BOOLEAN FUNCTIONS MINIMIZATION BY THE METHOD OF FIGURATIVE TRANSFORMATIONS

Mykhailo Solomko
PhD, Associate Professor
Department of Computer Engineering
National University of Water and Environmental Engineering
11 Soborna str., Rivne, Ukraine, 33028
doctrinas@ukr.net

Abstract: The object of research is the method of figurative transformations for Boolean functions minimization. One of the most problematic places to minimize Boolean functions is the complexity of the minimization algorithm and the guarantee of obtaining a minimal function.

During the study, the method of equivalent figurative transformations was used, which is based on the laws and axioms of the algebra of logic; minimization protocols for Boolean functions that are used when the truth table of a given function has a complete binary combinatorial system with repetition or an incomplete binary combinatorial system with repetition.

A reduction in the complexity of the minimization process for Boolean functions is obtained, new criteria for finding minimal functions are established. This is due to the fact that the proposed method of Boolean functions minimization has a number of peculiarities of solving the problem of finding minimal logical functions, in particular:

– mathematical apparatus of the block diagram with repetition makes it possible to obtain more information about the orthogonality, adjacency, uniqueness of the truth table blocks;

– equivalent figurative transformations due to the greater information capacity are capable of replacing verbal procedures of algebraic transformations;

– result of minimization is estimated based on the sign of the minimum function;

– minimum DNF or CNF functions are obtained regardless of the given normal form of the logical function, which means that it is necessary to minimize the given function for two normal forms – DNF and CNF using the full truth table;

This ensures that it is possible to obtain an optimal reduction in the number of variables of a given function without losing its functionality. The effectiveness of the use of equivalent figurative transformations for Boolean functions minimization is demonstrated by examples of minimization of functions borrowed from other methods for the purpose of comparison.

Compared with similar well-known methods of Boolean functions minimization, this provides:

– less complexity of the minimization procedure for Boolean functions;

– guaranteed Boolean functions minimization;

– self-sufficiency of the specified method of Boolean functions minimization due to the introduction of features of the minimal function and minimization of two normal forms – DNF and CNF on the complete truth table of a given Boolean function.

Keywords: Boolean functions minimization by figurative transformations, DNF, CNF, combinatorial block diagram with repetition, minterm, maxterm.

1. Introduction

Boolean functions minimization is popular in several areas of digital technology, in particular, such as PLA design, built-in self test (BIST), design of control systems, and the like.

The problems and shortcomings of the known methods for Boolean functions minimization is popular in several areas of digital technology, in particular, such as PLA design, built-in self test (BIST), design of control systems, and the like.

The problems and shortcomings of the known methods for Boolean functions minimization are associated with a rapid increase in the amount of computation, which results in an increase in the digit capacity of computational operations, and, consequently, an increase in the number of variable logical functions.

Boolean function $f(x_1,...,x_n)$ describing a logical device can be implemented using a disjunctive normal form (DNF) or conjunctive normal form (CNF), which in this case will describe the scheme of the corresponding logical device. The problem of minimizing DNF or CNF is one of the most flexible logical combinatorial problems and reduces to the optimal reduction of the number of logical elements of the gate circuit without losing its functionality. The speed of the computing device, its reliability and energy saving depend on the result of Boolean functions minimization.

Karnaugh map is usually difficult to recognize with an increase in the number of variables more than four or five, so this method is not advisable to use with a large number of variables. Despite the great perfection of the Quine-McCluskey method compared to Karnaugh maps, it also has limited practical applications due to the exponential increase in the computation time with an increase in the number of variables. It can be shown that for a function of n variables the upper limit of the number of basic implicants is equal to $3^n \ln(n)$ [1]. For example, it is known that for n=32 the number of basic implicants can exceed $6{,}5\times10^{15}$.

This paper presents the Boolean functions minimization in the class of DNF and CNF by the method of figurative transformations, the application of which gives the rules of logic algebra, allows to set the sign of the minimal function, gives a minimization hyperparameter of two normal forms – DNF and CNF of a given function using the full truth table.

Analysis of publications and problem statement. In [2], a procedure for simplifying a logic function is considered, after which, at the end of each stage, the truth table is shortened. It is shown that the method is systemic and unconditionally leads to a minimal function. It is simpler than based only on Boolean place names, Karnaugh maps, Quine – McCluskey and can handle any number of variables. This is explained by several examples.

The use of a genetic algorithm to select side objects of the procedure for minimizing a logical function using the Karnaugh map is demonstrated in [3].

In [4], the minimization of CNF of Boolean functions is considered and the complexity of solving this problem is analyzed. It is known that the question of whether there is a shorter CNF for a function defined as a CNF has n^2 complexity for general formulas. However, for certain classes of formulas, the complexity of minimizing CNF is different. In [4], a class of CNF formulas that can be recognized in polynomial time is considered.

In contrast to the considered literature sources, in this work, the object of solving the problem is Boolean functions minimization by the method of figurative transformations, if there is a complete or incomplete binary combinatorial systems with repetition in the structure of the truth table.

The mathematical apparatus of the block diagram with repetition makes it possible to obtain more information about the orthogonality, adjacency, uniqueness of the truth table blocks. Equivalent figurative transformations in the form of two-dimensional matrices by their properties have a large information capacity, therefore they are capable of replacing verbal procedures of algebraic transformations with effect.

The evolution of methods to simplify logical functions is the result of relentless optimization, so research remains relevant, in particular, to improve factors such as the methodology for minimizing logical functions in the class of DNF and CNF, establishing signs of a minimum function, and cost of minimizing technology of logical functions.

The aim of research is simplification of the Boolean functions minimization process using the figurative transformation method.

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

1. To establish the adequacy of the application of the method of figurative transformations to minimize the DNF and CNF of the Boolean functions.
2. To determine the equivalent figurative transformations of the minterm and the disjunctive monomial to minimize the DNF and CNF of the Boolean functions.
3. To set the sign of obtaining the minimum logical function.
4. To establish the feasibility of minimizing the two normal forms – DNF and CNF of a given Boolean function, using the full truth table.

2. Methods of research

2. 1. Binary combinatorial system with repetition

The number of all k-element subsets of a set of n elements is:

$$N(M_k(A)) = C_n^k = \frac{n!}{k!(n-k)!}.$$

Equality also holds:

$$\sum_{k=0}^{n} C_n^k = 2^n. \quad (1)$$

Since C_n^k – the number of k-element subsets of a set of n elements, the sum on the left side of expression (1) is the number of all subsets.

Let's note that the set A={a, b, c, d}, in addition to recalculating its elements, can also determine the position numbers at which the element is located. So, for example, a can mean the first position, b can mean the second position of the set A=={a, b, c, d}, and so on. D. A subset of the set A={a, b, c, d}, in this case, there will be subsets containing the element a at k positions, k=0,..., n, where n is the number of positions of the set A. In the general case, the element can occupy several positions on the set A, thus the element is repeated on the set A.

Let α=1, then positions at which the element α is absent are affected by zero.

Example 1. For the set A={a, b, c, d}, which determines the position numbers, take α=1. Then the subsets of the set A will have the following form:

$$\begin{matrix} (0,0,0,0); & (0,1,0,0); & (1,0,0,0); & (1,1,0,0); \\ (0,0,0,1); & (0,1,0,1); & (1,0,0,1); & (1,1,0,1); \\ (0,0,1,0); & (0,1,1,0); & (1,0,1,0); & (1,1,1,0); \\ (0,0,1,1); & (0,1,1,1); & (1,0,1,1); & (1,1,1,1). \end{matrix} \quad (2)$$

The configuration (2) is a complete combinatorial system with the element α repeated, which denote as:

2-(*n*, *b*)-design, (3)

where n – the digit capacity of the system block; b – the number of blocks of the complete system, which is determined by the formula – $b = 2^n$, the number 2 in front of the parentheses means the binary structure of the configuration (2). For example, 2 – (4, 16)-design is a complete binary combinatorial system with repetition, consisting of 4-bit blocks, the number of blocks – 16.

In the general case, the configuration of the truth table of this function, besides the submatrix of a complete combinatorial system with repetition (3), also contains submatrices of an incomplete combinatorial system with repetition (4)

2-(n, x/b)-design, (4)

where x – the number of blocks of an incomplete combinatorial system with repetition. The properties of an incomplete combinatorial system with repetition (4) also make it possible to establish rules that ensure the effective Boolean functions minimization.

2. 2. Equivalent figurative transformations of DNF and CNF of Boolean functions

The procedure of minimization by the method of figurative transformations uses the following rules of algebra of logic:

– gluing of variables – $ab + a\bar{b} = a$,
– generalized gluing of variables – $xy + \bar{x}z = xy + \bar{x}z + yz$,
– variable substitution – $a + \bar{a}b = a + b$,
– variable absorption – $ab + a = a(b+1) = a$,
– idempotency of variables – $a + a = a, \; aa = a$,
– variable addition – $a + \bar{a} = 1, \; a\bar{a} = 0$,
– repeating a constant – $a + 0 = a, \; a \cdot 1 = a$,
and other.

Algebraic transformations can be replaced by equivalent figurative transformations in the form of submatrices of the truth table.

The rule of gluing of variables for a DNF of logical expression.

$$x_1x_2 + \overline{x_1}x_2 + x_2(x_1 + \overline{x_1}) = x_2. \quad (5)$$

Equivalent figurative transformations for the rule of gluing of DNF of logical expression (5) have an illustration of the image (6):

$$\begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix} \to \begin{vmatrix} \sim & 1 \\ \sim & 1 \end{vmatrix} \to | \sim \; 1 |$$

or

$$\begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix} \to | \sim \; 1 |. \quad (6)$$

The rule of gluing of variables for a CNF of logical expression.

This rule is a consequence of the distribution law of the 2^{nd} kind.

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

Equivalent figurative transformations for the rule of the CNF gluing of a logical expression (7) have an illustration of the image (8):

$$\begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{vmatrix} = | 1 \; 1 \; 0 \; \sim |. \quad (8)$$

Equivalent figurative transformations can also be represented for other algebraic operations of the algebra of logic [5, 6].

If in conjunctive normal form (CNF) of logical function (9)

$$F = (\bar{x}_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_1 + \bar{x}_2 + x_3)(\bar{x}_1 + x_2 + x_3) \times$$
$$\times (x_1 + \bar{x}_2 + \bar{x}_3)(x_1 + \bar{x}_2 + x_3)(x_1 + x_2 + x_3). \quad (9)$$

variables with inversion replace with "0", and variables without inversion replace with "1", then it is possible to obtain the binary equivalent of the expression of a logical function (10)

$$F=(0+0+0)(0+0+1)(0+1+1)(1+0+0)(1+0+1)(1+1+1). \quad (10)$$

The expression (10) is represented by the matrix (11).

$$F_{CNF} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{vmatrix}. \quad (11)$$

Disjunctive normal form (DNF) of logic function (12)

$$F = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2 x_3 + \bar{x}_1 x_2 x_3 +$$
$$+ x_1\bar{x}_2\bar{x}_3 + x_1\bar{x}_2 x_3 + x_1 x_2 x_3 \quad (12)$$

can be represented by binary codes (13)

$$F=000+001+011+100+101+111 \quad (13)$$

or matrix (14)

$$F_{DNF} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{vmatrix}. \quad (14)$$

Considering the records (11) and (14), it can be seen that CNF and DNF of logical functions are represented by matrices with the same combinatorial structures. The difference between these matrices lies in the hermeneutics of logical operations. The matrix (11), which reflects the CNF of a logical function, provides the disjunctive monomial of the function and conjunction operations for them. The matrix (14), which reflects the DNF of a logical function, provides the minterm functions and disjunction operations for them.

2. 3. Minimization protocols for Boolean functions

For 4-bit logic functions, the protocols for super-gluing of variables will be as follows:

– first protocol:

$$\begin{vmatrix} 0 & 0 & 0 & x \\ 0 & 0 & 1 & x \\ 0 & 1 & 0 & x \\ 0 & 1 & 1 & x \\ 1 & 0 & 0 & x \\ 1 & 0 & 1 & x \\ 1 & 1 & 0 & x \\ 1 & 1 & 1 & x \end{vmatrix} = x; \quad (15)$$

– second protocol:

$$\begin{vmatrix} 0 & 0 & x & y \\ 0 & 1 & x & y \\ 1 & 0 & x & y \\ 1 & 1 & x & y \end{vmatrix} = xy; \quad (16)$$

– third protocol:

$$\begin{vmatrix} 0 & x & y & z \\ 1 & x & y & z \end{vmatrix} = xyz. \quad (17)$$

The first protocol uses 2 (3, 8)-design. The second protocol uses 2 (2, 4)-design. The third rule uses 2 (1, 2)-design.

The procedure for reducing the total perfect disjunctive normal form (PDNF) of the logical function gives one. For example, the abbreviation of 3-bit full PDNF looks like this:

$$\bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2 x_3 + \bar{x}_1 x_2\bar{x}_3 + \bar{x}_1 x_2 x_3 +$$
$$+ x_1\bar{x}_2\bar{x}_3 + x_1\bar{x}_2 x_3 + x_1 x_2\bar{x}_3 + x_1 x_2 x_3 =$$
$$= \bar{x}_1\bar{x}_2(\bar{x}_3 + x_3) + \bar{x}_1 x_2(\bar{x}_3 + x_3) + x_1\bar{x}_2(\bar{x}_3 + x_3) +$$
$$+ x_1 x_2(\bar{x}_3 + x_3) = \bar{x}_1\bar{x}_2 + \bar{x}_1 x_2 + x_1\bar{x}_2 + x_1 x_2 =$$
$$= \bar{x}_1(\bar{x}_2 + x_2) + x_1(\bar{x}_2 + x_2) = \bar{x}_1 + x_1 = 1.$$

Since a complete DNF uniquely identifies a complete combinatorial system with 2 (n, b)-design repetition and vice versa, this gives grounds to delete all the blocks of the complete combinatorial system from matrices that demonstrate super-gluing protocols (15)–(17). Further, by applying the law of idempotency to the variable, the remaining – x (xy; xyz) let's obtain the result of the reduction according to the protocol of super-gluing of the variables. Protocol (17) manifests itself as simple gluing of variables and is a special case of protocols (15) and (16).

The variables x, y, z, forming a complete combinatorial system with a repetition of 2 (n, b)-design, can occupy any bit of the minterm of a logic function.

Similar to the protocols of super-gluing of variables (15)–(17) for 4-bit functions, one can imagine super gluing protocols for functions of five or more variables.

In the general case, the configuration of the truth table of this function, in addition to the submatrix of a complete combinatorial system with repetition 2 (n, b)-design, also contains submatrices of an incomplete combinatorial system with repetition 2 (n, x/b)-design. The properties of an incomplete combinatorial system with the repetition of 2 (n, x/b)-design also make it possible to establish protocols that ensure the effective Boolean functions minimization [5–7].

3. Research results

3. 1. Minimization of DNF of logical functions by figurative transformations

Example 2. Minimize the logical function (18) by the algebraic method.

$$F(a,b,c,d) = \Sigma(3,7,11,12,13,14,15). \quad (18)$$

Note: the value in Σ is the minterm for rows when the function $F(a,b,c,d)$ returns "1" on the output (**Table 1**).

Table 1
Truth table of the logical function $F(a,b,c,d)$

No.	a	b	c	d	a	b	c	d	F
3	0	0	1	1	$\bar{a}$	$\bar{b}$	c	d	1
7	0	1	1	1	$\bar{a}$	b	c	d	1
11	1	0	1	1	a	$\bar{b}$	c	d	1
12	1	1	0	0	a	b	$\bar{c}$	$\bar{d}$	1
13	1	1	0	1	a	b	$\bar{c}$	d	1
14	1	1	1	0	a	b	c	$\bar{d}$	1
15	1	1	1	1	a	b	c	d	1

$$
\begin{aligned}
&F(a,b,c,d)=\Sigma(3,7,11,12,13,14,15)=\\
&=\bar{a}\bar{b}cd+\bar{a}bcd+a\bar{b}cd+ab\bar{c}\bar{d}+ab\bar{c}d+abc\bar{d}+abcd=\\
&=cd(\bar{a}\bar{b}+\bar{a}b+a\bar{b})+ab(\bar{c}\bar{d}+\bar{c}d+c\bar{d}+cd)=\\
&=cd(\bar{a}[\bar{b}+b]+a\bar{b})+ab(\bar{c}[\bar{d}+d]+c[\bar{d}+d])=\\
&=cd(\bar{a}[1]+a\bar{b})+ab(\bar{c}[1]+c[1])=ab+a\bar{b}cd+\bar{a}cd=\\
&=ab+cd(a\bar{b}+\bar{a})=ab+cd(a+\bar{a})(\bar{a}+\bar{b})=ab+\bar{a}cd+\bar{b}cd=\\
&=ab+cd(\bar{a}+\bar{b})=ab+cd.
\end{aligned}
$$

Minimization of the logical function (18) by figurative transformations looks like this:

$$
F=\begin{vmatrix}3 & 0 & 0 & 1 & 1\\ 7 & 0 & 1 & 1 & 1\\ 11 & 1 & 0 & 1 & 1\\ 12 & 1 & 1 & 0 & 0\\ 13 & 1 & 1 & 0 & 1\\ 14 & 1 & 1 & 1 & 0\\ 15 & 1 & 1 & 1 & 1\end{vmatrix}=\begin{vmatrix} & 0 & 1 & 1\\ 0 & 1 & 1 & 1\\ 1 & 1 & & \end{vmatrix}=
$$

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

Minimized function:

$$F=ab+cd.$$

The operation of super-gluing of variables in the first matrix is carried out for blocks 12-15, highlighted in red. The result of minimization by the combinatorial method coincides with the result of minimization obtained using the algebraic method; however, the process of minimizing a function by the combinatorial method is simple.

Example 3. Minimize the logical function $F(x_1,x_2,x_3,x_4)$, which are given by the following truth table, by figurative transformations [8].

$$F(x_1,x_2,x_3,x_4)=\Sigma(0,1,2,3,5,7,8,10,11,12,13).$$

In [8], the minimization of a function is reduced to the synthesis of an infimum disjunctive normal form (IDNF) of a logical function, using the perfect matrix placement (PMP) of a 4-dimensional cube E^4 (**Fig. 1**). The vertices of the cube E^4 of a given function, on which $F(x_1,x_2,x_3,x_4)=1$ are highlighted by shading. The shaded vertices correspond to the blocks of the truth table Σ (0, 1, 2, 3, 5, 7, 8, 10, 11, 12, 13) of a given logical function.

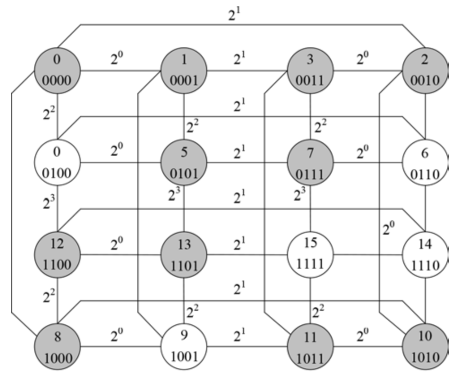

Fig. 1. Perfect matrix placement of a 4-dimensional cube E^4

The minimization of the function $F(x_1,x_2,x_3,x_4)$ by figurative transformations is reduced to the following procedure:

$$
F=\begin{vmatrix}0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 1\\ 2 & 0 & 0 & 1 & 0\\ 3 & 0 & 0 & 1 & 1\\ 5 & 0 & 1 & 0 & 1\\ 7 & 0 & 1 & 1 & 1\\ 8 & 1 & 0 & 0 & 0\\ 10 & 1 & 0 & 1 & 0\\ 11 & 1 & 0 & 1 & 1\\ 12 & 1 & 1 & 0 & 0\\ 13 & 1 & 1 & 0 & 1\end{vmatrix}=\begin{vmatrix} & 0 & 0 & 0\\ 0 & & 0 & 1\\ & 0 & 1 & \\ 0 & 1 & 1 & 1\\ 1 & 1 & 0 & \end{vmatrix}=
$$

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

Minimized function:

$$F=\bar{x}_2\bar{x}_4+\bar{x}_1x_4+\bar{x}_2x_3+x_1x_2\bar{x}_3. \qquad (19)$$

Blocks 2, 3, 10, 11 (highlighted in red) are minimized by the protocol of super-gluing of variables. Other blocks are minimized according to the protocols of simple gluing and semi-gluing [5, 6]. The result of the minimization of (19) coincides with the result of the synthesis of the infimum disjunctive normal form of the logical function [8], but the method of figurative transformations is a simple procedure.

Other comparative examples of Boolean functions minimization are presented in [5–7].

3. 2. Minimization of CNF of logical functions by figurative transformations

Example 4. Minimize the CNF of the function given by the PCNF:

$$
\begin{aligned}
&F(x_1,x_2,x_3,x_4,x_5)=\\
&=(x_1+x_2+x_3+x_4+x_5)(x_1+x_2+x_3+x_4+\bar{x}_5)(x_1+x_2+\bar{x}_3+x_4+x_5)\&\\
&\&(x_1+x_2+\bar{x}_3+\bar{x}_4+x_5)(x_1+\bar{x}_2+\bar{x}_3+x_4+x_5)(x_1+\bar{x}_2+\bar{x}_3+\bar{x}_4+x_5)\&\\
&\&(\bar{x}_1+x_2+x_3+x_4+x_5)(\bar{x}_1+x_2+x_3+x_4+\bar{x}_5)(\bar{x}_1+\bar{x}_2+\bar{x}_3+x_4+x_5)\&\\
&\&(\bar{x}_1+x_2+\bar{x}_3+\bar{x}_4+\bar{x}_5).
\end{aligned}
$$

This function returns zero in such sets: (0,0,0,0,0), (0,0,0,0,1), (0,0,1,0,0), (0,0,1,1,0), (0, 1, 1, 0, 0), (0, 1, 1, 1, 0), (1, 0, 0, 0, 0), (1, 0, 0, 0, 1), (1, 1, 1,0,0), (1, 0, 1, 1, 1).

Minimize the CNF of a given function $F(x_1, x_2, x_3, x_4, x_5)$ by figurative transformations.

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

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

Minimized CNF of the function $F(x_1, x_2, x_3, x_4, x_5)$.

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

The operation of super-gluing of variables in the first matrix is carried out for blocks, highlighted in red and blue. In the second matrix, the operation of semi-gluing of variables is performed.

3. 3. Sign of the minimum logical function

The establishment of signs of the minimum logical function is reduced to the minimization of a function from sets of truth tables for which the function returns "1" at the output and for sets of truth tables for which the function returns "0" at the output. With error-free calculations of the minimum function in two cases, the result of minimization will be the same. For this comparison, it is necessary to take into account the fact that a given logical function can have several minimal functions. In this connection, in some cases, the results of minimization of the logical function in DNF and CNF may differ, for example, in one variable, however, both minimized functions will be minimal.

Example 5. Minimize the logical function $F(x_1, x_2, x_3, x_4)$ specified in DNF (20) [9] and set the sign of the minimal function.

$$F(x_1, x_2, x_3, x_4) = \Sigma(0,1,2,3,5,8,10,12,13,14,15). \qquad (20)$$

$$F_{DNF} = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 1 \\ 5 & 0 & 1 & 0 & 1 \\ 8 & 1 & 0 & 0 & 0 \\ 10 & 1 & 0 & 1 & 0 \\ 12 & 1 & 1 & 0 & 0 \\ 13 & 1 & 1 & 0 & 1 \\ 14 & 1 & 1 & 1 & 0 \\ 15 & 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & & \\ 0 & 1 & 0 & 1 \\ 1 & 0 & & 0 \\ 1 & 1 & & \end{vmatrix} = \begin{vmatrix} 0 & 0 & & \\ & 1 & 0 & 1 \\ 1 & & & 0 \\ 1 & 1 & & \end{vmatrix}.$$

Minimized DNF of the function:

$$F_{DNF} = x_1 x_2 + \overline{x_1}\,\overline{x_2} + x_1 \overline{x_4} + x_2 \overline{x_3} x_4. \qquad (21)$$

The operation of super-gluing of variables in the first matrix is carried out for blocks 0–3 (highlighted in red) and 12–15 (highlighted in blue). A simple gluing operation is performed for blocks 8–11. In the second matrix semi-gluing of the variables.

Table 2 presents the results of minimization of the function $F(x_1, x_2, x_3, x_4)$ by the Quine method [9] and the method of figurative transformations.

Table 2

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

Minimization by Quine methods	Minimization by figurative transformations
$F = x_1x_2 + \overline{x_1}\,\overline{x_2} + \overline{x_2}\,\overline{x_4} +$ $+ x_1\overline{x_4} + \overline{x_1}\,\overline{x_3}x_4 + x_2\overline{x_3}x_4$	$F = x_1x_2 + \overline{x_1}\,\overline{x_2} +$ $+ x_1\overline{x_4} + x_2\overline{x_3}x_4$

Given the **Table 2** it is possible to see that figurative transformations give a minimal function with a smaller number of input variables.

Now let's minimize the given function in CNF (22) and by the Nelson method transform the minimum CNF into DNF.

$$F(x_1,x_2,x_3,x_4) = \Pi(4, 6, 7, 9, 11). \qquad (22)$$

Note: the value in Π is a maxterms for rows when the function $F(x_1, x_2, x_3, x_4)$ returns "0" on output.

$$F_{CNF} = \begin{vmatrix} 4 & 0 & 1 & 0 & 0 \\ 6 & 0 & 1 & 1 & 0 \\ 7 & 0 & 1 & 1 & 1 \\ 9 & 1 & 0 & 0 & 1 \\ 11 & 1 & 0 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix} =$$

$$= \begin{vmatrix} 1 & 0 & & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & & 0 \end{vmatrix} = \begin{vmatrix} 1 & 0 & & 1 \\ 1 & 0 & 0 & \\ 0 & 1 & & 0 \end{vmatrix}. \qquad (23)$$

According to the Nelson method, the result of minimization (the last matrix) of the entry (23) will be represented in the CNF of the minimal function

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

open brackets and transform it into a DNF of a minimal function.

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

$$F_{DNF} = (x_1 + \overline{x_2} + \overline{x_3}x_4)(\overline{x_1} + x_2 + \overline{x_4}) =$$
$$= x_1x_2 + x_1\overline{x_4} + \overline{x_1}\,\overline{x_2} + \overline{x_2}\,\overline{x_4} +$$
$$+ \overline{x_1}\,\overline{x_3}x_4 + x_2\overline{x_3}x_4 = x_1x_2 + x_1\overline{x_4} +$$
$$+ \overline{x_1}\,\overline{x_2} + \overline{x_2}\,\overline{x_4} + \overline{x_1}\,\overline{x_3}x_4 + x_2\overline{x_3}x_4. \qquad (24)$$

Simplification of the last expression (24) is carried out by the method of figurative transformations.

$$F_{DNF}\left|\begin{array}{cccc}1&1&&\\1&&&0\\0&0&&\\&0&&0\\0&&0&1\\&1&0&1\end{array}\right|=\left|\begin{array}{cccc}1&1&&\\1&&&0\\0&0&&\\0&&0&1\\&1&0&1\end{array}\right|=\left|\begin{array}{cccc}1&1&&\\1&&&0\\0&0&&\\&1&0&1\end{array}\right|.$$

It is obviously to use twice generalized gluing of variables. As a result, let's obtain a simplified logical expression (25).

$$F_{DNF}=x_1x_2+\overline{x_1x_2}+x_1\overline{x}_4+x_2\overline{x}_3x_4. \quad (25)$$

The expressions (21) and (25) coincide that, according to the sign of a minimal function, it means obtaining a procedure for minimizing of the minimum Boolean function.

3. 4. Boolean functions minimization on the complete truth table

The minimization of a DNF or CNF of Boolean functions is performed on the corresponding sets of truth table variables. Comparison of the results of minimization of the DNF and CNF of the function shows that the minimal function can be both in the DNF and in the CNF. It follows that the minimization of a given Boolean function must be carried out in two normal forms, DNF and CNF, using the full truth table of this function. A complete truth table contains sets of variables for which the function returns "1" and/or "0" at the output. The minimal function should be chosen according to the results of minimization of two normal forms – DNF and CNF.

Example 6. Minimize the logical function $F(x_1,x_2,x_3,x_4)$ on the complete truth table by figurative transformations in two normal forms – DNF and CNF, which is given in canonical form [10]:

$$F(x_1,x_2,x_3,x_4)=\Sigma(0,1,6,8,11,14,15). \quad (26)$$

The minimal function is chosen according to the results of minimization of two normal forms – DNF and CNF.

Minimization of DNF of a given function:

$$F_{DNF}=\left|\begin{array}{c}0\\1\\6\\8\\11\\14\\15\end{array}\right|\begin{array}{cccc}0&0&0&0\\0&0&0&1\\0&1&1&0\\1&0&0&0\\1&0&1&1\\1&1&1&0\\1&1&1&1\end{array}\right|=\left|\begin{array}{cccc}0&0&0&1\\&0&0&0\\&1&1&0\\1&&1&1\end{array}\right|=\left|\begin{array}{cccc}0&0&0&\\&0&0&0\\&1&1&0\\1&&1&1\end{array}\right|.$$

Minimized DNF of the function $F(x_1,x_2,x_3,x_4)$:

$$F_{DNF}=\overline{x}_1\overline{x}_2\overline{x}_3+\overline{x}_2\overline{x}_3\overline{x}_4+x_2x_3\overline{x}_4+x_1x_3x_4. \quad (27)$$

The results of the minimization of the DNF of the function $F(x_1,x_2,x_3,x_4)$ using parallel splitting of conjunctions [10] and the method of figurative transformations are presented in **Table 3.**

From **Table 3** it can be seen that the results of minimizing the two compared methods are the same. The minimization exponent coincides $k_\theta/k_1=4/12$ where k_θ – the number of simple implicants, k_1 – the number of input variables. However, the computational complexity of minimizing a Boolean function by figurative transformations is less.

Table 3

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

The method of parallel splitting of conjunctions	The method of figurative transformations
$\{(000\sim),(\sim000),(\sim110),(1\sim11)\}$	$\left\|\begin{array}{cccc}0&0&0&\\&0&0&0\\&1&1&0\\1&&1&1\end{array}\right\|$

To minimize the CNF of a given function, the Nelson method is used [11].

$$F_{CNF}=\left|\begin{array}{c}2\\3\\4\\5\\7\\9\\10\\12\\13\end{array}\right|\begin{array}{cccc}0&0&1&0\\0&0&1&1\\0&1&0&0\\0&1&0&1\\0&1&1&1\\1&0&0&1\\1&0&1&0\\1&1&0&0\\1&1&0&1\end{array}\right|=\left|\begin{array}{cccc}1&1&0&1\\1&1&0&0\\1&0&1&1\\1&0&1&0\\1&0&0&0\\0&1&1&0\\0&1&0&1\\0&0&1&1\\0&0&1&0\end{array}\right|=$$

$$=\left|\begin{array}{cccc}&1&0&1\\1&&0&0\\&0&1&\\0&1&1&0\end{array}\right|=\left|\begin{array}{cccc}&1&0&1\\1&&0&0\\&0&1&\\0&&1&0\end{array}\right|.$$

Minimized CNF of the function $F(x_1,x_2,x_3,x_4)$:

$$F_{CNF}=(x_1+\overline{x}_3+x_4)(\overline{x}_1+x_3+\overline{x}_4)(x_2+\overline{x}_3+x_4)(\overline{x}_2+x_3). \quad (28)$$

The minimum CNF of the function $F(x_1,x_2,x_3,x_4)$ (28) compared with the minimum DNF of the function $F(x_1,x_2,x_3,x_4)$ (27) contains a smaller number of literals. Thus, with the same functionality of expressions (27) and (28) (**Table 4**), the latter is a simple structure (**Fig. 2**).

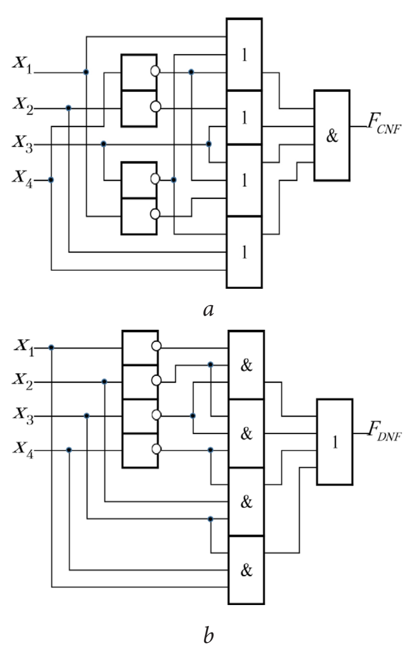

Fig. 2. The implementation of the minimum functions of the combinational circuit: *a* – CNF; *b* – DNF

Given the **Fig. 2** it is possible to see that the implementation of the combinational circuit of the minimum CNF of the Boolean function (a) is simple, because it contains a 2-input logical element OR, which is absent in the circuit that implements the minimal DNF of the Boolean function (b).

Table 4 presents the truth table of minimized CNF and DNF of the functions, which are given the canonical form (26).

Table 4

Truth table of minimized CNF and DNF of the functions

$$F_{CNF}(x_1,x_2,x_3,x_4)=$$
$$=(x_1+\overline{x_3}+\overline{x_4})(\overline{x_1}+x_3+\overline{x_4})(x_2+\overline{x_3}+x_4)(\overline{x_2}+x_3),$$
$$F_{DNF}(x_1,x_2,x_3,x_4)=\overline{x_1}\,\overline{x_2}\,\overline{x_3}+\overline{x_2}\,\overline{x_3}\,\overline{x_4}+x_2x_3\overline{x_4}+x_1x_3x_4$$

No. b/o	X_1	X_2	X_3	X_4	F_{CNF}	F_{DNF}
0	0	0	0	0	1	1
1	0	0	0	1	1	1
6	0	1	1	0	1	1
8	1	0	0	0	1	1
11	1	0	1	1	1	1
14	1	1	1	0	1	1
15	1	1	1	1	1	1
No. b/o	X_1	X_2	X_3	X_4	F_{CNF}	F_{DNF}
2	0	0	1	0	0	0
3	0	0	1	1	0	0
4	0	1	0	0	0	0
5	0	1	0	1	0	0
7	0	1	1	1	0	0
9	1	0	0	1	0	0
10	1	0	1	0	0	0
12	1	1	0	0	0	0
13	1	1	0	1	0	0

Given the **Table 4**, it is possible to see that the minimum CNF and DNF of the functions have the same functionality, but the minimum CNF of the function has one less literal. According to the results of minimization of two normal forms – DNF and CNF of a given function, the minimum function is selected in CNF (28).

4. Discussion of results

Equivalent figurative transformations by their properties have a large information capacity; therefore, they are capable of replacing verbal procedures of algebraic transformations with effect, which, in particular, simplifies the process of Boolean functions minimization. The method of figurative transformations allows to focus on the minimization principle within the protocol for calculating a logical function (within the truth table of a function) and, thus, dispense with auxiliary objects like the Karnaugh map, the Veitch diagram, the acyclic graph, the cubic representation, etc.

This distinguishes this method compared with peers for the following factors:

– an increase in the productivity of mental labor (the intellectual component) while minimizing Boolean functions, which contributes to the improvement of the algorithm for minimizing logical functions, expanding the control functions of the minimization method and a deeper understanding of logical transformations;

– a decrease in the amount of computation in the case of using the signs of the minimum function and a decrease in the computation volume in the case of Boolean function minimization on the full truth table;

– lower cost of development and implementation by reducing the need for the use of hardware-software automation tools.

The weak side of the method of figurative transformations with manual Boolean function minimization is associated with a small practice of application, therefore, the prospect of applying the method is based on the practical chances of optimal minimization of logical functions. Negative internal factors inherent in the process of Boolean function minimization by figurative transformations are the increase in the time for obtaining the minimum function with an insufficient library of minimization protocols for Boolean functions.

Additional possibilities that the implementation of the figurative transformation method for Boolean function minimization can bring in the new conditions of the minimal function are determined by the sign of the minimal function and Boolean function minimization on the full truth table.

An analogue of figurative transformations for Boolean function minimization is the algebraic method [12]. The algebraic method of Boolean function minimization is best in that for it the already predetermined laws of simplification, the discovered properties and algorithms for Boolean function minimization are created. However, the algebraic method is a verbal procedure for operational transformations, which gives a lesser effect to the quality of minimization compared to the method of figurative transformations.

The prospect of further research on the method of figurative transformations may be the development of a protocol for minimizing symmetric Boolean functions.

References

1. Quine–McCluskey algorithm. Wikipedia. Available at: https://en.wikipedia.org/wiki/Quine–McCluskey_algorithm
2. Rathore, T. S. (2014). Minimal Realizations of Logic Functions Using Truth Table Method with Distributed Simplification. IETE Journal of Education, 55 (1), 26–32. doi: http://doi.org/10.1080/09747338.2014.921412
3. Nosrati, M., Karimi, R., Nariri, M. (2012). Minimization of boolean functions using genetic algorithm. Anale. Seria Informatica, X fasc. 1, 73–77. Available at: https://pdf.semanticscholar.org/c53d/2240a2aa5531832a7707ad186dee23129ed8.pdf
4. Gursky, S. (2011). Minimization of Matched Formulas. WDS'11 Proceedings of Contributed Papers, Part I, 101–105.

5. Riznyk, V., Solomko, M. (2017). Minimization of Boolean functions by combinatorial method. Technology Audit and Production Reserves, 4 (2 (36)), 49–64. doi: http://doi.org/10.15587/2312-8372.2017.108532
6. Riznyk, V., Solomko, M. (2017). Application of super-sticking algebraic operation of variables for Boolean functions minimization by combinatorial method. Technology Audit and Production Reserves, 6 (2 (38)), 60–76. doi: http://doi.org/10.15587/2312-8372.2017.118336
7. Riznyk, V., Solomko, M. (2018). Research of 5-bit boolean functions minimization protocols by combinatorial method. Technology Audit and Production Reserves, 4 (2 (42)), 41–52. doi: http://doi.org/10.15587/2312-8372.2018.140351
8. Ivanov, Yu. D., Zakharova, O. S. (2004). Algorithm for synthesizing infimum disjunctive normal forms of logical functions. Proceedings of the Odessa Polytechnic University, 2 (22), 1–7. Available at: http://pratsi.opu.ua/app/webroot/articles/1313074117.pdf
9. Quine's construction algorithm for an abbreviated DNF (2014). Available at: https://studfiles.net/preview/952550/page:15/
10. Rytsar, B. Ye. (2015). New minimization method of logical functions in polynomial set-theoretical format. 1. Generalized rules of conjuncterms simplification. Control systems and machines, 2, 39–57. Available at: http://dspace.nbuv.gov.ua/handle/123456789/87194
11. Nelson's method (2016). Available at: https://studfiles.net/preview/5475041/page:26/
12. Martynyuk, O. M. (2008). Basics of discrete mathematics. Odessa National Polytechnic University: Science and Technology, 300.