numerical values of the analytical calculations obtained during the research.

References

1. Bondarev, D. I. Modelling of group flights of unmanned aerial vehicles using graph theory [Text] / D. I. Bondarev, D. P. Kucherov, T. F. Shmelova // Scientific Works of Kharkiv National Air Force University. – 2016. – No. 3 (48). – P. 61–66.
2. Aldoshin, D. V. Spatial planning routes for UAVs using search on graphs [Electronic resource] / D. V. Aldoshin // Youth Science and Technology Herald of the Bauman MSTU. – 2013. – No. 2. – Available at: \www/URL: http://sntbul.bmstu.ru/doc/551948.html
3. Gurnik, A. Use of intellectual sensor technics for monitoring and search-and-rescue operations [Electronic resource] / A. Gurnik, S. Valuiskii // Eastern-European Journal of Enterprise Technologies. – 2013. – Vol. 3, No. 9 (63). – P. 27–32. – Available at: \www/URL: http://journals.uran.ua/eejet/article/view/14845
4. Podlipian, P. E. Kombinirovannyi algoritm resheniia transportnoi zadachi v sisteme planirovaniia poleta gruppy bespilotnyh letatel'nyh apparatov [Text] / P. E. Podlipian, N. A. Maksimov // Tezisy dokladov 9 Mezhdunarodnoi konferentsii «Aviatsiia i kosmonavtika-2010». – St. Petersburg: Masterskaia pechati, 2010. – P. 138–139.
5. Bopardikar, S. D. Dynamic Vehicle Routing for Translating Demands: Stability Analysis and Receding-Horizon Policies [Text] / S. D. Bopardikar, S. L. Smith, F. Bullo, J. P. Hespanha // IEEE Transactions on Automatic Control. – 2010. – Vol. 55, No. 11. – P. 2554–2569. doi:10.1109/tac.2010.2049278
6. Sariel-Talay, S. Multiple Traveling Robot Problem: A Solution Based on Dynamic Task Selection and Robust Execution [Text] / S. Sariel-Talay, T. R. Balch, N. Erdogan // IEEE/ASME Transactions on Mechatronics. – 2009. – Vol. 14, No. 2. – P. 198–206. doi:10.1109/tmech.2009.2014157
7. Gao, P.-A. Evolutionary Computation Approach to Decentralized Multi-robot Task Allocation [Text] / P.-A. Gao, Z.-X. Cai, L.-L. Yu // 2009 Fifth International Conference on Natural Computation. – IEEE, 2009. – P. 415–419. doi:10.1109/icnc.2009.123
8. Pehlivanoglu, Y. V. A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV [Text] / Y. V. Pehlivanoglu // Aerospace Science and Technology. – 2012. – Vol. 16, No. 1. – P. 47–55. doi:10.1016/j.ast.2011.02.006
9. Rahimi-Vahed, A. A path relinking algorithm for a multi-depot periodic vehicle routing problem [Text] / A. Rahimi-Vahed, T. G. Crainic, M. Gendreau, W. Rei // Journal of Heuristics. – 2013. – Vol. 19, No. 3. – P. 497–524. doi:10.1007/s10732-013-9221-2
10. Murray, C. C. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery [Text] / C. C. Murray, A. G. Chu // Transportation Research Part C: Emerging Technologies. – 2015. – Vol. 54. – P. 86–109. doi:10.1016/j.trc.2015.03.005
11. Hawary, A. F. Routeing Strategy for Coverage Path Planning in Agricultural Monitoring Activity using UAV [Text] / A. F. Hawary, A. J. Chipperfield // Eminent Association of Pioneers (EAP) August 22-24, 2016 Kuala Lumpur (Malaysia). – Eminent Association of Pioneers (EAP), 2016. – P. 68–74. doi:10.17758/eap.eap816005
12. Johnson, D. S. The Traveling Salesman Problem: A Case Study in Local Optimization [Text] / D. S. Johnson, L. A. McGeoch. – November 20, 1995. – Available at: \www/URL: http://www.uniriotec.br/~adriana/files/TSPchapter.pdf

ПЛАНИРОВАНИЕ МАРШРУТОВ ПОЛЕТА БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ ПУТЕМ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА

Рассмотрены методы решения задачи коммивояжера для планирования маршрутов полета беспилотных летательных аппаратов и проанализированы результаты работы. Показано, что метод усредненных коэффициентов решает задачу оптимально по критерию расстояния, использование которого обеспечивает минимальные эксплуатационные расходы полета беспилотных летательных аппаратов и дает существенный выигрыш (5–10 %) в сравнении с другими методами.

Ключевые слова: задача коммивояжера, минимальный маршрут, планирование маршрутов, беспилотные летательные аппараты.

Vorotnikov Vladimir, *Doctor of Technical Science, Associate Professor, Department of Computer Integrated Technologies and Cyber Security, Korolyov Zhytomyr Military Institute, Ukraine, e-mail: vvvorotnik@ukr.net, ORCID: http://orcid.org/0000-0001-8584-3901*

Gumenyuk Igor, *Adjunct, Department of Computer Integrated Technologies and Cyber Security, Korolyov Zhytomyr Military Institute, Ukraine, e-mail: ig_gum@ukr.net, ORCID: http://orcid.org/0000-0001-5853-3238*

Pozdniakov Pavel, *PhD, Chief of Research Laboratory, Research Centre, Korolyov Zhytomyr Military Institute, Ukraine, e-mail: pozdner86@gmail.com, ORCID: http://orcid.org/0000-0003-4188-9486*

UDC 681.325
DOI: 10.15587/2312-8372.2017.108532

Riznyk V.,
Solomko M.

MINIMIZATION OF BOOLEAN FUNCTIONS BY COMBINATORIAL METHOD

Розглянуто поширення принципу мінімізації за допомогою алгебричних перетворень на метод мінімізації з використанням комбінаторної блок-схеми з повтореннями. Математичний апарат блок-схеми з повтореннями дає більше інформації стосовно ортогональності, суміжності, однозначності блоків комбінаторної системи, якою є власне таблиця істинності заданої функції, тому застосування такої системи мінімізації функції є більш ефективним.

Ключові слова: *булева функція, метод мінімізації, мінімізація логічної функції, блок-схема з повтореннями, мінтерм.*

1. Introduction

The problems and shortcomings of the known methods for minimizing Boolean functions are associated with a rapid growth in the amount of computation, which results in an increase in the number of computational operations, and, consequently, in the increase in the number of variables of the logical function.

ISSN 2226-3780

The following methods for minimizing Boolean functions are known [1–5]:

– Blake-Poretsky method;
– Nelson method;
– Karnaugh map method;
– Quine method;
– Quine-McCluskey method;
– Veitch diagram method;
– method of algebraic transformations;
– Petrik method;
– Roth method;
– a method of minimizing functions in bases YES-NO and OR-NOT (Schaeffer and Pierce basis);
– method of undetermined coefficients;
– hypercube method;
– functional decomposition method;
– heuristic algorithm for Espresso minimization.

The Boolean function $f(x_1,...,x_n)$ that describes the operation of a logical device can be realized with the help of a disjunctive normal form (DNF), which in this case describe the scheme of the corresponding logical device. The problem of minimizing DNF is one of the multiextremal logical-combinatorial problems and is reduced to optimal reduction of the number of logical elements of the gate system without loss of its functionality.

Functions with a large number of variables (more than 16 variables) can be minimized only in a certain sense, not guaranteeing the achievement of the optimal solution with the help of the heuristic Espresso algorithm, which today is documented by the world standard [6].

The result of minimizing the Boolean function depends on the speed of computing device, its reliability and energy savings. Since Espresso minimization algorithm does not guarantee optimal minimization of Boolean function with increasing number of variables, the search for new minimization methods remains relevant. Carrying out the minimization of the logical function is one of the central and practically important problems that arises during development of the computing attachments.

2. The object of research and its technological audit

The object of research is the problem of minimizing the Boolean function by a combinatorial method – a block-diagram with repetition. Since the block-diagram with repetition is actually the truth table of this function, it allows concentrating the minimization principle within the function calculation protocol. The tabular organization of the mathematical apparatus of the repetition block-diagrams also makes it possible to obtain more information about the orthogonality, contiguity, uniqueness of truth table blocks (combinatorial system). Equivalent transformations by graphic images, in their properties have a large information capacity, capable of effectively replacing verbal procedures of algebraic transformations, in particular using the library of submatrices. This efficiency of the combinatorial method makes it possible to carry out manual minimization of 4, 5-bit Boolean functions without difficulty.

The graphical properties of the combinatorial method make it possible to obtain a minimal function by several variants of the search, reduces the search, the search for the function becomes more definite, and, consequently, the complexity of the minimization algorithm decreases.

The complexity of the search algorithm by the combinatorial method is $O(n)$ and is linear – the execution time of the algorithm with increasing bit depth of the function n grows linearly.

Combinatorial method allows automation by its protocol and is able to support aggregated minimization systems by combining with other apparatus of other methods for minimizing Boolean functions.

The disadvantages of the combinatorial method of manual minimization are associated with the growth of the number of variables (more than seven or eight) of the logical function. Minimizing a function with a large number of variables requires updating the library of submatrices on which the figurative calculus of the combinatorial method is based.

3. The aim and objectives of research

The aim of research is development of a method for minimizing a logical function, using a combinatorial device of a block-diagram with repetition and establishing the properties of such method.

To achieve this aim, it is necessary to solve the following tasks:

1. To establish the adequacy of using a combinatorial block-diagram device with repetition to create a method for minimizing the Boolean function.

2. To determine the properties of the apparatus of a combinatorial method for minimizing Boolean functions, in particular, to represent the apparatus of figurative calculus for equivalent transformations of conjunctors.

3. To determine the verification of the combinatorial method and obtain an estimate of the complexity of the algorithm for finding the minimal function by a combinatorial method.

4. To conduct a comparable analysis of the performance and quality of minimization of Boolean functions obtained by the combinatorial method, with examples of minimizing the function by other methods.

4. Research of existing solutions of the problem

In [7], the conditions of logical minimization of the Boolean function represented in DNF are considered. If the function satisfies the following conditions, then to simplify it, the classical Quine-McCluskey minimization algorithm is applied, which allows automation. It is noted that the number of function variables for the program code is limited by the computer's memory.

In [8], generalized rules for simplifying the conjunctors in a polynomial set-theoretic format are considered, based on the proposed theorem for various initial conditions for the transformation of a pair of conjunctors, the gemming distance between which can be arbitrary. These rules can be useful for minimizing in the polynomial set-theoretic format arbitrary logical functions of n variables. The effectiveness of the proposed rules is demonstrated by examples of minimization of functions borrowed from the work of well-known authors for the purpose of comparison. Given the comparative examples, the proposed rules give grounds for confirming the expediency of applying them in the procedures for minimizing any logical function of n variables in polynomial form.

In [9], a simple and systematic method for minimizing a logical function is proposed. The method consists in reducing the truth table from N variables N–1, N – from 1 to N–2, and so on in a sequence until all variables are exhausted with built-in all possible simplifications, after each reduction. The obtained resultant expression for F will be minimal.

In [10], an algorithm and program for minimizing combinational logic functions up to 20 variables is presented, but the number of variables is limited only by the memory of the computer system. The algorithm is based on the sequential clustering of terms, beginning with the grouping of terms with one change. The clustering algorithm ends when the variables can no longer be grouped. This algorithm is analogous to the Quine-McClusky algorithm, but it is more simplistic, since it eliminates a number of actions necessary for implementation of the Quine-McCluskey algorithm

In [11], a discussion is presented on the role of the autosymmetry degree of variables in a Boolean function and why it deserves attention on minimizing a logical function. The regularity of the variables of a Boolean function can be expressed by the degree of autosymmetry, which in the end gives a new tool for effective minimization.

In [12], the method of logical-minimized image compression, which depends on the logical function, is demonstrated. The minimization process treats neighboring pixels of the image as separate minterms representing a logic function and compresses 24-bit color images using the function minimization procedure. The compression ratio of such method is on average 25 % larger than the existing methods of image compression.

The paper [13] demonstrates how to increase the efficiency of minimizing a logical function by applying M-terms. It is noted that implementation of the method is possible for any number of variables.

Work [14] demonstrates the use of a genetic algorithm for selecting side objects of the procedure for minimizing a logical function using the Karnaugh map.

A new heuristic algorithm is proposed in [15] for maximum minimization of Boolean functions. Graphic data is used to implement the proposed algorithm. There are also some conditions for achieving the maximum level of minimization of the Boolean function.

In [16], the optimal simplification of Boolean functions by means of Karnaugh maps is considered, using the object-oriented minimization algorithm. Analysis of the performance of the proposed algorithm is presented.

In contrast to [7–16], in this paper, the object of solving the problem of minimizing a Boolean function is a combinatorial block-diagram with repetition, which allows to concentrate the minimization principle within the truth table of a given function. The peculiarities of the combinatorial method consist in greater informativeness of the process of solving the problem in comparison with the algebraic method of minimizing the function, due to tabular organization and the introduction of figurative calculus apparatus. In this regard, the procedure for minimizing the function becomes more tangible, and, therefore, more reliable, simplified. Combinatorial method allows its automation and is able to support aggregated minimization systems by combining with other apparatus of other methods for minimizing Boolean functions.

5. Methods of research

5.1. Minimization of Boolean functions by means of an acyclic graph. To minimize the function that simulates the operation of a logical device is possibly using the method in which an acyclic graph is used [17]. To do this, two arcs are drawn from the initial vertex of G (the root of the graph): the left arc corresponds to the value of the variable x_j, and the right – variable $\bar{x}_j$. From each vertex of the first and subsequent levels, two arcs are drawn again according to the same rule, where each vertex forms two child vertices of the lowest level. Thus, from each vertex of the i-th level, two arcs are drawn, where the left arcs correspond to the direct value of the variable, and the right ones to the inverted one. The number of such entries is equal to the number of variables entering the created minterm (and, consequently, the number of levels of the acyclic graph) (Fig. 1).

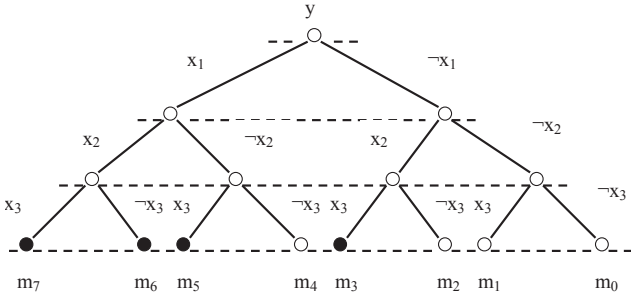


Fig. 1. The acyclic graph G for a function of three variables Y

It can be seen from Fig. 1 that each path in this graph from a finite vertex (in this case from the third level) to the root of the graph (to 0-level) identifies a certain minterm: $m_0 = x_1 x_2 x_3$, $m_1 = x_1 x_2 x_3, \ldots$, $m_7 = x_1 x_2 x_3$.

In the general case, each logical function can be represented by an acyclic graph of the form:

$$G = \{M, X\},$$

where $M = \{m_1, m_2, \ldots, m_k\}$, $X = \left\{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots, x_n, \overline{x_n}\right\}$.

The acyclic graph G with n levels is to be decomposed into components $G_1, G_2, \ldots, G_i \ldots, G_n$ in order to identify the possibility of gluing the i-th variable according to the dependence:

$$G_i x_i \vee G_i \overline{x_i} = G_i.$$

The minterms, which values are equal to one on the graph G, are denoted by black circles (Fig. 1).

The division of the acyclic graph G must begin from the terminal vertices to the root of the graph. For example, for the graph in Fig. 1:

$$G_3 = \left\{M_3, \left(x_3 \vee \overline{x_3}\right)\right\},$$

$$G_2 = \left\{M_2, \left(x_2 \vee \overline{x_2}\right)\right\}, \; M_2 = \left\{m_0^2, m_1^2, m_2^2, m_3^2\right\}, \qquad (1)$$

$$G_1 = \left\{M_1, \left(x_1 \vee \overline{x_1}\right)\right\}, \; M_1 = \left\{m_0^1, m_1^1\right\}.$$

ISSN 2226-3780

For the n-th variable:

$$G_n = \left\{M_n, \left(x_n \vee \overline{x_n}\right)\right\},\ M_n = \left\{m_0^n, m_1^n, ..., m_n^n\right\}.$$

Analyzing the system of equations (1) obtained by dividing the acyclic graph into three ($n=3$) components, it is easy to see that the process of minimizing the perfect disjunctive normal form (PDNF) of a Boolean function reduces to passing the path from a finite vertex of the third level to the root of the graph. PDNF minimization of a Boolean function is accomplished by gluing variables at appropriate levels.

The acyclic graph G for a function:

$$Y = x_1x_2x_3 + \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3}$$

is shown in Fig. 1. Black minterm is shaded, from which the function Y actually consists. As a result of the separation of graph G, at the 3-rd level the gluing procedure will pass between the variables of minterms $m_7^3 - m_8^3$ – get x_1x_2. For variable of minterm m_5 gluing will take place at the 2-nd level – get x_1x_3. For variable of minterm m_2 gluing will take place at the 1-st level – get x_2x_3. In the end we find a minimized function:

$$Y = x_1x_2 \vee x_1x_3 \vee x_2x_3. \qquad (2)$$

The function (2) satisfies the given truth table (Table 1).

Table 1

The truth table of a logical function $Y = x_1x_2x_3 + \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3}$

No.	x_1	x_2	x_3	Y	No.	x_1	x_2	x_3	Y
0	0	0	0	**0**	4	1	0	0	**0**
1	0	0	1	**0**	5	1	0	1	**1**
2	0	1	0	**0**	6	1	1	0	**1**
3	0	1	1	**1**	7	1	1	1	**1**

5.2. Combinatorial method for minimization of Boolean functions. The concept of Boolean functions and DNF are closely connected with many concepts of combinatorial analysis, in particular, with the notion of covering. Let $C=(X1, ..., Xn)$ be some family of subsets of X, and let $Y \subseteq X$. Then Y is a covering for C if the condition $X_i \cap Y \neq \varnothing$ holds for any X_i with C. Covering of Y is said to be reduced for C if any of its proper subsets is not a cover for C. The set of all reduced covering for C is denoted by $P(C)$.

With combinatorial analysis, it is known that a graph can be represented in the form of an appropriate block-diagram [18]. Hence it follows that the acyclic graph for a logical function can be analyzed from the truth table in the form of a block-diagram with a repetition (Table 1). Consequently, the principle of minimization with the help of an acyclic graph can be extended to a minimization method using combinatorial block-diagrams (Table 2).

It should be noted that, unlike the PDNF minimization by the acyclic graph method, where the minimization procedure reduces to the passage of the path from the final vertex of the lower level to the root of the graph in order to glue variables at the appropriate levels, using the block-diagram with repetition, the minimization process in the part of the gluing of the variables reduces to search for blocks with the same variables in the corresponding bits, except for one variable. Given the tabular organization of the combinatorial method, this makes it possible to improve the search efficiency of the minimal function.

Table 2

Thesauri of minimization methods

No.	Thesaurus of minimization. With the help of an acyclic graph	Thesaurus of minimization. Using a block-diagram with repetition
1	Acyclic graph	Combinatorial system with repetition
2	Path (branch) in the graph	Block of variables of a combinatorial system
3	Variables	Variables
4	Laws of algebra of logic	Laws of algebra of logic
5	–	Figurative calculus

Minimization of the logical function by combinatorial method is carried out as follows. At the first stage, blocks (constituents) are identified with variables, and can be glued together. The next step is search for sets of pairs of blocks (implicants) with the possibility of minimizing them by replacing (gluing, absorbing) the variables in these pairs. The resulting sets of blocks are again minimized in a similar way, etc., until a deadlock DNF (DDNF) is obtained. Among the set of DDNFs there are also minimal functions (MDNF). At the last step, the minimized function is verified by applying the optimality criterion and the specified truth table.

The process of minimizing by a combinatorial method a logical function is shown in Fig. 1 looks like this.

$$Y = x_1x_2x_3 + \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3}.$$

1-st option:

$$Y = \begin{vmatrix} 1&1&1 \\ 0&1&1 \\ 1&0&1 \\ 1&1&0 \end{vmatrix} = \begin{vmatrix} &1&1 \\ &1&1 \\ 1&0&1 \\ 1&1&0 \end{vmatrix} = \begin{vmatrix} &&\\ &1&1 \\ 1&0&1 \\ 1&1&0 \end{vmatrix} = \begin{vmatrix} &1&1 \\ 1&&1 \\ 1&1& \end{vmatrix}.$$

In the first matrix, the variables are glued together in the 1-st and 2-nd blocks, in the second matrix, the variables are replaced in the 2-nd and 4-th blocks.

2-nd option:

$$Y = \begin{vmatrix} 1&1&1 \\ 0&1&1 \\ 1&0&1 \\ 1&1&0 \end{vmatrix} = \begin{vmatrix} 1&1& \\ 0&1&1 \\ 1&0&1 \\ 1&1& \end{vmatrix} = \begin{vmatrix} &&\\ 0&1&1 \\ 1&0&1 \\ 1&1& \end{vmatrix} = \begin{vmatrix} &1&1 \\ 1&&1 \\ 1&1& \end{vmatrix}.$$

In the first matrix, the variables are glued together in the first and fourth blocks, the second matrix replaces the variables in the second and third blocks.

As a result, the function for the two variants is minimized and has the form $Y = x_1x_2 \vee x_1x_3 \vee x_2x_3$ that coincides with (2).

In the general case, when minimized by a combinatorial method, the following rules of the algebra of logic are used:

– gluing of variables – $ab + a\bar{b} = a$;

– generalized gluing of the variables – $xy + \bar{x}z = xy + \bar{x}z + yz$;

– substitution of the variable – $a+\bar{a}b=a+b$;
– absorbing of the variable – $ab+a=a(b+1)=a$;
– idempotency of the variables – $a+a=a$, $aa=a$;
– addition of the variable – $a+\bar{a}=1$, $a\bar{a}=0$;
– repetition of the constant – $a+0=a$, $a\cdot 1=a$ and others.

Algebraic transformations of the combinatorial method for minimizing the Boolean function can be replaced by equivalent transformations by means of submatrices (graphical images). The procedure for gluing using submatrices can be illustrated as follows:

$$\bar{x}_1\bar{x}_2+\bar{x}_1x_2=\bar{x}_1(\bar{x}_2+x_2)=\bar{x}_1,$$

$$\begin{matrix}0&0\\0&1\end{matrix}\rightarrow\begin{matrix}0\\0\end{matrix}\rightarrow\begin{matrix}0\end{matrix},$$

$$x_1x_2+\bar{x}_1x_2+x_2(x_1+\bar{x}_1)=x_2,$$

$$\begin{matrix}1&1\\0&1\end{matrix}\rightarrow\begin{matrix}1\\1\end{matrix}\rightarrow\begin{matrix}1\end{matrix}.$$

Applying a graphic image, it is possible to illustrate other algebraic transformations.

Generalized gluing:

$$x_1x_2+x_1x_3+x_2\bar{x}_3=x_1x_3+x_2\bar{x}_3,$$

$$\begin{matrix}1&1&\\1&&1\\&1&0\end{matrix}\rightarrow\begin{matrix}1&&1\\&1&0\end{matrix},$$

$$x_1x_3+x_2\bar{x}_3=x_1x_3+x_2\bar{x}_3+x_1x_2,$$

$$\begin{matrix}&&1&1\\1&&1&\\&1&0&\end{matrix}\rightarrow\begin{matrix}1&&1\\&1&0\end{matrix}.$$

Substitution of the variable:

$$x_1x_2+x_1\bar{x}_2x_3=x_1(x_2+\bar{x}_2x_3)=$$
$$=x_1(x_2+x_3)=x_1x_2+x_1x_3,$$

$$\begin{matrix}1&1&\\1&0&1\end{matrix}\rightarrow\begin{matrix}1&1\\1&\end{matrix}.$$

Absorbing of the variable:

$$x_1x_3+x_1\bar{x}_2x_3=x_1x_3(1+\bar{x}_2)=x_1x_3,$$

$$\begin{matrix}1&&1&\\1&0&1&\end{matrix}\rightarrow\begin{matrix}1&1\\&\end{matrix}.$$

Idem potency of the variables:

$$x_1x_2+x_1x_2=x_1x_2,$$

$$\begin{matrix}1&1\\1&1\end{matrix}\rightarrow\begin{matrix}1&1\end{matrix}.$$

Since the graphic images give more information about the orthogonality, contiguity, uniqueness of the blocks of the combinatorial system, so using them when searching for objects for equivalent transformations, in the process of minimizing the logical function, is effective.

5.3. Minimization of 4-bit Boolean functions. The combinatorial method easily minimizes 4-bit Boolean functions.

Example 1. To minimize the logic function $F(x_1,x_2,x_3,x_4)$ by the combinatorial method given by the following truth table (Table 3) [17].

Table 3

The truth table of a logical function $F(x_1,x_2,x_3,x_4)$

No.	x_1	x_2	x_3	x_4	F	No.	x_1	x_2	x_3	x_4	F
0	0	0	0	0	**1**	8	1	0	0	0	**0**
1	0	0	0	1	**1**	9	1	0	0	1	**1**
2	0	0	1	0	**1**	10	1	0	1	0	**1**
3	0	0	1	1	**1**	11	1	0	1	1	**1**
4	0	1	0	0	**0**	12	1	1	0	0	**0**
5	0	1	0	1	**0**	13	1	1	0	1	**1**
6	0	1	1	0	**1**	14	1	1	1	0	**1**
7	0	1	1	1	**0**	15	1	1	1	1	**1**

Let's compose the perfect disjunctive normal form (PDNF) of the given function with respect to the blocks for which the function gets the value of unity, that is, for the sets 0, 1, 2, 3, 6, 9, 10, 11, 13, 14, 15.

$$F(x_1,x_2,x_3,x_4)=\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4+\bar{x}_1\bar{x}_2\bar{x}_3x_4+$$
$$+\bar{x}_1\bar{x}_2x_3\bar{x}_4+\bar{x}_1\bar{x}_2x_3x_4+\bar{x}_1x_2x_3\bar{x}_4+$$
$$+x_1\bar{x}_2\bar{x}_3x_4+x_1\bar{x}_2x_3\bar{x}_4+x_1\bar{x}_2x_3x_4+$$
$$+x_1x_2\bar{x}_3x_4+x_1x_2x_3\bar{x}_4+x_1x_2x_3x_4.$$

The first step is gluing of the constituent (first blocks).

The variables for the first gluing are in blocks 1 and 2, 3 and 4, 6 and 9, 7 and 8 and 10 and 11 (3). In (3), the first step is shown on the first three matrices.

$$F=\left|\begin{matrix}0&0&0&0\\0&0&0&1\\0&0&1&0\\0&0&1&1\\0&1&1&0\\1&0&0&1\\1&0&1&0\\1&0&1&1\\1&1&0&1\\1&1&1&0\\1&1&1&1\end{matrix}\right|=\left|\begin{matrix}0&0&0&\\0&0&0&\\0&0&1&\\0&0&1&\\0&1&1&0\\1&&0&1\\1&0&1&\\1&0&1&\\1&&0&1\\1&1&1&\\1&1&1&\end{matrix}\right|=\left|\begin{matrix}&&&\\&&&\\0&0&0&\\0&0&1&\\0&1&1&0\\&&&\\&&&\\1&0&1&\\1&&0&1\\&&&\\1&1&1&\end{matrix}\right|=\left|\begin{matrix}&&&\\&&&\\&&&\\&&&\\0&1&1&0\\&&&\\&&&\\1&&1&\\1&&0&1\\&&&\\1&1&1&\end{matrix}\right|=\left|\begin{matrix}&&&\\&&&\\&&&\\&&&\\0&1&1&0\\&&&\\&&&\\&&&\\1&&0&1\\&&&\\1&&1&\end{matrix}\right|=\left|\begin{matrix}&&&\\&&&\\&&&\\&&&\\&&&\\&&&\\&&&\\&&&\\1&&0&1\\&&&\\1&&1&\end{matrix}\right|\quad(3)$$

The second step is to glue together the implicants – to search for pairs of blocks that have matching variables in the corresponding bits, except for one variable, followed by replacement (gluing, absorbing) in these pairs.

Obviously, the substitution of variables for the 4-th, 5t-h and 9-th, 11-th blocks of the fourth matrix (3):

$$\bar{x}_1\bar{x}_2+\bar{x}_1x_2x_3\bar{x}_4=\bar{x}_1(\bar{x}_2+x_2x_3\bar{x}_4)=$$
$$=\bar{x}_1(\bar{x}_2+x_3\bar{x}_4)=\bar{x}_1\bar{x}_2+\bar{x}_1x_3\bar{x}_4,\quad(4)$$

ISSN 2226-3780

$$\begin{matrix}0&0\\0&1&1&0\end{matrix}\rightarrow\begin{matrix}0&0\\0&&1&0\end{matrix},$$

$$x_1\overline{x_3}x_4 + x_1x_3 = x_1(\overline{x_3}x_4 + x_3) = \\ = x_1(x_4 + x_3) = x_1x_4 + x_1x_3, \quad (5)$$

$$\begin{matrix}1&0&1\\1&&1\end{matrix}\rightarrow\begin{matrix}1&&1\\1&1\end{matrix}.$$

After the procedures (4) and (5) let's obtain an intermediate matrix:

$$=\begin{vmatrix}0&0&\\0&&1&0\\1&&&1\\1&&1&\end{vmatrix}=$$

Obviously, the substitution of variables for the 2-nd and 4-th blocks of the intermediate matrix:

$$\overline{x_1}x_3\overline{x_4} + x_1x_3 = x_3(\overline{x_1}\,\overline{x_4} + x_1) = \\ = x_3(\overline{x_4} + x_1) = x_1x_3 + x_3\overline{x_4}, \quad (6)$$

$$\begin{matrix}0&&1&0\\1&&1\end{matrix}\rightarrow\begin{matrix}&&1&0\\1&&1\end{matrix}.$$

After the procedure (6), let's get the last intermediate matrix:

$$=\begin{vmatrix}0&0&\\&&1&0\\1&&&1\\1&&1&\end{vmatrix}=$$

The generalized substitution of variables for the 2-nd, 3-rd and 4-th blocks of the last intermediate matrix is obvious:

$$x_3\overline{x_4} + x_1x_4 + x_1x_3 = x_3\overline{x_4} + x_1x_4.$$

$$\begin{matrix}&&1&0\\1&&&1\\1&&1&\end{matrix}\rightarrow\begin{matrix}&&1&0\\1&&&1,\end{matrix}$$

$$=\begin{vmatrix}0&0&\\&&1&0\\1&&&1\\&&&\end{vmatrix}.$$

As a result, let's get the minimum function:

$$F = \overline{x_1}\,\overline{x_2} + x_1x_4 + x_3\overline{x_4}. \quad (7)$$

The third step is the verification of the obtained minimized function (7) using the original truth table (Table 3).

The minimized logic function (7) satisfies the original truth table.

Table 4 shows the results of minimizing the function $F(x_1,x_2,x_3,x_4)$ using an acyclic graph [17] and the combinatorial method.

Table 4

Results of minimizing the function $F(x_1,x_2,x_3,x_4)$

Minimization by acyclic graph	Minimization by combinatorial method
$F = \overline{x_1}\,\overline{x_2} + x_1x_4 + x_1x_3\overline{x_4} + \overline{x_1}x_2x_3\overline{x_4}$	$F = \overline{x_1}\,\overline{x_2} + x_1x_4 + x_3\overline{x_4}$

Considering Table 4 we see that the combinatorial method gives a function with a smaller number of input variables.

5.4. Minimization of 5-bit Boolean functions. The combinatorial structure of the truth table of the 5-bit function is more complex compared to the 4-bit function, because of which there are more minimization options from the first step. For example, in the first step of minimizing the 5-bit function, it is possible to detect pair of blocks and sets of three blocks that allow for gluing and replacing variables. In the general case, the minimization of the combinatorial method of the 5-bit function is analogous to minimizing the 4-bit function.

Example 2. To minimize the logical function by the combinatorial method specified by the following table of truth (Table 5) [19].

Table 5

The truth table of the logical function $F(x_1,x_2,x_3,x_4,x_5)$

No.	x_1	x_2	x_3	x_4	x_5	F	No.	x_1	x_2	x_3	x_4	x_5	F
0	0	0	0	0	0	**0**	16	1	0	0	0	0	**1**
1	0	0	0	0	1	**1**	17	1	0	0	0	1	**1**
2	0	0	0	1	0	–	18	1	0	0	1	0	**1**
3	0	0	0	1	1	–	19	1	0	0	1	1	**0**
4	0	0	1	0	0	**1**	20	1	0	1	0	0	–
5	0	0	1	0	1	**1**	21	1	0	1	0	1	**0**
6	0	0	1	1	0	**0**	22	1	0	1	1	0	**1**
7	0	0	1	1	1	**1**	23	1	0	1	1	1	**0**
8	0	1	0	0	0	**0**	24	1	1	0	0	0	**0**
9	0	1	0	0	1	**1**	25	1	1	0	0	1	**0**
10	0	1	0	1	0	**0**	26	1	1	0	1	0	–
11	0	1	0	1	1	**1**	27	1	1	0	1	1	**0**
12	0	1	1	0	0	**1**	28	1	1	1	0	0	**1**
13	0	1	1	0	1	**1**	29	1	1	1	0	1	**0**
14	0	1	1	1	0	–	30	1	1	1	1	0	–
15	0	1	1	1	1	–	31	1	1	1	1	1	**1**

Using Table 5, let's compose the PDNF of the given 5-bit function from the blocks for which the function receives the value of 1, that is, for the sets 1, 4, 5, 7, 9, 11, 12, 13, 16, 17, 18, 22, 28, 31.

$$F(x_1,x_2,x_3,x_4,x_5) = \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}x_5 + \overline{x_1}\,\overline{x_2}x_3\,\overline{x_4}\,\overline{x_5} + \\ + \overline{x_1}\,\overline{x_2}x_3\,\overline{x_4}x_5 + \overline{x_1}\,x_2\,\overline{x_3}\,\overline{x_4}x_5 + \overline{x_1}\,x_2\,\overline{x_3}x_4\,\overline{x_5} + \\ + \overline{x_1}x_2\,\overline{x_3}x_4x_5 + \overline{x_1}x_2x_3\,\overline{x_4}\,\overline{x_5} + \overline{x_1}x_2x_3\,\overline{x_4}x_5 + \\ + x_1\,\overline{x_2}\,\overline{x_3}x_4\,\overline{x_5} + x_1\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}x_5 + x_1\,\overline{x_2}\,\overline{x_3}x_4\,\overline{x_5} + \\ + x_1\overline{x_2}x_3\,\overline{x_4}\,\overline{x_5} + x_1x_2\,\overline{x_3}\,\overline{x_4}\,\overline{x_5} + x_1x_2x_3x_4x_5. \quad (8)$$

Let's recall that the value of «–» of F function means an arbitrary state indicating that such set of input variables is not expected and the value of the function can be arbitrary – zero or one in the process of minimization.

Let's define function $F(x_1,x_2,x_3,x_4,x_5)$ by replacing the value of the «–» function by one. After replacing the value of the «–» function $F(x_1,x_2,x_3,x_4,x_5)$ by one, the truth table (Table 5) takes the following form (Table 6).

Table 6

The truth table of a logical function $F(x_1,x_2,x_3,x_4,x_5)$ after the value of the «–» function is replaced by one

No.	x_1	x_2	x_3	x_4	x_5	F	No.	x_1	x_2	x_3	x_4	x_5	F
0	0	0	0	0	0	**0**	16	1	0	0	0	0	**1**
1	0	0	0	0	1	**1**	17	1	0	0	0	1	**1**
2	0	0	0	1	0	**1**	18	1	0	0	1	0	**1**
3	0	0	0	1	1	**1**	19	1	0	0	1	1	**0**
4	0	0	1	0	0	**1**	20	1	0	1	0	0	**1**
5	0	0	1	0	1	**1**	21	1	0	1	0	1	**0**
6	0	0	1	1	0	**0**	22	1	0	1	1	0	**1**
7	0	0	1	1	1	**1**	23	1	0	1	1	1	**0**
8	0	1	0	0	0	**0**	24	1	1	0	0	0	**0**
9	0	1	0	0	1	**1**	25	1	1	0	0	1	**0**
10	0	1	0	1	0	**0**	26	1	1	0	1	0	**1**
11	0	1	0	1	1	**1**	27	1	1	0	1	1	**0**
12	0	1	1	0	0	**1**	28	1	1	1	0	0	**1**
13	0	1	1	0	1	**1**	29	1	1	1	0	1	**0**
14	0	1	1	1	0	**1**	30	1	1	1	1	0	**1**
15	0	1	1	1	1	**1**	31	1	1	1	1	1	**1**

Using Table 6, let's compose the PDNF of the 5-bit function from the blocks for which the function receives the value of unity, that is, for the sets 1, 2, 3, 4, 5, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 20, 22, 26, 28, 30, 31.

$$F(x_1,x_2,x_3,x_4,x_5)=\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4x_5+\overline{x}_1\overline{x}_2\overline{x}_3x_4\overline{x}_5+$$
$$+\overline{x}_1\overline{x}_2\overline{x}_3x_4x_5+\overline{x}_1\overline{x}_2x_3\overline{x}_4\overline{x}_5+\overline{x}_1\overline{x}_2x_3\overline{x}_4x_5+$$
$$+\overline{x}_1\overline{x}_2x_3x_4x_5+\overline{x}_1x_2\overline{x}_3\overline{x}_4x_5+\overline{x}_1x_2\overline{x}_3x_4\overline{x}_5+$$
$$+\overline{x}_1x_2x_3\overline{x}_4\overline{x}_5+\overline{x}_1x_2x_3\overline{x}_4x_5+\overline{x}_1x_2x_3x_4\overline{x}_5+$$
$$+\overline{x}_1x_2x_3x_4x_5+x_1\overline{x}_2\overline{x}_3\overline{x}_4\overline{x}_5+x_1\overline{x}_2\overline{x}_3\overline{x}_4x_5+$$
$$+x_1\overline{x}_2\overline{x}_3x_4\overline{x}_5+x_1\overline{x}_2x_3\overline{x}_4\overline{x}_5+x_1\overline{x}_2x_3x_4\overline{x}_5+$$
$$+x_1x_2\overline{x}_3\overline{x}_4\overline{x}_5+x_1x_2x_3\overline{x}_4\overline{x}_5+x_1x_2x_3x_4\overline{x}_5+x_1x_2x_3x_4x_5. \quad (9)$$

Let's consider two options for minimization of the 5-bit Boolean function (9).

In the first variant, pairs of blocks that allow procedures for pasting and replacing variables are first identified.

At the first step, the constituents are glued together and the variables are replaced.

Algebraic transformations of the 1-st matrix (the result of the transformation is written in the 2-nd matrix):

$F=$

No.	1	2	3	4
1	0 0 0 0 1	0 0 0 0 1	0 0 0 0 1	0 0 0 1
2	0 0 0 1 0			
3	0 0 0 1 1	0 0 0 1	0 0 0 1	0 0 0 1
4	0 0 1 0 0	0 0 1 0	0 0 1 0	0 0 1 0
5	0 0 1 0 1	0 0 1 1	0 0 1 1	0 0 1 1
6	0 0 1 1 1			
7	0 1 0 0 1			
8	0 1 0 1 1	0 1 0 1	0 1 0 1	0 1 0 1
9	0 1 1 0 0			
10	0 1 1 0 1	0 1 1 0	0 1 1 0	0 1 1
11	0 1 1 1 0			
12	0 1 1 1 1	0 1 1 1		
13	1 0 0 0 0			
14	1 0 0 0 1	1 0 0 0	1 0 0 0	1 0 0 0
15	1 0 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0
16	1 0 1 0 0	1 0 1 0	1 0 1 0	1 0 1 0
17	1 0 1 1 0			
18	1 1 0 1 0	1 1 0 1 0	1 1 0 1 0	1 1 1 0
19	1 1 1 0 0	1 1 1 0 0	1 1 1 0 0	1 1 1 0
20	1 1 1 1 0			
21	1 1 1 1 1	1 1 1 1	1 1 1	1 1 1

=

– gluing of variable in 2 and 3 blocks:

$$\overline{x}_1\overline{x}_2\overline{x}_3x_4\overline{x}_5+\overline{x}_1\overline{x}_2\overline{x}_3x_4x_5=\overline{x}_1\overline{x}_2\overline{x}_3x_4(\overline{x}_5+x_5)=\overline{x}_1\overline{x}_2\overline{x}_3x_4,$$

– gluing and replacing of variables in 4, 5 and 6 blocks:

$$\overline{x}_1\overline{x}_2x_3\overline{x}_4\overline{x}_5+\overline{x}_1\overline{x}_2x_3\overline{x}_4x_5+\overline{x}_1\overline{x}_2x_3x_4x_5=\overline{x}_1\overline{x}_2x_3\times$$
$$\times(\overline{x}_4\overline{x}_5+\overline{x}_4x_5+x_4x_5)=\overline{x}_1\overline{x}_2x_3(\overline{x}_4\overline{x}_5+\overline{x}_4x_5+\overline{x}_4+x_4x_5)=$$
$$=\overline{x}_1\overline{x}_2x_3(\overline{x}_4\overline{x}_5+\overline{x}_4x_5+\overline{x}_4+x_5)=\overline{x}_1\overline{x}_2x_3\times$$
$$\times(\overline{x}_4(\overline{x}_5+x_5)+\overline{x}_4+x_5)=\overline{x}_1\overline{x}_2x_3(\overline{x}_4+\overline{x}_4+x_5)=$$
$$=\overline{x}_1\overline{x}_2x_3(\overline{x}_4+x_5)=\overline{x}_1\overline{x}_2x_3\overline{x}_4+\overline{x}_1\overline{x}_2x_3x_5,$$

– gluing of variables in 7 and 8 blocks:

$$\overline{x}_1x_2\overline{x}_3\overline{x}_4x_5+\overline{x}_1x_2\overline{x}_3x_4x_5=\overline{x}_1x_2\overline{x}_3x_5(\overline{x}_4+x_4)=\overline{x}_1x_2\overline{x}_3x_5,$$

– gluing of variables in 9 and 10 blocks:

$$\overline{x}_1x_2x_3\overline{x}_4\overline{x}_5+\overline{x}_1x_2x_3\overline{x}_4x_5=\overline{x}_1x_2x_3\overline{x}_4(\overline{x}_5+x_5)=\overline{x}_1x_2x_3\overline{x}_4,$$

– gluing of variables in 11 and 12 blocks:

$$\overline{x}_1x_2x_3x_4\overline{x}_5+\overline{x}_1x_2x_3x_4x_5=\overline{x}_1x_2x_3x_4(\overline{x}_5+x_5)=\overline{x}_1x_2x_3x_4,$$

– gluing of variables in 13 and 14 blocks:

$$\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4\overline{x}_5+\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4x_5=\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4(\overline{x}_5+x_5)=\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4,$$

ISSN 2226-3780

– gluing and replacing of variables in 15, 16 and 17 blocks:

$$\overline{x_1}\overline{x_2}x_3x_4\overline{x_5}+x_1\overline{x_2}x_3\overline{x_4}\,\overline{x_5}+x_1\overline{x_2}x_3x_4\overline{x_5}=x_1\overline{x_2}\,\overline{x_5}\times$$
$$\times(\overline{x_3}x_4+x_3\overline{x_4}+x_3x_4)=x_1\overline{x_2}\,\overline{x_5}(\overline{x_3}x_4+x_3x_4+x_3\overline{x_4})=$$
$$=x_1\overline{x_2}\,\overline{x_5}(\overline{x_3}x_4+x_3x_4+x_4+x_3\overline{x_4})=$$
$$=x_1\overline{x_2}\,\overline{x_5}(x_4(\overline{x_3}+x_3)+x_4+x_3)=$$
$$=x_1\overline{x_2}\,\overline{x_5}(x_4+x_4+x_3)=x_1\overline{x_2}\,\overline{x_5}(x_4+x_3)=$$
$$=x_1\overline{x_2}x_3\overline{x_5}+x_1\overline{x_2}x_4\overline{x_5},$$

– gluing of variables in 20 and 21 blocks:

$$\overline{x_1}\overline{x_2}\overline{x_3}\overline{x_4}\overline{x_5}+x_1\overline{x_2}\overline{x_3}\overline{x_4}x_5=x_1\overline{x_2}\overline{x_3}\overline{x_4}(\overline{x_5}+x_5)=x_1\overline{x_2}\overline{x_3}\overline{x_4}.$$

At the second step, the implicants are glued together and the replacing the variables is made.

Gluing of variables in 12 and 21 blocks.

Algebraic transformations of the 2-nd matrix (the result of the transformation is written in the 3-rd matrix):

$$\overline{x_1}x_2x_3x_4+x_1x_2x_3x_4=x_2x_3x_4(\overline{x_1}+x_1)=x_2x_3x_4.$$

Algebraic transformations of the 3-rd matrix (the result of the transformation is written in the 4-th matrix):

– replacing of variables in 10, 18, 19 and 21 blocks:

$$\overline{x_1}x_2x_3\overline{x_4}+x_1x_2\overline{x_3}x_4\overline{x_5}+x_1x_2x_3\overline{x_4}\,\overline{x_5}+x_2x_3x_4=$$
$$=x_2x_3x_4+\overline{x_1}x_2x_3\overline{x_4}+x_1x_2x_3\overline{x_4}\,\overline{x_5}+x_2x_3x_4+$$
$$+x_1x_2\overline{x_3}x_4\overline{x_5}=x_2x_3(x_4+\overline{x_1}\,\overline{x_4}+x_1\overline{x_4}\,\overline{x_5})+$$
$$+x_2x_4(x_3+x_1\overline{x_3}\,\overline{x_5})=x_2x_3(x_4+\overline{x_1}+x_1\overline{x_5})+$$
$$+x_2x_4(x_3+x_1\overline{x_5})=x_2x_3(x_4+\overline{x_1}+x_1\overline{x_5})+$$
$$+x_2x_4(x_3+x_1\overline{x_5})=x_2x_3x_4+\overline{x_1}x_2x_3+$$
$$+x_1x_2x_3\overline{x_5}+x_2x_3x_4+x_1x_2x_4\overline{x_5}=x_2x_3x_4+$$
$$+\overline{x_1}x_2x_3+x_1x_2x_3\overline{x_5}+x_2x_3x_4+x_1x_2x_4\overline{x_5}=$$
$$=x_2x_3x_4+\overline{x_1}x_2x_3+x_1x_2x_3\overline{x_5}+x_1x_2x_4\overline{x_5}.$$

– replacing of variables in 1 and 3 blocks:

$$\overline{x_1}\overline{x_2}\overline{x_3}\overline{x_4}x_5+\overline{x_1}\overline{x_2}\overline{x_3}x_4=\overline{x_1}\overline{x_2}\overline{x_3}(\overline{x_4}x_5+x_4)=$$
$$=\overline{x_1}\overline{x_2}\overline{x_3}(x_5+x_4)=\overline{x_1}\overline{x_2}\overline{x_3}x_5+\overline{x_1}\overline{x_2}\overline{x_3}x_4.$$

Algebraic transformations of the 4-th matrix (the result of the transformation is written in the 5-th matrix):

– gluing of variables in 15 and 18 blocks:

$$x_1\overline{x_2}x_4\overline{x_5}+x_1x_2x_4\overline{x_5}=x_1x_4\overline{x_5}(\overline{x_2}+x_2)=x_1x_4\overline{x_5},$$

– gluing of variables in 16 and 19 blocks:

$$x_1\overline{x_2}x_3\overline{x_5}+x_1x_2x_3\overline{x_5}=x_1x_3\overline{x_5}(\overline{x_2}+x_2)=x_1x_3\overline{x_5},$$

– gluing of variables in 8 and 10 blocks:

$$\overline{x_1}x_2\overline{x_3}x_5+\overline{x_1}x_2x_3=\overline{x_1}x_2(\overline{x_3}x_5+x_3)=$$
$$=\overline{x_1}x_2(x_5+x_3)=\overline{x_1}x_2x_5+\overline{x_1}x_2x_3,$$

– replacing of variables in 4 and 10 blocks:

$$\overline{x_1}\,\overline{x_2}x_3\overline{x_4}+\overline{x_1}x_2x_3=\overline{x_1}x_3(\overline{x_2}\,\overline{x_4}+x_2)=$$
$$=\overline{x_1}x_3(\overline{x_4}+x_2)=\overline{x_1}x_3\overline{x_4}+\overline{x_1}x_2x_3,$$

– gluing of variables in 1 and 5 blocks:

$$\overline{x_1}\overline{x_2}\overline{x_3}x_5+\overline{x_1}\overline{x_2}x_3x_5=\overline{x_1}\overline{x_2}x_5(\overline{x_3}+x_3)=\overline{x_1}\overline{x_2}x_5.$$

=

No.	5	6	7	8
	$x_1x_2x_3x_4x_5$	$x_1x_2x_3x_4x_5$	$x_1x_2x_3x_4x_5$	$x_1x_2x_3x_4x_5$
1				
2				
3	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
4	0 1 0	0 1 0	0 1 0	
5	0 0 1			
6				
7				
8	0 1 1	0 1	0 1	0 1
9				
10	0 1 1	0 1 1		
11				
12				
13				
14	1 0 0 0	1 0 0 0	1 0 0 0	1 0 0 0
15				
16			1 0 0	1 0 0
17				
18	1 1 0	1 1 0	1 1 0	1 1 0
19	1 1 0	1 1 0	1 1 0	
20				
21	1 1 1	1 1 1	1 1 1	1 1 1

Algebraic transformations of the 5-th matrix (the result of the transformation is written in the 6-th matrix):

– gluing of variables in 5 and 8 blocks:

$$\overline{x_1}\,\overline{x_2}x_5+\overline{x_1}x_2x_5=\overline{x_1}x_5(\overline{x_2}+x_2)=\overline{x_1}x_5.$$

– generalized replacing of variables in 4, 8 and 16 blocks:

$$\overline{x_1}x_3\overline{x_4}+\overline{x_1}x_2x_3+x_2x_3x_4=\overline{x_1}x_3\overline{x_4}+$$
$$+x_2x_3x_4+\overline{x_1}x_2x_3=\overline{x_1}x_3\overline{x_4}+x_2x_3x_4,$$

– generalized replacing of variables in 4 and 19 blocks:

$$\overline{x_1}x_3\overline{x_4}+x_1x_3\overline{x_5}=\overline{x_1}x_3\overline{x_4}+x_1x_3\overline{x_5}+$$
$$+x_3x_3\overline{x_4}\,\overline{x_5}=\overline{x_1}x_3\overline{x_4}+x_1x_3\overline{x_5}+x_3\overline{x_4}\,\overline{x_5}.$$

Algebraic transformations of the 7-th matrix (the result of the transformation is written in the 8-th matrix):

– generalized replacing of variables in 4, 10 and 21 blocks:

$$\overline{x_1}x_5+x_3\overline{x_4}\,\overline{x_5}+\overline{x_1}x_3\overline{x_4}=\overline{x_1}x_5+x_3\overline{x_4}\,\overline{x_5},$$

– generalized replacing of variables in 16, 18 and 19 blocks:

$$x_3\overline{x}_4\overline{x}_5 + x_1x_4\overline{x}_5 = x_3\overline{x}_4\overline{x}_5 + x_1x_4\overline{x}_5 + x_1x_3\overline{x}_5\overline{x}_5 =$$
$$= x_3\overline{x}_4\overline{x}_5 + x_1x_4\overline{x}_5 + x_1x_3\overline{x}_5 = x_3\overline{x}_4\overline{x}_5 + x_1x_4\overline{x}_5.$$

The third step involves testing of each simple implicant in PDNF for redundancy to remove it and verifying the resulting function using a truth table (Table 6).

Attempts to further apply algebraic transformation operations do not yield a result (matrix 8). So, the resulting DDNF of the function $F(x_1,x_2,x_3,x_4,x_5)$ is presented in Table 6. Further, the problem of finding the minimal DNF is solved on the basis of the covering table (Table 7). In general, in order to obtain the minimum DNF, it is necessary to remove all superfluous simple implicants from the DDNF.

Table 7

Covering table of the function $F(x_1,x_2,x_3,x_4,x_5)$

Constituents	$\overline{x}_1x_5$	$x_1\overline{x}_2\overline{x}_3\overline{x}_4$	$x_1x_4\overline{x}_5$	$x_1x_2x_3x_4$	$x_2x_3x_4$	$x_3\overline{x}_4\overline{x}_5$
00001	●	–	–	–	–	–
00010	–	–	–	●	–	–
00011	●	–	–	–	–	–
00100	–	–	–	–	–	●
00101	●	–	–	–	–	–
00111	●	–	–	–	–	–
01001	●	–	–	–	–	–
01011	●	–	–	–	–	–
01100	–	–	–	–	–	●
01101	●	–	–	–	–	–
01110	–	–	–	–	●	–
01111	●	–	–	–	●	–
10000	–	●	–	–	–	–
10001	–	●	–	–	–	–
10010	–	–	●	–	–	–
10100	–	–	–	–	–	●
10110	–	–	●	–	–	–
11010	–	–	●	–	–	–
11100	–	–	–	–	–	●
11110	–	–	●	–	●	–
11111	–	–	–	–	●	–

In the columns of Table 7 there are simple implicants of the reduced DNF of the function (matrix 8). The rows of Table 7 represent the constituents of a unit of the PDNF function, represented by Table. 6.

A simple implicant absorbs some constituent of the unit when it is its own part. The corresponding cell of Table 7 at the intersection of the column (with the simple implicant under consideration) and the line (with the constituent of the unit) is affected by the ● icon of black color.

Considering Table 7 we see that there are no superfluous implicants, and, consequently, Table 7 represents MDNF of the function (9), presented in Table 6.

$$F(x_1,x_2,x_3,x_4,x_5) = \overline{x}_1x_5 + x_1\overline{x}_2\overline{x}_3\overline{x}_4 +$$
$$+x_1x_4\overline{x}_5 + \overline{x}_1\overline{x}_2\overline{x}_3x_4 + x_2x_3x_4 + x_3\overline{x}_4\overline{x}_5. \quad (10)$$

The truth table (Table 6) is created in order to obtain a more convenient minimization process. However, the original logical function (8) is represented by a truth table (Table 5), in which there are sets of variables, is not expected. The value of the function F for such sets is affected by «–» and means an arbitrary state.

In this regard, the search for MDNF of the function, represented by the original truth table (Table 5), is solved using the covering table (Table 7), removing sets of variables from its rows is not expected. Table 7 after removing the sets that are not expected to take the form of Table 8.

Table 8

Covering table of the function $F(x_1,x_2,x_3,x_4,x_5)$ with remote sets of variables that are not expected

Constituents	$\overline{x}_1x_5$	$x_1\overline{x}_2\overline{x}_3\overline{x}_4$	$x_1x_4\overline{x}_5$	$\overline{x}_1\overline{x}_2x_3x_4$	$x_2x_3x_4$	$x_3\overline{x}_4\overline{x}_5$
00001	●	–	–	–	–	–
00100	–	–	–	–	–	●
00101	●	–	–	–	–	–
00111	●	–	–	–	–	–
01001	●	–	–	–	–	–
01011	●	–	–	–	–	–
01100	–	–	–	–	–	●
01101	●	–	–	–	–	–
10000	–	●	–	–	–	–
10001	–	●	–	–	–	–
10010	–	–	●	–	–	–
10110	–	–	●	–	–	–
11100	–	–	–	–	–	●
11111	–	–	–	–	●	–

Considering Table 8 we see that the implicant $\overline{x}_1\overline{x}_2\overline{x}_3x_4$ is superfluous, which we remove from the expression for the function (10).

$$F(x_1,x_2,x_3,x_4,x_5) = \overline{x}_1x_5 + x_1\overline{x}_2\overline{x}_3\overline{x}_4 +$$
$$+x_1x_4\overline{x}_5 + x_2x_3x_4 + x_3\overline{x}_4\overline{x}_5. \quad (11)$$

Expression (11) represents DDNF and MDNF of the initial function (8) that is presented in Table. 5.

Table 9 shows the results of minimization by the method of «symmetric maps» [19] and combinatorial method.

Table 9

The result of minimization of the function $F(x_1,x_2,x_3,x_4,x_5)$

Minimization by the method of «symmetric maps»
$F(x_1,x_2,x_3,x_4,x_5) = \overline{x}_1x_5 + x_1\overline{x}_2\overline{x}_3\overline{x}_4 + x_1\overline{x}_2\overline{x}_5 + x_2x_3x_4 + x_3\overline{x}_4\overline{x}_5.$
$F(x_1,x_2,x_3,x_4,x_5) = \overline{x}_1x_5 + \overline{x}_2\overline{x}_3\overline{x}_4x_5 + x_1\overline{x}_2\overline{x}_5 + x_2x_3x_4 + x_3\overline{x}_4\overline{x}_5.$
Minimization by combinatorial method
$F(x_1,x_2,x_3,x_4,x_5) = \overline{x}_1x_5 + x_1\overline{x}_2\overline{x}_3\overline{x}_4 + x_1x_4\overline{x}_5 + x_2x_3x_4 + x_3\overline{x}_4\overline{x}_5.$

ISSN 2226-3780

The main difference between the minimal functions of Table 9 shows the 3-rd implicant. For a function minimized by the method of «symmetrical cards» implicant – $x_1\overline{x_2}\overline{x_5}$ to maintain its functionality requires two inverters. For a function minimized by the combinatorial method of an implicant $x_1x_4\overline{x_5}$, one inverter is required to maintain its functionality. Thus, using, for example, C-MOS technology (the complementary metal-oxide-semiconductor structure), the hardware implementation of the function (11) will require one inverter less.

The minimized logic function (11) satisfies the given truth table (Table 5).

The second option is to minimize the 5-bit logic function (9). At the first stage, sets of three blocks are identified that allow procedures for gluing and replacing of variables.

The first step is gluing of the constituent and replacing of the variables.

Algebraic transformations are presented only for the first matrix.

– gluing and replacing of variables in 1, 2 and 3 blocks:

$$\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4x_5+\overline{x}_1\overline{x}_2\overline{x}_3x_4\overline{x}_5+\overline{x}_1\overline{x}_2x_3\overline{x}_4\overline{x}_5=\overline{x}_1\overline{x}_2\overline{x}_3\times$$
$$\times(\overline{x}_4x_5+x_4\overline{x}_5+x_4x_5)=\overline{x}_1\overline{x}_2\overline{x}_3(\overline{x}_4x_5+x_4x_5+x_4\overline{x}_5)=$$
$$=\overline{x}_1\overline{x}_2\overline{x}_3(\overline{x}_4x_5+x_4x_5+x_5+x_4\overline{x}_5)=\overline{x}_1\overline{x}_2\overline{x}_3\times$$
$$\times(\overline{x}_4x_5+x_4x_5+x_5+x_4)=\overline{x}_1\overline{x}_2\overline{x}_3(x_5(\overline{x}_4+x_4)+x_5+x_4)=$$
$$=\overline{x}_1\overline{x}_2\overline{x}_3x_4+\overline{x}_1\overline{x}_2x_3x_5,$$

– gluing and replacing of variables in 4, 5 and 6 blocks:

$$\overline{x}_1\overline{x}_2x_3\overline{x}_4\overline{x}_5+\overline{x}_1\overline{x}_2x_3\overline{x}_4x_5+\overline{x}_1\overline{x}_2x_3x_4x_5=\overline{x}_1\overline{x}_2x_3\times$$
$$\times(\overline{x}_4\overline{x}_5+\overline{x}_4x_5+x_4x_5)=\overline{x}_1\overline{x}_2x_3(\overline{x}_4\overline{x}_5+\overline{x}_4x_5+\overline{x}_4+x_4x_5)=$$
$$=\overline{x}_1\overline{x}_2x_3(\overline{x}_4\overline{x}_5+\overline{x}_4x_5+\overline{x}_4+x_5)=\overline{x}_1\overline{x}_2x_3(\overline{x}_4(\overline{x}_5+x_5)+\overline{x}_4+x_5)=$$
$$=\overline{x}_1\overline{x}_2x_3(\overline{x}_4+\overline{x}_4+x_5)=\overline{x}_1\overline{x}_2x_3(\overline{x}_4+x_5)=\overline{x}_1\overline{x}_2x_3\overline{x}_4+\overline{x}_1\overline{x}_2x_3x_5,$$

– gluing of variables in 7 and 8 blocks:

$$\overline{x}_1x_2\overline{x}_3\overline{x}_4x_5+\overline{x}_1x_2\overline{x}_3x_4x_5=\overline{x}_1x_2\overline{x}_3x_5(\overline{x}_4+x_4)=\overline{x}_1x_2\overline{x}_3x_5,$$

– gluing and replacing of variables in 9, 10 and 11 blocks:

$$\overline{x}_1x_2x_3\overline{x}_4\overline{x}_5+\overline{x}_1x_2x_3\overline{x}_4x_5+\overline{x}_1x_2x_3x_4\overline{x}_5=\overline{x}_1x_2x_3\times$$
$$\times(\overline{x}_4\overline{x}_5+\overline{x}_4x_5+x_4\overline{x}_5)=\overline{x}_1x_2x_3(\overline{x}_4\overline{x}_5+\overline{x}_4x_5+\overline{x}_4+x_4\overline{x}_5)=$$
$$=\overline{x}_1x_2x_3(\overline{x}_4\overline{x}_5+\overline{x}_4x_5+\overline{x}_4+\overline{x}_5)=$$
$$=\overline{x}_1x_2x_3(\overline{x}_4(\overline{x}_5+x_5)+\overline{x}_4+\overline{x}_5)=\overline{x}_1x_2x_3(\overline{x}_4+\overline{x}_4+\overline{x}_5)=$$
$$=\overline{x}_1x_2x_3(\overline{x}_4+\overline{x}_5)=\overline{x}_1x_2x_3\overline{x}_4+\overline{x}_1x_2x_3\overline{x}_5,$$

– gluing of variables in 12 and 21 blocks:

$$\overline{x}_1x_2x_3x_4x_5+x_1x_2x_3x_4x_5=x_2x_3x_4x_5(\overline{x}_1+x_1)=x_2x_3x_4x_5,$$

– gluing of variables in 13 and 14 blocks:

$$x_1\overline{x}_2\overline{x}_3\overline{x}_4\overline{x}_5+x_1\overline{x}_2\overline{x}_3\overline{x}_4x_5=x_1\overline{x}_2\overline{x}_3\overline{x}_4(\overline{x}_5+x_5)=x_1\overline{x}_2\overline{x}_3\overline{x}_4,$$

– gluing and replacing of variables in 15, 16 and 17 blocks:

$$x_1\overline{x}_2\overline{x}_3x_4\overline{x}_5+x_1\overline{x}_2x_3\overline{x}_4\overline{x}_5+x_1\overline{x}_2x_3x_4\overline{x}_5=x_1\overline{x}_2\overline{x}_5\times$$
$$\times(\overline{x}_3x_4+x_3\overline{x}_4+x_3x_4)=x_1\overline{x}_2\overline{x}_3x_4\overline{x}_5+x_1\overline{x}_2x_3\overline{x}_4\overline{x}_5+$$
$$+x_1\overline{x}_2x_3x_4\overline{x}_5=x_1\overline{x}_2\overline{x}_5(\overline{x}_3x_4+x_3\overline{x}_4+x_3x_4)=$$
$$=x_1\overline{x}_2\overline{x}_5(x_4(\overline{x}_3+x_3)+x_4+x_3)=x_1\overline{x}_2\overline{x}_5(x_4+x_4+x_3)=$$
$$=x_1\overline{x}_2\overline{x}_5(x_4+x_3)=x_1\overline{x}_2x_3\overline{x}_5+x_1\overline{x}_2x_4\overline{x}_5,$$

$F=$

00001	000 1				
00010					
00011	0001	0001	0001	0001	0001
00100	0010				
00101	001 1	00 1	00 1	00 1	00 1
00111					
01001					
01011	010 1	010 1	0 0 1	0 0 1	0 0 1
01100	0110	0 10	0 10	0 10	0 10
01101					
01110	011 0	011 0	011 0	011 0	011 0
01111					011
10000					
10001	1000	1000	1000	1000	1000
10010					
10100	101 0				100
10110	10 10		1110		
11010					
11100	111 0	1 1 0	1 1 0	1 1 0	1 1 0
11110	11 10	1 10	1 10	1 10	1 10
11111	1111	1111	1111	111	111

=

0001	0001	0001	0001	0001
00 1	00 1			
0 1 1				
0 0 1	0 1	0 1	0 1	0 1
0 10	0 10	0 10	0 10	
011	011	011		
1000	1000	1000	1000	1000
100	100	100	100	100
1 10	1 10	1 10	1 10	1 10
111	111	111	111	111

– gluing and replacing of variables in 18, 19 and 20 blocks:

$$x_1x_2\overline{x_3}x_4\overline{x_5}+x_1x_2x_3\overline{x_4}\,\overline{x_5}+x_1x_2x_3x_4\overline{x_5}=x_1x_2\overline{x_5}\times$$
$$\times(\overline{x_3}x_4+x_3\overline{x_4}+x_3x_4)=x_1x_2\overline{x_5}(\overline{x_3}x_4+x_3x_4+x_3\overline{x_4})=$$
$$=x_1x_2\overline{x_5}(\overline{x_3}x_4+x_3x_4+x_4+x_3\overline{x_4})=$$
$$=x_1x_2\overline{x_5}(\overline{x_3}x_4+x_3x_4+x_4+x_3)=$$
$$x_1x_2\overline{x_5}(x_4(\overline{x_3}+x_3)+x_4+x_3)=x_1x_2\overline{x_5}(x_4+x_4+x_3)=$$
$$=x_1x_2\overline{x_5}(x_4+x_3)=x_1x_2x_3\overline{x_5}+x_1x_2x_4\overline{x_5}.$$

The peculiarity of the second variant of minimization is the change in the initial state of each constituent of the logical function in the first stage of minimization.

Attempts to further apply the operations of algebraic transformation in the second version of minimization do not give a result. So, DDNF of the function $F(x_1,x_2,x_3,x_4,x_5)$ is obtained (9).

To obtain the minimum DNF of the function, represented by the original truth table (Table 5), it is necessary to carry out actions similar to the first variant of minimization.

Example 3. To minimize the logic function $F(x_1,x_2,x_3,x_4,x_5)$ by combinatorial method, given by the following truth table (Table 10) [20].

Table 10

The truth table of a logical function $F(x_1,x_2,x_3,x_4,x_5)$

No.	x_1	x_2	x_3	x_4	x_5	F	No.	x_1	x_2	x_3	x_4	x_5	F
0	0	0	0	0	0	**1**	8	1	0	1	0	1	**1**
1	0	0	1	0	1	**1**	9	1	0	1	1	0	**1**
2	0	0	1	1	1	**1**	10	1	0	1	1	1	**1**
3	0	1	0	1	1	**1**	11	1	1	0	0	0	**1**
4	0	1	1	0	0	**1**	12	1	1	1	0	0	**1**
5	0	1	1	0	1	**1**	13	1	1	1	0	1	**1**
6	0	1	1	1	1	**1**	14	1	1	1	1	0	**1**
7	1	0	0	0	0	**1**	15	1	1	1	1	1	**1**

An figurative calculus (without demonstrating algebraic transformations) of the combinatorial method of minimizing the Boolean function looks like this:

$$F=\left|\begin{array}{l}00000\\00101\\00111\\01011\\01100\\01101\\01111\\10000\\10101\\10110\\10111\\11000\\11100\\11101\\11110\\11111\end{array}\right|=\left|\begin{array}{l}\\\\00\,\,1\,\,1\\01011\\0110\\011\,\,1\\\\0000\\101\,\,1\\1011\\\\11000\\\\1110\\\\1111\end{array}\right|=\left|\begin{array}{l}\\\\\\01011\\0110\\0\,\,1\,\,1\\\\0000\\101\,\,1\\1011\\\\11000\\\\111\\\\\end{array}\right|=\left|\begin{array}{l}\\\\\\01011\\\,\,110\\0\,\,1\,\,1\\\\0000\\1\,\,1\,\,1\\1\,\,11\\\\11\,\,00\\\\111\\\\\end{array}\right|=\left|\begin{array}{l}\\\\01\,\,11\\\,\,110\\\\\\\\0000\\\,\,1\,\,1\\1\,\,11\\\\11\,\,00\\\\\\\\\end{array}\right|.$$

In the first matrix 6 procedures of gluing and replacing of variables were carried out, 2 procedures for gluing of variables were performed in the second matrix, 4 gluing procedures were carried out in the third matrix, 1 gluing procedure was carried out in the fourth matrix and 1 generalized gluing procedure was performed – 14 transformations in all.

Superfluous implicants in the obtained minimal logical function (12) are absent.

$$F(x_1,x_2,x_3,x_4,x_5)=x_3x_5+x_1x_3x_4+x_2x_3\overline{x_4}+$$
$$+\overline{x_1}x_2x_4x_5+x_1x_2\overline{x_4}\,\overline{x_5}+\overline{x_2}\,\overline{x_3}\,\overline{x_4}\,\overline{x_5}. \qquad (12)$$

The result of minimization by combinatorial method (12) coincides with the result of minimization obtained by means of the Karnaugh map [20].

The minimized logic function (12) satisfies the given truth table (Table 10).

The process of minimizing the function of Example 3 demonstrates the hardware compactness of the combinatorial method.

6. Research results

The problems of reducing the Boolean function and establishing an assessment of the complexity of DNF minimization have been studied since the 50s of the 20th century [21–25]. A typical difficulty of such problems is that, on the one hand, procedures for minimizing Boolean functions can't be performed without brute force [22], and on the other hand, the power of busting is usually very large. As noted in [23, 24], the maximum number of DDNFs of a logical function of n variables is of the order greater than 2^{2^n}.

In the process of these studies, algorithms were developed that perform a much less exhaustive search than the algorithm for enumerating all the DDNF of Boolean functions from the selected class. The mathematical apparatus of such algorithm is the interval graph [25].

Algorithms for finding the MDNF for a class of Boolean functions, the so-called simplified graph of intervals, were described. The complexity of the constructed algorithms turned out to be linearly dependent on the number of vertices-conjunctions in the original graph.

The peculiarity of the combinatorial minimization method is obtaining of a minimal function by several variants of the search that reduces the search. This feature is the rationale for development of an appropriate minimization protocol.

The complexity of the algorithm is a quantitative characteristic that reflects the resources consumed by the algorithm during its execution. The main resources that are evaluated are the execution time (that is, the maximum number of operations required by the algorithm to obtain the response) and the memory space.

To estimate the algorithmic complexity of the search for a minimal function by a combinatorial method, let's take the operations of algebraic transformations as consumed algorithmic resources, which are performed when the function is minimized. For example, the absorption, substitution or idempotency of variables is one operation, but the number of elementary operations in the indicated algebraic transformations can be different.

ISSN 2226-3780

Minimization of the logical function by the Quine-McCluskey method involves splitting all sets of variables into groups according to the number of units in them. The gluing operation can only be in sets of neighboring groups that differ in a single-digit variable. Putting the sets into groups, at the first stage, carry out all possible gluing of variables. In the second stage, again, all the sets of variables are separated after gluing into groups according to the number of units in them and taking into account the coordinate sign (for example, «~»). Conduct all possible gluing of the second stage.

After receiving DDNF, a table of covering is constructed, the columns of which are called constituents of the original function, and the lines correspond to the obtained implicants at the stages of gluing of the variables. Using the covering table, the minimum function (MDNF) is found.

The main resources that are spent using the Quine-McCluskey search algorithm are algebraic operations in the stages of gluing together variables. In this connection, the estimation of the algorithmic complexity of the search for a minimal function by the Quine-McCluskey method will be determined by calculating the amount of consumed resources (the number of algebraic operations) at the stages of gluing, absorbing and idempotency of the variables that are performed when the function is minimized.

Example 4. To minimize the 3-bit Boolean function by the Quine-McCluskey method. The output function is given by the following truth table (Table 11) [23].

Table 11

The truth table of a logical function $F(x_1, x_2, x_3)$

No.	x_3	x_2	x_1	F	No.	x_3	x_2	x_1	F
0	0	0	0	**1**	4	1	0	0	**1**
1	0	0	1	**0**	5	1	0	1	**1**
2	0	1	0	**1**	6	1	1	0	**1**
3	0	1	1	**1**	7	1	1	1	**0**

To minimize the given function, let's select sets of variables for which the function receives the values of unit:

$$F = \{000, 010, 011, 100, 101, 110\}. \quad (13)$$

Let's separate the sets of variables into groups depending on the number of units in them and carry out the gluing of variables in neighboring groups (Fig. 2).

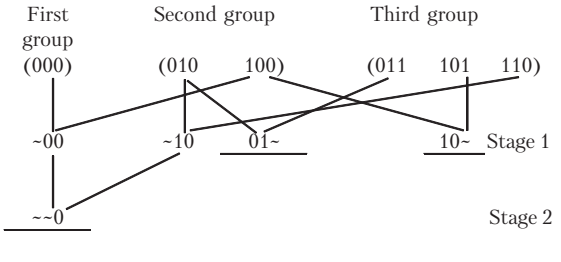


Fig. 2. Separation of the sets of variables of function (13) into groups by number of units in them

From Fig. 2 we see that in the two stages of gluing of variables, 5 algebraic transformations are consumed. Underlined implicants form a *Z*-covering:

$$Z = \begin{cases} 01\sim; \\ 10\sim; \\ \sim\sim 0. \end{cases}$$

Thus, DDNF (14) of the function *F* (13) is obtained:

$$F_{DNF} = \overline{x_3}x_2 + x_3\overline{x_2} + \overline{x_1}. \quad (14)$$

To remove excess implicants and to obtain MDNF, a covering table is constructed (Table 12).

Table 12

Covering table of the function $F(x_1, x_2, x_3)$

Implicants	Constituents					
	000	010	100	011	101	110
01~	–	●	–	●	–	–
10~	–	–	●	–	●	–
~~0	●	●	●	–	–	●

From the covering table it turns out that all the obtained implicants are part of the core of the function. Thus, the MDNF of a given function has the following form:

$$F_{DNF} = \overline{x_3}x_2 + x_3\overline{x_2} + \overline{x_1}. \quad (15)$$

The minimization of the logical function (13) by combinatorial method looks like this:

$$F = \begin{vmatrix} 0\ 0\ 0 \\ 0\ 1\ 0 \\ 0\ 1\ 1 \\ 1\ 0\ 0 \\ 1\ 0\ 1 \\ 1\ 1\ 0 \end{vmatrix} = \begin{vmatrix} 0\ 0\ 0 \\ \\ 0\ 1 \\ \\ 1\ 0 \\ 1\ 1\ 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 \\ \\ 0\ 1 \\ \\ 1\ 0 \\ 1 & 0 \end{vmatrix} = \begin{vmatrix} 0\ 1 \\ \\ 1\ 0 \\ & 0 \end{vmatrix}.$$

In the first matrix, 2 procedures for gluing of variables are performed, the second matrix contains the 2 procedure for the replacing of variables, in the third matrix there is 1 procedure for gluing of variables. Total: 5 algebraic transformations. The minimized function by combinatorial method coincides with expression (15).

Example 5. To minimize the 4-bit Boolean function by the Quine-McCluskey method. The output function is specified by the following truth table (Table 13) [24].

Table 13

The truth table of a logical function $F(x_1, x_2, x_3, x_4)$

No.	x_1	x_2	x_3	x_4	F	No.	x_1	x_2	x_3	x_4	F
0	0	0	0	0	**0**	8	1	0	0	0	**0**
1	0	0	0	1	**0**	9	1	0	0	1	**1**
2	0	0	1	0	**0**	10	1	0	1	0	**0**
3	0	0	1	1	**1**	11	1	0	1	1	**0**
4	0	1	0	0	**1**	12	1	1	0	0	**0**
5	0	1	0	1	**1**	13	1	1	0	1	**1**
6	0	1	1	0	**0**	14	1	1	1	0	**1**
7	0	1	1	1	**1**	15	1	1	1	1	**1**

To minimize the given function, let's select sets of variables for which the function receives the values of unit:

$$F = \{0011, 0100, 0101, 0111, 1001, 1101, 1110, 1111\}. \quad (16)$$

Let's separate the sets of variables into groups depending on the number of units in them and carry out the gluing of variables in neighboring groups (Fig. 3).

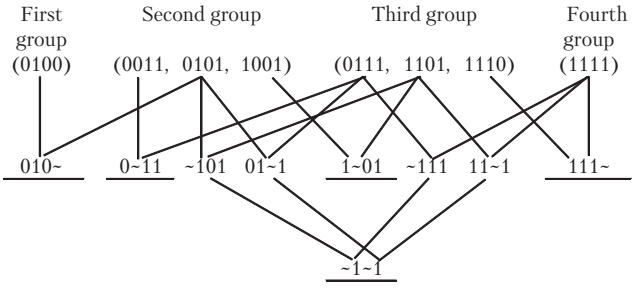


Fig. 3. Separation of the sets of variables of function (16) into groups by number of units in them

From Fig. 3, we see that 11 algebraic transformations (10 gluing and 1 idempodentity of variables) are used in two stages of gluing of variables. Emphasized implicants form a *Z*-covering:

$$Z = \begin{cases} 010\sim; \\ 0\sim 11; \\ 1\sim 01; \\ 111\sim; \\ \sim 1\sim 1. \end{cases}$$

Thus, DDNF of the function F (16) is obtained:

$$F_{DNF} = \overline{x}_1 x_2 \overline{x}_3 + \overline{x}_1 x_3 x_4 + x_1 \overline{x}_3 x_4 + x_1 x_2 x_3 + x_2 x_4.$$

To remove excess implicants and to obtain MDNF, a covering table is constructed (Table 14).

Table 14

Covering table of the function $F(x_1, x_2, x_3, x_4)$

Implicants	Constituents							
	0011	0100	0101	0111	1001	1101	1110	1111
010~	–	●	●	–	–	–	–	–
0~11	●	–	–	●	–	–	–	–
1~01	–	–	–	–	●	●	–	–
111~	–	–	–	–	–	–	●	●
~1~1	–	–	●	●	–	●	–	●

It is necessary to select the minimum number of rows that cover all columns. The solution to the problem of selecting columns containing one label begins. This is column 0011: in the decision it is necessary to accept implicants 0~11, otherwise constituent 0011 will not be covered (will not enter the solution).

From the covering table 14 it turns out that the implicant ~1~1 is superfluous. Thus, the MDNF of the given function (16) has the form (17):

$$F_{DNF} = \overline{x}_1 x_2 \overline{x}_3 + \overline{x}_1 x_3 x_4 + x_1 \overline{x}_3 x_4 + x_1 x_2 x_3. \quad (17)$$

The minimization of the logical function (16) by combinatorial method looks like this:

$$F = \begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} & & & \\ & & & \\ 0 & 1 & 0 & \\ 0 & & 1 & 1 \\ & & & \\ 1 & & 0 & 1 \\ & & & \\ 1 & 1 & 1 & \end{vmatrix}.$$

In the first matrix there are 4 procedures of gluing of variables – only 4 algebraic transformations. The minimized function by combinatorial method coincides with the expression (17).

Example 6. To minimize the 5-bit Boolean function by the Quine-McCluskey method. The output function is specified by a truth table (Table 10). To minimize the given function, let's select sets of variables for which the function receives the values of unity

$$F = \{00000, 00101, 00111, 01011, 01100, 01101, 01111, 10000, 10101, 10110, 10111, 11000, 11100, 11101, 11110, 11111\}. \quad (18)$$

Let's separate the sets of variables into groups depending on the number of units in them and carry out the gluing of variables in neighboring groups (Fig. 4).

The resulting DDNF of the function F (18) has the form (19):

$$F_{DNF} = \overline{x}_2 \overline{x}_3 \overline{x}_4 \overline{x}_5 + x_1 \overline{x}_3 \overline{x}_4 \overline{x}_5 + x_1 x_2 \overline{x}_4 \overline{x}_5 + \overline{x}_1 x_2 x_4 x_5 + x_3 x_5 + x_2 x_3 \overline{x}_4 + x_1 x_3 x_4 + x_1 x_2 x_3. \quad (19)$$

To remove excess implicants and to obtain MDNF, a covering table is constructed (Table 15).

It is necessary to select the minimum number of columns that cover all rows. From the covering Table 15 it turns out that the implicants 1~000 and 111~ are superfluous. The corresponding cells of Table 15 at the intersection of the column (with implicants 1~000, 111~) and the row (with constituent unit) are indicated by the green symbol ●. The cells of Table 15 at the intersection of the column with implicants, which are included in the minimum function and the rows (with constituent unit) are indicated by the blue symbol ●.

Thus, MDNF of the given function (18) has the form (20), which coincides with (12):

$$F_{DNF} = \overline{x}_2 \overline{x}_3 \overline{x}_4 \overline{x}_5 + x_1 x_2 \overline{x}_4 \overline{x}_5 + \overline{x}_1 x_2 x_4 x_5 + x_3 x_5 + x_2 x_3 \overline{x}_4 + x_1 x_3 x_4. \quad (20)$$

Based on the results of examples 4–6, let's calculate the number of consumed algebraic transformations by the Quine-McCluskey method performed at the stages of gluing, absorption, and idempotence of variables, while minimizing the logical function. Let's also establish an estimate of the algorithmic complexity of the search for a minimal function by this method.

ISSN 2226-3780

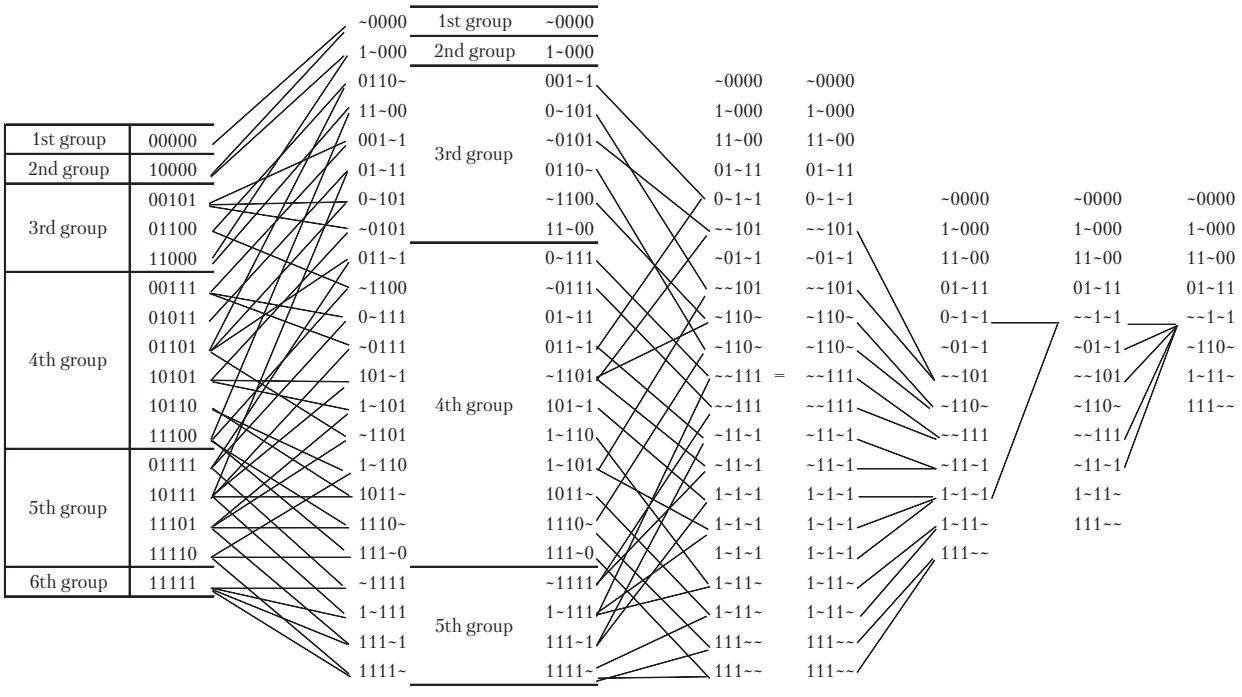


Fig. 4. Separation of the sets of variables of the function (18) into groups with the following procedures for gluing, absorbing and idem potency of implicants

Table 15

Covering table of the function $F(x_1, x_2, x_3, x_4, x_5)$

Constituents	Implicants							
	~ 0000	1 ~ 000	11 ~ 00	01 ~ 11	~~ 1 ~ 1	~ 110 ~	1 ~ 11 ~	111 ~~
00000	●	–	–	–	–	–	–	–
10000	●	●	–	–	–	–	–	–
00101	–	–	–	–	●	–	–	–
01100	–	–	–	–	–	●	–	–
11000	–	●	●	–	–	–	–	–
00111	–	–	–	–	●	–	–	–
01011	–	–	–	●	–	–	–	–
01101	–	–	–	–	●	●	–	–
10101	–	–	–	–	●	–	–	–
10110	–	–	–	–	–	–	●	–
11100	–	–	●	–	–	●	–	●
01111	–	–	–	–	●	–	–	–
10111	–	–	–	–	●	–	●	–
11101	–	–	–	–	●	●	–	●
11110	–	–	–	–	–	–	●	●
11111	–	–	–	–	●	–	●	●

The results of calculating the number of algebraic transformations are presented in Table 16.

Fig. 5 shows the dynamics of the growth in the number of algebraic transformations that occur when the logical function is minimized by the Quine-McCluskey method and by the combinatorial method with increasing the bit capacity of the function.

Table 16

Comparative table of spent algebraic transformations for two methods of minimizing the function $F(x_1, x_2, x_3, x_4, x_5)$

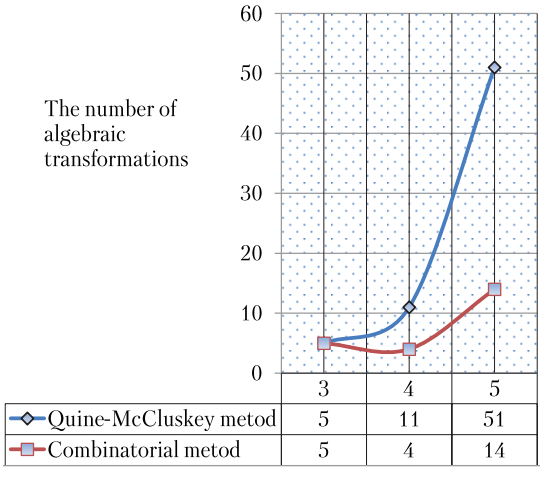
Bit depth of function	The number of algebraic transformations	
	Quine-McCluskey method	Combinatorial method
3	5	5
4	11	4
5	51	14

Taking into account Fig. 5 we see that the dynamics of the growth of the number of algebraic transformations, with the increase in the bit depth of the logical function, for a combinatorial minimization method by a slower process in comparison with dynamics.

The growth of the number of algebraic transformations in the Quine-McCluskey method. Thus, the search for a minimal function by a combinatorial method is more efficient than searching using the Quine-McCluskey method.

The complexity of the Quine-McCluskey minimization method increases exponentially with the increase in the bit of the input variables [25].

According to the data that we have at our disposal, the complexity of the search algorithm by a combinatorial method can be described as a first approximation linearly on the number of algebraic transformations with an $O(n)$ complexity rating.



	3	4	5
Quine-McCluskey metod	5	11	51
Combinatorial metod	5	4	14

Fig. 5. Dynamics of growth of algebraic transformations when the function is minimized by the Quine-McCluskey method and by the combinatorial method, with the increase in the bit of the function

7. SWOT analysis of research results

Strengths. The strength of the combinatorial method is that the object of solving the problem of minimizing a Boolean function is a block-diagram with repetition, what is the truth table of this function. This allows to concentrate the minimization principle within the function calculation protocol and, thus, dispense with auxiliary objects like the Karnaugh map, Veitch diagram, acyclic graph, etc. The equivalent transformations by graphic images have a large information capacity, with the properties replace the verbal procedures of algebraic transformations. The increased information capacity of the combinatorial method makes it possible to carry out manual minimization of 4, 5-bit Boolean functions quite easily.

This is more advantageous in comparison with analogues for the following factors:

– lower cost of development and implementation, since the principle of minimization of the method remains within the truth table of this function and does not require other auxiliary objects;

– Increasing the performance of the manual minimization procedure for 4-, 5-bit functions and increasing the performance of automated minimization with a greater number of variable functions, in particular due to the fact that several search options give the same minimum function.

Weaknesses. The weak side of the combinatorial method with manual minimization is associated with an increase in the number of variables (more than seven or eight) of the logical function. With such number of variables, the laboriousness of calculating manual minimization increases.

Negative internal factors inherent in the combinatorial method of manual minimization of the Boolean function consist in increasing the time of obtaining the minimal function with increasing number of variables of the given function.

Opportunities. The opportunity of further studies of the combinatorial method can be the development of a protocol for the optimal alternation of algebraic transformations over implicit Boolean functions, with the aim of further optimizing the execution time of the search algorithm of the minimal function.

Additional features that connected with implementation of the combinatorial method for minimizing Boolean functions is the use and support of the submatrix library, which will help optimize the response time for the search algorithm for minimizing the function.

Threats. The minimization protocol for the Boolean function of the combinatorial method is independent of the protocols of other minimization methods, therefore there is no threat of negative impact on the object of research of external factors.

To a certain extent, the Quine-McClusky method is an analog of the combinatorial method for minimizing the Boolean function. At the moment, the Quine-McCluskey method is the best because an algorithm for automating the search for a minimal function has already been created for it.

8. Conclusions

1. It is established that the object of solving the problem of minimizing a Boolean function is a combinatorial block-diagram with repetition, what is the truth table of this function. This allows to focus the minimization principle within the function calculation protocol and, thus, do without auxiliary objects of the minimum function search.

2. It is revealed that the tabular organization of the mathematical apparatus of the block-diagram with repetition allows obtain more information about the orthogonality, contiguity, uniqueness of the blocks of the combinatorial system, and, consequently, blocks of the truth table of the given function. Equivalent transformations by graphic images, in their properties have a large information capacity, capable of effectively replacing verbal procedures of algebraic transformations, in particular using the library of submatrices.

3. It is established that the results of verification of the minimized function obtained by the combinatorial method satisfy the output protocol of calculating the given function and, therefore, indicate the optimal decrease in the number of function variables without losing its functionality. The complexity of the algorithm for searching for a minimal function by a combinatorial method is $O(n)$ and is linear – the execution time of the algorithm increases linearly with increasing the bit length of the function n.

4. The efficiency of the combinatorial method is demonstrated by examples of minimizing functions borrowed from the work of other authors for the purpose of comparison: Example 1 [17, art. 184], – minimization of the 4-bit Boolean function, examples 2 [19], 3 [20] – minimization of 5-bit Boolean functions. Taking these examples into account, the combinatorial method of minimizing a function gives grounds for the expediency of applying it in procedures for minimizing a logical function.

It is established that the growth dynamics of the number of algebraic transformations, with increasing bit depth of the logical function, for a combinatorial minimization method by a slower process in comparison with the growth dynamics of the number of algebraic transformations by the Quine-McCluskey method. In this regard, the com-

ISSN 2226-3780

binatorial method is more efficient than the search using the Quine-McCluskey method.

References

1. Matviienko, M. P. Kompiuterna lohika [Text] / M. P. Matviienko. – Kyiv: TOV »Tsentr navchalnoi literatury», 2012. – 288 p.
2. Igoshin, V. I. Matematicheskaia logika i teoriia algoritmov [Text]: Handbook / V. I. Igoshin. – Moscow: Izdatel'skii tsentr «Akademiia», 2007. – 304 p.
3. Kutiura, L. Algebra logiki [Text] / L. Kutiura. – Moscow: Librokom, 2011. – 128 p.
4. Kolmogorov, A. N. Matematicheskaia logika [Text] / A. N. Kolmogorov, A. G. Dragalin. – Ed. 3. – Moscow: KomKniga, 2006. – 240 p.
5. Venn, J. I.On the diagrammatic and mechanical representation of propositions and reasonings [Text] / J. Venn // Philosophical Magazine Series 5. – 1880. – Vol. 10, No. 59. – P. 1–18. doi:10.1080/14786448008626877
6. Nelson, V. P. Digital Logic Circuit Analysis and Design [Text] / V. P. Nelson, H. Troy Nagle, B. D. Carroll, D. Irwin. – Pearson, 1995. – 842 p.
7. Manojlovic, V. Minimization of Switching Functions using Quine-McCluskey Method [Text] / V. Manojlovic // International Journal of Computer Applications. – 2013. – Vol. 82, No. 4. – P. 12–16. doi:10.5120/14103-2127
8. Rytsar, B. The Minimization Method of Boolean Functions in Polynomial Set-theoretical Format [Electronic resource] / B. Rytsar // Conference: Proc. 24th Inter. Workshop, CS@P'2015, Sept. 28–30, 2015. – Poland, Rzeszow, 2015. – Vol. 2. – P. 130–146. – Available at: \www/URL: http://dspace.nbuv.gov.ua/handle/123456789/87194
9. Rathore, T. S. Minimal Realizations of Logic Functions Using Truth Table Method with Distributed Simplification [Text] / T. S. Rathore // IETE Journal of Education. – 2014. – Vol. 55, No. 1. – P. 26–32. doi:10.1080/09747338.2014.921412
10. Rotar, D. Software for The Minimization of The Combinational Logic Functions [Text] / D. Rotar // The Romanian Review Precision Mechanics, Optics & Mechatronics. – 2010. – Vol. 37. – P. 95–99.
11. Bernasconi, A. Three-level logic minimization based on function regularities [Text] / A. Bernasconi, V. Ciriani, F. Luccio, L. Pagli // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2003. – Vol. 22, No. 8. – P. 1005–1016. doi:10.1109/tcad.2003.814950
12. Zolfaghari, B. A New Case for Image Compression Using Logic Function Minimization [Text] / B. Zolfaghari, H. Sheidaeian // The International journal of Multimedia & Its Applications. – 2011. – Vol. 3, No. 2. – P. 45–62. doi:10.5121/ijma.2011.3204
13. Mohana Ranga Rao, R. An Innovative procedure to minimize Boolean function [Text] / R. Mohana Ranga Rao // International Journal of Advanced Engineering Sciences and Technologies. – 2011. – Vol. 3, No. 1. – P. 12–14.
14. Nosrati, M. Minimization of Boolean Functions Using Genetic Algorithm [Text] / M. Nosrati, R. Karimi, M. Nariri // Annals. Computer Science Series. – 2012. – Vol. 10, No. 1. – P. 73–77.
15. Nosrati, M. An Algorithm for Minimizing of Boolean Functions Based on Graph DS [Text] / M. Nosrati, M. Nariri // World Applied Programming. – 2011. – Vol. 1, No. 3. – P. 209–214.
16. Solairaju, A. Optimal Boolean Function Simplification through K-Map using Object-Oriented Algorithm [Text] / A. Solairaju, R. Periyasamy // International Journal of Computer Applications. – 2011. – Vol. 15, No. 7. – P. 28–32. doi:10.5120/1959-2621
17. Buniak, A. Elektronika ta mikroskhemotekhnika [Text] / A. Buniak. – Ternopil: Aston, 2001. – 382 p.
18. Tonchev, V. Kombinatornye konfiguratsii. Blok-shemy, kody, grafy [Text] / V. Tonchev. – Kyiv: Holovne vydavnytstvo vydavnychoho obiednannia «Vyshcha shkola», 1988. – 178 p.
19. Plehanov, A. Simmetrichnye karty kak sredstvo minimizatsii bulevyh funktsii [Electronic resource] / A. Plehanov // Geektimes. – March 8, 2016. – Available at: \www/URL: https://geektimes.ru/post/272294/
20. Sudnitson, A. Diskretnaia matematika. F.4. Minimizatsiia bulevyh funktsii [Electronic resource] / A. Sudnitson. – 2008. – Available at: \www/URL: http://ati.ttu.ee/~alsu/DM%20_MinBF_2008_lecture.pdf. – 15.07.2017.
21. Kudriavtsev, V. B. O slozhnosti algoritmov [Electronic resource] / V. B. Kudriavtsev, A. E. Andreev // Intellektual'nye sistemy. – 2006. – Vol. 10, No. 1-4. – P. 695–760. – Available at: \www/URL: http://intsys.msu.ru/magazine/archive/v10(1-4)/andreev-695-760.pdf
22. Nechiporuk, E. I. O korrektnosti obryvov v ventil'nyh i kontaktnyh shemah [Text] / E. I. Nechiporuk // Kibernetika. – 1968. – Vol. 5. – P. 40–48.
23. Redkin, N. P. O samokorrektirovanii kontaktnyh shem [Text] / N. P. Redkin // Problemy kibernetiki. – 1978. – Vol. 33. – P. 119–138.
24. Kirienko, G. I. Sintez samokorrektiruiushchihsia shem iz funktsional'nyh elementov dlia sluchaia rastushchego chisla oshibok v sheme [Text] / G. I. Kirienko // Diskretnyi analiz. – 1970. – Vol. 16. – P. 38–43.
25. Serikov, Yu. A. Algebraicheskii metod resheniia logicheskih uravnenii [Text] / Yu. A. Serikov // Izvestiia AN SSSR. Tehnicheskaia kibernetika. – 1972. – Vol. 2. – P. 114–124.
26. Zhabin, V. I. Prykladna teoriia tsyfrovykh avtomativ [Text] / V. I. Zhabin, I. A. Zhukov, I. A. Klymenko, V. V. Tkachenko. – Ed. 2. – Kyiv: Vydavnytstvo Natsionalnoho aviatsiinoho universytetu «NAU – druk», 2009. – 360 p.
27. Bitiutskii, V. P. Minimizatsiia perekliuchatel'nyh funktsii [Electronic resource]: Educational electronic text edition / V. P. Bitiutskii, S. V. Grigorieva. – Ekaterinburg: UrFU, 2012. – 22 p. – Available at: \www/URL: http://docplayer.ru/46853788-Minimizaciya-pereklyuchatelnyh-funkciy.html
28. Amerbaev, V. M. Library implementation of modular arithmetic operations, based on logic functions minimization algorithms [Electronic resource] / V. M. Amerbaev, R. A. Solovyev, D. V. Telpukhov // Izvestiya SFedU. Engineering Sciences. – 2013. – Vol. 7. – P. 221–225. – Available at: \www/URL: http://izv-tn.tti.sfedu.ru/wp-content/uploads/2013/7/40.pdf

МИНИМИЗАЦИЯ БУЛЕВЫХ ФУНКЦИЙ КОМБИНАТОРНЫМ МЕТОДОМ

Рассмотрено распространение принципа минимизации с помощью алгебраических преобразований на метод минимизации с использованием комбинаторной блок – схемы с повторением. Математический аппарат блок-схемы с повторением даёт больше информации относительно ортогональности, смежности, однозначности блоков комбинаторной системы, которой собственно является таблица истинности заданной функции, поэтому применение такой системы минимизации функции есть более эффективным.

Ключевые слова: булева функция, метод минимизации, минимизация логической функции, блок-схема с повторением, минтерм.

Riznyk Volodymyr, *Doctor of Technical Sciences, Professor, Department of Control Aided Systems, Lviv Polytechnic National University, Ukraine, e-mail: rvv@polynet.lviv.ua, ORCID: http://orcid.org/0000-0002-3880-4595*

Solomko Mykhailo, *PhD, Associate Professor, Department of Computer Engineering, National University of Water and Environmental Engineering, Rivne, Ukraine, e-mail: doctrinas@ukr.net, ORCID: http://orcid.org/0000-0003-0168-5657*