ, v_{dc} is the number of vertices after linearization of the orthogonal terms under the constraint $w=3$. We notice that for example rd73 both linearization algorithms lead to an increased size of the decision diagram. However, the complexity function μ was improved, which means that on the average linearization enables one to obtain smaller decision diagrams and the complexity function does not always ideally correspond to the minimal decision diagram.

There are areas of application of the linearization method where it is only natural to do representation on orthogonal terms. For example, the sequential circuits described in terms of the *algorithm flowgraphs* [3]. On the other hand, in the cases where the original representation of a function consists of an assembly of nonorthogonal terms, the orthogonal representation can be obtained with the use of standard programs such as ESPRESSO.

Table 5 presents data about the sizes of the original assembly of term, number of terms, as well as the data about the impact of linearization on the complexity of the original function in terms of the parameters of the corresponding binary decision diagrams. The first column corresponds to

the name of the example function, the second, to the number of input (n) and output (k) variables. Column 3 corresponds to the number of terms (N) in the original representation of the function and to the number of terms (N_{dc}) in the representation of this function. Column μ corresponds to the value of the complexity function of the original ($orig$) and linearized (dc) functions. The rest of the columns correspond to the number of vertices (v), number of paths, and the mean length of the path of the binary decision diagrams, respectively, in the original and linearized functions.

We notice that the permutation matrix was obtained as the result of linearization of function d27. Therefore, the complexity function of the original function is equal to the complexity function of the linearized function ($\mu_{orig} = \mu_{dc}$). Nevertheless, the new order of the basic vectors enabled one to obtain a binary decision diagram with improved characteristics.

The proposed algorithm is applicable to the functions where representation by the orthogonal terms has a small number of terms (see Section 9). However, the set of nonorthogonal terms can be divided into several subsets of orthogonal terms, and to each of them the linearization algorithm can be applied.

9. COMPLEXITY OF THE LINEARIZATION PROCEDURE

Linearization of functions defined in orthogonal terms is useful if the number of inputs or the number of the considered characteristic functions is great. In these cases, linearization of functions defined by the truth table is inefficient.

For linearization by the truth table, complexity of the linearization procedure (K-procedure) is calculated as follows. The algorithm optimizes the number of vertices at the levels beginning from the roots of the binary tree. There are at least n levels. At each level, the autocorrelation function R is calculated, then the best τ is selected, and the function is convoluted. At the ith level, the output is convoluted i times; therefore, the output has $k2^i$ bits. Let $C(i)$ denote the number of different characteristic functions at the ith level, $C(i) \le 2^{k2^i}$. Complexity of the Walsh transformation of the function of z variables is $z2^z$; therefore, complexity of the K-procedure based on calculating autocorrelation from the Wiener–Khinchin theorem at the ith level is equal to $\max_i O\left((n-1)C(i)2^{n-1}\right)$.

The algorithm for linearization on the orthogonal terms calculates the autocorrelation function for τ with the greatest Hamming weight w. The exact number $W = \sum_{k=0}^{w} \left(\begin{array}{c} n \\ k \end{array} \right) - \left(\begin{array}{c} i \\ k \end{array} \right)$ of the candidates for calculation of $R_u(\tau)$ is known at the ith step. We denote by $N^{(i)}$ the number of terms at the ith step, and let $\tilde{N}$ reflect the greatest number of terms in the course of the procedure, that is, $\tilde{N} = \max_i(N^{(i)})$.

Complexity of calculation of $R_u(\tau)$ has the order of WN_u^2, where N_u is the maximal number of terms having the characteristic u. It is clear that in distinction to [2], the greater the number of characteristic functions, the smaller the calculation complexity $R(\tau) = \sum_u R_u(\tau)$. The total complexity of the algorithm does not exceed $O(nW\tilde{N}^2)$.

We notice that for the systems with many logical functions the linearization of functions defined by the orthogonal terms is more efficient than for those defined by the truth table because complexity of the proposed procedure is independent of k. Moreover, even to one function, if $N < \sqrt{2^n/n^3}$, then representation of the function by the set of orthogonal terms is preferable because for $w = 3$

$$O_{dc} < O(n^4N^2) < O(n2^n) < O_k,$$

where O_{dc} is the complexity of the proposed linearization procedure in the space of orthogonal terms and O_k is the complexity of the K-procedure.

10. CONCLUSIONS

The linear transformation of variables proved to be efficient for reduction of complexity of the logical functions. In the present paper, the linearization-based approach was extended so as to work for the functions defined by the orthogonal terms and not only the truth tables or the decision diagrams. The main results obtained are as follows.

(1) Proposed was a method for calculation and compact representation of the autocorrelation function for the functions defined by a set of orthogonal terms.

(2) Proposed was a method to construct the corresponding linear transformation matrix.

(3) Proposed was a method of linear transformation of variables in the system of Boolean functions with the use of the linear transformation matrix.

(4) It was demonstrated that complexity of calculation for the proposed method does not exceed $O(nW\tilde{N}^2)$. As the result, the calculations based on the representation of functions by orthogonal terms have the edge over the calculation based on the truth tables in the cases where these functions are described by less than $N < \sqrt{2^n/n^3}$ orthogonal terms.

The experiments corroborate efficiency of the proposed methods. The present authors are sure that the above results will extend applicability of the method of linearization owing to its efficiency.

REFERENCES

1. Karpovsky, 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.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.

3. Baranov, S.I., *Sintez mikroprogrammnykh avtomatov: graf-skhemy i avtomaty* (Design of Microprogram Automata: Flowcharts and Automata), Leningrad: Energiya, 1979.

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