ISSN 1064-2307, Journal of Computer and Systems Sciences International, 2019, Vol. 58, No. 2, pp. 167–182. © Pleiades Publishing, Ltd., 2019.
Russian Text © The Author(s), 2019, published in Izvestiya Akademii Nauk, Teoriya i Sistemy Upravleniya, 2019, No. 2, pp. 14–29.

Decomposing a System of Boolean Functions into Subsystems of Connected Functions

P. N. Bibilo

United Institute of Informatics Problems, National Academy of Sciences of Belarus, Minsk, Belarus

e-mail: bibilo@newman.bas-net.by

Received July 10, 2018; revised November 22, 2018; accepted November 26, 2018

Abstract—This paper suggests some algorithms for decomposing a system of Boolean functions into subsystems of connected functions for representations such as truth tables (TTs), systems of disjunctive normal forms (DNFs), and binary decision diagrams (BDDs). The connectivity of functions consists of the presence of identical parts in the domains of functions from a given system. The algorithms are heuristic and can be used in computer-aided synthesis systems for real-dimension problems with several hundred functions, each having several tens of arguments. The experiments described below prove the efficiency of this decomposition approach in the logic optimization of a system of Boolean functions based on Shannon's decomposition with the possible use of subfunction inversions.

DOI: 10.1134/S1064230719020059

INTRODUCTION

Logical circuits are synthesized from library elements using the optimized two-level or multilevel representations of systems of Boolean functions. In the literature, two-level representations (AND-OR) are the ones involving disjunctive normal forms (DNFs), while multilevel representations are the ones involving different forms of functional decompositions [1–4]. The idea to employ the connectivity (commonality) of the domains of Boolean functions for multioutput combination circuits synthesis was proposed long ago [5, 6]. The connectivity of incompletely defined Boolean functions described by the values sets of Boolean variables was considered in the book [6]. As was demonstrated there, by extracting connected functions it is possible to extend them to the completely defined ones; moreover, the common parts in their domains correspond to subfunctions in the joint disjunctive and conjunctive decompositions.

Good connectivity of the functions has a significant influence on the appearance of identical structural units (conjunctions, algebraic expressions, subfunctions, etc.) in the optimized two-level or multilevel representations of functions, which are used to synthesize logical circuits in a certain technological basis. The higher the connectivity of the functions the more likely their representations include more identical subexpressions and the synthesized circuits have smaller complexity. As a matter of fact, the extraction of connected functions is a logic optimization tool for the multilevel representations of systems of functions. For the connected subsystems of functions, there are more efficient solutions of logic optimization problems, e.g., DNF optimization [2, 3], binary decision diagram (BDD) optimization [4, 7–9], and different types of decompositions, e.g., joint functional decompositions [10]. In the Russian-language literature, BDDs are also called binary solution diagrams, binary decision diagrams, binary decision diagrams, etc. The BDD representations of Boolean functions correspond to the multilevel representations based on Shannon's decomposition [4].

This paper formulates the concept of connectivity of Boolean functions (and also connectivity up to inversion) and develops extraction algorithms for the connected subsystems of functions with different representations of systems of completely defined functions. The algorithms are heuristic and intended for high-dimensional problems with several hundred functions, each having several tens of arguments, in the systems whose representations have to be optimized. As was shown by the experiments on the extraction of connected subsystems of functions, this procedure should be performed before BDD optimization, which is currently the most widespread logic optimization method for synthesizing logical circuits from library elements. Joint BDDs are preferable for some systems of functions, while separate ones are preferable for the others [11]. By extracting all the connected functions we can integrate within the same subsystem the functions that should be minimized based on joint BDDs.

1. MAIN DEFINITIONS AND REPRESENTATIONS OF SYSTEMS OF BOOLEAN FUNCTIONS

Consider a given system $\mathbf{f}(\mathbf{x}) = (f^1(\mathbf{x}),\ldots,f^m(\mathbf{x}))$ of Boolean functions. Denote by $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ the vector of all arguments x_1, x_2, ..., x_n. The characteristic set $M^1_{f^i}$ of a component function $f^i(\mathbf{x})$ of a system $\mathbf{f}(\mathbf{x})$ is the set of all collections in the Boolean space on which the function $f^i(\mathbf{x})$ takes value 1. In a similar way, denote by $M^0_{f^i}$ the set of all collections on which a function $f^i(\mathbf{x})$ takes the value 0.

Hereinafter, the *connectivity* of Boolean functions is understood in the sense that some subdomains of their domains coincide with each other. Denote by $|A|$ the cardinality of a set A. A system $\mathbf{f}(\mathbf{x})$ of functions is called S_p*-connected* if $p^n_{\max} \ge p$, where

$$p^n_{\max} = \left|\bigcap_{i=1}^{m} M^1_{f^i}\right| + \left|\bigcap_{i=1}^{m} M^0_{f^i}\right|.$$

In other words, there are p collections in the n-dimensional Boolean space on which all component functions $f^i(\mathbf{x})$ of the system simultaneously take value 1 or 0. Obviously, for a given system of Boolean functions of n variables, the parameter $p^n_{\max}$ specifies the maximum value of p. If a system of functions is S_p*-connected*, then it is also S_q*-connected* for all q such that $0 < q < p$. The value $p^n_{\max}$ will be called *the weight of connectivity* for a system of functions. For a single Boolean function of n variables, the weight of connectivity is the sum of the cardinalities of the sets $M^1_{f^i}$ and $M^0_{f^i}$; to put it differently, this is the number 2^n of all elements of the Boolean space constructed over the variables of the Boolean vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. The elements of this space are the n-component collections (vectors) $\mathbf{x}^*$ of zeros and unities.

For a system $\mathbf{f}(\mathbf{x})$ of functions, *the measure (share) of connectivity* $\rho^n(f^1, \ldots, f^m)$ is the ratio

$$\rho^n(f^1,\ldots,f^m) = \frac{p^n_{\max}}{2^n}.$$

As easily seen, the measure of connectivity is bounded:

$$0 \le \rho^n(f^1, \ldots, f^m) \le 1.$$

If a system of functions contains a single component function f^1 (or all component functions coincide with each other), then $\rho^n(f^1) = 1$. If a system of functions consists, e.g., of a pair $\{f^1, \bar{f}^1\}$ of mutually inverse functions, then $\rho^n(f^1, \bar{f}^1) = 0$. Clearly, this is not the only example of the systems of functions with the zero measure of connectivity; such systems will be called *unconnected*. Some examples will be given below.

Introduce an m-component Boolean vector $\alpha = (\alpha^1, \alpha^2, \ldots, \alpha^m)$, hereinafter called the polarization vector for the component functions $f^i(\mathbf{x})$. Let $\alpha^i = 1$ if a function f^i is considered, and let $\alpha^i = 0$ if the inversion $\bar{f}^i$ of f^i is considered. A system $\mathbf{f}(\mathbf{x})$ of functions is called S^α_p*-connected* if there exists at least one polarization vector $\alpha = (\alpha^1, \alpha^2, \ldots, \alpha^m)$ for which the corresponding system of functions is S_p*-connected.* The concept of S^α_p*-connectivity* matches the standard connectivity of functions up to the inversion of the component functions of a system.

The systems of Boolean functions may have different representations. In what follows, the matrix representations—truth tables (see Table 1) and systems of DNFs (see Table 2)—as well as the representations with algebraic formulas defining Shannon's decompositions (BDD representations) will be used. For the system of functions described by Table 1, the measure of connectivity is $\rho^4(f^1, f^2, f^3, f^4) = 0$, because the values of all four component functions are not simultaneously equal to 1 or 0 on any collection of values

Table 1. System of completely defined Boolean functions

x_1 x_2 x_3 x_4	f^1 f^2 f^3 f^4
0 0 0 0	0 0 1 1
0 0 0 1	1 1 0 0
0 0 1 0	1 0 0 0
0 0 1 1	1 1 0 0
0 1 0 0	0 0 1 1
0 1 0 1	1 1 0 0
0 1 1 0	0 1 1 1
0 1 1 1	1 1 1 0
1 0 0 0	0 0 1 1
1 0 0 1	1 0 0 0
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 0
1 1 0 0	0 0 1 1
1 1 0 1	1 0 1 1
1 1 1 0	0 0 0 1
1 1 1 1	1 0 1 0

Table 2. System of DNFs of Boolean functions

T^x	B^f
x_1 x_2 x_3 x_4	f^1 f^2 f^3 f^4
– 0 1 –	1 0 0 0
0 1 1 –	0 1 1 0
– – – 1	1 0 0 0
1 0 1 –	0 1 0 0
1 1 0 –	0 0 1 1
– 1 – 0	0 0 0 1
– 1 1 1	0 0 1 0
0 – – 1	0 1 0 0
– – 0 0	0 0 1 1

of the variables vector $\mathbf{x} = (x_1, x_2, x_3, x_4)$. As will be demonstrated below, this system actually contains subsystems of connected functions.

In the matrix representation, the system of DNFs (Table 2) is defined by a pair of matrices as follows: the rows of the ternary matrix T^x represent the elementary conjunctions (the ternary vectors–intervals of the Boolean space [3]), while the unit values of the elements in the Boolean matrix B^f indicate the entries of the corresponding conjunctions into the DNFs of the functions:

$$D^1 = \bar{x}_2 x_3 \vee x_4;$$

$$D^2 = \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_4;$$

$$D^3 = \bar{x}_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_2 x_3 x_4 \vee \bar{x}_3 \bar{x}_4;$$

$$D^4 = \bar{x}_1 x_2 x_3 \vee x_2 \bar{x}_4 \vee \bar{x}_3 \bar{x}_4.$$

Note that this system of DNFs describes the same system of Boolean functions as Table 1. If an elementary conjunction does not contain q literals, then its ternary vector has q unknown elements ("–"). By definition, a ternary vector with q unknown elements has *weight* 2^q. In other words, the weight of a ternary vector is the number of all binary vectors obtained by replacing its unknown elements with elements 0 or 1.

The systems of DNFs are often minimized [2, 3], in order to reduce the number of elementary conjunctions and (or) the number of literals in conjunctions. In the resulting minimal systems of DNFs, elementary conjunctions (ternary vectors) have different relations with each other. In what follows, the relation of *orthogonality* will be used. If all ternary vectors of a matrix T^x are pairwise orthogonal, then the system of DNFs is called *orthogonalized* [12]. Ternary vectors $a = (x_1^a, \ldots, x_n^a)$ and $b = (x_1^b, \ldots, x_n^b)$ are orthogonal if there exists at least one component $i \in \{1, \ldots, n\}$ such that x_i^a and x_i^b are well-defined and not equal to each other. For example, the ternary vectors $a = (0 - 10)$ and $b = (-100)$ are orthogonal since the orthogonality condition holds for $i = 3$: $x_3^a = 1$ and $x_3^b = 0$. If ternary vectors are orthogonal, then the conjunction (logical product) of the corresponding elementary conjunctions is 0: $a \,\&\, b = ab = \bar{x}_1 x_3 \bar{x}_4 \& x_2 \bar{x}_3 \bar{x}_4 = 0$.

The BDDs are the graph representations of Boolean functions. As was shown in [4], such representations mean that each component function $f^i(\mathbf{x})$ is defined as a pair of orthogonalized DNFs. One such DNF determines the unit-value domain $M^1_{f^i}$ of a component function, while the other determines the zero-value domain $M^0_{f^i}$. For a system of functions, the BDD representation is compact because defining a graph is not as cumbersome as listing all paths from root vertices to the leaf vertices 0 and 1. For graph simplification the leaf vertices are often duplicated. Each path from a root vertex of a BDD labeled by a function f^i to the leaf vertex 1 is identified with the elementary conjunction containing the arcs-literals x_i and $\bar{x}_i$ on this path. In addition, an arc labeled by 0 is identified with the negative literal $\bar{x}_i$; an arc labeled by 1, with the positive literal x_i; and all paths between the above vertices, with the disjunction of the resulting elementary conjunctions that forms the orthogonalized DNF $D^1_{f^i}$. The paths from a root vertex f^i to the leaf vertex 0 define the orthogonalized DNF $D^0_{f^i}$.

The BDD in Fig. 1 represents the same system of Boolean functions as Tables 1 and 2. Next, Tables 3–6 give the orthogonalized DNFs obtained by the graph's BDD representation (Fig. 1). Each functional vertex of the BDD is identified with Shannon's decomposition formula. For example, the functional vertex f^1 is identified with the formula $f^1 = \bar{x}_2 s_9 \vee x_2 x_4$; the functional vertex s_{13}, with the formula $s_{13} = \bar{x}_3 \bar{x}_4 \vee x_3$; etc. Therefore, functional vertices are often not indicated on graphs, while labels 0 and 1 on the arcs are replaced by different lines. For example, if an arc is labeled 0, then the corresponding line is a dashed line [8, 9]. The entire BDD graph is identified with the 13 formulas (1.1)–(1.4), which incorporate 29 logical disjunctions and conjunctions:

$$f^1 = \bar{x}_2 s_9 \vee x_2 x_4; \quad f^2 = \bar{x}_2 s_3 \vee x_2 s_4; \quad f^3 = \bar{x}_2 s_{12} \vee x_2 s_6; \tag{1.1}$$

$$f^4 = \bar{x}_2 s_{12} \vee x_2 s_8; \quad s_9 = \bar{x}_3 x_4 \vee x_3; \quad s_3 = \bar{x}_1 x_4 \vee x_1 x_3; \quad s_4 = \bar{x}_1 s_9; \tag{1.2}$$

$$s_{12} = \bar{x}_3 \bar{x}_4; \quad s_6 = \bar{x}_1 s_{13} \vee x_1 s_{14}; \quad s_8 = \bar{x}_1 \bar{x}_4 \vee x_1 s_{16}; \quad s_{13} = \bar{x}_3 \bar{x}_4 \vee x_3; \tag{1.3}$$

$$s_{14} = \bar{x}_3 \vee x_3 x_4; \quad s_{14} = \bar{x}_3 \vee x_3 x_4; \quad s_{16} = \bar{x}_3 \vee x_3 \bar{x}_4. \tag{1.4}$$

2. CONNECTIVITY CHECK FOR A SYSTEM OF BOOLEAN FUNCTIONS

Problem 1. Given a system $\mathbf{f}(\mathbf{x}) = (f^1(\mathbf{x}), \ldots, f^m(\mathbf{x}))$ of Boolean functions and a parameter p, where $0 < p < 2^n$, check the S_p-connectivity of $\mathbf{f}(\mathbf{x})$.

Problem 2. Given a system $\mathbf{f}(\mathbf{x}) = (f^1(\mathbf{x}), \ldots, f^m(\mathbf{x}))$ of Boolean functions, a polarization vector $\alpha = (\alpha^1, \alpha^2, \ldots, \alpha^m)$, and a parameter p, where $0 < p < 2^n$, check the S_p^α-connectivity of $\mathbf{f}(\mathbf{x})$.

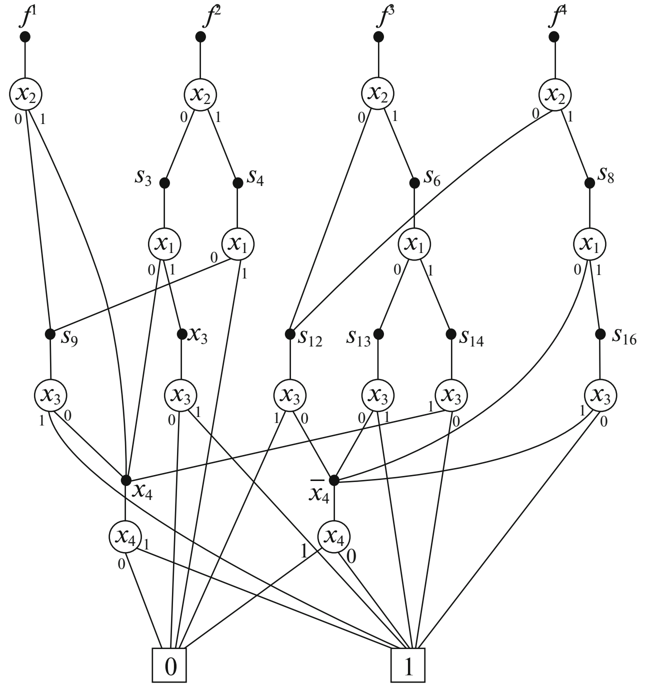

Fig. 1. BDD representation for system of Boolean functions.

If the functions of this system are defined by 2^n-bit Boolean vectors of their values on all collections in the Boolean space (i.e., by the column vectors of their values in truth tables), the connectivity of the system can be successfully checked using the technique of Boolean calculations suggested in [13]. For the subsystem of functions defined by Table 7, the S_p-connectivity check includes the following operations: (1) the module 2 addition (XOR) of the column vectors describing the values of these functions and (2) the calculation of the number of unit values in the resulting vector. The S_p^{α}-connectivity check requires the preliminary inversion of those column vectors of system functions that correspond to the zero components in the polarization vector.

Note that in the truth table the Boolean function is identified with the full DNF (FDNF), which contains the full elementary conjunctions that are mutually orthogonal.

The next subsections will illustrate the solution of Problems 1 and 2 by an example of the subsystems of a system of functions defined by the truth table (Table 1), the DNF (Table 2), and the BDD representation (Fig. 1 and formulas (1.1)–(1.4)).

2.1. Connectivity Check for Systems of DNFs of Functions

If functions are defined by a system of DNFs, Problem 1 is restated as follows.

Table 3. Function f^1 defined by orthogonalized DNFs

DNF	x_1 x_2 x_3 x_4	f^1
$D^0_{f^1}$	– 0 0 0	0
	– 1 – 0	0
$D^1_{f^1}$	– 1 – 1	1
	– 0 1 –	1
	– 0 0 1	1

Table 4. Function f^2 defined by orthogonalized DNFs

DNF	x_1 x_2 x_3 x_4	f^2
$D^0_{f^2}$	1 0 0 –	0
	0 0 – 0	0
	0 1 0 0	0
	1 1 – –	0
$D^1_{f^2}$	1 0 1 –	1
	0 1 1 –	1
	0 0 – 1	1
	0 1 0 1	1

Table 5. Function f^3 defined by orthogonalized DNFs

DNF	x_1 x_2 x_3 x_4	f^3
$D^0_{f^3}$	– 0 1 –	0
	– 0 0 1	0
	0 1 0 1	0
	1 1 1 0	0
$D^1_{f^3}$	0 1 1 –	1
	– 0 0 0	1
	1 1 0 –	1
	0 1 0 0	1
	1 1 1 1	1

Table 6. Function f^4 defined by orthogonalized DNFs

DNF	x_1 x_2 x_3 x_4	f^4
$D^0_{f^4}$	– 0 1 –	0
	0 1 – 1	0
	1 1 1 1	0
	– 0 0 1	0
$D^1_{f^4}$	1 1 0 –	1
	0 1 – 0	1
	1 1 1 0	1
	– 0 0 0	1

Problem 1a. Given a system of DNFs $\mathbf{D}(\mathbf{x}) = (D^1(\mathbf{x}), \dots , D^m(\mathbf{x}))$ representing a system of Boolean functions $\mathbf{f}(\mathbf{x}) = (f^1(\mathbf{x}), \dots, f^m(\mathbf{x}))$ of n variables, check the S_p-connectivity of $\mathbf{f}(\mathbf{x})$.

With the DNF representation of a system of functions on the general set of elementary conjunctions, the S_p-connectivity check for this system should be preceded by its orthogonalization and general orthogonalization. The corresponding algorithms were described in [12] (orthogonalization) and in [14] (general orthogonalization).

Now, perform the orthogonalization of the system of DNFs (Table 2) for obtaining its orthogonalized form (Table 8), in which all pairs of ternary vectors are orthogonal. Orthogonalized DNFs are more compact than truth tables; in the current example, Table 8 contains 12 rows against 16 rows of Table 1. However, the minimal DNFs (Table 2) are even more compact than the orthogonalized systems of DNFs:

Table 7. Subsystems of Boolean functions

x_1 x_2 x_3 x_4	f^1 f^2	f^1 $\overline{f}^2$
0 0 0 0	**0 0**	0 1
0 0 0 1	**1 1**	1 0
0 0 1 0	1 0	**1 1**
0 0 1 1	**1 1**	1 0
0 1 0 0	**0 0**	0 1
0 1 0 1	**1 1**	1 0
0 1 1 0	0 1	**0 0**
0 1 1 1	**1 1**	1 0
1 0 0 0	**0 0**	0 1
1 0 0 1	1 0	**1 1**
1 0 1 0	**1 1**	1 0
1 0 1 1	**1 1**	1 0
1 1 0 0	**0 0**	0 1
1 1 0 1	1 0	**1 1**
1 1 1 0	**0 0**	0 1
1 1 1 1	1 0	**1 1**
Weight of connectivity	11	5
Measure of connectivity	11/16	5/16

Table 8. Orthogonalized system of DNFs

Weight of ternary vector	x_1 x_2 x_3 x_4	f^1	f^2	f^3	f^4
2	0 – 0 0	0	0	1	1
2	1 – 0 0	0	0	1	1
2	0 0 – 1	1	1	0	0
1	0 1 0 1	1	1	0	0
2	1 0 1 –	1	1	0	0
1	0 0 1 0	1	0	0	0
1	1 0 0 1	1	0	0	0
1	0 1 1 0	0	1	1	1
1	0 1 1 1	1	1	1	0
1	1 1 0 1	1	0	1	1
1	1 1 1 0	0	0	0	1
1	1 1 1 1	1	0	1	0

Table 2 contains 9 rows only. For each ternary vector (elementary conjunction), the right-hand part of Table 8, i.e., the vector of entries of all conjunctions in the DNF, is 0. Hence, $p^n_{\max} = 0$ and the system of functions is unconnected. For the orthogonalized form (like for truth tables or the systems of FDNFs), the weight of $p^n_{\max}$-connectivity can be easily found for the subsystems of functions directly form (Table 9). In the current example, there exist no collections in the domain of functions on which all these functions are 0; nonetheless, the calculation of this domain is important for the S^{α}_p-connectivity check. For extracting the S^{α}_p*-connected* subsystems, we need the domains in which all component functions are 0. This domain can be large, and its explicit representation (as a DNF) can be difficult due to the complexity of the corresponding DNF. However, the orthogonalized forms possibly include many elementary conjunc-

Table 9. Subsystems of orthogonalized system of DNFs

Weight of ternary vector	$x_1\ x_2\ x_3\ x_4$	$f^1\ f^2$	$f^1\ \overline{f}^2$
2	0 – 0 0	**0 0**	0 1
2	1 – 0 0	**0 0**	0 1
2	0 0 – 1	**1 1**	1 0
1	0 1 0 1	**1 1**	1 0
2	1 0 1 –	**1 1**	1 0
1	0 0 1 0	1 0	**1 1**
1	1 0 0 1	1 0	**1 1**
1	0 1 1 0	0 1	**0 0**
1	0 1 1 1	**1 1**	1 0
1	1 1 0 1	1 0	**1 1**
1	1 1 1 0	**0 0**	0 1
1	1 1 1 1	1 0	**1 1**
Weight of connectivity		11	5
Measure of connectivity		11/16	5/16

Table 10. Minimal disjunctive basis for system of DNFs

$S(\Pi^i)$	Π^i	$x_1\ x_2\ x_3\ x_4$
4	Π^1	– – 00
5	Π^2	0 0 – 1
		0 1 0 1
		1 0 1 –
2	Π^3	0 0 1 0
		1 0 0 1
1	Π^4	0 1 1 0
1	Π^5	0 1 1 1
1	Π^6	1 1 0 1
1	Π^7	1 1 1 0
1	Π^8	1 1 1 1

tions, which makes them cumbersome too. For reducing the number of elementary conjunctions, a reasonable approach is to define a system of functions on the set of pairwise orthogonal DNFs Π^i (*disjunctive basis*). In [14] this representation was called the general orthogonalized form; among other results, the cited book solved the general orthogonalization problem for a system of DNFs using the multiplication and inversion of the DNFs associated with the unit and zero value domains of its component functions.

For the system of DNFs in Table 2, the minimal disjunctive basis (i.e., the one with a minimal number of DNFs) includes the eight DNFs Π^i; see Table 10. The matrix representation of the general orthogonalized form

$$f^1 = \Pi^2 \vee \Pi^3 \vee \Pi^6 \vee \Pi^7 \vee \Pi^8;$$

$$f^2 = \Pi^2 \vee \Pi^4 \vee \Pi^5;$$

$$f^3 = \Pi^1 \vee \Pi^4 \vee \Pi^5 \vee \Pi^7 \vee \Pi^8;$$

Table 11. General orthogonalized representation of system of DNFs

$S(\Pi^i)$	Π^i	$f^1\ f^2\ f^3\ f^4$
4	Π^1	0 0 1 1
5	Π^2	1 1 0 0
2	Π^3	1 0 0 0
1	Π^4	0 1 1 1
1	Π^5	1 1 1 0
1	Π^6	1 0 1 1
1	Π^7	0 0 0 1
1	Π^8	1 0 1 0

Table 12. Connectivity of subsystems for system of DNFs with general orthogonalized representation

$S(\Pi^i)$	Π^i	$f^1\ f^2$	$f^1\ \overline{f}^2$
4	Π^1	**0 0**	0 1
5	Π^2	**1 1**	1 0
2	Π^3	1 0	**1 1**
1	Π^4	0 1	**0 0**
1	Π^5	**1 1**	1 0
1	Π^6	1 0	**1 1**
1	Π^7	**0 0**	0 1
1	Π^8	1 0	**1 1**
Weight of connectivity		11	5
Measure of connectivity		11/16	5/16

$$f^4 = \Pi^1 \vee \Pi^4 \vee \Pi^6 \vee \Pi^7$$

of the system of DNFs (Table 2) is given in Table 11. In addition, Table 10 shows the weights of DNFs defined as follows. The *weight* $S(\Pi^i)$ of a DNF Π^i is the number of full elementary conjunctions in the FDNF equal to the DNF Π^i. For example, the weight of the DNF Π^1 is 4, since the DNF $\Pi^1 = \bar{x}_3\bar{x}_4$ contains one elementary conjunction representing four full elementary conjunctions:

$$\Pi^1 = \bar{x}_3\bar{x}_4 = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_3\bar{x}_4 \vee x_1\bar{x}_2\bar{x}_3\bar{x}_4 \vee x_1x_2\bar{x}_3\bar{x}_4.$$

The weights of other DNFs from Table 10 can be calculated by analogy. If the general orthogonalized form is successfully obtained for a system of functions, then the S_p-connectivity check for its subsystems causes no difficulty, being similar to the connectivity check for the orthogonalized form. For example, Table 12 defines two subsystems, and their weight of connectivity is easily calculated. Recall that, in this example, all component functions do not take value 0 simultaneously on any domain in the Boolean space of their input variables. Thus, the inversion of a component function can be constructed using the column of its values on the intervals of the Boolean space of the DNF Π^i.

Table 13. Connectivity of subsystem $\{f^1, f^2\}$ obtained by BDD representation of functions f^1, f^2 (Fig. 1)

DNF	$x_1\ x_2\ x_3\ x_4$	$f^1\ f^2$
$D^0_{f^1}\ D^0_{f^2}$	1 0 0 0	**0 0**
	0 0 0 0	**0 0**
	0 1 0 0	**0 0**
	1 1 – 0	**0 0**
$D^0_{f^1}\ D^1_{f^2}$	0 1 1 0	0 1
$D^1_{f^1}\ D^0_{f^2}$	1 1 – 1	1 0
	0 0 1 0	1 0
	1 0 0 1	1 0
$D^1_{f^1}\ D^1_{f^2}$	0 1 – 1	**1 1**
	1 0 1 –	**1 1**
	0 0 1 1	**1 1**
	0 0 0 1	**1 1**
Weight of connectivity		11
Measure of connectivity		11/16

Table 14. Connectivity of subsystem $\{f^1, \overline{f}^2\}$ obtained by BDD representation of functions f^1, $\overline{f}^2$ (see Fig. 1)

DNF	$x_1\ x_2\ x_3\ x_4$	$f^1\ \overline{f}^2$
$D^0_{f^1}\ D^1_{\overline{f}^2}$	1 0 0 0	0 1
	0 0 0 0	0 1
	0 1 0 0	0 1
	1 1 – 0	0 1
$D^0_{f^1}\ D^0_{\overline{f}^2}$	0 1 1 0	**0 0**
$D^1_{f^1}\ D^1_{\overline{f}^2}$	1 1 – 1	**1 1**
	0 0 1 0	**1 1**
	1 0 0 1	**1 1**
$D^1_{f^1}\ D^0_{\overline{f}^2}$	0 1 – 1	1 0
	1 0 1 –	1 0
	0 0 1 1	1 0
	0 0 0 1	1 0
Weight of connectivity		5
Measure of connectivity		5/16

2.2. Connectivity Check for Systems of Functions with BDD Representations

The logical operations over Boolean functions with BDD representations were described in [4]; they involve the universal operation for obtaining the orthogonalized DNF of a system of two functions using the orthogonalized DNFs of the functions participating in an operation. This approach can be adopted for constructing the orthogonalized form of a subsystem of functions. In addition, it is easy to take into account the polarization vector and inversion of a function in subsystem design.

The orthogonalized form of a subsystem is obtained from the orthogonalized forms of its component functions through the multiplication of $D^{\alpha}_{f^i}$ by $D^{\alpha}_{f^j}$, where $\alpha \in \{0,1\}$.

Calculate the orthogonalized form of the subsystem $\{f^1, f^2\}$ by multiplying the corresponding DNFs. The orthogonalized form is described by Table 13. For the subsystem $\{f^1, \overline{f^2}\}$, we should consider

$D^0_{f^2} = D^1_{\overline{f}^2}$ and $D^1_{f^2} = D^0_{\overline{f}^2}$; in this case, the representation of this subsystem with four pairwise orthogonal DNFs is easily derived (see Table 14).

The data structures and efficient algorithms of logical operations over them [15] can be used for the software implementation of the BDD representations of different systems of Boolean functions. Note that the operation ◊ (the merging of two BDDs, see [15, p. 259]) is the basic operation for the connectivity check of a subsystem in the BDD representation within a system of functions using their separate BDD representations.

3. EXTRACTION OF CONNECTED SUBSYSTEMS: PROBLEM STATEMENT AND ALGORITHM

Problem 3. Given a parameter p $(0 < p < 2^n)$ and a system $\mathbf{f}(\mathbf{x}) = (f^1(\mathbf{x}), \ldots, f^m(\mathbf{x}))$ of Boolean functions of n variables, decompose $\mathbf{f}(\mathbf{x})$ into the minimal number of S_p-connected subsystems.

The exact solution of Problem 3 can be reduced as follows:

—extracting all maximal S_p-connected subsystems of functions in terms of cardinality;

—constructing the Boolean entry matrix R in which each row describes the entry of a component function into the corresponding subsystem;

—obtaining the smallest row coverage of the matrix R (the corresponding methods were considered in detail in [3]);

—ensuring the mutual non-intersection of the subsystems in the shortest coverage by eliminating some functions from the appropriate subsystems.

The exact solution of Problem 3 can be found for the systems of Boolean functions consisting of a finite number of component functions of at most 20 variables. Consider some algorithms for solving Problem 3 with different representations of Boolean functions that can be implemented in logic optimization systems (e.g., *FLC* [16]).

The heuristic algorithm in the solution of Problem 3 consists in the sequential (iterative) formation of the next subsystem of S_p-connected functions using the current (residual) system of functions. At the first iteration, the current system of functions coincides with the functions of the original system.

Each iteration includes three steps as follows.

Step 1. Study C_m^2 unordered pairs $\{f^i, f^j\}$, $i \neq j$, $m > 1$, of functions in the current system and *find a pair L of functions that are S_q-connected with a maximal number of the parameter q*, where $q \geq p$; if there are several such pairs, then choose any of them (the heuristic EVR1). Otherwise (the pair L does not exist) the solution of Problem 3 is the subsystems each of which includes only one function from the current system.

Step 2. From the functions of the pair L (see Step 1) construct a subsystem of two connected functions by eliminating this pair from the current system, and sequentially supplement this subsystem with the functions obtained using the heuristic EVR2 as follows.

The heuristic EVR2: "from the set of all functions of the current system choose the one with the maximal value of the parameter q for the resulting S_q-connected subsystem. If there are several such functions, choose any of them".

Step 3. If none of the functions of the current system can be included in the resulting S_q-connected subsystem ($q \geq p$) without violating the constraint $q \geq p$, then stop subsystem formation, choose all functions beyond the current system as the new current system and go back to Step 1 (form the next subsystem).

Step 4. Terminate subsystem formation when all functions of the current system are included in the resulting subsystems (i.e., when the current system becomes empty). End of the algorithm.

The heuristic algorithm reliably gives the exact solution of Problem 3 in two cases as follows: (1) all functions of the original system form a subsystem of connected functions; (2) each of the subsystems of connected functions contains only one function of the original system.

Problem 4. Given a parameter p $(0 < p < 2^n)$ and a system $\mathbf{f}(\mathbf{x}) = (f^1(\mathbf{x}), \ldots, f^m(\mathbf{x}))$ of Boolean functions of n variables, decompose $\mathbf{f}(\mathbf{x})$ into the minimal number of S_p^α-connected subsystems.

Table 15. Weights of connectivity for subsystems consisting of pairs of functions

Component functions	f^2	f^3	f^4
f^1	**11**	4	1
f^2		5	4
f^3			**13**

Obviously, Problem 3 is a special case of Problem 4. The solution of Problem 4 should be found in the class of all connected subsystems of functions up to inversion.

The exact solution of Problem 4 is similar to that of Problem 3 and requires knowledge of all maximal connected subsystems (taking into account the inversions of functions). Each time this leads to the exhaustive search of 2^{m_i} polarization vectors $\alpha = (\alpha^1, \alpha^2, \ldots, \alpha^{m_i})$ for each resulting subsystem composed of m_i functions, which is computationally intensive for real problems with tens or hundreds of functions in a system.

The heuristic algorithm for solving Problem 4 has the following distinctions from its counterpart for Problem 3.

The current system of functions is formed by some functions of the original system and also by their inversions.

At Step 1, each of the functions f^i is considered in its direct (f^i) and also inverse ($\overline{f}^i$) forms, with the exhaustive search of all $C_{2m}^2 - m$, $m > 1$, possible unordered pairs of functions in the current system (the pairs $\{f^i, f^i\}$, $\{\overline{f}^i, \overline{f}^i\}$, and $\{f^i, \overline{f}^i\}$ are eliminated from consideration); the goal is to find a pair of S_q^{α}-connected functions with the maximal value $q_{\max}$ of the parameter q, $q \geq p$. If there are several such pairs, choose any of them (the heuristic EVR3). Form a polarization vector $\alpha = (\alpha^1, \alpha^2)$ for the subsystem of two functions.

At Step 2, the next function from the current system is added in the resulting subsystem using the heuristic EVR4 as follows.

The heuristic EVR4: "from the set of all functions (and their inversions) of the current system choose the one (or its inversion) with the maximal value of the parameter q for the resulting S_p^{α}-connected subsystem. If there are several such functions, choose any of them".

At each iteration, the heuristics EVR1 and EVR2 allow us to find a subset of connected functions of the maximum possible cardinality, thereby reducing the number of subsystems that are S_p-connected (Problem 3). The same property applies to the heuristics EVR3 and EVR4 used for solving Problem 4.

4. EXTRACTION OF S_p-CONNECTED SUBSYSTEMS OF FUNCTIONS: AN EXAMPLE

It is required to decompose the system of functions (Table 1) into the minimal number of S_{11}-connected subsystems of functions ($p = 11$).

Iteration 1. Form the first subsystem of connected functions.

Step 1. The weights of connectivity for different pairs of functions in this system are given in Table 15. The pair of functions with the maximal parameter value $q = 13$ is $\{f^3, f^4\}$.

Step 2. Form the first S_{13}-connected subsystem $\{f^3, f^4\}$, sequentially supplementing it with other functions. As can be easily seen, the functions of the subsystem $\{f^1, f^3, f^4\}$ take value 1 on the collection 1101 only ($q = 1$), and hence this subsystem is S_1-connected. Add the function f^2 in the subsystem $\{f^3, f^4\}$ and check for which value of the parameter q it satisfies the S_q-connectivity. There are only three collections on which all functions of the subsystem $\{f^2, f^3, f^4\}$ take value 1 or 0; thus, the subsystem $\{f^1,$

Table 16. Weights of connectivity for subsystems consisting of pairs of functions (with function inversions)

Component functions	f^2	$\overline{f}^2$	f^3	$\overline{f}^3$	f^4	$\overline{f}^4$
f^1	11	5	4	12	1	**15**
$\overline{f}^1$	5	11	12	4	15	1
f^2			5	11	4	12
$\overline{f}^2$			**11**	5	12	4
f^3					13	3
$\overline{f}^3$					3	13

f^2, f^4} is S_3-connected and, obviously, not S_{11}-connected. This completes the formation of the first subsystem. Further the subsystem {f^1, f^2} is considered as the current subsystem of functions.

Iteration 2. Form the second subsystem of connected functions.

Step 1. The subsystem {f^1, f^2} is S_{11}-connected. End of the algorithm.

Now, perform BDD optimization separately for each of the resulting subsystems. The joint BDD for {f^1, f^2} (see Fig. 2) is described by the five equations (4.1) and (4.2), which contain 12 logical operations of disjunction and conjunction:

$$f^1 = \overline{x}_2 s_5 \vee x_2 x_4; \quad f^2 = \overline{x}_2 s_3 \vee x_2 s_4; \quad s_5 = \overline{x}_3 x_4 \vee x_3; \tag{4.1}$$

$$s_3 = \overline{x}_1 x_4 \vee x_1 x_3; \quad s_4 = \overline{x}_1 s_5. \tag{4.2}$$

The joint BDD for {f^3, f^4} (see Fig. 3) is described by the seven equations (4.3) and (4.4), which contain 15 logical operations of disjunction and conjunction:

$$f^3 = \overline{x}_2 s_1 \vee x_2 s_2; \quad f^4 = \overline{x}_2 s_1 \vee x_2 s_4; \quad s_1 = \overline{x}_3 \overline{x}_4; \quad s_2 = \overline{x}_3 s_6 \vee x_3 s_7; \tag{4.3}$$

$$s_4 = \overline{x}_3 s_6 \vee x_3 \overline{x}_4; \quad s_6 = \overline{x}_1 \overline{x}_4 \vee x_1; \quad s_7 = \overline{x}_1 \vee x_1 x_4. \tag{4.4}$$

5. EXTRACTION OF S_p^α-CONNECTED SUBSYSTEMS OF FUNCTIONS: AN EXAMPLE

It is required to decompose the system of functions (Table 1) into the minimal number of S_{11}^α-connected subsystems of functions ($p = 11$).

Iteration 1. Form the first subsystem of connected functions.

Step 1. The weights of connectivity for different pairs of functions in this system are given in Table 16. The pair of functions with the maximal parameter value $q = 15$ is {f^1, $\overline{f}^4$} and the polarization vector is $\alpha = (\alpha^1, \alpha^4) = (1, 0)$. Adding any function from the set {f^2, $\overline{f}^2$, f^3, $\overline{f}^3$} in the subsystem {f^1, $\overline{f}^4$} will not make the supplemented subsystem S_{11}^α-connected as desired.

Iteration 2. Form the second subsystem of connected functions.

Step 1. The second subsystem is {$\overline{f}^2$, f^3}, which satisfies the S_{11}^α-connectivity with the polarization vector $\alpha = (\alpha^2, \alpha^3) = (0, 1)$. End of the algorithm.

As a result of the BDD optimization of the original system (without extracting the subsystems of connected functions) and its synthesis from library CMOS elements, the total number of transistors in the logic circuit was 68. The decomposition of the original system into two subsystems with the subsequent BDD optimization of these subsystems [4] (see Figs. 2, 3) reduced the total number of transistors in the logic circuit to 66. Note that the decomposition into the subsystems {f^1,$\overline{f}^4$} and {$\overline{f}^2$,f^3} of connected functions with their BDDI optimization [17] yielded the best solution—a logic circuit with 62 transistors.

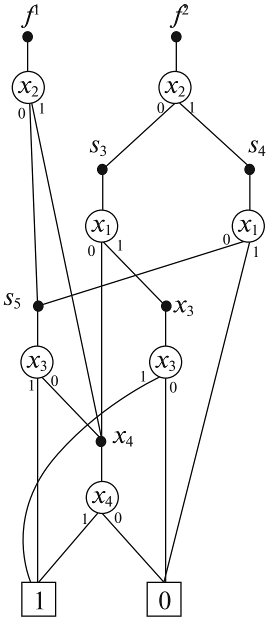

Fig. 2. BDD representation for subsystem $\{f^1, f^2\}$ of Boolean functions.

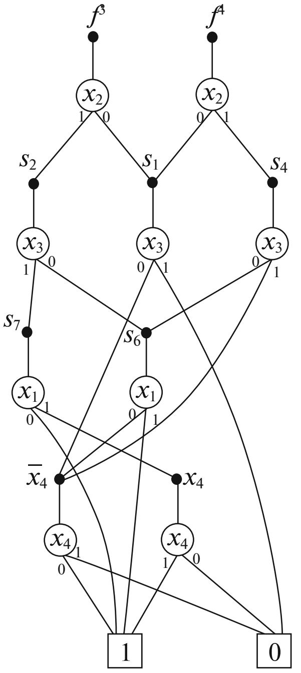

Fig. 3. BDD representation for subsystem $\{f^3, f^4\}$ of Boolean functions.

(Binary Decision Diagram with Inverse (BDDI) cofactors [17] is a representation of systems of Boolean functions based on Shannon's decompositions with the possible use of subfunction inversions.)

6. RESULTS OF A NUMERICAL EXPERIMENT

A numerical experiment was conducted to study how the extraction procedure of connected subsystems of functions affects the complexity of the resulting logic circuits (i.e., their area).

For this purpose, some systems of functions were taken from the collection of test examples [18]. For each of them, the truth table was constructed in the SDF format using *FLC* [16]. Three circuit versions were synthesized for each example based on different preliminary logic optimization programs as follows.

Version 1. Joint BDDI optimization, the conversion of the minimized BDDI representation to the VHDL description and circuit implementation.

Version 2. Separate BDDI optimization, the conversion of the minimized BDDI representation to the VHDL description and circuit implementation.

Version 3. The extraction of connected subsystems of functions, joint BDDI optimization of each subsystem, elimination of a hierarchical description, conversion to the VHDL description, and circuit implementation.

The connected subsystems were formed using almost the same parameter values $\rho^n(f^1, \ldots, f^m)$ expressed in % to the total number (2^n) of all collections in the corresponding Boolean space. For the system P82, $\rho^n(f^1, \ldots, f^m) = 60\%$ (see Table 17), which meant that the functions of the connected subsystem took the same values on $p = \lceil 0.6 \times 32 \rceil = 20$ collections in the Boolean space (here $\lceil a \rceil$ denotes the least integer that is greater or equal to a). For this example, $n = 5$ and the number of all collections in the Boolean space is 32.

The joint BDDIs were constructed in BDD_Builder [17] using the heuristic choice of the next variable in the decomposition as the one with the minimal number of different subfunctions (up to their inversions). For version 2, BDDIs were constructed in FLC using a combined approach—a successive execution of the extraction procedure of connected subsystems, with further implementation of strategies such as the BDDI optimization of the project's leaves and hierarchy elimination, and the final conversion of the resulting optimized representations to the VHDL descriptions. For all versions, the logic circuits were

Table 17. Experimental results

Example	Parameters of system of functions		Version 1. Joint BDDIs	Version 2. Separate BDDIs	Version 3. Decomposition into subsystems of connected functions and joint BDDIs for subsystems		
	n	m	S_{ASIC}	S_{ASIC}	$\rho^n(f^1, \ldots, f^m)$ %	Number of subsystems of connected functions	S_{ASIC}
P82	5	14	19988	**18481**	60	2	21282
Z5XP1	7	10	**18442**	19296	50	1	18665
M2	8	16	**45064**	46532	60	1	**45064**
M3	8	16	**52580**	58456	60	3	52781
ROOT	8	5	26109	27113	65	4	**26092**
ADDM4	9	8	80782	**77210**	55	3	87154
MP2D	11	14	**17471**	17666	60	1	18018
ADD6	12	7	**12806**	20518	55	6	21929
T3	12	8	17276	17170	60	4	**16941**
TIAL	14	8	**255531**	441523	60	5	295043
B12	15	9	18358	**16266**	55	2	17934
INTB	15	7	273532	248338	55	3	**221688**
M181	15	9	18849	**16344**	60	2	17549
IN0	15	11	94620	**93655**	60	3	130706
B2	16	17	**192655**	313925	60	1	**192655**
Number of best solutions			7	5	–	–	5

synthesized in *LeonardoSpectrum* [19] from library CMOS elements using VHDL descriptions with the same synthesis settings.

The experimental results are illustrated in Table 17 with the following notations: n as the number of variables; m as the number of functions in the system; and S_{ASIC} as the area of the logical circuit expressed in the conditional units of area of its logical elements. The best solutions—the circuits of the minimal area—are set in bold in Table 17. In some cases, a single subsystem of connected functions was formed for version 3 (the test examples Z5XP1, M2, MP2D, B2), while the area of the logical circuit was changed in comparison with version 1. This meant that some functions of the original system were inverted (see the example Z5XP1), and the synthesized circuit differed from version 1.

As indicated by the numerical experiment with the above collection of test examples, the extraction of connected subsystems can be an efficient procedure in the case of the logic optimization of multilevel BDDI representations (the subsequent synthesis produced logical circuits covering a smaller area). If the functions of a system are strongly connected, the best result is obtained by the logic optimization with joint BDDIs; and if they are weakly connected, with separate BDDIs. The advantages of decomposing a system of functions into subsystems of connected functions rely on these characteristics of the extracted subsystems of functions. Furthermore, the numerical experiment showed that in practice the corresponding extraction programs need to specify an appropriate value for the measure of connectivity, and this issue should be studied separately.

CONCLUSIONS

The logic optimization of multilevel representations for the high-dimensional systems of Boolean functions is a topical problem that arises in the course of reducing the total area of all functional blocks of very-large-scale integration circuits. A smaller area (fewer transistors in the circuits) has a positive effect on the power saving of logical circuits. In many cases, the extraction of subsystems of connected functions with further optimization of the BDDI representations of subsystems actually decreases the complexity of functional descriptions, thereby reducing the area and power consumption of the logical circuits synthesized using the optimized functional descriptions.

REFERENCES

1. K. R. Brayton, G. D. Hactel, and A. L. Sangiovanni-Vincentelli, "Synthesis of multi-level combinational logic circuits," TIIER **78** (2) (1990).
2. K. R. Brayton, G. D. Hactel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli, *Logic Minimization Algorithm for VLSI Synthesis* (Kluwer Academic, Dordrecht, 1984).
3. A. D. Zakrevskii, Yu. V. Pottosin, and L. D. Cheremisinova, *The Logical Basis of the Designing the Discrete Tools* (Fizmatlit, Moscow, 2007) [in Russian].
4. P. N. Bibilo, *Application of Binary Choice Diagrams in the Synthesis of Logic Circuits* (Belarus. Navuka, Minsk, 2014) [in Russian].
5. M. A. Gavrilov, "Composition and decomposition of combinational automata," in *Automata Theory* (Nauka, Moscow, 1976) [in Russian].
6. M. A. Gavrilov, V. V. Devyatkov, and E. I. Pupyrev, *Logical Design of Discrete Automata* (Nauka, Moscow, 1977) [in Russian].
7. O. P. Kuznetsov, "On software implementation of logic functions and automata," Avtom. Telemekh., No. 7 (1977).
8. R. E. Bryant and C. Meinel, "Ordered binary decision diagrams," in *Logic Synthesis and Verification,* Ed. by S. Hassoun, T. Sasao, and R. K. Brayton (Kluwer Academic, Dordrecht, 2002).
9. S. Yang and M. Ciesielski, "BDS: a BDD-based logic optimization system," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **21** (7) (2002).
10. P. N. Bibilo, *Decomposition of Boolean Functions Based on the Solution of Logical Equations* (Belarus. Navuka, Minsk, 2009) [in Russian].
11. P. N. Bibilo, N. A. Kirienko, and Yu. Yu. Lankevich, "Logical optimization the multilevel representations of systems of Boolean functions based on partitioning into blocks and Shannon decomposition," Informatika **15** (3), 56–70 (2018).
12. Yu. V. Pottosin and E. A. Shestakov, "Orthogonalization of a system of fully defined boolean functions," in *Logical Design, Collection of Articles* (Inst. Tekh. Kibern. NAN Belarusi, Minsk, 2000), No. 5 [in Russian].
13. A. D. Zakrevskii, *Calculations in Multidimensional Boolean Space* (OIPI NAN Belarusi, Minsk, 2011) [in Russian].
14. P. N. Bibilo, *Synthesis of Combination PLA-Structures for VLSI* (Nauka Tekh., Minsk, 1992) [in Russian].
15. D. E. Knuth, *The Art of Computer Programming* (Addison-Wesley, Reading, MA, 2011), Vol. 4.
16. P. N. Bibilo and V. I. Romanov, *Logical Design of Discrete Devices Using Production-Frame Model of Knowledge Representation* (Belarus. Navuka, Minsk, 2011) [in Russian].
17. P. N. Bibilo and Yu. Yu. Lankevich, "The use of Zhegalkin polynomials for minimization of multilevel representations of Boolean functions based on Shannon expansion," Program. Inzhen. **8**, 369–384 (2017).
18. C. Jeong, Computer-Aided Design of Digital Systems, Department of Computer Science. http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex.
19. P. N. Bibilo and N. A. Avdeev, *VHDL. Effective Use in the Design of Digital Systems* (SOLON-Press, Moscow, 2006) [in Russian].

Translated by A. Mazurov