Walsh Spectral Techniques for Logic Synthesis FPGA

Nhan Khanh Huu NGUYEN

Department of Electronics and Telecommunications, Faculty of Electrical & Electronics Engineering,
Ton Duc Thang University, Nguyen Tho Street, Tan Phong Ward, District 7,
Ho Chi Minh City, Vietnam

nguyenhuukhanhnhan@tdt.edu.vn

DOI: 10.15598/aeee.v13i2.1117

Abstract. *The implementation value of multi-output Boolean functions in logic synthesis FPGA can be reduced by using Walsh spectral representation. This paper proposes an algorithm for calculating the maximum coefficient of the autocorrelation function of BF without generating a truth table, using the heuristic procedure limits the maximum autocorrelation coefficients of sorting on a small subset of the function. We also suggest a spectral technique of the linear function transformation defined by disjoint cubes. This method for decomposition of BF, which allows to reducing the complexity of the linear part of the corresponding blocks about 25–55 %, and the complexity of the nonlinear part of the blocks do not increase more than 10 %, compared to the traditional approach.*

Keywords

Autocorrelation, Boolean function, disjoints cubes, logic synthesis, Walsh spectrum.

1. Introduction

There are two popular categories of field programmable gate array (FPGA) block structures, namely Look-Up Table-based (LUT) and multiplexor-based (MB); the resulting architectures are called LUT-based and MUX-based architectures respectively [1].

The basic block of an LUT architecture is a look-up table that can implement any Boolean function of up to m inputs, $m \geq 2$. For a given LUT architecture, m is a fixed number. In commercial architectures, m is typically between 3 and 6. An m-LUT is typically implemented by static random access memory (SRAM) that has m address lines and 1 data line. An m-LUT can implement any Boolean function of up to m inputs.

In MB architecture core logic element has a structure consisting of configuration multiplexers. An example is the architecture proposed by Actel [5], in which the base unit has a configuration comprising three elements of the multiplexer; and/or series of logical blocks separated trace channels, consisting of the actual trace of the system and global synchronization [1], [2] and [3].

In this paper will be used architecture LCA (Logic Cell Array) type TLU of Xilinx Company, base on configurable logic blocks (CLBs), are bigger and more complex than the Actel or QuickLogic cells. The Xilinx LCA basic logic cell is an example of a coarse-grain architecture. The Xilinx CLBs contain both combinational logic and flip-flops [4].

The first generation of LCA devices appeared in 1985. They consist of logical blocks that include the generator combination of functions that implements the 4-input foundation, and the only element of memory and the trigger. Family of crystals marked with the symbol XC2000 and had two structures with conventional equivalent complexity from 1200 to 1800 two-input elements (gates).

The second generation of LCA devices, which appeared on the market in 1987, included the logic blocks, extended to implement the 5-input foundation, as well as containing two triggers. Corresponding family of crystals marked with the symbol XC3000 and had five structures, ranging in complexity from 1200 to 5000 gates. Clock frequency XC3000 reaches 125 MHz, which is equivalent to clock frequency of the system in the 30–40 MHz.

The third generation of LCA devices appeared in 1991. It further increased the possibilities of this architecture. In addition, in this series for the first time it was possible to reconcile the memory of a random sample and a combinational logic on a single chip. Corresponding family of crystals be marked XC4000, has ten structures, ranging in complexity from 2000 to 20000 gates. System clock frequency is 60–70 MHz, which is approximately higher than two times in the previous series.

The main drawback of the crystals XC4000 is the underutilization of their resources. The maximum "occupancy" of the crystal does not exceed 70–80 %, since the greater utilization of the crystal having trouble tracing. To solve this problem, the fourth generation architecture of the crystals (series XC5200), submitted in 1996, was redesigned in the direction of greater "traceability" and the possibility of more "waste" of resources. XC5200 family has five structures, ranging in complexity from 2000 to 23000 gates with the system clock frequency of 50 MHz.

As already mentioned in this paper, the main difference between the latest developments in the area of the LCA devices, will based on static memory technology is to improve the characteristics of the trace of the crystal. Thus, the analysis of architecture and technology of FPGA allows us to conclude that, in addition to common for the entire microelectronics industry trends to increase the degree of integration, improving overall performance, reduce costs, etc., the new trend is the increased ease of design and debug circuits. However, increasing complexity of both the integrated density and application requirements become higher every pasing day. Those are questions of design and development of algorithms for automatic logic synthesis. It follows that the main problems of logic synthesis in the FPGAs minimize the number of used logic blocks and reduce the complexity of the trace.

2. Spectral and Correlation Analysis of Boolean Functions

We use the definition of BF in the monographs [6], where they are treated as multi-dimensional functions with m-inputs and k-outputs, and carry out mapping of the form f:{0,1}^{m} → {0,1}^{k}. Set of outputs is denoted as BF f_{k-1}, ... f_0, and used the decimal indices $x = (X_{m-1}$, ... $X_0) \in 0{,}1^m$ is calculated the formula:

$$x = \sum_{i=0}^{m-1} x_i 2^i, \tag{1}$$

$$f = (f_{k-1}, ...f_0) \in 0, 1^k, f = \sum_{i=0}^{k-1} f_i 2^i, \tag{2}$$

where x and f can be interpreted as the coordinates of the binary vectors to decimal numbers. Note that the Eq. (1) and Eq. (2) describe the BF as a piecewise constant function *F(x)* of real argument on the half-open interval [0.2 m]. With this notation system of BF can be represented as a lattice of y = *f(x)*, defined at the points 0, 1, ... , 2^m -1 interval [0, 2^m]. Extend the function y = *f(x)* to piecewise constant function *F(x)* as follows:

$$F(x) = f(\delta) \text{ variations } x \in [\delta, \delta + 1]. \tag{3}$$

We say that a piecewise constant function *F(x)* represents the original system of BF, if it satisfies the condition in Eq. (3) and *f(x)* is constructed by equations Eq. (1) and Eq. (2). Thus, the foundation can be described as a vector F=[f(0),f(1), ..., f(2^m-1)]^T, where x=(x_{m-1}, ..., X_0), ($0 < x < 2^m - l$) - a set of input vectors, and *f(x)* is an integer value, here F_i=[f_i(0), f_i(1), ..., f_i(2^m-1)]^T, and $f_i(x)$, 0< $i < k - l$, is a binary value.

It is known that between BF and Walsh functions, there is a relationship, which explains the possibility of effective use of spectral analysis in the basis of Walsh functions to analyze the fleet. In order to determine this relationship, we consider details of the Walsh function. These functions are piecewise constant and are given on the half-open interval [0, 2^m] expression:

$$W_\omega(x) = (-1)^{\sum_{i=0}^{\omega-1} \omega_{(m-l-i)^2}}, \tag{4}$$

where 0< $\omega < 2^m - 1, m \in N$, and ω_i and x_i are determined from the binary representations ω and x.

Autocorrelation function of BF $f(x_0$, x_1, ..., $x_{m-1})$ is determined on the basis of relations:

$$B_2^{f;f}(\tau) = \sum_{x=0}^{2^m-1} f(x) f(x \oplus \tau), \tag{5}$$

where $\tau \in$ 0, 1, ..., 2^m-1. As seen from Eq. (5), the original function is related to the autocorrelation function of convolution transforms. Cross-correlation or simply the correlation function of two BF $f_1(x)$ and $f_2(x)$ is the function:

$$B_2^{f_1;f_2}(\tau) = \sum_{x=0}^{2^m-1} f_1(x) f_2(x \oplus \tau), \tag{6}$$

where $\tau \in$ 0, 1, ..., 2^m-1. Establish a connection between the correlation functions and features considered earlier Walsh, also known as Wiener-Khinchin theorem [7] and [8]:

$$B_{2,2}^{f_1;f_2} = 2^{2m} W(W(f_1)W(f_2)). \tag{7}$$

Properties of the correlation characteristics of BF determined by the properties of convolution transforms of the original features. In particular, the form of these transformations implies the invariance of the correlation characteristics to shift the argument of the original. Converse is also true that the autocorrelation function of the original function can be restored up to a shift of the argument.

The complexity of BF is usually understood as the minimum number of two-input elements necessary for the construction of the scheme; it realizes that the complexity criteria are now known a lot. The simplest and most natural criterion of BF $f(x_0, x_1, ..., x_{m-1})$, $x_i \in \{0, 1\}$, i=1, ..., n-1 is the number $\mu_0(f)$, ($\mu_0(f) \leq m$), which equals the number of arguments to this function, from which it depends, it is assumed that the function essentially depends on the arguments x_i, if there are $\alpha, \beta \in \{0, 1\}$, such that for any set of arguments (x_0, ..., x_{i-1}, x_{i+1}, ..., x_{m-1}) value with $f(x_0$, ..., x_{i-1}, α, x_{i+1}, ..., $x_{m-1}) \neq f(x_0$, ..., x_{i-1}, β, x_{i+1}, ..., x_{m-1}), [9].

This criterion is called the μ_0, we note that this assessment is quite easy to get, but it is μ_0 criterion of BF very weakly associated with specific properties of the original BF.

Frequently uses criterion of BF μ_1. To determine this, we use the notion of Hamming distance in the discrete Euclidean space, i.e. if $x_1 = (x_1^0$, ..., $x_1^{m-1})$ and $x_2 = (x_2^0$, ..., $x_2^{m-1})$; ($x_1^{(i)}$, $x_2^{(i)} \in \{0,1\}$) then the Hamming distance between x_1 and x_2 will be:

$$d(x_1, x_2) = \sum_{i=0}^{m-1} \mid x_1^{(i)} - x_2^{(i)} \mid . \tag{8}$$

Then the complexity of BF $\mu_1(f)$, we mean the number of vectors pairs $\{x_1, x_2\}$ with Hamming distance between them $d(x_1, x_2) = 1$ such that $f(x_1) \neq f(x_2)$. Similarly, we introduce criteria of BF μ_r, where $r = d(x_1, x_2)$. Strength criteria with increasing r, but also increases the complexity of their calculation are determined by C_m^r 2^m. Note that μ-criteria of BF may be related to their correlation functions. Indeed, since the number of true minterm at a distance, for example, 1 corresponds to the values of the autocorrelation function of BF in points τ = 1, 2, 4, ..., 2^m-1, then the function

$$\psi(f) = \sum_{\tau=1,2,4,\ldots,2^{m-1}} B^{(f,f)}(\tau), \tag{9}$$

can be regarded as a measure of simplicity of this function, and, as shown in [7], $\mu(f) = km2^m$-1-$\psi(f)$. Consider a set of m linear transformations of the arguments of the original BF $f(z)$. BF obtained to be denoted as $f_i(z)$, and their autocorrelation functions - as $B_i(\tau)$; moreover:

$$B(\tau) = \sum_{i=0}^{m-1} B_i(\tau). \tag{10}$$

Denote

$$B(T) = \sum_{i=0}^{m-1} B\left(\sum_{q}^{m-1} \tau_{q,s} 2^{m-1-q}\right), \tag{11}$$

where $T = (\tau_{qs})$, $\tau \in \{0, 1\}$ and (q, s = 0, l, ..., m-l). It is obvious that the function $B(T)$ holds Karpovsky theorem [7], whose formulation is given below.

Let max$_{T\neq 0}B(T)$=$B(T_\eta)$ then σ_η+T_η=E_m(mod2).

Here | T | - determinant T, $\mathbf{E}_m$ - identity matrix size $m \times m$. The importance of this theorem is due to the fact that its use can introduce the concept of an optimal linear transformation of the arguments given BF σ_η. It consists of the following: conversion σ_η, corresponding theorem Karpovsky, considered the optimal linear transformation of the arguments of BF by the criterion η.

3. Decomposition of the Boolean Function

Assume that the BF is implemented using a logic block, shown in Fig. 1 and its decomposition - a block in Fig. 2. Thus, under the decomposition of BF realize its expansion on the linear σ and non-linear f_σ part.

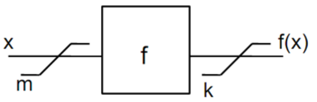

Fig. 1: Original function.

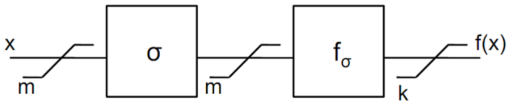

Fig. 2: Linear decomposition.

In the literature [10] is used and the more generalized notion of decomposition, called splitting decomposition or disjoint decomposition. This kind of decomposition illustration in Fig. 3 (in all figures the input variables are designated as $x = (x_{m-1}, ..., x_0)$, and the termination as $f = (f_{k-1}, ..., f_0)$; study of disjoint decomposition foundation is dedicated to monograph [7].

Consider the disjoint-decomposition for different numbers of input variables.

- For 2 variables, there is only one type of decomposition, shown in the Fig. 4a.
- For the 3 variables are known, as illustrated in Fig. 4b, has two types of decomposition, the total

number of functions involved in it will be $C_3^2 + C_3^2$.

- For 4-input variables, number of types of decomposition is three with the total number of functions $C_4^3 + C_4^2 + C_4^1$, as shown in Fig. 4c.
- For 5 variables will be four types of decomposition, the total number of functions involved $C_5^4 + C_5^3 + C_5^2 + C_5^1$, as shown in Fig. 4d.

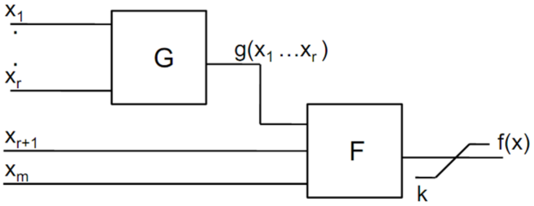

Fig. 3: Splitting decomposition.

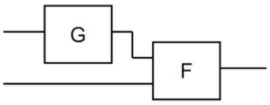

Fig. 4a: Disjoint-decomposition for 2 input variables.

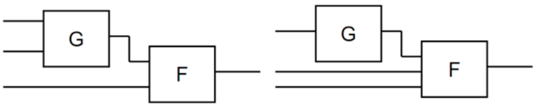

Fig. 4b: Disjoint-decomposition for 3 input variables.

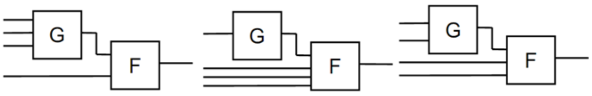

Fig. 4c: Disjoint-decomposition for 4 input variables.

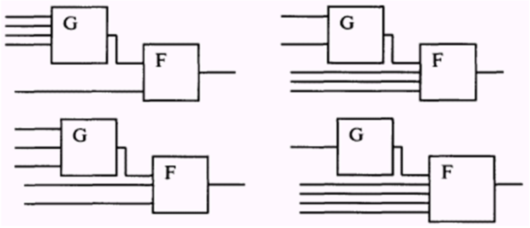

Fig. 4d: Disjoint-decomposition for 5 input variables.

From the analysis of above sections, it follows that for any m input variables exist (m-1) BF type (variant), its decomposition, which involves functions $\sum_{i=1}^{m-1} C_m^i$.

Using the rule of common geometric progression, we estimate the upper limit of the functions mentioned below. Then we obtain:

$$\sum_{i=1}^{m-1} C_m^i \leq \sum_{i=1}^{m} C_m^i = 2^m, \qquad (12)$$

$$\sum_{i=1}^{m} C_m^i = 2^{m+1} - 1.$$

Note that this number is negligible to compare with the total number of BF in m variables.

As shown in [8], the proportion of linear BF volume sets, these functions are involved in the disjoint decomposition and all of BF can be roughly illustrated by Fig. 5, where the set of BF, close to the line, indicated by a dotted line .

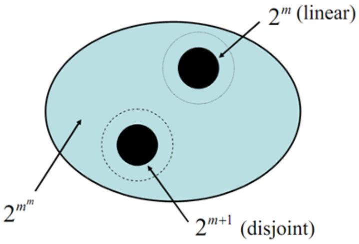

Fig. 5: The conditional distribution functions representation.

4. Linearization Algorithm

The theorem Karpovsky [7] allows finds the linear transformation σ_η, construct the autocorrelation function $B(t)$ is the original BF and m linearly independent samples of its arguments, such that the sum of $B(t)$ for these samples is maximal. These m samples can be found as follows: if we find s samples ($1 \leq s \leq m$), which is denoted as $t_0, t_1, ..., t_{s-1}$, then $(s+1)$ - reading ts from the condition:

$$B(\tau_s) = max_{r \notin \varrho_s} B(\tau). \qquad (13)$$

Here Q_s - the set of all linear combinations of vectors $\vec{\tau}_0, \vec{\tau}_1, ..., \vec{\tau}_{s-1}$ and vector (00...0) modulo 2; τ_i - vector of the binary expansion of t_i. We can show that found $\vec{\tau}_0, \vec{\tau}_1, ..., \vec{\tau}_{s-1}$ thus form the columns of $\mathbf{T}_\eta$. Then the transformation matrix σ_η, the optimal criterion η, can be determined by $\mathbf{T}_\eta$. In addition, to find a linear transformation σ_η, the optimal criterion η, can use the following recursive procedure: Let $\mathbf{T}$ - matrix of size $m \times m$, such that $\mathbf{T} = \tau_0, \tau_1, ..., \tau_{m-1}$ (τ_i − raw size $m \times 1$), other τ_0 is found from the expression:

- Calculated autocorrelation coefficient of BF $B(\tau_0) = max_{|r| \neq 0} B(\tau)$.

- Take $L_0 = \{C_0\tau_0 \mid C_0 \in \{0;1\}\}$, so $L_0 = \{0;\tau_0\}$. τ_1 is as $B(\tau_1) = max_{|r| \neq L_0} B(\tau)$.
- When $\tau_0, \tau_1, ..., \tau_{s-2}$ found, take $L_{s-2} = \{\oplus_{i=0}^{s-2} C_i\tau_i\}$, $C_iin\{0;1\}$ and τ_{s-1} defined as $B(\tau_{s-1}) = max_{|r| \neq L_{s-1}} B(\tau)$.
- Desired transformation $\sigma_\eta = T^{-1}$. Thus, the linear transformation of the BF arguments, the optimal criterion η, is given by:

$$z_i = \oplus_{j=0}^{m-1} \sigma_{ij} x_{m-1-j}, \quad 0 \leq i \leq m-1. \tag{14}$$

In this case, the sum modulo 2 can be realized in z_i, require many inputs, how many units contained in the *i-th* row, that is σ_η, in the worst case complexity of linear part of the BF is proportional to the square of the number of input variables, since the matrix σ_η can contain $m \times (m-1)$ of non-zero values. The nonlinear part f_σ of BF can be calculated by multiplying each minterm $(x_{m-1}, ..., x_0)$ in the matrix σ. The resulting vector will be minterm nonlinear part f_σ of the BF.

To illustrate this fact consider the following example. Assume that the operation described by summing the decimal function $f(x_3, x_2, x_1, x_0) = 2(x_3 + x_1) + x_2 + x_0$. Construct a function $F = [f(0), f(1), ..., f(2^m - l)]^T = [0,1,2,3,1,2,3,4,2,3,4,5,3,4,5,6]^T$; We note that the *i-th* column of F - is the decimal representation of the binary digital signal in the output of three bit adder contained in the *i-th* column of the truth table.

Calculating the autocorrelation function F on Wiener-Khinchin theorem [8], we have $B = [22,8,10,6,8,16,6,14,10,6,18,4,6,14,4,12]$. Next, we use the linearization procedure of BF, as described above: after deleting the coefficient *B(0)*, we find that the maximum coefficient of the autocorrelation function of BF is 18 with the number of columns (address) $\tau_0 = 10$, which corresponds to the binary representation of 1010. Thus, $L_0 = \{0,\ 10\}$. Then, strike out from the vector in the term l_0 find that following its maximum rate is 16 and is located at 5. Thus, $\tau_1 = 5 = 0101$, $L_1 = \{0,\ 10,\ 5,\ 15\}$. Similarly, we find that $\tau_2 = 7$ or 13. Arbitrarily choose a value. Let it be 13 (1101); $L_2 = \{0,\ 10,\ 5,\ 15,\ 13,\ 7,\ 8,\ 2\}$. Note that the L_2 will remain the same regardless of the choice, because it contains a linear combination of 13 and 7. Similarly, we have $\tau_3 = 1 = 0001$. Then:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [\tau_0, \tau_1, \tau_2, \tau_3], \tag{15}$$

$$\sigma = \mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}. \tag{16}$$

Thus, during decomposition of BF initially implemented block σ, which translates as if the original set of input variables x in a different set of z, conversion between them is as follows: $f_\sigma(z) = f_\sigma(x) = f_\sigma(\sigma x)$, $f_0(\sigma x) = f(T_z)$. As an example, consider z = (0010) = 2;

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}. \tag{17}$$

Thus, $f(1101) = f_\sigma(0010) = f_\sigma(2) = 4$, a F_σ described as $[0,1,4,3,2,1,2,3,4,5,4,3,6,5,2,3]^T$.

The main drawback of the above methods of decomposition of a BF is the fact that for the calculation of the autocorrelation function for the Wiener-Khinchin theorem requires a truth table of this function, resulting in a double need to apply the procedure of transformation that requires a $m \cdot 2^{m+1}$ elementary operations. And after the construction of the matrix σ, you want to convert the original truth table of f in the truth table of the function f_σ, resulting in the computational complexity of the problem of decomposition increases exponentially with the number of variables, and memory requirements, as well as high-speed computers have become unacceptably large.

This paper proposes a procedure for calculating the maximum coefficient of the autocorrelation function of BF without generating a truth table, using the heuristic procedure limits the maximum autocorrelation coefficients of sorting on a small subset of the function on the basis of the Varma-Trachtenberg method [11].

5. Disjoint Cubes Performance Analysis

The intersection of two cubes C_i and C_j is the cubic C_l, whose coordinates are defined as follows [9]:

Tab. 1: Intersection coordinates C_i.

			x_p^i	
	x_i^l	0	1	-
x_p^j	0	0	∅	0
	1	∅	1	1
	-	0	1	-

Tab. 2: Intersection coordinates C_j.

			x_p^i
	x_i^l	0	1
x_p^j	0	0	0
	1	0	1

C_l is empty (there is no intersection), if at least one $x_i^l = \varnothing$ or $z_j^l = 0$ for all $l (0 \leq l \leq k-1)$.

A set of cubes representing the function *f*, called a covering of *C(f)*, and the number of elements of the covering C is its size.

Pair-wise intersection (PWI) the set of cubes C - is a set of non-empty cubes are pair-wise intersection of all the cubes of a given set of C_i and C_j, $i \neq j$. In this case, PWI (C) covers all values of the original features that are included in more than one cube of the function.

Weight $w(C_i)$ of the cube C_i is the number of values of the original function, its covering. That is, $w(C_i) = \mid z \mid 2^d$, where $\mid z \mid$ - number of units in the C_i and d - a number of uncertainties in C_i.

$\sum C_i \oplus \tau = \left(x_{m-1}^i \oplus \tau_{m-1}, ..., x_0^i \oplus \tau_0, z_{k-1}^i, ..., z_0^i\right)$, where $\tau = (\tau_{m-1}, ..., \tau_1, \tau_0)$ and

$$x_i^j \oplus \tau_j = \left\{ \begin{array}{c} -; x_i^j = -; \\ x_i^j \oplus \tau_j^i; x_j^i \in \{0; 1\} \end{array} \right\}, \tag{18}$$

for a set of cubes $C, C \oplus \tau = \{C_i \oplus \tau \mid l \leq i \leq C \mid\}$.

Define the excess covering logical function *f*, which can be obtained from the minimum cover the following logical. $C = \bigcup_{i=0}^{N} C^i$, here $C^0 = C(f)$ by definition, and $C^{i+1} = PWI(C^i)$, $C^{N+1} = C^N$ or $C^{N+1} = \varnothing$.

Define a symbol for each cube in C:

$$sign(w(C_i)) = \left\{ \begin{array}{l} +, \text{if } C_i \in C^j \text{ and } j - \text{even} \\ -, \text{if } C_i \in C^j \text{ and } j - \text{even} \end{array} \right\}. \tag{19}$$

Note that the properties of cubes, described above, provide an opportunity to perform arithmetic operations on the logical surface, where the elementary set may include more than one value of the original function. At the same time, the operations of calculating the spectrum and autocorrelation functions are arithmetic functions with sets of given values in elementary form. This fact allows us to obtain a definite advantage in computational complexity by using operations on the cubes to decompose BF. For example, to calculate the number of unit values of BF simply adds the weights of all cubes in *C*, since this amount is characterized by the number of elementary sets contained in a cube. The sign of a cube shows whether the weight is added to or subtracted from the weight of the other cubes, as a result, removes duplicate sets, and the situation becomes as if each elementary set was presented once.

Coefficient of the autocorrelation function of BF with $B(\tau)$ can be calculated for any $\tau (0 \leq \tau \leq 2^m - 1)$ by adding (with sign) the weights of all cubes in $C(f(x) \cap C(f(x \oplus \tau))$. The proof of this fact is considered in detail [13]. From this analysis that is possible to calculate the autocorrelation function of BF without a truth table. To illustrate the above assertion, consider the following example.

Let BF f is presented in the following cover: $C^1 = \{C_1^0 \cap C_2^0, C_2^0 \cap C_3^0, C_2^0 \cap C_2^0\} = \{\varnothing, [1100], \varnothing\} = \{[1100]\}$; $C^2 = \varnothing$.

$$C(f) = \left[\begin{array}{cccc} - & 1 & 0 & - \\ 0 & 0 & - & - \\ 1 & - & - & 0 \\ 1 & 1 & 0 & 0 \end{array} \right]. \tag{20}$$

And the weight of all the cubes $w(C_1) = 4,\ w(C_2) = 4,\ w(C_3) = 4,\ w(C_1) = -1$.

Then $B(\tau)$ with $\tau = 0$ equals the total number of minterm, i.e. $B(0) = 4+4+4-1$ at $\tau = 1$ point $B(1)$ equal to the sum $w(C_i), C_i \in C(f(x)) \cap C(f(x \oplus\ 1))$:

$$C(f \oplus 1) = C(f) \oplus (0001) \left[\begin{array}{cccc} - & 1 & 0 & - \\ 0 & 0 & - & - \\ 1 & - & - & 1 \end{array} \right], \tag{21}$$

$$C(f) \cap C(f \oplus 1) = \left[\begin{array}{cccc} - & 1 & 0 & - \\ 0 & 0 & - & - \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{array} \right], \tag{22}$$

$$C(f(x) \cap f(x \oplus 1)) = \left[\begin{array}{cccc} - & 1 & 0 & - \\ 0 & 0 & - & - \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{array} \right] \left[\begin{array}{c} +4 \\ +4 \\ +1 \\ +1 \\ -1 \\ -1 \end{array} \right], \tag{23}$$

$$B(l) = 4 + 4 + 1 + 1 - 1 - 1 = 8. \tag{24}$$

It is obvious that finding the maximum coefficients of the autocorrelation function requires going through all the values of their coordinates. However, there is a way to limit the enumeration to only those coordinates, the values of autocorrelation coefficients which are maximal with the highest probability. That is, should examine only those τ, whose units are in the positions corresponding to the uncertain positions in the cubes of the original function with a maximum value of weights, since the intersection of *f(x)* and $f(x \oplus \tau)$ vector τ should make minimal changes to the original cover for the largest weights of cubes in the cover.

For example, for:

$$C = \left[\begin{array}{cccc} - & 1 & 0 & - \\ 0 & 0 & - & - \\ 1 & - & - & 0 \\ 1 & 1 & 0 & 1 \end{array} \right]. \tag{25}$$

All τ species $(\tau_3, 0, 0, \tau_0)$ have no effect on C_1. Similarly, $(0, 0, \tau_1, \tau_0)$ and $(0, \tau_2, \tau_1, 0)$ have no effect on

C_2 and C_3, respectively. The result should be considered only cubes with more weight, so that C_4 is not taken into account. In this case, two non-zero vector is most likely to correspond to the coordinates of the maximum values of autocorrelation function – is $(0, 0, \tau_1, \tau_0) \cap (\tau_3, 0, 0, \tau_0) = (0, 0, 0, 1)$, $(\tau_0 = 1)$ and $(0, 0, \tau_1, \tau_0) \cap (0, \tau_2, \tau_1, 0) = (0, 0, 1, 0)$, $(\tau_1 = 1)$. Consequently $\tau = 1$ and $\tau = 2$ - is the argument values the autocorrelation function, where it can have a maximum value.

This simple heuristic can be used to limit the number of coordinates, searched to find the maximum values of the autocorrelation function (Nagayama et al., 2005), even though a relatively small part of the original truth table. In practice, have BF, in which the size of cubes *C(f)* grows exponentially. In these cases, the procedure does not apply.

6. Numerical Results

In this section we provide simulation results on the benchmarks with wide-AND/OR architectures. The complexity of the logic blocks which are general PLAs with several inputs (from about 20 to 100) connected together by some kind of bus structure is high. The performance of the suggested logic synthesis is examined in terms of the cost function and the execution time. Table 3 refers to the benchmark function S420.

Tab. 3: FSM S420.

i	N	$\mathrm{L_{orig}}$	$\mathrm{L_{lin}}$	% improvement	(s)
1	24	66	49	25.8	7.8
2	24	49	32	34.7	7.4
3	24	32	15	53.1	7.3
4	26	32	32	reordering	8.1
18	69	151	151	reordering	23.9

The S420 represents a Finite State Machine (FSM) that has 19 input variables, 16 state variables and two output bits.

The FSM is defined by a set of 18 Boolean functions *d(i)* of 35 variables. In the table: N is the number of disjoint cubes in the representation of *f(i)* and L_{orig} and L_{lin} stand for the number of literals in SOP representation of the original function and the linearized function as computed by ESPRESSO [14].

Figure 6 shows the average execution time of the linearization procedures of [8] and the proposed method with $w = 3$ as a function of the number of imputs. The execution time was measured in Intel-Corei3, 2.5 Ghz, 2 GB RAM. For the statistics we used random PLA'S of four outputs and 50 products. The variance of the measurements was less than 3 %. It is clear from Fig. 1 that linearization over disjoint cubes is more efficient in terms of execution time than linearization based on Wiener-Khinchin theorem (W-K).

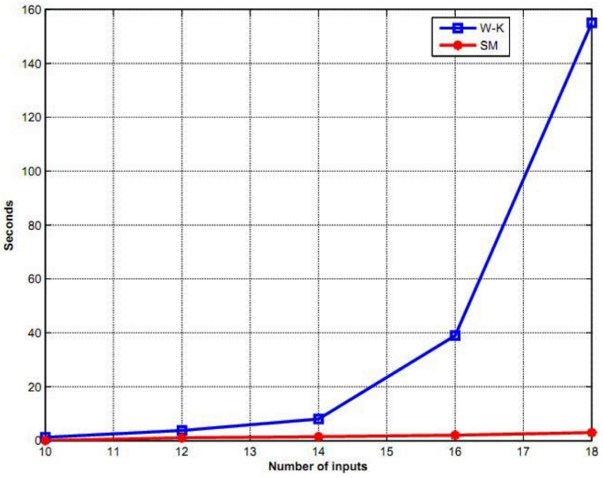

Fig. 6: The average execution time in seconds of Wiener-Khinchin theorem (labeled as W-K) [8], and the proposed spectral algorithm (labeled as SM) as a function of the number of inputs of randomly generated PLAs (4 outputs and 50 products).

Table 4 compares the average execution time of the linearization procedure of [9] and the suggested method (SM) (both with $w = 3$) for randomly generated PLAs having 10 to 40 inputs, four outputs 50 products.

Tab. 4: Average execution-time in seconds for 4-outputs and 50-products PLAs.

Inputs	[9]	SM
10	2.64	0.33
15	7.69	0.69
20	21.87	1.55
25	56.38	3.59
30	151.42	8.37
35	339.95	16.97
40	738.03	31.62

One example of short realization for simulation results of the linear part with the selection method of using Trachtenberg and Varma's algorithm.

Tab. 5: The input file.

6	4				
1	1	2	0	1	2
2	0	0	0	2	2
2	2	2	1	0	1
1	0	1	0	1	0

7. Conclusion and Future Extension

This paper proposed a heuristic algorithm for the first stage of decomposition of the BF, which uses treatment

Tab. 6: The output file.

Formula	**After the mini-mizing**	**Algorithm Trachtenberg--Varma**		**Substi-tutions**
		Time execution	Matrix	
1112012	1112012		$\sigma=$	
1200022	1200022	t=1 s.	1000100	$x_0 = x_4$
1222101	1222101		1000010	$x_1 = x_4$
1101010	1101010		1010000	$x_2 = x_4$
			1000001	$x_3 = x_0$
			1001000	$x_4 = x_3$
			1100000	$x_5 = x_5$

Tab. 7: The transformed function.

Transformed function	
The ternary matrix	Function
1011221	f_1
1020202	$x_0 \ \& \ x_3 \ \& \ x_4 \ \& \sim x_5 \mid$
1102122	$\sim x_1 \ \& \sim x_3 \ \& \sim x_5 \mid$
1010011	$x_2 \ \& \sim x_4 \ \& \ x_5 \mid$
	$x_0 \ \& \ x_1 \ \& \sim x_2 \ \& \sim x_3 \ \& \ x_4 \ \& \sim x_5$

of the Walsh spectrum of the original BF, linearization technique Karpovsky, as well as general properties of the autocorrelation functions. This algorithm involves finding the maximum autocorrelation coefficient of BF and to determine its address, i.e. serial number. Then, this number appears in the binary system, and location of units produced binary number determined by variables that are involved in its formation. Further search is carried out only in the variables of which was formed by the maximum rate. This greatly reduces the volume of the entire procedures. Then, by typing the required number of input variables, coefficients are deleted from the table for further search; the algorithm terminates the current step and starts a new one.

In future, the proposed technique is verified over standard benchmark functions and randomly generated Boolean functions for different number of variables and products. The experimental results will clearly demonstrate more efficiency.

References

[1] MURGAI, R., Y. NISHIZAKI, N. SHENOY, R. K. BRAYTON and A. SANGIOVANNI-VINCENTELLI. Logic synthesis for programmable gate arrays. In: *27th ACM/IEEE Design Automation Conference*. Orlando: IEEE, 1990, pp. 620–625. ISBN 0-89791-363-9. DOI: 10.1109/DAC.1990.114928.

[2] DAMARLA, T. R., M. KARPOVSKY, N. SHENOY, R. K. BRAYTON and A. SANGIOVANNI-VINCENTELLI. Fault detection in combinational networks by Reed-Muller transforms. *IEEE Transactions on Computers*. 1990, vol. 38, iss. 6, pp. 788–797. ISSN 0018-9340. DOI: 10.1109/12.24287.

[3] HARKING, B. Efficient algorithm for canonical Reed-Muller expansion of Boolean functions. *Computers and Digital Techniques*. 1990, vol. 137, iss. 5, pp. 366–370. ISSN 0143-7062.

[4] SMITH, M. and J. SEBASTIAN. *Application-specific integrated circuits*. Boston: Addison-Wesley, 1997. ISBN 02-015-0022-1.

[5] GAMAL, A., J. GREENE, J. REYNERI, E. ROGOYSKI, K. A. EL-AYAT and A. MOHSEN. An architecture for electrically configurable gate arrays. *IEEE Journal of Solid-State Circuits*. 1990, vol. 24, iss. 2, pp. 394–398. ISSN 0018-9200. DOI: 10.1109/4.18600.

[6] DARRINGER, J. A., W. H. JOYNER, C. L. BERMAN and L. TREVILLYAN. Logic Synthesis Through Local Transformations. *IBM Journal of Research and Development*. 1981, vol. 25, iss. 4, pp. 272–280. ISSN 0018-8646. DOI: 10.1147/rd.254.0272.

[7] KARPOVSKY, M. G. *Finite orthogonal series in the design of digital devices*. New York: Wiley, 1976. ISBN 04-701-5015-7.

[8] KARPOVSKY, M. G., R. S. STANKOVIC and J. T. ASTOLA. Reduction of sizes of decision diagrams by autocorrelation functions. *IEEE Transactions on Computers*. 2003, vol. 52, iss. 5, pp. 592–606. ISSN 0018-9340. DOI: 10.1109/TC.2003.1197126.

[9] KEREN O., I. LEVIN and R. S. STANKOVIC. Linearization of Functions Represented as a Set of Disjoint Cubes at the Autocorrelation Domain. In: *Proc. of the 7th International Workshop on Boolean Problems*. Freiberg: Freiberg University, 2006, pp. 137–144.

[10] FALKOWSKI, B. J. and S. KANNURAO. Calculation of sign Walsh spectra of Boolean functions from disjoint cubes. In: *The 2001 IEEE International Symposium on Circuits and Systems*. Sydney: IEEE, 2001, pp. 61–64. ISBN 0-7803-6685-9. DOI: dx.doi.org/10.1109/ISCAS.2001.921985.

[11] VARMA, D. and E. A. TRACHTENBERG. Design automation tools for efficient implementation of logic functions by decomposition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1989, vol. 8, iss. 8, pp. 901–916. ISSN 0278-0070. DOI: 10.1109/43.31549.

[12] STANKOVIC, R. S. and B. J. FALKOWSKI. Spectral Transforms Calculation through Decision Diagrams. *VLSI Design.* 2002, vol. 14, iss. 1, pp. 5–12. ISSN 1065-514x. DOI: 10.1080/10655140290009765.

[13] MAILHOT, F. and G. DI MICHELI. Algorithms for technology mapping based on binary decision diagrams and on Boolean operations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* 1993, vol. 12, iss. 5, pp. 599–620. ISSN 0278-0070. DOI: 10.1109/43.277607.

[14] BRAYTON, R. K. *Logic minimization algorithms for VLSI synthesis.* Boston: Kluwer Academic Publishers, 1984. ISBN 08-983-8164-9.

About Authors

Nhan Khanh Huu NGUYEN was born in Ho Chi Minh city, Vietnam. He received his B.Eng. degrees in Electrical and Electronic Engineering from University of Technical education Ho Chi Minh City in 1996, received his M.Sc. degrees in Nano materials and electronic devices from Ho Chi Minh City National University, Ho Chi Minh city, Vietnam in 2007, and Studied Ph.D. degree at Institute of researchs and experiments for electrical and electronic equipments, Moscow – Russia in 2012. Now, he is teaching at department of electrical and electronics engineering, Ton Duc Thang University, Ho Chi Minh city, Vietnam. His research interests include VLSI, MEMS and RF chip.