ISSN 1063-7397, Russian Microelectronics, 2015, Vol. 44, No. 5, pp. 338–354. © Pleiades Publishing, Ltd., 2015.
Original Russian Text © N.A. Avdeev, P.N. Bibilo, 2015, published in Mikroelektronika, 2015, Vol. 44, No. 5, pp. 383–399.

Logical Optimization Efficiency in the Synthesis of Combinational Circuits

N. A. Avdeev and P. N. Bibilo

United Institute of Informatics Problems, National Academy of Sciences of Belarus, ul. Surganova 6, Minsk, 220012 Belarus

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

Received May 5, 2014

Abstract—The results of the experimental comparison of minimization programs for various forms of representations of the systems of completely specified Boolean functions are described. The experiments show that the minimization programs of multilevel representations based on the Shannon expansion and decomposition are preferential when synthesizing combinational logic circuits from library elements compared to the minimization programs in a class of disjunctive normal forms.

DOI: 10.1134/S1063739715050029

INTRODUCTION

The synthesis of combinational circuits in the specified basis (library) of logic elements is traditionally divided into two large stages: the technologically independent optimization of the implemented sets of Boolean functions and technological mapping—covering the optimized representations by descriptions of library logic elements. The first stage exerts a decisive influence on the main parameters (complexity, performance, and power consumption) of logic circuits. Methods of separate and joint minimization of the set of Boolean functions in a class of disjunctive normal forms (DNFs) have so far been used at this stage as the main optimization method. Recently, they were complemented by optimization methods of multilevel representations based on the Shannon expansion, the binary decision diagrams (BDDs) and decomposition methods, which allow us to reduce the number of arguments of the implemented sets of functions.

The creation of effective expert systems of the automated synthesis of logic circuits requires the knowledge on the preferential use of the fields of programs of the technologically independent optimization for the functional descriptions of logic circuits [1]. This study is devoted to the experimental comparison of the efficiency of optimization programs of representations for sets of Boolean functions applied when synthesizing logic circuits from elements of the designing library of very large domestic custom integrated circuits (VLSIs). The advantage of applying the optimization programs based on the Shannon expansion and decomposition programs compared to the traditional minimization programs of functions in the DNF class for the synthesis of circuits is shown by a set of widely known examples.

1. MATRIX FORM OF SPECIFICATION OF THE SET OF BOOLEAN FUNCTIONS

The binary (0, 1) functions $f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_n)$ of binary (Boolean) variables $x_1, x_2, \ldots x_n$ are called the Boolean functions. As the vector Boolean function $\boldsymbol{f}(\boldsymbol{x})$, we understand the ordered set of Boolean functions $\boldsymbol{f}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x}))$. The matrix form of specification of the vector of the completely specified function $\boldsymbol{f}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), f^2(\boldsymbol{x}), f^3(\boldsymbol{x}))$, where

$$f^1 = x_1x_2\bar{x}_4x_5\bar{x}_6 \vee \bar{x}_1x_4\bar{x}_5x_6 \vee x_2\bar{x}_3x_5;$$

$$f^2 = \bar{x}_1\bar{x}_4x_5\bar{x}_6 \vee \bar{x}_1\bar{x}_3x_5 \vee x_1x_2x_3x_5\bar{x}_6 \vee x_1\bar{x}_2x_4\bar{x}_5x_6;$$

$$f^3 = x_1\bar{x}_2\bar{x}_3x_6 \vee x_1\bar{x}_2x_4x_6 \vee x_1\bar{x}_3x_4x_6 \vee \bar{x}_1x_2\bar{x}_4x_5\bar{x}_6 \vee x_1\bar{x}_2x_5 \vee x_2\bar{x}_3x_5$$

is presented in Table 1. This form consists of ternary matrix T^x of the specification of elementary conjunctions in the form of ternary vectors and Boolean matrix B^f of entries of conjunctions in the DNFs of component functions $f^j(\boldsymbol{x}), j = 1, \ldots, m$. We will also call the representation of the vector function by the pair of matrices $\langle T^x, B^f \rangle$ as the matrix form of the set of DNF Boolean functions or simply the DNF set.

2. LOGICAL OPTIMIZATION OF THE SET OF BOOLEAN FUNCTIONS

Joint Minimization of the Set of Boolean Functions in the DNF Class

This method of the logical optimization is classical and is used to minimize the total number of elementary conjunctions for which all functions of the initial system are specified and reduce the number of literals $\bar{x}_i$, x_i in the elementary conjunctions. Under the matrix representation of the functions, the joint minimization allows us to decrease the number of the lines in matrices T^x and B^f, increase the number of indefinite values "–" in the specification of the conjunctions by ternary vectors, and possibly reduce the number of unity values in the Boolean matrix B^f. The methods of the joint minimization of the set of Boolean functions are well known in the literature [2–5]. If you hold a joint minimization system DNF (Table 1), it is possible to obtain 11 (instead of 12) elementary conjunctions of DNF for the job functions of the system, minimized system DNP is presented in Table 2.

Joint BDD Minimization of Sets of Boolean Functions in a Class of Multilevel Representation Based on the Shannon Expansion

Representation $f(x_1,\ldots,x_n) = f$ in the form

$$f = \bar{x}_i f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n) \vee x_i f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n). \quad (1)$$

is called the Shannon expansion of the completely specified Boolean function $f(x_1,\ldots,x_n)$ with respect to variable x_i. Functions $f(x_1,\ldots, x_{i-1}, 0, x_{i+1}, \ldots x_n)$ and $f(x_1,\ldots, x_{i-1}, 1, x_{i+1}, \ldots x_n)$ in (1) are called expansion coefficients. They are derived from function $f(x_1,\ldots,x_n)$ by the substitution of constant 0 or 1, respectively, instead of variable x_i. The BDD specifies the sequence of Shannon expansions of the initial function and acquired expansion coefficients in the form of a graph. The minimization of the BDD complexity is based on the fact that identical expansion coefficients not only of one but also of several (or even of all) component functions can appear in the course of the expansion. We understand the optimization of multilevel representations of the sets of Boolean functions corresponding to reduced ordered BDDs (ROBDDs) as the joint BDD minimization. A similar description of the ordered BDDs is given in [6] and that of ROBDDs is given in [7].

Let us denote the sequence of variables over which the Shannon expansion is performed for vector function $\boldsymbol{f}(\boldsymbol{x})$, which consists of three component functions f^1, f^2, and f^3 (Table 1), as $\langle x_1, x_2, x_3, x_4, x_5, x_6\rangle$. The designed BDD is shown in Fig. 1, and this BDD corresponds to a multilevel representation

$$f^1 = \bar{x}_1\psi^1 \vee x_1\psi^2;\ \ f^2 = \bar{x}_1\varphi^2 \vee x_1\psi^3;\ \ f^3 = \bar{x}_1\psi^2 \vee x_1\psi^4;$$
$$\psi^1 = \bar{x}_2\varphi^1 \vee x_2\varphi^1;\ \psi^2 = x_2\varphi^2;\ \psi^3 = \bar{x}_2 s^1 \vee x_2\varphi^3;\ \psi^4 = \bar{x}_2\varphi^4 \vee x_2\varphi^5;\ \varphi^1 = \bar{x}_3 s^2 \vee x_3 s^1;\ \varphi^2 = \bar{x}_3\lambda^3 \vee x_3 s^3;$$
$$\varphi^3 = x_3\lambda^4;\ \varphi^4 = \bar{x}_3\lambda^2 \vee x_3 s^2;\ \varphi^5 = \bar{x}_3 s^2;\ s^1 = x_4\lambda^1;\ s^2 = \bar{x}_4\lambda^3 \vee x_4\lambda^2;\ s^3 = \bar{x}_4\lambda^4;$$
$$\lambda^1 = \bar{x}_5\omega^1;\ \lambda^2 = \bar{x}_5\omega^1 \vee x_5;\ \lambda^3 = x_5;\ \lambda^4 = x_5\omega^2;\ \omega^1 = x_6;\ \omega^2 = \bar{x}_6.$$

The BDD complexity is evaluated by the number of vertices marked by symbols of functions, and the vertices corresponding to arguments are not taken into account when evaluating the BDD complexity. For example, the complexity of BDD (Fig. 1) is 21. The main problem when designing the lower-complexity BDDs is the selection of the permutation of variables over which the BDD is constructed.

Separate Decomposition of Sets of Boolean Functions

Let the partition Y/Z of set $X = \{x_1,\ldots,x_n\}$ of variables of the vector Boolean function $\boldsymbol{f}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x}))$ into two nonintersecting subsets $Y = \{y_1,\ldots,y_r\}$, $Z = \{z_1,\ldots,z_{n-r}\}$, $2 \le r \le n-1$, $n \ge 3$ be specified. Let us denote the vector acquired by ordering the variables from subset $Y = \{y_1,\ldots,y_r\}$ through $\boldsymbol{y} = (y_1,\ldots,y_r)$ and the vector acquired by ordering the variables from subset $Z = \{z_1,\ldots,z_{n-r}\}$ through $\boldsymbol{z} = (z_1,\ldots,z_{n-r})$.

We will call the construction of functional expansions

$$\begin{cases} f^1(\boldsymbol{x}) = f^1(\boldsymbol{y},\boldsymbol{z}) = g^1(\boldsymbol{h}^1(\boldsymbol{y}),\boldsymbol{z}), \\ \ldots \\ f^m(\boldsymbol{x}) = f^m(\boldsymbol{y},\boldsymbol{z}) = g^m(\boldsymbol{h}^m(\boldsymbol{y}),\boldsymbol{z}), \end{cases} \quad (2)$$

where $\boldsymbol{h}^j(\boldsymbol{y}) = (h_1^j(\boldsymbol{y}), \ldots, h_{p_j}^j(\boldsymbol{y}))$, $j = 1, \ldots, m$, as the separate decomposition of the set of Boolean func-

Table 1. DNF set of Boolean functions

T^x						B^f		
x_1	x_2	x_3	x_4	x_5	x_6	f^1	f^2	f^3
1	1	–	0	1	0	1	0	0
0	–	–	1	0	1	1	0	0
0	–	–	0	1	0	0	1	0
0	–	0	–	1	–	0	1	0
1	1	1	–	1	0	0	1	0
1	0	–	1	0	1	0	1	0
1	0	0	–	–	1	0	0	1
1	0	–	1	–	1	0	0	1
1	–	0	1	–	1	0	0	1
0	1	–	0	1	0	0	0	1
1	0	–	–	1	–	0	0	1
–	1	0	–	1	–	1	0	1

Table 2. Minimized DNF set

T^x						B^f		
x_1	x_2	x_3	x_4	x_5	x_6	f^1	f^2	f^3
1	1	1	–	1	0	0	1	0
1	1	–	0	1	0	1	0	0
0	–	–	1	0	1	1	0	0
0	1	–	0	1	0	0	0	1
1	0	–	1	0	1	0	1	1
0	–	–	0	1	0	0	1	0
1	–	0	1	–	1	0	0	1
1	0	0	–	–	1	0	0	1
0	–	0	–	1	–	0	1	0
1	0	–	–	1	–	0	0	1
–	1	0	–	1	–	1	0	1

tions $\boldsymbol{f}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), ..., f^m(\boldsymbol{x}))$ by the specified partition Y/Z of the multitude of variables X, and the number of intermediate variables (components of vector function $\boldsymbol{h}^j(\boldsymbol{y}) = (h_1^j(\boldsymbol{y}), \dots, h_{p_j}^j(\boldsymbol{y})))$ is minimal for each component function f^j.

The separate decomposition of the vector function specified in Table 1 by the partition of variables $Y = \{x_1, x_2, x_3, x_4\}$ and $Z = \{x_5, x_6\}$ allows us to find seven intermediate functions at the first stage (input unit) of the circuit (Fig. 2):

$$h_1^1 = x_1x_2x_3\bar{x}_4;$$

$$h_2^1 = x_1x_2\bar{x}_3x_4 \vee x_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_3;$$

$$h_3^1 = \bar{x}_1x_4;$$

$$h_1^2 = x_1x_3\bar{x}_4 \vee x_1x_2x_3 \vee \bar{x}_1\bar{x}_3;$$

$$h_2^2 = \bar{x}_1\bar{x}_2x_4 \vee \bar{x}_1x_3\bar{x}_4 \vee x_1x_2x_3;$$

$$h_1^3 = x_1x_2\bar{x}_3x_4 \vee \bar{x}_1x_2x_3\bar{x}_4 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_4;$$

$$h_2^3 = x_1\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_1x_2x_3\bar{x}_4 \vee x_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_3.$$

We note that these functions are jointly minimized in the DNF class, while the decomposition was performed using the method described in [8]. The matrix form of these functions is presented in Table 3. Output functions minimized in the DNF class, which depend on variables x_5, x_6, and intermediate variables, are specified in Table 4. It is easy to see that each of the functions $g^1(h_1^1, h_2^1, h_3^1, x_5, x_6) = f^1$, $g^2(h_1^2, h_2^2, x_5, x_6) = f^2$, and $g^3(h_1^3, h_2^3, x_5, x_6) = f^3$ with a separate decomposition depends on its subset of intermediate variables.

The functions of both the input and output block of decomposition can be minimized in the class of BDD representations. The representation of functions of the input block after the joint BDD minimization has the form

$$h_1^1 = x_1\varphi_1;\ h_2^1 = x_2\bar{x}_3;\ h_3^1 = \bar{x}_1x_4;\ h_1^2 = \bar{x}_1\varphi_{11} \vee x_1\varphi_2;\ h_2^2 = \bar{x}_1\varphi_7 \vee x_1\varphi_3;\ h_1^3 = \bar{x}_1\varphi_1 \vee x_1\varphi_4;\ h_2^3 = \bar{x}_1\varphi_5 \vee x_1\varphi_6;$$
$$\varphi_1 = x_2\varphi_7;\ \varphi_2 = x_2x_3;\ \varphi_3 = \bar{x}_2x_4 \vee x_2x_3;\ \varphi_4 = \bar{x}_2\varphi_8 \vee x_2\varphi_9;\ \varphi_5 = x_2\varphi_{11};\ \varphi_6 = \bar{x}_2\varphi_7 \vee x_2\varphi_{10};$$
$$\varphi_7 = \bar{x}_4x_3;\ \varphi_8 = \bar{x}_4\bar{x}_3 \vee x_4;\ \varphi_9 = x_4\bar{x}_3;\ \varphi_{10} = \bar{x}_4\bar{x}_3;\ \varphi_{11} = \bar{x}_4 \vee x_4\bar{x}_3.$$

A multilevel BDD representation of the functions of the output block after the joint BDD minimization has the following form:

$$f^1 = \bar{h}_1^1s_1 \vee h_1^1s_2;\ f^2 = \bar{h}_1^2s_3 \vee h_1^2s_4;\ f^3 = \bar{h}_1^3s_5 \vee h_1^3s_6;\ s_1 = \bar{h}_2^1s_7 \vee h_2^1s_8;\ s_2 = \bar{h}_2^1s_9;\ s_3 = h_2^2s_{10};\ s_4 = \bar{h}_2^2x_5 \vee h_2^2s_{12};$$
$$s_5 = h_2^3x_5;\ s_6 = \bar{h}_2^3s_{11} \vee h_2^3s_{12};\ s_7 = h_3^1s_{10};\ s_8 = \bar{h}_3^1x_5 \vee h_3^1s_{11};\ s_9 = \bar{h}_3^1s_{12};\ s_{10} = x_6\bar{x}_5;\ s_{11} = \bar{x}_6x_5 \vee x_6;\ s_{12} = \bar{x}_6x_5.$$

In this (and previous) multilevel representations, the literals of the variables are not redesignated at lower BDD stages.

In the example under consideration, the separate decomposition allowed us to reduce the number of arguments of the implemented functions; however, the

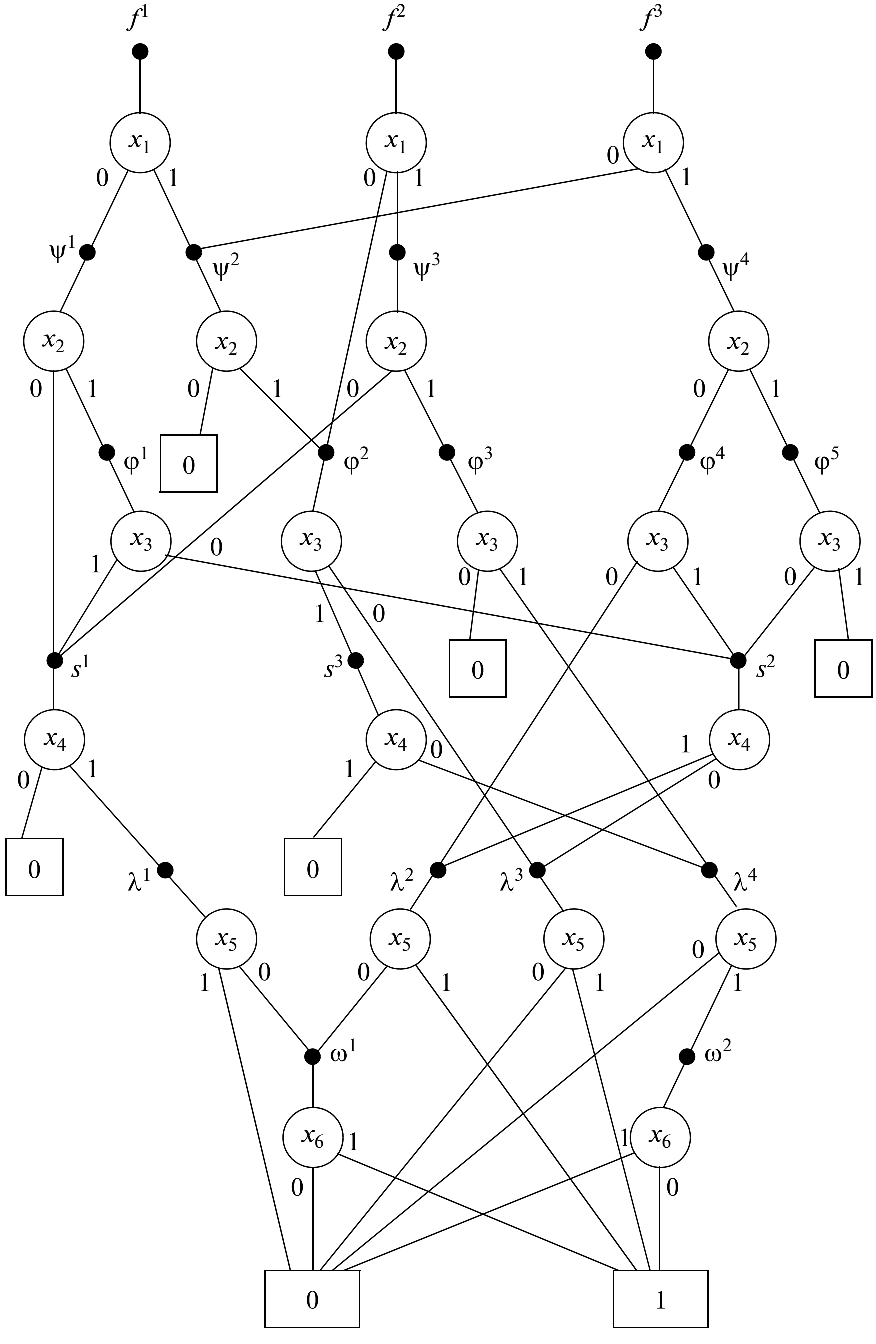

Fig. 1. Binary decision diagram (BDD).

summary complexity of the BDD representations of the two blocks—input and output—did not decrease after the decomposition. For other examples (of larger dimensionality) of DNF systems, the decomposition can lead not only to a decrease in the number of variable subfunctions but also to a decrease in the complexity of the BDD representations of subfunctions h and g of the functional decomposition.

Joint Decomposition of Sets of Boolean Functions

We will call the construction of the functional expansion

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{y}), \boldsymbol{z}), \tag{3}$$

where $\boldsymbol{h}(\boldsymbol{y}) = (h_1(\boldsymbol{y}), \ldots, h_p(\boldsymbol{y}))$ is the vector function having the minimal number p of components, as the joint decomposition of set of Boolean functions $\boldsymbol{f}(\boldsymbol{x}) =$

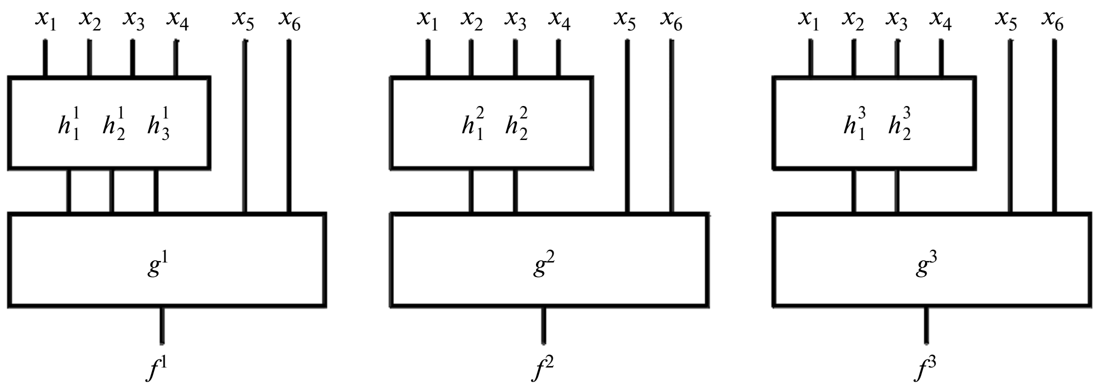

Fig. 2. Separate decomposition of the multitude of arguments by partition $Y = \{x_1, x_2, x_3, x_4\}$, $Z = \{x_5, x_6\}$.

Table 3. Minimized DNF set of the input block functions

x_1	x_2	x_3	x_4	h_1^1	h_2^1	h_3^1	h_1^2	h_2^2	h_1^3	h_2^3
1	1	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	1
1	1	0	1	0	1	0	0	0	1	0
0	1	1	0	0	0	0	0	0	1	1
1	0	0	–	0	0	0	0	0	1	0
1	0	–	1	0	0	0	0	1	1	0
0	–	–	1	0	0	1	0	0	0	0
0	–	1	0	0	0	0	1	1	0	0
1	1	1	–	0	0	0	1	1	0	0
–	1	0	0	0	1	0	0	0	0	1
0	1	0	–	0	1	0	0	0	0	1
0	–	0	–	0	0	0	1	0	0	0

Table 4. Minimized DNF set of the output block functions

h_1^1	h_2^1	h_3^1	h_1^2	h_2^2	h_1^3	h_2^3	x_5	x_6	f^1	f^2	f^3
0	1	–	–	–	–	–	1	–	1	0	0
0	–	1	–	–	–	–	0	1	1	0	0
1	0	0	–	–	–	–	1	0	1	0	0
–	–	–	0	1	–	–	0	1	0	1	0
–	–	–	1	–	–	–	1	0	0	1	0
–	–	–	1	0	–	–	1	–	0	1	0
–	–	–	–	–	0	1	1	–	0	0	1
–	–	–	–	–	1	–	1	0	0	0	1
–	–	–	–	–	1	0	–	1	0	0	1

$(f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x}))$ by the specified partition Y/Z of the multitude of variables X.

Decomposition (3) of vector function $\boldsymbol{f}(\boldsymbol{x})$ is called joint [8] since all component functions f^j of the decomposed vector Boolean function $\boldsymbol{f}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x}))$ have common (jointly used) intermediate subfunctions $h_1(\boldsymbol{y}), \ldots, h_p(\boldsymbol{y})$.

The result of the joint decomposition of the vector function specified in Table 1 by partition $Y = \{x_3, x_4, x_5, x_6\}$, $Z = \{x_1, x_2,\}$ of the set of variables is shown in Fig. 3. With the joint decomposition, there are three rather than seven intermediate variables; however, each of them is used as an intermediate variable of each of the component functions f^1, f^2, and f^3 of the decomposed vector function.

The methods of performance of the joint decomposition for sets of functions differ by the used mathematical apparatus. The decomposition of the vector function with a minimal number of intermediate functions is considered in the publications for the case of the specification of vector functions by matrix forms—DNF systems [2, 9, 10], and the decomposition of BDD representations of vector functions is described in [8]. Various formal apparatuses are applied—spectral representations [11], logical equations [12], matrix logic equations [2], and final predicates [13].

The main problem of the decomposition is the selection of the partition of the variables by which it is performed. It is shown in [8] that the use of the BDD apparatus for the decomposition of the set of functions facilitates the solution of the problem of selecting the partition of variables.

3. PROBLEMS OF TECHNOLOGICALLY INDEPENDENT LOGICAL OPTIMIZATION

3.1. The ESPRESSO program (version 2.3) of the joint DNF minimization of sets of Boolean functions $\boldsymbol{f}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x}))$, $\boldsymbol{x} = (x_1,\ldots,x_n)$ in the DNF class is a well-known program of minimization available in the Internet, which is described in monograph [3].

The input and resulting data of this program are the text matrix representations of the initial and minimized sets of DNF Boolean functions, respectively.

3.2. The TIE_BDD program of the joint BDD minimization of the sets of Boolean functions implements the minimization algorithm [14] of multilevel representations of the set of Boolean functions based on the Shannon expansion. The initial data are the DNF sets and the results are the multilevel formula BDD representations. When performing the experiments, this program constructed the BDD no larger than by 5000 of randomly selected permutations of variables and selected the BDD of the lowest complexity among the considered variants.

3.3. The SEPT_BDD program of the separate decomposition of DNF sets based on the apparatus of BDD representations of the sets of Boolean functions implements the method described in [8]. The initial data are the DNF sets and the results are superpositions of functions of form (2), and the functions of the input block are presented in the form of the DNF set, while the output block is presented in the BDD form.

The initial data for the joint decomposition programs are the DNF sets, and the results are superpositions of form (3), and each of the vector functions $\boldsymbol{h}(\boldsymbol{y})$ (input block) and $\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{y}), \boldsymbol{z})$ (output block) are represented in the form of DNF sets.

3.4. The DECU_BDD program of the joint decomposition of the DNF sets based on the apparatus of BDD representations of the sets of Boolean functions is described in [8].

The algorithms for the selection of the partition of variables, which are implemented in SEPT_BDD and DECU_BDD programs, are described in [8].

3.5. The DEC_FT program of the joint decomposition of matrix representations of DNF sets of Boolean functions implements the decomposition method based on compact tables and is described in [10].

3.6. The DEC_HIE program of the joint decomposition of matrix representations of DNF sets of Boolean functions is described in [15] and acquires the intermediate variables by the method of the arrangement of codes (values of intermediate variables) in hypercube vertices.

3.7. The DEC_TRIV program of the joint decomposition of matrix representations of DNF sets of Boolean functions is a simplified variant of the DEC_HIE program since it uses the trivial DNF coding, for which the intermediate variables are specified, to accelerate its operation.

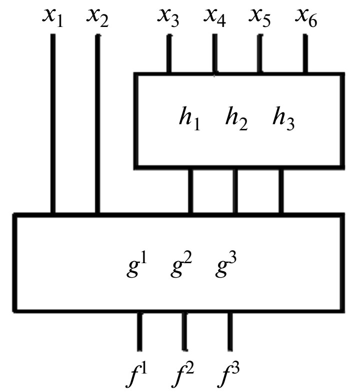

Fig. 3. Separate decomposition of the multitude of arguments by partition $Y = \{x_3, x_4, x_5, x_6\}$, $Z = \{x_1, x_2,\}$.

Programs DEC_HIE and DEC_TRIV searches all $2^n - n - 2$ nontrivial partitions of the multitude of variables and find the best variant of decomposition (3) evaluated by formula $O(2^r/r = 2^{n-r+p}/(n-r+p))$, where n is the number of arguments of the set of functions, r is the number of arguments in block Y, and p is the number of variables. These programs are applied for sets of functions which depend on $n \le 20$ variables.

Table 5. Functions and areas of elements of the logical circuit (Fig. 4)

Element	Function	Area
N	$y = \overline{A}$	223
NA	$y = \overline{AB}$	307
NA3	$y = \overline{ABC}$	407
NAO	$y = \overline{(A \vee B)C}$	435
NA4	$y = \overline{ABCD}$	491
NAOO	$y = \overline{(A \vee B)(C \vee D)}$	525
NOA	$y = \overline{(AB) \vee C}$	363
NO3	$y = \overline{A \vee B \vee C}$	396
NO3A	$y = \overline{(AB) \vee C \vee D}$	491
NAO3	$y = \overline{(A \vee B \vee C)D}$	491
NA3O3	$y = \overline{(A \vee B \vee C)DE}$	586

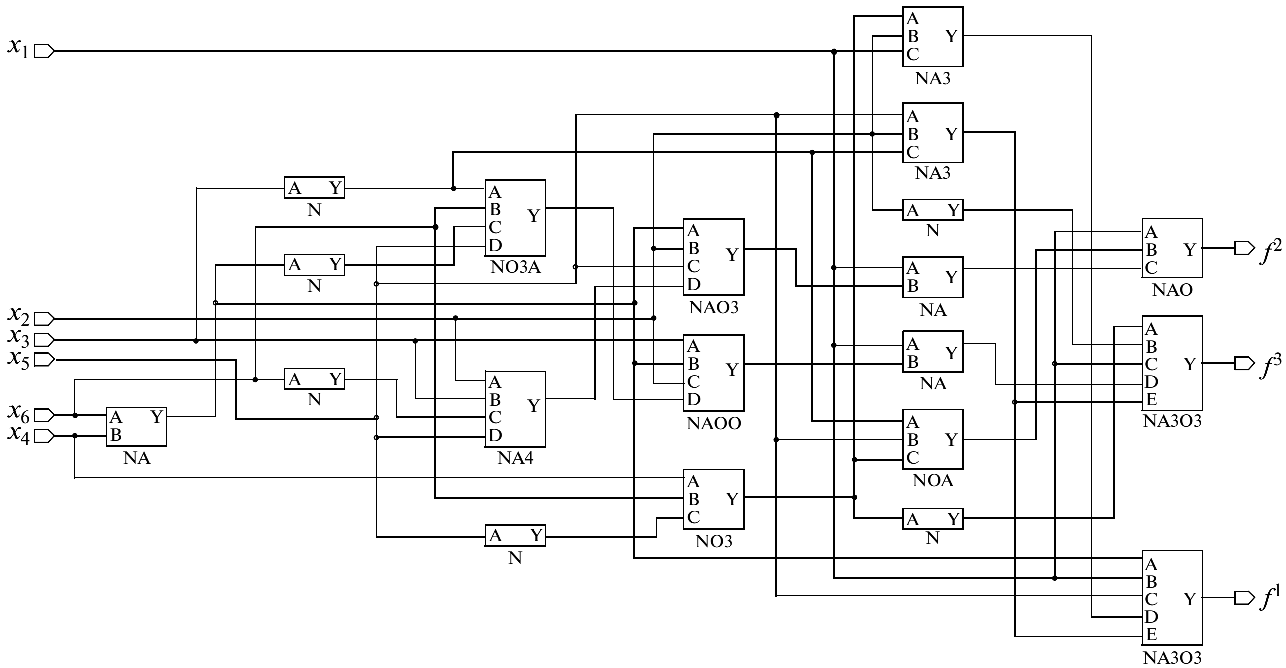

Fig. 4. Logical circuit based on library elements of the custom VLSI.

4. CIRCUIT AREA AT THE LOGICAL DESIGN STAGE

As the circuit area at the logical design stage, we usually understand the summary area of the crystal required to arrange the circuit elements. Although this criterion of complexity is approximate (the area for interconnections of circuit elements often is not taken into account), it is often used at the logical design stage of circuits in contrast with the topological design stage, when the total area for the elements and interconnections (links) of elements is understood as the area. In further experiments, area S_{ASIC} of the circuit from library elements was calculated as the sum of the areas of the elements constituting the circuit. Functions and areas of logical elements for the circuit (Fig. 4) acquired as a result of the logical synthesis according to the DNF set, which is presented in Table 1, are presented in Table 5. The logical circuit (Fig. 4), which consists of 20 library elements, has the complexity

$$S_{ASIC} = 7437 \text{ (conditional units)},$$

since $(223 \times 6) + (307 \times 3) + (407 \times 2) + (586 \times 2) + (491 \times 3) + 435 + 525 + 396 + 363 = 7437$.

This calculation can be easily performed using the areas (see Table 5) of the elements entering the circuit: there are six inverters, three NA elements, two NA3 elements, etc., in the circuit.

5. EXPERIMENTS

Circuit implementations of various representations of the same set of Boolean functions under the synthesis in industrial synthesizers can have various areas since these synthesizers are sensitive to the specification form of initial data. The experiments described below consisted of various methods of the preliminary technologically independent optimization of representations of sets of Boolean functions, by which the logical circuits were synthesized in the same synthesis library and under the same synthesis modes.

Figure 5 shows the general organization of the experiments. The initial data in all experiments were the DNF sets of completely specified Boolean functions from the set of examples available at http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex/

The LeonardoSpectrum synthesizer (version 2011a.4) was used in all experiments as the industrial system for the synthesis of logical circuits [16], and the domestic library for the design of custom digital CMOS VLSIs was used as the target synthesis library. The input data of the LeonardoSpectrum synthesizer in the performed experiments were the VHDL descriptions of various representations of vector functions acquired using optimization programs described in Section 3. The translation of text representations of sets of Boolean functions into VHDL descriptions were performed by the FormatSF program. The VHDL language is the language of specification of projects of digital systems implemented based on VLSIs, including the input language of the LeonardoSpectrum synthesizer. For example, the matrix form of the functions (see Table 3) has the following description in VHDL:

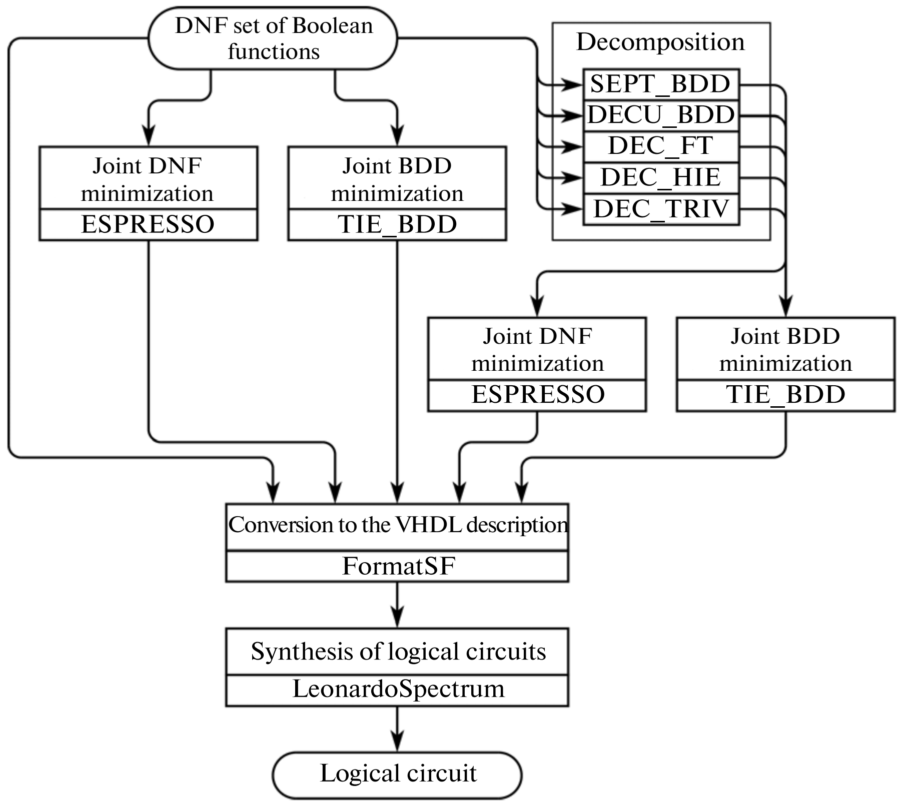

Fig. 5. Organization of experiments.

```
library ieee;
use ieee.std_logic_1164.all;
entity DECOMP1 is
  port(x1, x2, x3, x4 : in std_logic;
       h0, h1, h2, h3, h4, h5, h6 : out std_logic);
end DECOMP1;
architecture DECOMP1_arch of DECOMP1 is
begin
  h0 < = x1 and x2 and x3 and not x4;
  h1 < = ((x1 and x2 and not x3 and x4) or (x2 and not x3 and not x4)
         or (not x1 and x2 and not x3));
  h2 < = not x1 and x4;
  h3 < = ((not x1 and x3 and not x4) or (x1 and x2 and x3)
         or (not x1 and not x3));
  h4 < = ((x1 and not x2 and x4) or (not x1 and x3 and not x4)
         or (x1 and x2 and x3));
  h5 < = ((x1 and x2 and x4 and not x3) or (not x1 and x2 and x3 and not x4)
         or (x1 and not x2 and not x3) or (x1 and not x2 and x4));
  h6 < = ((x1 and not x2 and x3 and not x4)
         or (not x1 and x2 and x3 and not x4) or (x2 and not x3 and not x4)
         or (not x1 and x2 and not x3));
end DECOMP1_arch.
```

Table 6. Area of circuits

Circuit name	n	m	k	Experiment 1		
				no optimization	ESPRESSO	TIE_BDD
ADD6	12	7	1092	451913	85843	20222
ADDM4	9	8	512	311225	124663	89269
ADR4	8	5	256	131102	10948	7661*
ALU1	12	8	19	7109*	7109*	7109*
B12	15	9	431	29027	21879	16617*
B2	16	17	110	229840	207241	197967*
B9	16	5	123	26148	30227	24770*
BR1	12	8	34	32615	26243	27911
BR2	12	8	35	22632	21472*	21634
CLPL	11	5	20	2929*	2929*	2929*
CO14	14	1	47	12449	12449	12449
DC2	8	7	58	32375	27314	21946*
DIST	8	5	256	110060	87980	68601
EX7	16	5	123	26148	30227	24770*
F51M	8	8	256	276350	30668	25344
gary	15	11	442	107627	105429	98911*
IN0	15	11	138	133786	105535	98911*
IN1	16	17	110	229840	207241	197967*
IN2	19	10	137	141336	128368	92410*
INTB	15	7	664	402686	408300	227848*
LIFE	9	1	512	24641	23972	14977
LOG8MOD	8	5	47	35673	27906	25953*
M1	6	12	32	21773	22398	18715
M181	15	9	430	29546	20981	17225*
M2	8	16	96	85960	62334	61095
M3	8	16	128	97120	83901	60325
M4	8	16	256	222447	116131	82266
MAX1024	10	6	1024	423288	194072	140862*
MAX46	9	1	46	38602	37174	36878
MAX512	9	6	512	173220	101695	71446*
MLP4	8	8	256	164967	88895	71854*
MP2D	14	14	123	36471	18542	17968*
NEWAPLA	12	10	17	11763	11796	10189*
NEWAPLA1	12	7	10	7315	7315	6869*
NEWAPLA2	6	7	7	5569	5569	4944*
NEWBYTE	5	8	8	5569	5569	5569
NEWCOND	11	2	31	18899	15540	13414*
NEWCPLA1	9	16	38	29262	28625	28045
NEWCPLA2	7	10	19	13359*	14659	17438
NEWILL	8	1	8	5134	5134	5312
NEWTAG	8	1	8	1864*	1864*	2126
NEWTPLA	15	5	23	15044	15044	14229*
NEWTPLA1	10	2	4	4324	4324	3577

Table 6. (Contd.)

Circuit name	n	m	k	Experiment 1		
				no optimization	ESPRESSO	TIE_BDD
NEWTPLA2	10	4	9	9263	8588	7310
P82	5	14	24	23124	22008	19971*
RADD	8	5	120	39824	14982	8465*
RD53	5	3	32	13571	12142	9843
RD73	7	3	147	23347	23414	15925
ROOT	8	5	256	55750	34429	26717*
RYY6	16	1	112	4754	3337	4224
SEX	9	14	23	13124*	14692	13928
soar	83	94	529	175564	172483	167484*
SQN	7	3	96	43167	28319	23743
SQR6	6	12	64	66011	33262	27069*
SYM10	10	1	837	77785	69906	19882
T3	12	8	152	20105	18866*	19223
TIAL	14	8	640	323690	336022	295952*
vtx1	27	6	110	18386*	18386*	20216
x9dn	27	7	120	19195*	19195*	22342
Z4	7	4	128	52379	7527	6992
Z5XP1	7	10	128	148121	43172	26499*
Z9SYM	9	1	420	59896	42994	18018
Number of best solutions				7	7	31

The logical operators in VHDL are denoted as follows: not (negation), and (conjunction), and or (disjunction). Input variables are written as input (in) ports and implemented functions correspond to output (out) ports.

Furthermore, the following notations are used in the tables with the experimental results: n is the number of input variables, m is the number of functions, and k is the number of conjunctions in the initial set of DNF functions.

The synthesized logical circuits in the basis of library elements serve as the resulting data in the experiments. An example of such a circuit is shown in Fig. 4. The information on the summary area of the elements of the circuit is given by the LeonardoSpectrum synthesizer after the fulfillment of the technological synthesis stage. The following characteristics of the synthesized logical circuits are used in the tables with the experimental results: S_{ASIC} is the circuit area, τ is the delay of circuit (ns).

Experiment 1. This experiment consisted of an investigation of the influence of procedures of global optimization on the area and delay of a circuit from library elements. We compared the synthesis without the preliminary optimization and with the preliminary global joint DNF minimization (ESPRESSO) and BDD minimization (TIE_BDD).

The results of the first experiment on the evaluation of the area of the circuits are presented in Table 6, and Table 7 gives the delays of the circuits synthesized in this experiment.

Experiment 2. This experiment consisted of the influence of decomposition (programs SEPT_BDD, DECU_BDD, DEC_FT, DEC_HIE, DEC_TRIV) and subsequent DNF-minimization (ESPRESSO) of the formed decomposition blocks on the area and delay of the circuit from library elements.

Experiment 3. This experiment consisted of the influence of decomposition (programs SEPT_BDD, DECU_BDD, DEC_FT, DEC_HIE, DEC_TRIV) and subsequent BDD-minimization (TIE_BDD) of the formed decomposition blocks on the area and delay of the circuit from library elements.

The results of experiments 2 and 3 are presented in Tables 8 and 9. It resulted in each of the 62 initial examples being implemented 13 times—thrice in the first experiment and five times each in the second and third experiments.

The best results in the tables with the experimental results (with a smaller area or with a smaller delay) are marked by symbol *. For example, for the first example of the add6 circuit (see the first lines in Table 6 and Table 8), the circuit with the smallest area S_{ASIC} = 18414 was acquired with the preliminary optimization

Table 7. Delay of circuits (ns)

Circuit name	n	m	k	Experiment 1		
				no optimization	ESPRESSO	TIE_BDD
ADD6	12	7	1092	11.03	7.46	5.50
ADDM4	9	8	512	10.41	6.49	7.00
ADR4	8	5	256	7.38	3.50	3.90
ALU1	12	8	19	1.11*	1.11*	1.11*
B12	15	9	431	4.05	4.24	3.37*
B2	16	17	110	10.05*	11.05	10.24
B9	16	5	123	3.92	4.59	3.64*
BR1	12	8	34	6.54	5.47	6.08
BR2	12	8	35	6.44	6.11	6.43
CLPL	11	5	20	3.44*	3.44*	3.44*
CO14	14	1	47	4.73*	4.73*	4.73*
DC2	8	7	58	4.73	4.09*	4.11
DIST	8	5	256	7.97	5.92	5.51
EX7	16	5	123	3.92	4.59	3.64*
F51M	8	8	256	11.17	5.74	5.49
gary	15	11	442	6.66*	7.09	7.52
IN0	15	11	138	9.39	7.60	7.52*
IN1	16	17	110	10.05*	11.05	10.24
IN2	19	10	137	7.77	8.10	7.49*
INTB	15	7	664	11.03	9.02*	9.77
LIFE	9	1	512	5.54	4.90*	4.99
LOG8MOD	8	5	47	4.94	4.55	4.38
M1	6	12	32	5.34	4.59	3.53
M181	15	9	430	3.72	2.96*	3.43
M2	8	16	96	7.51	6.28	5.53
M3	8	16	128	7.86	7.69	5.18*
M4	8	16	256	11.01	8.07	5.88
MAX1024	10	6	1024	10.77	7.90	7.33
MAX46	9	1	46	5.68	5.13*	5.13*
MAX512	9	6	512	8.30	5.81*	6.80
MLP4	8	8	256	8.06	6.17	5.67*
MP2D	14	14	123	7.57	4.46	4.68
NEWAPLA	12	10	17	2.99	2.75*	4.08
NEWAPLA1	12	7	10	2.49*	2.51	3.39
NEWAPLA2	6	7	7	2.79	2.79	2.45*
NEWBYTE	5	8	8	2.57	2.57	2.52
NEWCOND	11	2	31	4.73	3.88*	5.71
NEWCPLA1	9	16	38	4.91	4.91	4.52*
NEWCPLA2	7	10	19	3.13*	3.63	3.64
NEWILL	8	1	8	3.54	3.54	3.43
NEWTAG	8	1	8	1.71*	1.71*	1.90
NEWTPLA	15	5	23	3.52*	3.52*	4.60
NEWTPLA1	10	2	4	2.60	2.60	3.16

Table 7. (Contd.)

Circuit name	n	m	k	Experiment 1		
				no optimization	ESPRESSO	TIE_BDD
NEWTPLA2	10	4	9	4.52	4.94	3.61*
P82	5	14	24	4.17	3.46	2.85*
RADD	8	5	120	5.65	4.04	4.44
RD53	5	3	32	3.42	4.05	2.85*
RD73	7	3	147	4.75	4.70	3.94*
ROOT	8	5	256	6.47	4.55	4.81
RYY6	16	1	112	1.77*	2.43	2.23
SEX	9	14	23	2.32*	3.12	2.85
soar	83	94	529	5.63*	5.88	6.12
SQN	7	3	96	5.32	3.70*	4.34
SQR6	6	12	64	5.97	5.04	3.40*
SYM10	10	1	837	9.60	7.85	5.71*
T3	12	8	152	5.63	4.20*	4.31
TIAL	14	8	640	8.37*	8.67	9.56
vtx1	27	6	110	4.08*	4.08*	5.15
x9dn	27	7	120	5.24	5.24	4.57*
Z4	7	4	128	6.32	5.05	3.09
Z5XP1	7	10	128	7.03	5.18	3.85*
Z9SYM	9	1	420	9.90	6.07	5.39
Number of best solutions				15	16	21

using the SEPT_BDD program (experiment 3); it also turned out that this circuit has the shortest delay τ = 4.27 (ns) among all the 13 fulfilled circuit implementations of this example (see first lines in Table 7 and Table 9). For certain examples, programs DEC_FT, DEC_HIE, and DEC_TRIV failed to perform the decomposition; therefore, the area and delay parameters are absent in the corresponding places of the tables.

6. DISCUSSION OF EXPERIMENTAL RESULTS

Let us initially analyze the results according to the area criterion of the circuits using Tables 6 and 8. The largest number of best solutions by area (example 31) was provided by the TIE_BDD program of global BDD minimization (experiment 1). The SEPT_BDD program of the separate decomposition (experiment 3), which is fulfilled along with the BDD minimization of the found expansion blocks, turned out to the second-most efficient program. The route of the circuit implementation without the preliminary optimization also turned out to be rather successful since the optimization programs implemented in the LeonardoSpectrum synthesizer was competitive.

The analysis of the results of small-dimensionality examples, i.e., sets of functions with a small number n of arguments and a small number k of conjunctions, shows that the preliminary optimization is not required for such examples, and the synthesizer builds "good" circuits for them both by area and by delay.

Programs DEC_FT, DEC_HIE, and DEC_TRIV for the decomposition of the matrix forms gives good results for examples of small and average dimensionality; therefore, we can apply the combined approach for the decomposition fulfilling initially the "large-block" decomposition using programs SEPT_BDD and DECU_BDD, while the decomposition of comparatively small blocks can be achieved using the decomposition programs of the matrix forms.

The analysis of Tables 7 and 9 shows that the design of circuits with a smaller delay is also provided by the TIE_BDD program of the global BDD minimization (21 examples) and the ESPRESSO program of the joint DNF minimization. The fulfillment of the synthesis without optimization also allows us to design small-delay circuits. It is also possible to design small-delay circuits after the decomposition; however, the circuit delay increases as a rule due to the decomposition.

CONCLUSIONS

Our experimental comparison of the efficiency of optimization programs and analysis of the experimen-

Table 8. Area of circuits

Circuit name	Experiment 2					Experiment 3				
	SEPT_BDD	DECU_BDD	DEC_FT	DEC_HIE	DEC_TRIV	SEPT_BDD	DECU_BDD	DEC_FT	DEC_HIE	DEC_TRIV
ADD6	23403	65833	52837	36571	46950	18414*	32587	24502	21689	29658
ADDM4	162478	140772	126538	228306	241279	141754	97047	86747*	212034	200584
ADR4	11924	14720	9726	19318	22571	10931	13180	7661*	11740	11740
ALU1	7109*	31254	61944	60074	124015	7109*	31471	65107	56391	102984
B12	28285	42977	77183	39914	50555	25194	34886	86412	24898	34713
B2	404896	292152	269425	365490	1172855	370752	268738	259682	259766	380048
B9	29714	104151	212230	91601	183738	30338	86557	168287	80246	147279
BR1	32515	25947*	42559	33173	33402	32464	31449	39880	32978	38190
BR2	24647	21957	35333	23218	33809	26868	24435	35729	25969	32135
CLPL	5563	2929*	12198	2929*	5418	7806	2929*	12572	2929*	7550
CO14	12416	11545*	12416	12125	12125	13353	12895	13353	11779	11779
DC2	36582	33117	45890	43742	67819	33982	25132	45644	43502	58769
DIST	96339	81624	87327	91618	105138	92572	69956	67501*	89793	98359
EX7	29714	90898	212230	91601	183738	30338	76792	168287	80246	147279
F51M	30394	31706	24820	–	–	23637*	31812	26131	–	–
gary	109162	139612	176925	190579	274620	122258	155861	200517	170251	222045
IN0	109162	148629	137525	198408	244315	122258	147864	170759	177868	194513
IN1	404896	352645	269425	368085	1087509	370752	353962	259682	243695	339956
IN2	148796	123915	176629	420693	1878942	128586	109535	149550	257238	872528
INTB	369418	629837	759589	422389	651242	285344	377342	563184	337902	453459
LIFE	16355	19709	28185	19100	15976	13961*	16617	18353	15574	15719
LOG8MOD	34563	28558	33034	30439	31097	32933	29741	34959	29602	28664
M1	28090	22794	21963	22220	22800	20004	18554*	23330	20825	22839
M181	28770	43200	87623	41275	47224	24714	35600	87796	24379	35645
M2	58975	62089	63701	65498	59912	54851	60515	55448	59438	52681*
M3	87751	76820	80570	78276	71837	75944	78332	75358	59818*	60236
M4	129378	122129	132731	146263	150359	127486	83081	82232*	129138	112108
MAX1024	188972	226665	189603	236637	292877	182031	146564	142050	237513	248935
MAX46	36555	36795	43948	38982	42793	35980*	37587	41621	41376	42179
MAX512	98169	103699	101695	122509	162456	98855	83761	71446*	120801	136063
MLP4	94977	102521	88700	–	–	81953	84197	71854*	–	–
MP2D	18955	33960	101941	35026	53914	19011	20819	101997	36125	44021

Table 8. (Contd.)

Circuit name	Experiment 2					Experiment 3				
	SEPT_BDD	DECU_BDD	DEC_FT	DEC_HIE	DEC_TRIV	SEPT_BDD	DECU_BDD	DEC_FT	DEC_HIE	DEC_TRIV
NEWAPLA	17477	11852	32810	13956	12957	13292	13760	31109	14078	13844
NEWAPLA1	7594	7332	8961	11930	12639	6869*	8119	9062	11300	12030
NEWAPLA2	5569	5569	8337	5569	5569	4944*	4944*	8030	5569	5569
NEWBYTE	5569	5569	5128*	–	–	5569	5569	5569	–	–
NEWCOND	17393	18314	34401	18604	43251	14541	14687	35321	15657	33452
NEWCPLA1	33184	33201	29708	29719	31114	28598	32431	34256	24831*	29585
NEWCPLA2	25093	17019	18459	17276	18079	27671	22582	18141	17778	18252
NEWILL	5122	5396	8588	7656	7293	5028*	5134	7985	7840	7723
NEWTAG	2003	2265	3409	2126	2126	2126	2126	3036	2126	2126
NEWTPLA	15418	28882	35154	19619	20825	16863	25919	33681	19156	19245
NEWTPLA1	3694	3917	5932	4453	3856	3342*	3577	3510	3566	3566
NEWTPLA2	8911	10184	9045	8454	9519	9296	10970	9068	6808*	12170
P82	30640	22331	22114	–	–	32213	22270	19971*	–	–
RADD	11450	19831	12616	16160	16333	10083	14614	9157	10496	10675
RD53	7410	5558	9631	5574	5574	7912	7399	7483	5312*	5312*
RD73	19552	22722	19519	19541	19541	13766	13046	12734*	13046	13046
ROOT	43641	46766	34150	40209	55504	39172	28692	26717*	42241	43608
RYY6	3181	3080*	5491	4615	4615	3197	3298	5814	5563	6021
SEX	17047	16980	24206	15384	14938	16539	16762	25065	14106	18805
soar	183018	4034982	–	–	–	188063	2268666	–	–	–
SQN	36700	25445	29423	15685	10702*	36298	24887	23124	12990	12053
SQR6	43228	33156	30316	–	–	40550	30099	27069*	–	–
SYM10	23358	28860	26081	21583	28899	24825	18966*	24279	20088	23548
T3	20931	23012	56966	26845	22649	20055	22387	52965	26505	23866
TIAL	409483	658507	1036820	449313	813062	350441	369708	690358	362109	502853
vtx1	23118	62552	404818	–	–	24011	66145	199580	–	–
x9dn	23380	109586	279017	–	–	23899	89224	169548	–	–
Z4	10066	6986*	23933	7516	7516	9519	7405	6992	7405	7405
Z5XP1	43825	51431	43172	–	–	40589	30690	26862	–	–
Z9SYM	29635	20082	21344	20434	23910	14614*	16333	17544	17767	20367
Number of best solutions	1	5	1	1	1	10	4	10	5	2

Table 9. Delay of circuits (ns)

Circuit name	Experiment 2					Experiment 3				
	SEPT_BDD	DECU_BDD	DEC_FT	DEC_HIE	DEC_TRIV	SEPT_BDD	DECU_BDD	DEC_FT	DEC_HIE	DEC_TRIV
ADD6	4.62	9.93	8.67	10.54	11.61	4.27*	9.80	8.62	8.01	8.10
ADDM4	10.06	11.23	7.07	12.57	14.74	8.63	9.32	6.42*	13.74	12.45
ADR4	2.94	4.28	2.49*	5.66	5.99	3.53	3.84	3.90	3.98	3.98
ALU1	1.11*	6.97	8.65	9.32	11.90	1.11*	7.89	7.44	9.02	11.97
B12	5.31	7.18	10.09	6.23	7.84	5.78	8.23	9.38	5.76	9.81
B2	14.24	14.11	18.54	14.32	19.32	10.51	12.85	13.78	12.76	15.55
B9	4.12	13.45	13.67	9.67	13.40	4.61	9.00	12.45	11.66	12.06
BR1	5.73	6.46	8.18	7.75	8.68	4.78*	7.92	8.16	7.11	8.21
BR2	7.34	7.52*	8.90	7.24	8.49	4.78*	6.95	6.79	6.52	10.16
CLPL	3.65	3.44*	5.23	3.44*	4.51	3.75	3.44*	4.79	3.44*	4.52
CO14	7.20	5.92	7.20	8.06	8.06	5.80	4.84	5.77	6.39	6.39
DC2	6.03	4.92	6.14	8.00	9.05	5.30	4.44	5.94	6.96	8.44
DIST	8.43	7.30	5.79	10.64	10.96	8.44	6.75	5.39*	9.08	9.97
EX7	4.12	10.88	13.67	9.67	13.40	4.61	7.98	12.45	11.66	12.06
F51M	4.79	3.36*	4.23	–	–	4.94	6.30	5.49	–	–
gary	7.89	12.27	12.52	11.12	14.25	7.97	10.03	11.99	13.37	14.61
IN0	7.89	12.25	13.30	11.29	13.69	7.97	12.70	11.50	14.47	14.08
IN1	14.24	16.10	18.54	16.23	18.12	10.51	13.85	13.78	13.63	16.65
IN2	10.78	11.63	14.42	17.26	24.40	7.87	8.96	13.60	16.12	21.53
INTB	13.01	17.15	17.84	16.23	20.81	12.84	13.43	12.35	15.25	14.82
LIFE	6.17	6.04	7.38	6.63	5.68	6.21	5.13	7.38	5.24	6.95
LOG8MOD	5.80	4.14	5.94	5.45	4.71	4.89	3.59*	8.47	7.18	4.62
M1	5.00	4.48	5.47	5.95	6.17	4.10	3.41*	4.87	5.94	5.02
M181	4.98	7.18	10.06	7.14	7.30	4.92	7.01	10.08	5.74	8.80
M2	7.55	7.18	7.38	8.79	7.86	6.31	5.65	5.07*	6.00	7.32
M3	8.09	7.17	9.11	6.79	8.44	6.52	6.11	8.10	5.58	6.35
M4	9.45	8.51	7.28	12.00	13.38	8.41	5.30*	5.88	9.91	11.41
MAX1024	11.02	9.10	8.11	12.66	14.58	10.49	7.46	6.69*	12.78	14.18
MAX46	8.28	5.66	8.22	8.24	8.22	6.43	5.49	8.62	9.19	8.76
MAX512	7.95	8.55	6.27	10.16	12.63	8.22	8.58	6.80	10.28	11.38
MLP4	10.05	9.77	6.96	–	–	7.80	7.09	5.72	–	–
MP2D	3.85*	5.91	10.41	8.93	10.79	4.19	4.19	9.98	12.10	9.66

Table 9. (Contd.)

Circuit name	Experiment 2					Experiment 3				
	SEPT_BDD	DECU_BDD	DEC_FT	DEC_HIE	DEC_TRIV	SEPT_BDD	DECU_BDD	DEC_FT	DEC_HIE	DEC_TRIV
NEWAPLA	5.19	3.61	6.10	6.02	4.23	4.93	4.03	6.47	5.13	4.09
NEWAPLA1	3.26	2.88	3.60	5.70	5.96	3.44	3.44	3.81	5.98	5.50
NEWAPLA2	2.79	2.68	4.11	2.89	2.89	2.45*	2.45*	3.39	2.79	2.79
NEWBYTE	2.52	2.57	1.31*	–	–	2.52	2.57	2.52	–	–
NEWCOND	7.13	4.83	8.36	6.23	9.52	5.49	5.29	7.13	4.94	6.62
NEWCPLA1	5.52	6.47	6.03	5.77	6.54	5.10	6.17	9.03	5.39	6.83
NEWCPLA2	5.06	3.98	4.03	6.02	4.22	4.30	5.09	4.31	4.37	4.53
NEWILL	3.61	3.56	4.49	4.79	4.93	3.08	2.68*	5.00	4.71	5.45
NEWTAG	1.80	2.00	2.83	1.90	1.90	1.90	1.90	2.60	1.90	1.90
NEWTPLA	4.56	6.95	8.57	6.05	7.65	4.12	5.66	8.18	6.05	6.40
NEWTPLA1	2.08*	2.96	3.61	3.29	2.39	2.26	3.16	2.81	2.30	2.30
NEWTPLA2	4.43	4.64	5.56	4.68	5.72	4.51	4.56	3.78	3.83	4.06
P82	4.77	3.63	3.87	–	–	3.69	3.29	2.85*	–	–
RADD	4.14	5.30	5.23	4.85	5.89	3.03*	5.13	4.85	3.43	4.25
RD53	4.21	3.32	3.76	3.52	3.52	4.08	2.88	3.22	4.43	4.43
RD73	6.67	5.91	5.42	6.13	6.13	4.92	4.94	5.17	4.94	4.94
ROOT	7.10	5.68	4.43	8.41	9.98	6.93	4.08*	4.81	10.43	11.54
RYY6	2.38	2.41	3.47	2.69	2.69	2.50	2.47	3.47	3.08	4.35
SEX	3.70	3.82	6.15	2.55	3.06	3.47	3.38	6.76	2.92	5.15
soar	8.73	27.93	–	–	–	5.63*	21.53	–	–	–
SQN	6.17	5.16	4.12	4.14	6.15	6.29	4.89	4.71	3.73	4.15
SQR6	6.67	3.87	4.26	–	–	4.88	4.65	3.40*	–	–
SYM10	7.27	9.65	6.32	6.47	7.41	8.48	6.20	9.10	7.37	7.43
T3	4.90	5.82	8.88	8.50	6.11	6.11	5.42	9.94	5.54	5.72
TIAL	15.02	17.19	19.64	15.13	19.02	11.14	16.55	15.49	14.63	15.26
vtx1	4.89	9.85	17.91	–	–	6.73	10.44	13.68	–	–
x9dn	5.05	12.44	14.78	–	–	5.23	11.26	13.12	–	–
Z4	3.27	4.50	4.07	4.73	4.73	3.04*	5.78	3.09	5.78	5.78
Z5XP1	5.59	4.97	5.18	–	–	6.21	5.17	3.98	–	–
Z9SYM	6.73	6.70	6.19	5.19*	5.66	6.04	5.88	8.01	6.25	9.96
Number of best solutions	3	2	2	2	0	8	7	6	1	0

tal results allowed us to rank the optimization programs by area and delay criteria and reveal the regions of their preferential use. The results of this study are used in the expert system of logical design [1] to form efficient combined design routes of functional units of custom digital VLSIs.

REFERENCES

1. Bibilo, P.N. and Romanov, V.I., *Logicheskoe proektirovanie diskretnykh ustroistv s ispol'zovaniem produktsionno-freimovoi modeli predstavleniya znanii* (Logical Design of Discrete Devices with Product-Frame Model of Knowledge), Minsk: Belarus. navuka, 2011.
2. Zakrevskii, A.D., *Logicheskii sintez kaskadnykh skhem* (Logical Synthesis of Cascade Circuits), Moscow: Nauka, 1981.
3. Brayton, K.R., Hactel, G.D., McMullen, C.T., and Sangiovanni-Vincentelli, A.L., *Logic Minimization Algorithm for VLSI Synthesis,* Boston: Kluwer Academic, 1984.
4. Toropov, N.R., Minimization of systems of Boolean functions in DNF, in *Logicheskoe proektirovanie* (Logic Design), Minsk: Inst. Tekh. kibernetiki NAN Belarusi, 1999, no. 4, pp. 4–19.
5. Zakrevskii, A.D., Pottosin, Yu.V., and Cheremisinova, L.D., *Logicheskie osnovy proektirovaniya diskretnykh ustroistv* (Logical Fundamentals for Discrete Devices Design), Moscow: Fizmatlit, 2007.
6. Bryant, R.E. and Meinel, C., Ordered binary decision diagrams, in *Logic Synthesis and Verification*, Hassoun, S., Sasao, T., and Brayton, R.K., Dordrecht: Kluwer Academic, 2002, pp. 285–307.
7. Knuth, D.E., *The Art of Computer Programming*, vol. 4A: *Combinatorial Algorithms*, Upper Saddle River, New Jersey: Addison-Wesley, 2011, pt. 1.
8. Bibilo, P.N. and Leonchik, P.V., Decomposition of systems of Boolean functions determined by binary decision diagrams, *J. Comput. Syst. Sci. Int.*, 2011, vol. 50, no. 4, pp. 609–624.
9. Pottosin, Yu.V. and Shestakov, E.A., *Tablichnye metody dekompozitsii sistem polnost'yu opredelennykh bulevykh funktsii* (Tabular Methods for Decomposition of Systems of Completely Specified Boolean Functions), Minsk: Belarus. navuka, 2006.
10. Sadnikov, D.V., Development of tabular method for decomposition of system of completely specified Boolean functions, *Informatika*, 2005, no. 2, pp. 43–53.
11. Porwik, P. and Stankovic, R.S., Dedicated spectral method of Boolean function decomposition, *Int. J. Appl. Math. Comput. Sci.*, 2006, vol. 16, no. 1, pp. 271–278.
12. Bibilo, P.N., *Dekompozitsiya bulevykh funktsii na osnove resheniya logicheskikh uravnenii* (Decomposition of Boolean Functions Based on Solving Logical Equations), Minsk: Belarus. navuka, 2009.
13. Gol'dberg, E.I., Decomposition of PLM, *Preprint of Inst. of Tech. Cibernetics, Academy of Sciences BSSR*, Minsk, 1991, no. 6.
14. Bibilo, P.N. and Leonchik, P.V., The algorithm of the construction of a binary decision diagram for the system of completely defined Boolean functions, *Upravl. Sist. Mashiny*, 2009, no. 6, pp. 42–49.
15. Tagavi Afshord, S.Kh. and Pottosin, Yu.V., Study of decomposition properties of Boolean function systems, *Informatika*, 2013, no. 4, pp. 94–103.
16. Bibilo, P.N., *Sistemy proektirovaniya integral'nykh skhem na osnove yazyka VHDL. StateCAD, ModelSim, LeonardoSpectrum* (Systems of Design of Integrated Circuits Based on VHDL, StateCAD, ModelSim, LeonardoSpectrum), Moscow: SOLON-Press, 2005.

Translated by N. Korovin