UDC 519.718
DOI: 10.15587/1729-4061.2021.229786

DEVELOPING THE MINIMIZATION OF A POLYNOMIAL NORMAL FORM OF BOOLEAN FUNCTIONS BY THE METHOD OF FIGURATIVE TRANSFORMATIONS

M. Solomko
PhD, Associate Professor
Department of Computer Engineering*
E-mail: doctrinas@ukr.net

Iu. Batyshkina
PhD, Associate Professor
Department of Informational and Communal Technologies and Computes Science Teaching Methods**
E-mail: yuliia.batyshkina@rshu.edu.ua

N. Khomiuk
PhD
Department of International Economic Relations and Project Management
Lesya Ukrainka Volyn National University
Voli str., 13, Lutsk, Ukraine, 43000
E-mail: nataljabilous@gmail.com

Ya. Ivashchuk
PhD
Department of Higher Mathematics*
E-mail: zrsd@i.ua

N. Shevtsova
PhD
Department of Informatics and Applied Mathematics**
E-mail: natalia.shevtsova@rshu.edu.ua

* National University of Water and Environmental Engineering
Soborna str., 11, Rivne, Ukraine, 33028
**Rivne State University of Humanities
S. Bandery str., 12, Rivne, Ukraine, 33028

This paper reports a study that has established the possibility of improving the effectiveness of the method of figurative transformations in order to minimize Boolean functions on the Reed-Muller basis. Such potential prospects in the analytical method have been identified as a sequence in the procedure of inserting the same conjuncterms of polynomial functions followed by the operation of super-gluing the variables.

The extension of the method of figurative transformations to the process of simplifying the functions of the polynomial basis involved the developed algebra in terms of the rules for simplifying functions in the Reed-Muller basis. It was established that the simplification of Boolean functions of the polynomial basis by a figurative transformation method is based on a flowchart with repetition, which is actually the truth table of the predefined function. This is a sufficient resource to minimize functions that makes it possible not to refer to such auxiliary objects as Karnaugh maps, Weich charts, cubes, etc.

A perfect normal form of the polynomial basis functions can be represented by binary sets or a matrix that would represent the terms of the functions and the addition operation by module two for them.

The experimental study has confirmed that the method of figurative transformations that employs the systems of 2-(n, b)-design, and 2-(n, x/b)-design in the first matrix improves the efficiency of minimizing Boolean functions. That also simplifies the procedure for finding a minimum function on the Reed-Muller basis. Compared to analogs, this makes it possible to enhance the performance of minimizing Boolean functions by 100–200 %.

There is reason to assert the possibility of improving the efficiency of minimizing Boolean functions in the Reed-Muller basis by a method of figurative transformations. This is ensured by using more complex algorithms to simplify logical expressions involving a procedure of inserting the same function terms in the Reed-Muller basis, followed by the operation of super-gluing the variables

Keywords: minimization of Boolean functions in the Reed-Muller basis, figurative transformation method, singular function

Received date 24.02.2021
Accepted date 16.04.2021
Published date 30.04.2021

***How to Cite:** Solomko, M., Batyshkina, Iu., Khomiuk, N., Ivashchuk, Ya., Shevtsova, N. (2021). Developing the minimization of a polynomial normal form of boolean functions by the method of figurative transformations. Eastern-European Journal of Enterprise Technologies, 2 (4 (110)), 22–37. doi: https://doi.org/10.15587/1729-4061.2021.229786*

1. Introduction

An arbitrary Boolean function $f(x_1, x_2, ..., x_n)$ can be represented in an Exclusive-OR Sum-Of-Product form (ESOP), created by double conjunction operations (AND) and the amount by mod 2 (EXOR) and a unity constant; the inversion of an arbitrary variable is produced by the operation $x \oplus 1 = \bar{x}$. At the same time, depending on which variables of the ESOP conjuncterms f (all or some of them) have or do not have an inversion sign, which determines the so-called polarity of variables, the classes of AND/EXOR expressions of Boolean functions ESOP are distinguished. In a general case, they are termed Reed-Muller expressions (RM-polynomials) [1]. The taxonomy of RM-polynomials, the relationship between different classes, and the complexity of their implementation are considered in [1–6].

For comparison, we shall illustrate possible polynomials by following the examples:

– $x_1x_3 \oplus x_1x_2x_3$ – *PPRM*-polynomial (*Positive Polarity Reed-Muller expression*), that is, a polynomial of the *n*-th power by Zhegalkin, all variables of which have a direct polarity;

– $\overline{x}_1\,\overline{x}_3 \oplus \overline{x}_1\,\overline{x}_2\,\overline{x}_3$ – *NPRM*-polynomial (*Negative Polarity Reed-Muller expression*) whose all variables have inverted polarity;

– $\overline{x}_1x_3 \oplus \overline{x}_1x_2x_3$ – *FPRM*-polynomial, the ESOP expression of the Boolean function $f\,(x_1, x_2, ..., x_n)$, where each variable has a certain fixed (direct or inverted) polarity; it is called the *Fixed Polarity Reed-Muller expression*;

– $\overline{x}_1x_3 \oplus \overline{x}_1x_2\overline{x}_3$ – *MPRM*-polynomial (or Kronecker polynomial), *Mixed Polarity Reed-Muller expression*; in [3–5], it is called *Kronecker expression*, where $\widetilde{x}_i = \{\overline{x}_i, x_i\}$, that is, all variables have both polarities;

– $x_1 \oplus x_2 \oplus \overline{x}_1\,\overline{x}_2$ – *GRM*-polynomial (*Generalized Reed-Muller expression*), formed by the arbitrary choice of polarity *n* for variables of the Boolean function $f\,(x_1, x_2, ..., x_n)$.

The development of microelectronic technology has ensured the creation of elements that form multiple disjunctions with exception (EXOR elements). This, in turn, ensured the synthesis of similar two-level AND/EXOR circuits that contain the same elements in the second cascade. The structure of these schemes is described by formulas similar to a disjunctive normal form (DNF), in which disjunction operators with an exception are used instead of disjunction operators. Such formulas are called ESOP – the exclusive sum of products.

The advantages of these formulas are justified by the fact that the number of logic elements in their respective schemes is usually less. For example, after minimizing the DNF of arbitrary Boolean functions, the four variables contain an average of 4.13 conjuncterms, and ESOP contains only 3.66 [4, 7]. When considering Boolean functions typical of schemes that implement arithmetic operations, the win is even greater. In addition, AND/EXOR circuits are more easily diagnosed [7, 8].

In the case of representing Boolean functions by Zhegalkin polynomials, the optimization problem does not arise since the solution is unambiguous. The optimization problem appears for non-fully defined Boolean functions. If the value of the function remains undefined in *k* sets, 2^k different function determination and, accordingly, 2^k different Zhegalkin polynomials forming a given function are possible. The choice among them of the simplest polynomial is a complex combinatorial problem. The task becomes even more difficult when implementing functions in the form of ESOP (which contain literals with different inversions) or when it is necessary to implement a system of Boolean functions.

As stated in [2], effective algorithms for minimizing ESOP of the Boolean functions $f(x_1, x_2, ..., x_n)$ do not exist. Such a conclusion, however, could be some start, which would eventually move to its own opposite.

The evolution of the visual-matrix form of the analytical method of simplification of logic functions is the result of continuous optimization, in particular [9]. In this regard, theoretical studies on minimizing ESOP of Boolean functions remain relevant, in particular, to improve the following factors:

– the visual and matrix methods of minimizing logic functions in the ESOP class;

– the cost of a technique to minimize ESOP of logic functions.

2. Literature review and problem statement

Generalized rules for simplifying logic expressions in the format of the theory of polynomial sets are considered in paper [10]. The rules reported there are based on the proposed theorems for different initial conditions for the transformation of paired conjuncterms, the Hamming distance between which can be arbitrary. These rules can be useful for minimizing arbitrary logic functions with *n* variables in the theoretical format of polynomials. The advantages of the proposed simplification rules are illustrated by examples.

A method to search for the exact ESOP expression for a not fully specified arbitrary logic function up to six input variables is proposed in work [11]. To this end, the weights of all 5-variable functions are entered in the table, which is used in the proposed approach and speeds up the computation time. It is believed that this is the first paper concerning the exact solutions of minimization for not fully defined Boolean functions.

Work [12] reports the results of a study into minimizing AND-XOR expressions for functions with a large number of literals. The object of the study is the use of a simple greedy algorithm for finding a minimum function, based on a set of local transformations, to the expressions of the Boolean functions of the Reed-Müller basis with positive polarity. It is noted that experiments with large functions demonstrate good results. The number of literals was reduced by an average of 23 %. It is assumed that much better results could be achieved once a more complex non-greedy algorithm for finding minimal functions is applied.

Minimization of multilevel representation of Boolean functional systems based on Shannon extension with finding equal coefficients (exact within inversion), and using Zhegalkin polynomials for these purposes, is proposed in [13]. Zhegalkin polynomials are easy to compare and easy to obtain the inversion of a function, and that significantly reduces the calculation time. The application of the program that implements the proposed algorithms makes it possible to receive smaller areas of VLSI circuits compared to chains that are synthesized using minimized DNF and Shannon expansion schemes where coefficient inversion is not taken into consideration.

Paper [14] reports a method for determining the upper limit of complexity in the implementation of arbitrary Boolean functions, which can be implemented by Zhegalkin polynomials. A computational method for improving these boundaries is proposed.

The representation of Boolean functions by reverse circuits, on the elements of Toffoli, is considered in paper [15]. Interest in this issue is associated with the implementation of «cold» calculations. This means that when performing such calculations, there is no heat dissipation. In general, reversible schemes implement reversible functions. Therefore, the Toffoli-Fredkin method is used to represent a Boolean function by a reversible function. The paper describes an algorithm for finding the minimum representation of a Boolean function in the class of reversing chains built on Toffoli elements. The algorithm uses ESOP of Boolean functions, as well as the class of polarized Zhegalkin polynomials or the Reed-Muller forms. The results of the computational algorithm that minimizes Boolean functions in the class of reversible circuits are presented.

Minimizing logic has been attracting considerable attention lately, as it is important for many applications to have the most compact images possible. Paper [16] proposes a fast minimizing algorithm (FMA) for the Reed-Muller fixed polarity expressions (FPRM). The basic FMA idea is to find the minimum FPRM function with the fewest conjuncterms. This uses the proposed binary differential search evolution (BDE) algorithm. The authors described experimental results involving 24 MCNC control circuits, which show that FMA surpasses the genetically based effectiveness of finding the minimum Reed-Muller expressions. It is assumed that the use of the differential evolution algorithm to minimize FPRM was first considered in paper [16]. FMA can be expanded so that it is possible to obtain a minimum mixed polarity of the expression on the polynomial basis.

The above literary sources [10–16] mostly report algorithms and methods for minimizing Boolean functions in the Reed-Muller basis by using theoretical objects of adjacent theory. Specifically, Hamming distance tables, greedy (non-greedy) algorithms, Shannon extensions, genetic algorithm, differential evolution algorithm to minimize FPRM, etc. Not all methods provide an accurate solution to minimization. An obligatory technological point for these algorithms and methods are automated calculations. In the complex search for the optimal function, compensation may be an approximate synthesis – the tendency of logic synthesis, when some results of the logical specification change within the permissible optimality of the digital circuit to be designed.

A method of figurative transformations based on binary combinatorial systems with repeated 2-(n, b)-design, 2-(n, x/b)-design belongs to the classical analytical method by qualification. They do not allow for the approximate result of minimization and do not exclude the manual technique for minimizing Boolean functions, including in the Reed-Muller basis.

Thus, the algorithms and methods that employ theoretical objects of adjacent theory, software tools for them, which cover the general procedure for minimizing the logic functions of the polynomial basis [10–16] and a method of figurative transformations, follow different approaches (principles of minimization). Therefore, they consider different prospects regarding the possibility of algorithmic minimization of logic functions of the polynomial basis.

The prospect for a figurative transformation method, which is a descendant of the analytical method, regarding the proper minimization of logic functions in the Reed-Muller basis is to create the necessary algebra in terms of the rules for the equivalent transformation of polynomial functions. As well as identify the reserves of the analytical method, such as the sequence in the procedure of inserting the same conjuncterms of polynomial functions, followed by the operation of super-gluing the variables. Thus, the classical analytical method still has the prospect of increasing its hardware capabilities to minimize functions on the Reed-Muller basis. And this is the reason to believe that the software and technological base, represented by the algorithms and methods with theoretical objects of adjacent theories [10–16], is insufficient for theoretical research into the optimal minimization of Boolean functions on the Reed-Muller basis.

This determines the need to investigate equivalent figurative transformations in order to minimize logic functions on the Reed-Muller basis. In particular, the peculiarities of relatively complex algorithms of simplification of functions with the procedure of inserting the same conjuncterms of polynomial functions followed by the operation of super-gluing the variables, a stack of logical operations for the first binary matrix of a polynomial function [9], ways to simplify arbitrary functions in the Reed-Muller basis.

In practical terms, a figurative transformation method would provide an expansion of the capabilities of technology for designing digital components based on the basis $\Sigma_1 = \{\wedge, \oplus, 1\}$.

3. The aim and objectives of the study

The purpose of this work is to extend the method of figurative transformations to minimize Boolean functions in the class of Exclusive-OR Sum-Of-Product forms (ESOP) and perfect Exclusive-OR Sum-Of-Product forms (PESOP). That could simplify the performance of minimizing functions in the Reed-Muller basis by refining Reed-Muller's algebra in terms of algebraic rules for the equivalent transformation of ESOP functions.

To accomplish the aim, the following tasks have been set:

– to establish the hermeneutics of logic operations for the class of equivalent binary matrices of functions of the Reed-Muller basis;

– to establish patterns in the use of algorithms for the equivalent transformation of Boolean functions in the Reed-Muller basis, consisting of the procedure of inserting two identical ESOP conjuncterms with the following operation of super-gluing the variables;

– to devise a method for the orthogonalization of logic functions using figurative transformations in order to establish singular functions;

– to refine Reed-Muller's algebra in terms of the necessary algebraic rules for the equivalent transformation of ESOP functions;

– to analyze the results from simplifying functions in the Reed-Muller basis by a figurative transformation method and the examples of minimization of functions in the polynomial basis in order to compare the cost of the minimum function implementation and the number of procedural steps.

4. The Reed-Muller basis

Along with the well-known Boolean basis $\{\vee, \wedge, \neg\}$ and non-redundant bases $\{\vee, \neg\}$ and $\{\wedge, \neg\}$, an important role in the theory of logic functions and in its practical application belongs to the Reed-Muller basis $\{\wedge, \oplus, 1\}$, which includes the operation «sum of modulo 2» ($\oplus$). The completeness of this basis is proven by the following ratios, which demonstrate that it is reduced to a well-known full basis $\{\neg, \vee\}$:

$$\bar{a} = 1 \oplus a; \quad a \vee b = a \oplus b \oplus ab.$$

Similar to disjunction and conjunction operations, the sum by module two has the properties of commutativity and associativity, and is also generalized in the case of a large number of variables.

Multiple amount for module two of:

$$\oplus(a,b,s,d) = a \oplus b \oplus c \oplus d = mod2\ sum(a,b,s,d),$$

arbitrary elementary conjunctions is a *polynomial*. A separate case of the polynomial is the Zhegalkin polynomial, consisting of non-inverted variables.

Any Boolean function can be represented by a formula in the Reed-Muller basis (Zhegalkin) $\Sigma_1 = \{\wedge, \oplus, 1\}$.

5. Reed-Muller Algebra

An algebra over a set of logic functions with two binary operations & and $\oplus$ is the Reed-Muller (Zhegalkin) algebra. The following interrelations are satisfied in the Reed-Muller algebra:

$$x\oplus y=x\bar{y}+\bar{x}y=(x+y)(\bar{x}+\bar{y}); \tag{1}$$

$$x_1\bar{x}_2\oplus\bar{x}_1x_2=x_1\oplus x_2=\bar{x}_2\oplus\bar{x}_1. \tag{2}$$

Identity (2) has an illustration of the representation:

$$\begin{vmatrix}1 & 0\\ 0 & 1\end{vmatrix}=\begin{vmatrix}1 & \\ & 1\end{vmatrix}=x_1\oplus x_2,$$

or

$$\begin{vmatrix}1 & 0\\ 0 & 1\end{vmatrix}=\begin{vmatrix}0 & \\ & 0\end{vmatrix}=\bar{x}_1\oplus\bar{x}_2.$$

$$xy\oplus\bar{x}\;\bar{y}=\overline{x\oplus y}=\overline{\bar{x}\oplus\bar{y}}=x\oplus\bar{y}=\bar{x}\oplus y. \tag{3}$$

Identity (3) has an illustration of the representation:

$$\begin{vmatrix}1 & 1\\ 0 & 0\end{vmatrix}=\begin{vmatrix}1 & \\ & 0\end{vmatrix}=x\oplus\bar{y},$$

or

$$\begin{vmatrix}1 & 1\\ 0 & 0\end{vmatrix}=\begin{vmatrix} & 1\\ 0 & \end{vmatrix}=\bar{x}\oplus y.$$

$$xyz\oplus xy\oplus x=xy\bar{z}\oplus x=x\left(1\oplus y\bar{z}\right)=$$
$$=x\overline{y\bar{z}}=x(\bar{y}+z)=x\bar{y}+xz,$$

or

$$xyz\oplus xy\oplus x=xy\bar{z}\oplus x=$$
$$=x\overline{y\bar{z}}=x\left(\bar{y}+z\right)=x\bar{y}+xz.$$

For the function of addition modulo two, there are displaceable and connecting laws, as well as the distribution law with respect to the conjunction.

$$x\oplus y=y\oplus x;$$

$$x\oplus(y\oplus z)=(x\oplus y)\oplus z;$$

$$x(y\oplus z)=(xy)\oplus(xz).$$

The following valid interrelations are obvious [17]:

$$\begin{cases}x\oplus x=0;\\ x\oplus 0=x;\\ x\oplus 1=\bar{x};\\ x\oplus\bar{x}=1.\end{cases}$$

In addition, the following formulas hold:

$$x+y=\overline{\bar{x}\cdot\bar{y}}=(x\oplus 1)(y\oplus 1)\oplus 1=xy\oplus x\oplus y=x\bar{y}\oplus y; \tag{4}$$

$$x_1\cdot x_2=(x_1\oplus x_2)\oplus(x_1\oplus x_2). \tag{5}$$

Logic identities for two variables are given in Table 1.

Table 1

Equivalent logic expressions of two variables

Logical equality	$x\sim y$	$\bar{x}\bar{y}+xy=1\oplus x\oplus y$
Logical inequality	$x\oplus y$	$x\bar{y}+\bar{x}y=x\oplus y$
Disjunction	$x+y$	$x+y=x\oplus y\oplus xy$
Schaefer's stroke	$x\vert y$	$x\vert y=\bar{x}+\bar{y}=1\oplus xy$
Implication	$x\rightarrow y$	$x\rightarrow y=\bar{x}+y=1\oplus x\oplus xy$
Implication	$y\rightarrow x$	$y\rightarrow x=\bar{y}+x=1\oplus y\oplus xy$
Conjunction	xy	$xy=x\bar{y}\oplus x$
Conjunction	$x\bar{y}$	$x\bar{y}=x\oplus xy$
Conjunction	$\bar{x}y$	$\bar{x}y=y\oplus xy$
Pierce's arrow	$x\downarrow y$	$x\downarrow y=\bar{x}\bar{y}=1\oplus x\oplus y\oplus xy$

The Reed-Muller algebra provides the creation of classical rules of equivalent transformation to simplify logic expressions in the ESOP class by analytical method. The difference between a Reed-Muller polynomial and a Zhegalkin polynomial is that the Zhegalkin polynomial represents 2-level logic, and the Reed-Muller polynomial represents 3-level logic. However, a Reed-Muller polynomial generally contains fewer literals.

In some cases, the transformation of the Reed-Muller polynomial to the mixed basis produces 2-level logic. Such as:

$$1\oplus x_1x_2x_3\oplus\bar{x}_1\;\bar{x}_2\;\bar{x}_3=x_1x_2x_3\oplus\bar{\bar{x}}_1\;\bar{\bar{x}}_2\;\bar{\bar{x}}_3=$$
$$=x_1x_2x_3\oplus\left(x_1+x_2+x_3\right).$$

The 3-level logic of the Reed-Muller polynomial is transformed into 2-level mixed basis logic.

6. Interpretation of the truth table of a logic function by the binary combinatorial systems with repetition

For some set A, a new set $M(A)$ can be considered – a set of all its subsets – a Boolean. $M_k(A)$ denotes the set of all subsets A that have k elements.

Assume $A=\{a, b, c\}$, then:

$$M(A)=\begin{Bmatrix}\{a\},\{b\},\{c\},\{a,b\},\\ \{a,c\},\{b,c\},\{a,b,c\},\varnothing\end{Bmatrix};$$

$$M_2(A)=\{\{a,b\},\{a,c\},\{b,c\}\}.$$

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

$$N\left(M_k(A)\right)=C_n^k=\frac{n!}{k!(n-k)!}. \tag{6}$$

The following equality holds:

$$\sum_{k=0}^{n}C_n^k=2^n. \tag{7}$$

Since C_n^k is the number of k-element subsets of the set of n elements, the sum in the left-hand side of expression (7) is the number of all subsets.

The set $A=\{a, b, c\}$, in addition to recalculating its elements, can also determine the number of positions on which the element a is located. For example, a can mean the first position, b can mean the second position of the set $A=\{a, b, c\}$, etc. Subsets of the set $A=\{a, b, c\}$, in this case, are the subsets containing the element α at k positions, $k=0, ..., n$, where n is the number of positions of the set A. In a general case, the element a can occupy several positions on the set A, so the element a is repeated on the set A.

Let $\alpha=1$, then the positions where the element a is missing are denoted by zero.

For the set $A=\{a, b, c\}$, which defines position numbers, we accept $\alpha=1$. Then the subsets of set A take the following form:

$$\begin{array}{ll} (0,0,0); & (1,0,0); \\ (0,0,1); & (1,0,1); \\ (0,1,0); & (1,1,0); \\ (0,1,1); & (1,1,1). \end{array} \tag{8}$$

The number of all k-element subsets of the set $A=\{a, b, c\}$, which determines the position numbers, is calculated from formula (7).

Configuration (8) is a complete combinatorial system with a repeated element α, which is denoted:

2-(n, b)-design,

where n is the bit size of the system block; b is the number of blocks of the complete system, determined from formula $b=2^n$, the number 2 before the brackets denotes the binary structure of configuration (8). For example, 2-(4, 16)-design is the complete binary combinatorial system with a repetition consisting of 4-bit blocks, the number of blocks is 16.

It is easy to see that configuration (8), which makes up the complete combinatorial system with repetition, can be interpreted as a truth table of the logic function $f(a, b, c)$, with a full set of minterms or maxterms (Table 2).

Another interpretation variant is demonstrated by a truth table that contains a combinatorial system with repeated 2-(2, 4)-design in the table configuration variant when there is one column with the same variable values (Table 3).

Table 2

Logic function $f(a, b, c)$ truth table

No.	a	b	c	No.	a	b	c
0	0	0	0	4	1	0	0
1	0	0	1	5	1	0	1
2	0	1	0	6	1	1	0
3	0	1	1	7	1	1	1

Table 3

The truth table of the logic function $f(a, b, c)$ with a column of the same variable values

a	b	c
0	0	1
0	1	1
1	0	1
0	1	1

In this case, the combinatorial system with repeated 2-(2, 4)-design (combinatorial representation) would interpret the login operation, in a given case – super-gluing the variables.

The procedure of reducing the complete perfect disjunctive normal form (PESOP) of a logic function yields unity. For example, reducing a 2-variable full PESOP takes the following form:

$$\begin{aligned}&\overline{x_1}\,\overline{x_2}+\overline{x_1}x_2+x_1\overline{x_2}+x_1x_2=\\&=\overline{x_1}\left(\overline{x_2}+x_2\right)+x_1\left(\overline{x_2}+x_2\right)=\overline{x_1}+x_1=1.\end{aligned}$$

Since the complete PESOP uniquely determines the complete combinatorial system with repeated 2-(n, b)-design and vice versa, this gives reason to remove all blocks of the full combinatorial system with a repetition from the truth table of the assigned function.

The visual representation of the logic operation of super-gluing the variables involving 2-(2, 4)-design takes the following form:

$$\begin{vmatrix}0&0&1\\0&1&1\\1&0&1\\0&1&1\end{vmatrix}=|\qquad 1\,|=x_3.$$

The algebraic notation of the logic operation of super-gluing the variables is as follows:

$$\begin{aligned}&\overline{x_1}\,\overline{x_2}x_3+\overline{x_1}x_2x_3+x_1\overline{x_2}x_3+x_1x_2x_3=\\&=\left(\overline{x_1}\left(\overline{x_2}+x_2\right)+x_1\left(\overline{x_2}+x_2\right)\right)x_3=\left(\overline{x_1}+x_1\right)x_3=x_3.\end{aligned}$$

Similarly, other operations of equivalent transformation of logic expressions are interpreted.

In a general case, the truth table of the logic function, in addition to the configuration of the complete combinatorial system with repeated 2-(n, b)-design, can also contain the configuration of an incomplete combinatorial system with repeated 2-(n, x/b)-design. In this case, x is the number of blocks of an incomplete combinatorial system with repetition. The properties of an incomplete combinatorial system with repeated 2-(n, x/b)-design also make it possible to set rules that ensure the effective minimization of Boolean functions.

Configuration (8) can also be interpreted as a truth table of the logic function ESOP f (a, b, c), with a full set of conjuncterms of the function in the Reed-Muller basis and the operation of adding modulo two for them.

7. Results of minimizing Boolean functions in the Reed-Muller basis by a figurative transformation method

Equivalent figurative transformations when minimizing functions in the Reed-Muller basis yield the following result:

– determining the hermeneutics of logic operations for the class of equivalent binary matrices of functions in the Reed-Muller basis;

– a protocol with relatively complex algorithms for simplifying logic expressions, which consists of the procedure for inserting two identical conjuncterms of the functions of a polynomial basis with the following operation of super-gluing the variables. This protocol increases the efficiency of the procedure, which makes it possible, in particular, to simplify logic functions in the Reed-Muller basis with a relatively large number of input variables manually;

– ensuring the method of orthogonalization of specified logic functions for the establishment of singular functions;

– creating algebra in terms of the rules for equivalent transformation of Boolean functions in the Reed-Muller basis.

7. 1. Hermeneutics of logic operations for the class of equivalent binary matrices of functions in the Reed-Muller basis

Represent the logic function $f(x_1,x_2,...,x_n)$ in a perfect Exclusive-OR Sum-Of-Product form (PESOP):

$$f(x_1,x_2,...,x_n)=\underset{1}{\oplus} x_1^{\alpha_1}x_2^{\alpha_2}...x_n^{\alpha_n}, \tag{9}$$

where the symbol $\underset{1}{\oplus}$ means that the sum by modulo two is taken only on sets of variables $<\alpha_1,\alpha_2,...,\alpha_n>$, on which $f(x_1,x_2,...,x_n)=1$.

To represent PESOP (9) by a binary equivalent or a matrix, variables with inversion $\overline{x_n}$ must be replaced with 0_n, and variables without inversion x_n – with 1_n, where n is a numeric index that determines the bit size of the variable character «1» or «0» in the conjuncterms of the Reed-Muller basis function. Then PESOP (9) can be represented by the following binary sets (tuples).

$$F=(0_1 0_2 ... 0_n)(0_1 0_2 ... 1_n)(0_1 1_2 ... 1_n), \tag{10}$$

or the following matrix:

$$F=\begin{vmatrix} 0 & 0 & ... & 0 \\ 0 & 0 & ... & 1 \\ ... & ... & ... & ... \\ 0 & 1 & ... & 1 \end{vmatrix}. \tag{11}$$

We shall call matrix (11) an instance of the binary matrix class of the Reed-Muller basis functions.

The hermeneutics of logic operations for matrix (11) is that matrix (11) expresses the conjuncterms of the Reed-Muller basis function and the operation of adding modulo two for them. Our hermeneutics of logic operations in the Reed-Muller basis should be used when deriving the result of logic operations in the class of binary matrices of the Reed-Muller basis functions.

7. 2. Patterns of using the logic operation of super-gluing the variables for the polynomial normal form of Boolean functions

The implementation of logic operations on binary and algebraic function structures is to be highlighted in color if you know what we mean. That would provide for a better didactic of the method under consideration.

For the method of figurative transformations, the algorithm of function simplification involving the procedure of inserting two identical conjuncterms of polynomial functions with the following operation of super-gluing the variables may have the following variants, for example:

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

$$=x_1\overline{x_3}\oplus x_1x_2\overline{x_3}x_4\oplus x_1x_2x_3x_4. \tag{12}$$

$$x_1\overline{x_2}\ \overline{x_3}\ \overline{x_4}\oplus x_1\overline{x_2}\ \overline{x_3}x_4\oplus x_1x_2\overline{x_3}\ \overline{x_4}\oplus x_1x_2x_3x_4=$$
$$=x_1\overline{x_2}\ \overline{x_3}\ \overline{x_4}\oplus x_1\overline{x_2}\ \overline{x_3}x_4\oplus x_1x_2\overline{x_3}\ \overline{x_4}\oplus$$
$$\oplus x_1x_2\overline{x_3}x_4\oplus x_1x_2\overline{x_3}x_4\oplus x_1x_2x_3x_4=$$
$$=x_1\overline{x_3}\oplus x_1x_2\overline{x_3}x_4\oplus x_1x_2x_3x_4. \tag{13}$$

$$2.\ \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 0 & & \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix} =$$

$$=\overline{x_1}\oplus\overline{x_1}x_2x_3\oplus x_1\overline{x_2}x_3. \tag{14}$$

$$\overline{x_1}\ \overline{x_2}\ \overline{x_3}\oplus\overline{x_1}\ \overline{x_2}x_3\oplus\overline{x_1}x_2\overline{x_3}\oplus x_1\overline{x_2}x_3=$$
$$=\overline{x_1}\ \overline{x_2}\ \overline{x_3}\oplus\overline{x_1}\ \overline{x_2}x_3\oplus\overline{x_1}x_2\overline{x_3}\oplus$$
$$\oplus\overline{x_1}x_2x_3\oplus\overline{x_1}x_2x_3\oplus x_1\overline{x_2}x_3=$$
$$=\overline{x_1}\oplus\overline{x_1}x_2x_3\oplus x_1\overline{x_2}x_3. \tag{15}$$

$$3.\ \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{vmatrix} = \begin{vmatrix} & 0 & 0 \\ 0 & & \\ 0 & 1 & 1 \end{vmatrix} =$$

$$=\overline{x_2}\ \overline{x_3}\oplus\overline{x_1}\oplus\overline{x_1}x_2x_3. \tag{16}$$

$$x_1\overline{x_2}\ \overline{x_3}\oplus\overline{x_1}x_2\overline{x_3}\oplus\overline{x_1}\ \overline{x_2}x_3=$$
$$=x_1\overline{x_2}\ \overline{x_3}\oplus\overline{x_1}x_2\overline{x_3}\oplus\overline{x_1}\ \overline{x_2}x_3\oplus\overline{x_1}\ \overline{x_2}\ \overline{x_3}\oplus$$
$$\oplus\overline{x_1}x_2x_3\oplus\overline{x_1}\ \overline{x_2}\ \overline{x_3}\oplus\overline{x_1}x_2x_3=$$
$$=\overline{x_2}\ \overline{x_3}\oplus\overline{x_1}\oplus\overline{x_1}x_2x_3. \tag{17}$$

The results from the matrix (12), (14), (16), and algebraic (13), (15), (17) techniques of minimizing a logic expression coincide.

Comparatively complex algorithms for simplifying logic expressions involving the procedure of inserting two identical conjuncterms of polynomial basis functions with the following operation of super-gluing the variables (12), (14), (16) expand the variants of their application. This ensures an increase in the efficiency of the procedure for minimizing Boolean functions in ESOP by using a method of figurative transformations.

7. 3. Singular functions

To solve optimization problems of logic synthesis, it is necessary to have PESOP functions and RM-polynomials with a minimum number of conjuncterms of the assigned function $f(x_1, x_2, ..., x_n)$. At the same time, if it is possible to choose an *RM* polynomial (except for *PPRM*- and *NPRM*-polynomials), then in the case of the same number of conjuncterms, preference is given to the *RM*-polynomial with the minimum total number of literals. And when the number of the latter is the same, the minimum *RM* polynomial is the one with fewer inverted literals. Thus, the cost of implementing the *RM*-polynomial of the assigned function $f(x_1, x_2, ..., x_n)$ can be estimated by the numerical ratio $k_\theta / k_l / k_{in}$, where

k_θ, k_l, k_{in} is the number of conjuncterms, literals, and inverters, respectively [1]. A similar assessment of the cost of implementing a minimum function can be applied to the PESOP function and, to some extent, to logic functions on a mixed basis.

Transitions between the Bull and Reed-Muller bases are carried out using a singular (special) function, the conjuncterms of which are pairwise orthogonal.

To convert the PESOP function into a normal polynomial form, it is necessary to orthogonalize the assigned function.

Example 1: It is required to orthogonalize function $F(x_1,x_2,x_3,x_4)$ (17) that is set by PESOP.

$$F(x_1,x_2,x_3,x_4)=\overline{x_1}x_3+\overline{x_1}x_2x_4+x_1\overline{x_2}\,\overline{x_3}+x_1\overline{x_3}\,\overline{x_4}. \quad (18)$$

Solution. The procedure of $F(x_1,x_2,x_3,x_4)$ (19) orthogonalization using figurative transformations is as follows:

$$F(x_1,x_2,x_3,x_4)=$$

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

$$=\left|\begin{array}{cccc}0 & & 1 & \\ 0 & 1 & 0 & 1\\ 1 & 0 & 0 & \\ 1 & & 0 & 0\end{array}\right|=\left|\begin{array}{cccc}0 & & 1 & \\ 0 & 1 & 0 & 1\\ 1 & 0 & 0 & \\ 1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\end{array}\right|=\left|\begin{array}{cccc}0 & & 1 & \\ 0 & 1 & 0 & 1\\ 1 & 0 & 0 & \\ 1 & 1 & 0 & 0\end{array}\right|. \quad (19)$$

The conjuncterms of function (19) are pairwise orthogonal. In algebraic notation, function (19) takes the following form (20):

$$F(x_1,x_2,x_3,x_4)=\overline{x_1}x_3+\overline{x_1}x_2\overline{x_3}x_4+x_1\overline{x_2}\,\overline{x_3}+x_1x_2\overline{x_3}\,\overline{x_4}. \quad (20)$$

Functions (19), (20) are singular since the equivalent transformations for them can be carried out by choosing the algebra of one of the bases – Bull $\{\wedge,\vee,\neg\}$, or Reed-Muller $\{\wedge,\oplus,1\}$.

Orthogonalizing PESOP function (18) by the algebraic procedure is as follows:

$$\begin{aligned}
f&=\overline{x_1}x_3+\overline{x_1}x_2x_4+x_1\overline{x_2}\,\overline{x_3}+x_1\overline{x_3}\,\overline{x_4}=\\
&=\overline{x_1}x_3+\overline{x_1}x_2(x_3+\overline{x_3})x_4+x_1\overline{x_2}\,\overline{x_3}+x_1\overline{x_3}\,\overline{x_4}=\\
&=\overline{x_1}x_3+\overline{x_1}x_2x_3x_4+\overline{x_1}x_2\overline{x_3}x_4+x_1\overline{x_2}\,\overline{x_3}+x_1\overline{x_3}\,\overline{x_4}=\\
&=\overline{x_1}x_3+\overline{x_1}x_2\overline{x_3}x_4+x_1\overline{x_2}\,\overline{x_3}+x_1\overline{x_3}\,\overline{x_4}=\\
&=\overline{x_1}x_3+\overline{x_1}x_2\overline{x_3}x_4+x_1\overline{x_2}\,\overline{x_3}+x_1(x_2+\overline{x_2})\overline{x_3}\,\overline{x_4}=\\
&=\overline{x_1}x_3+\overline{x_1}x_2\overline{x_3}x_4+x_1\overline{x_2}\,\overline{x_3}+x_1x_2\overline{x_3}\,\overline{x_4}+x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}=\\
&=\overline{x_1}x_3+\overline{x_1}x_2\overline{x_3}x_4+x_1\overline{x_2}\,\overline{x_3}+x_1x_2\overline{x_3}\,\overline{x_4}.
\end{aligned} \quad (21)$$

The result of the orthogonalization of assigned function (18) by figurative transformations (19) and algebraic technique (21) is the same.

7. 4. Equivalent transformations of logic functions on the Reed-Muller basis

Since $x\oplus\overline{x}=1$, the Reed-Muller algebra allows the operation to glue variables:

$$x_1x_2x_3\oplus x_1\overline{x_2}x_3=x_1x_3. \quad (22)$$

Proof:

$$x_1x_2x_3\oplus x_1\overline{x_2}x_3=x_1x_3\left(x_2\oplus\overline{x_2}\right)=x_1x_3.$$

The rule of gluing the variables for ESOP (22) has an illustration of the combinatorial representation (23).

$$\left|\begin{array}{ccc}1 & 1 & 1\\ 1 & 0 & 1\end{array}\right|=\left|\begin{array}{ccc}1 & & 1\end{array}\right|=x_1x_3. \quad (23)$$

The logic operation of semi-gluing the variables for ESOP (24) is as follows:

$$x_1\oplus\overline{x_1}x_2=x_1\vee x_2. \quad (24)$$

Proving rule (24) is based on formula (4). And, since the left-hand side of (24) is singular, in order to implement the simplification procedure by means of the operation of semi-gluing the variables, it is necessary to switch from ESOP to PESOP of the logic function.

The operation of semi-gluing the variables (24) has an illustration of the representation (25).

$$\left|\begin{array}{cc}1 & \\ 0 & 1\end{array}\right|=\left|\begin{array}{cc}1 & 1\end{array}\right|=x_1+x_2. \quad (25)$$

The hermeneutics of the logic operation of semi-gluing the variables (25) implies a transition from the ESOP function to the PESOP function.

Other variants of semi-gluing the variables:

$$\overline{x_1}\oplus x_1x_2=\overline{x_1}\vee x_2; \quad (26)$$

$$\overline{x_1}\oplus x_1\overline{x_2}=\overline{x_1}\vee\overline{x_2}. \quad (27)$$

The operation of semi-gluing the variables for ESOP can take the following form, for example [18]:

$$\overline{x_1}\,\overline{x_2}\oplus x_1\overline{x_2}\,\overline{x_3}=\overline{x_2}\oplus x_1\overline{x_2}x_3. \quad (28)$$

Proof:

$$\begin{aligned}
&\overline{x_1}\,\overline{x_2}\oplus x_1\overline{x_2}\,\overline{x_3}=\overline{x_2}\left(\overline{x_1}\oplus x_1\overline{x_3}\right)=\overline{x_2}\left(\overline{x_1}+\overline{x_3}\right)=\\
&=\overline{x_2}\,\overline{x_1x_3}=\overline{x_2}(1\oplus x_1x_3)=\overline{x_2}\oplus x_1\overline{x_2}x_3.
\end{aligned} \quad (29)$$

The operation of semi-gluing the variables (28) has an illustration of the representation (30) [18].

$$\left|\begin{array}{ccc}0 & 0 & \\ 1 & 0 & 0\end{array}\right|=\left|\begin{array}{ccc} & 0 & \\ 1 & 0 & 1\end{array}\right|=\overline{x_2}\oplus x_1\overline{x_2}x_3. \quad (30)$$

Rule (28) in paper [18] is proved by using the Hamming distance.

The rule of semi-gluing the variables for PESOP logical functions.

For 4-variable conjuncterms of PESOP logic functions, the rule of super-gluing the variables can take the following form, for example:

$$\begin{aligned}
&x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}\oplus x_1\overline{x_2}\,\overline{x_3}x_4\oplus x_1\overline{x_2}x_3\overline{x_4}\oplus x_1\overline{x_2}x_3x_4=\\
&=x_1\overline{x_2}\left(\overline{x_3}\,\overline{x_4}\oplus\overline{x_3}x_4\oplus x_3\overline{x_4}\oplus x_3x_4\right)=\\
&=x_1\overline{x_2}\left(\overline{x_3}\left(\overline{x_4}\oplus x_4\right)\oplus x_3\left(\overline{x_4}\oplus x_4\right)\right)=\\
&=x_1\overline{x_2}\left(\overline{x_3}\oplus x_3\right)=x_1\overline{x_2}.
\end{aligned} \quad (31)$$

The equivalent transformations for the rule of semi-gluing the variables (31) have an illustration of representation (32):

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{vmatrix} = 10 = x_1\overline{x_2}. \tag{32}$$

The super-gluing rule for ESOP (32) is based on the use of a complete combinatorial system with repeated 2-(2, 4)-design [19].

The super-gluing variable rule for ESOP, which uses a complete combinatorial system with repeated 2-(3, 8)-design [19] may take the following form, for example:

$$F = \begin{vmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} & & 1 & & 1 \end{vmatrix} = x_3x_5. \tag{33}$$

The rule of incomplete super-gluing the variables for ESOP logic functions.

Combinatorial properties of an incomplete combinatorial system with repeated 2-$(n, x/b)$-design [19] ensure a rule of incomplete super-gluing the variables in the Reed-Muller basis.

For a 2-variable ESOP function's conjuncterms, the rule of incomplete super-gluing the variables may take the following form, for example:

$$\begin{aligned} & f(x_1, x_2) = \overline{x_1}x_2 \oplus x_1x_2 \oplus x_1\overline{x_2} = \\ & = x_2\left(\overline{x_1} \oplus x_1\right) \oplus x_1\overline{x_2} = \\ & = x_2 \oplus x_1\overline{x_2} = x_1 + x_2. \end{aligned} \tag{34}$$

Equivalent transformations for the rule of incomplete super-gluing the variables (34) have an illustration of representation (35):

$$\begin{vmatrix} \color{red}{0} & \color{red}{1} \\ \color{red}{1} & \color{red}{1} \\ 1 & 0 \end{vmatrix} = \begin{vmatrix} & \color{red}{1} \\ 1 & 0 \end{vmatrix} = \begin{vmatrix} & 1 \\ 1 & \end{vmatrix} = x_1 + x_2. \tag{35}$$

In the second matrix of (35), the operation of semi-gluing the variables for ESOP of logic functions (24) was applied, the result of which is represented in the third matrix (35).

Rule (35) uses an incomplete combinatorial system with repeated 2-(2, 3/4)-design [19].

Generalized gluing the variables in the Reed-Muller basis can be carried out using the following transformations:

$$1.\ x_1x_2 \oplus x_1x_3 \oplus x_2\overline{x_3} = \overline{x_1}x_2\overline{x_3} \oplus x_1\overline{x_2}x_3. \tag{36}$$

Proof:

$$\begin{aligned} & x_1x_2 \oplus x_1x_3 \oplus x_2\overline{x_3} = \\ & = x_1x_2x_3 \oplus x_1x_2\overline{x_3} \oplus x_1x_3 \oplus x_2\overline{x_3} = \\ & = x_1\overline{x_2}x_3 \oplus x_1x_2\overline{x_3} \oplus x_2\overline{x_3} = \overline{x_1}x_2\overline{x_3} \oplus x_1\overline{x_2}x_3. \end{aligned}$$

Equivalent transformations for the rule of generalized gluing the variables (36) have an illustration of representation (37):

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

$$= \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{vmatrix} = \overline{x_1}x_2\overline{x_3} \oplus x_1\overline{x_2}x_3. \tag{37}$$

The second and third matrices (37) involve an operation of absorbing the variables for PESOP (40) to (45).

$$2.\ x_1x_2 \oplus x_1x_3 \oplus x_2\overline{x_3} = x_1\left(x_2 \oplus x_3\right) \oplus x_2\overline{x_3}. \tag{38}$$

The rule of generalized gluing the variables (38) has an illustration of representation (39):

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

$$= x_1\left(x_2 \oplus x_3\right) \oplus x_2\overline{x_3}. \tag{39}$$

Both generalized variable gluing rules (36) and (38) have a 3-level logic. Applying rule (38) reduces the original expression by one literal.

The logic operation of variable absorption for ESOP is as follows:

$$x_1 \oplus x_1\overline{x_2} = x_1x_2. \tag{40}$$

The validity of logic expression (40) is confirmed by a truth table (Table 4).

Table 4

The truth table of the logic operation of absorbing the variables $x_1 \oplus x_1\overline{x_2} = x_1x_2$

x_1	x_2	$\overline{x_2}$	$x_1\overline{x_2}$	$x_1 \oplus x_1\overline{x_2}$	x_1x_2
0	0	1	0	0	0
0	1	0	0	0	0
1	0	1	1	0	0
1	1	0	0	1	1

Other variants of the operation of absorbing the variables for ESOP:

$$x_1 \oplus x_1x_2 = x_1\overline{x_2}; \tag{41}$$

$$x_1x_2x_3 \oplus x_1x_3 = x_1\overline{x_2}x_3; \tag{42}$$

$$x_1x_2 \oplus x_1 = x_1\overline{x_2}; \tag{43}$$

$$x_1x_2x_3 \oplus x_1x_2 = x_1x_2\overline{x_3}; \tag{44}$$

$$x_1 \oplus x_1x_2x_3 = x_1\overline{x_2x_3} = x_1\left(\overline{x_2} + \overline{x_3}\right). \tag{45}$$

The created classical rules for the equivalent transformation of logic functions in the Reed-Muller basis ensure their effective simplification by a figurative transformation method.

7. 5. Examples of minimizing Boolean functions in the Reed-Muller basis by a figurative transformation method

Proving the advantage of logic expressions in a polynomial format comes down to a smaller number of logic elements in their respective schemes, compared to the expressions of the disjunctive (conjunctive) form. It is also promising to use a mixed basis.

Example 2: It is required to simplify the Boolean function $f(x_1, x_2, x_3, x_4)$ in the polynomial form (ESOP), assigned by the Karnaugh map (Fig. 1) [20].

cd \ ab	00	01	11	10
00	1	1	1	1
01	0	0	1	1
11	1	1	1	1
10	1	0	0	1

Fig. 1. Minimizing Boolean functions using a Karnaugh map

Solution.

Since the conjuncterms of the initial function are pairwise orthogonal (a singular function), in order to simplify the assigned function, we select the Reed-Muller algebra. The minimization of $f(x_1, x_2, x_3, x_4)$ (Fig. 1) in a polynomial format is carried out using the following figurative transformations:

$$f=\begin{array}{c|cccc|}0&0&0&0&0\\2&0&0&1&0\\3&0&0&1&1\\4&0&1&0&0\\7&0&1&1&1\\8&1&0&0&0\\9&1&0&0&1\\10&1&0&1&0\\11&1&0&1&1\\12&1&1&0&0\\13&1&1&0&1\\15&1&1&1&1\end{array}=\left|\begin{array}{cccc}0&0&0&0\\0&0&1&0\\0&0&1&1\\0&1&0&0\\0&1&1&1\\0&0&0&1\\0&0&0&1\\0&1&0&1\\0&1&0&1\\0&1&1&0\\0&1&1&0\\1&0&0&0\\1&0&0&1\\1&0&1&0\\1&0&1&1\\1&1&0&0\\1&1&0&1\\1&1&1&1\\1&1&1&0\\1&1&1&0\end{array}\right|=$$

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

$$=1\oplus\left|\begin{array}{cccc}0&&0&1\\&1&1&0\end{array}\right|=1\oplus\overline{x}_1\,\overline{x}_3x_4\oplus x_2x_3\overline{x}_4=$$

$$=\overline{\overline{x}_1\,\overline{x}_3x_4}\oplus x_2x_3\overline{x}_4=(x_1+x_3+\overline{x}_4)\oplus x_2x_3\overline{x}_4.$$

Function $f(x_1,x_2,x_3,x_4)$ MPNF (Fig. 1):

$$f_{\text{MPNF}}=1\oplus\overline{x}_1\,\overline{x}_3x_4\oplus x_2x_3\overline{x}_4. \qquad (46)$$

The cost of (46) implementation is $k_\theta/k_l/k_{in}=3/6/3$.

To minimize the function in Fig. 1, an algorithm to simplify the function involving the procedure of inserting two identical ESOP conjuncterms with the following operation of super-gluing the variables (chapter 7.2) was applied.

The simplified function $f(x_1, x_2, x_3, x_4)$ (Fig. 1) in the mixed basis:

$$f_{\text{MFMB}}=\left(x_1+x_3+\overline{x}_4\right)\oplus x_2x_3\overline{x}_4. \qquad (47)$$

The cost of (47) implementation is $k_\theta/k_l/k_{in}=2/6/2$.

Both functions (46) and (47) represent a 3-level logic. On a mixed basis, the minimum function (47) has better implementation indicators.

Table 5 gives the results from minimizing the function $f(x_1, x_2, x_3, x_4)$ (Fig. 1) in ESOP using a Karnaugh map [20] and the method of figurative transformations.

Table 5

Results from minimizing the function $f(x_1, x_2, x_3, x_4)$ (Fig. 1) in ESOP

Karnaugh map	Method of figurative transformations
$P_{1000}(a,b,c,d)=1\oplus bc\oplus\overline{a}d\oplus\overline{a}cd\oplus bcd$ $P_{1011}(a,b,c,d)=1\oplus\overline{a}\,\overline{c}\oplus b\overline{c}\oplus\overline{a}\,\overline{c}\,\overline{d}\oplus b\overline{c}\,\overline{d}$	$f_{\text{MPNF}}=$ $=1\oplus\overline{x}_1\,\overline{x}_3x_4\oplus x_2x_3\overline{x}_4$

When contemplating Table 5, we see that the result of simplifying the function $f(x_1, x_2, x_3, x_4)$ (Fig. 1) by a figurative transformation method is a minimal function containing six literals. This is four literals less compared to [20].

The verification of the resulting MPNF (46) is given in Table 6.

Table 6

Verification of MPNF (46) – $1\oplus\overline{x}_1\,\overline{x}_3x_4\oplus x_2x_3\overline{x}_4$

No.	x_1	x_2	x_3	x_4	f (Fig. 1)	$1\oplus\overline{x}_1\,\overline{x}_3x_4\oplus x_2x_3\overline{x}_4$	f_{MPNF}
0	0	0	0	0	1	$1\oplus\overline{0}_1\,\overline{0}_30_4\oplus0_20_3\overline{0}_4$	1
1	0	0	0	1	0	$1\oplus\overline{0}_1\,\overline{0}_31_4\oplus0_20_3\overline{1}_4$	0
2	0	0	1	0	1	$1\oplus\overline{0}_1\,\overline{1}_30_4\oplus0_21_3\overline{0}_4$	1
3	0	0	1	1	1	$1\oplus\overline{0}_1\,\overline{1}_31_4\oplus0_21_3\overline{1}_4$	1
4	0	1	0	0	1	$1\oplus\overline{0}_1\,\overline{0}_30_4\oplus1_20_3\overline{0}_4$	1
5	0	1	0	1	0	$1\oplus\overline{0}_1\,\overline{0}_31_4\oplus1_20_3\overline{1}_4$	0
6	0	1	1	0	0	$1\oplus\overline{0}_1\,\overline{1}_30_4\oplus1_21_3\overline{0}_4$	0
7	0	1	1	1	1	$1\oplus\overline{0}_1\,\overline{1}_31_4\oplus1_21_3\overline{1}_4$	1
8	1	0	0	0	1	$1\oplus\overline{1}_1\,\overline{0}_30_4\oplus0_20_3\overline{0}_4$	1
9	1	0	0	1	1	$1\oplus\overline{1}_1\,\overline{0}_31_4\oplus0_20_3\overline{1}_4$	1
10	1	0	1	0	1	$1\oplus\overline{1}_1\,\overline{1}_30_4\oplus0_21_3\overline{0}_4$	1
11	1	0	1	1	1	$1\oplus\overline{1}_1\,\overline{1}_31_4\oplus0_21_3\overline{1}_4$	1
12	1	1	0	0	1	$1\oplus\overline{1}_1\,\overline{0}_30_4\oplus1_20_3\overline{0}_4$	1
13	1	1	0	1	1	$1\oplus\overline{1}_1\,\overline{0}_31_4\oplus1_20_3\overline{1}_4$	1
14	1	1	1	0	0	$1\oplus\overline{1}_1\,\overline{1}_30_4\oplus1_21_3\overline{0}_4$	0
15	1	1	1	1	1	$1\oplus\overline{1}_1\,\overline{1}_31_4\oplus1_21_3\overline{1}_4$	1

Table 6 demonstrates that MPNF (46) $1\oplus\overline{x_1}\,\overline{x_3}x_4\oplus x_2x_3\overline{x_4}$ satisfies the assigned logic function $f(x_1, x_2, x_3, x_4)$ (Fig. 1).

Example 3: It is required, by using the method of figurative transformations, to simplify the Boolean function $f(x_1, x_2, x_3, x_4)$ in ESOP, which is set in the canonical form [18]:

$$f=(0,6,14,15). \tag{48}$$

Solution:

$$f=\left|\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 1 & 1 & 0 \\ 14 & 1 & 1 & 1 & 0 \\ 15 & 1 & 1 & 1 & 1 \end{array}\right| = \left|\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & \end{array}\right| =$$

$$=\left|\begin{array}{cccc} 0 & 0 & & 0 \\ 0 & & 1 & 0 \\ 1 & 1 & 1 & \end{array}\right| = x_1x_2x_3\oplus\overline{x_1}\,\overline{x_4}\left(\overline{x_2\oplus x_3}\right)=$$

$$=x_1x_2x_3\oplus\overline{x_1}\,\overline{x_4}\left(\overline{x_2}\oplus x_3\right)=$$

$$=x_1x_2x_3\oplus\overline{x_1}\,\overline{x_2}\,\overline{x_4}\oplus\overline{x_1}x_3\overline{x_4}.$$

MPNF of function $f(x_1,x_2,x_3)$ (48):

$$f_{\mathrm{MPNF}}=x_1x_2x_3\oplus\overline{x_1}\,\overline{x_2}\,\overline{x_4}\oplus\overline{x_1}x_3\overline{x_4}. \tag{49}$$

When simplifying function (48), identity (3) is taken into consideration.

The results of minimizing (49) function (48) by the method of figurative transformations and by the method of splitting ESOP conjuncterms [18] coincide. The indicator of the implementation of function $k_\theta/k_l/k_{in}=3/9/5$, where k_θ is the number of simple implicants, k_l is the number of input variables, k_{in} is the number of inverted variables. However, the computational complexity of the procedure for minimizing the Boolean function with figurative transformations is less.

Function (48) can be simplified using the Zhegalkin polynomial.

Zhegalkin polynomials for the constituents of function (48) are given in Table 7.

Table 7

Zhegalkin polynomials for the constituents of function $f=(0,6,14,15)$ (48)

Constituents	Zhegalkin polynomials
$\overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}$ 0000	$(1\oplus x_1)(1\oplus x_2)(1\oplus x_3)(1\oplus x_4)=$ $=(1\oplus x_2\oplus x_1\oplus x_1x_2)(1\oplus x_4\oplus x_3\oplus x_3x_4)=$ $=1\oplus x_4\oplus x_3\oplus x_3x_4\oplus x_2\oplus x_2x_4\oplus x_2x_3\oplus$ $\oplus x_2x_3x_4\oplus x_1\oplus x_1x_4\oplus x_1x_3\oplus x_1x_3x_4\oplus$ $\oplus x_1x_2\oplus x_1x_2x_4\oplus x_1x_2x_3\oplus x_1x_2x_3x_4$
$\overline{x_1}x_2x_3\overline{x_4}$ 0110	$(1\oplus x_1)x_2x_3(1\oplus x_4)=$ $=(x_2x_3\oplus x_1x_2x_3)(1\oplus x_4)=$ $=x_2x_3\oplus x_2x_3x_4\oplus x_1x_2x_3\oplus x_1x_2x_3x_4$
$x_1x_2x_3\overline{x_4}$ 1110	$x_1x_2x_3(1\oplus x_4)=x_1x_2x_3\oplus x_1x_2x_3x_4$
$x_1x_2x_3x_4$ 1111	$x_1x_2x_3x_4$

The Zhegalkin polynomial corresponding to function (48) takes the following form:

$$\begin{aligned}Y=\;&1\oplus x_4\oplus x_3\oplus x_3x_4\oplus x_2\oplus x_2x_4\oplus x_2x_3\oplus\\ &\oplus x_2x_3x_4\oplus x_1\oplus x_1x_4\oplus x_1x_3\oplus x_1x_3x_4\oplus\\ &\oplus x_1x_2\oplus x_1x_2x_4\oplus x_1x_2x_3\oplus x_1x_2x_3x_4\oplus\\ &\oplus x_2x_3\oplus x_2x_3x_4\oplus x_1x_2x_3\oplus x_1x_2x_3x_4\oplus\\ &\oplus x_1x_2x_3\oplus x_1x_2x_3x_4\oplus\\ &\oplus x_1x_2x_3x_4.\end{aligned}$$

The simplification of the Zhegalkin polynomial is carried out with the help of figurative transformations.

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

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

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

In the simplification of Zhegalkin polynomial, Zhegalkin algebra (Reed-Muller) is used. The simplification of the Zhegalkin polynomial turns the latter into a Reed-Muller polynomial (50):

$$Y=x_1x_2x_3\oplus\overline{x_1}x_2\overline{x_4}\oplus\overline{x_1}\,\overline{x_3}\,\overline{x_4}. \tag{50}$$

Minimal functions (49) and (50) match.

Example 4: It is required, by using a figurative transformation method, to simplify the Boolean function $f(x_1, x_2, x_3, x_4)$ in ESOP, which is set in the canonical form [18]:

$$f=(0,3,5,6,7,8,9,10,12,15). \tag{51}$$

Solution:

$$f=\begin{array}{c|cccc|}0&0&0&0&0\\3&0&0&1&1\\5&0&1&0&1\\6&0&1&1&0\\7&0&1&1&1\\8&1&0&0&0\\9&1&0&0&1\\10&1&0&1&0\\12&1&1&0&0\\15&1&1&1&1\end{array}=\begin{vmatrix}0&0&0&0\\0&0&1&1\\0&1&0&1\\0&1&1&0\\0&1&1&1\\0&1&0&0\\0&1&0&0\\1&0&0&0\\1&0&0&1\\1&0&1&0\\1&0&1&1\\1&0&1&1\\1&1&0&0\\1&1&1&1\end{vmatrix}=\begin{vmatrix}0&0&0&0\\0&0&1&1\\0&1&&\\0&1&0&0\\1&0&&\\1&0&1&1\\1&1&0&0\\1&1&1&1\end{vmatrix}=$$

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

MPNF of function $f(x_1, x_2, x_3, x_4)$ (51):

$$f_{\mathrm{MPNF}} = x_1 \oplus x_2 \oplus \overline{x}_3 \oplus x_4 \oplus \overline{x}_1 x_2 x_3 x_4 \oplus x_1 \overline{x}_2\, \overline{x}_3\, \overline{x}_4. \qquad (52)$$

Table 8 gives the results from minimizing the function $f(x_1, x_2, x_3, x_4)$ (51) in ESOP by splitting the ESOP conjuncterms [18] and by a figurative transformation method.

Table 8

The result of minimizing
the function $f(x_1, x_2, x_3, x_4)$ (51) in ESOP

Method of splitting conjuncterms	Figurative transformation method
$f = x_1 \oplus x_2 \oplus \overline{x}_2 x_3 x_4 \oplus \overline{x}_1\, \overline{x}_3\, \overline{x}_4 \oplus$ $\oplus x_1 x_2 \overline{x}_3 \oplus x_1 x_2 x_4$	$f = x_1 \oplus x_2 \oplus \overline{x}_3 \oplus x_4 \oplus$ $\oplus \overline{x}_1 x_2 x_3 x_4 \oplus x_1 \overline{x}_2\, \overline{x}_3\, \overline{x}_4$

When contemplating Table 8, we see that the result of minimizing the function $f(x_1, x_2, x_3, x_4)$ (51) by a figurative transformation method is a minimal function (52), containing 12 literals. This is 2 literals less compared to [18] and 3 literals less compared to [22].

The verification of the resulting MPNF (52) is given in Table 9. Table 9 shows that MPNF (52) $x_1 \oplus x_2 \oplus \overline{x}_3 \oplus x_4 \oplus$ $\oplus \overline{x}_1 x_2 x_3 x_4 \oplus x_1 \overline{x}_2\, \overline{x}_3\, \overline{x}_4$ satisfies the assigned logic function $f(x_1, x_2, x_3, x_4)$ (51).

Example 5: It is required, by using a figurative transformation method, to simplify the Boolean function $f(x_1, x_2, x_3, x_4)$ in ESOP, which is set in the canonical form (53) [23]:

$$f = (0,2,4,7,9,10,12,13). \qquad (53)$$

Table 9

Verification of MPNF (52) $x_1 \oplus x_2 \oplus \overline{x}_3 \oplus x_4 \oplus \overline{x}_1 x_2 x_3 x_4 \oplus x_1 \overline{x}_2\, \overline{x}_3\, \overline{x}_4$

No.	x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$ (51)	$x_1 \oplus x_2 \oplus \overline{x}_3 \oplus x_4 \oplus \overline{x}_1 x_2 x_3 x_4 \oplus x_1 \overline{x}_2\, \overline{x}_3\, \overline{x}_4$	f_{MPNF}
0	0	0	0	0	1	$0_1 \oplus 0_2 \oplus \overline{0}_3 \oplus 0_4 \oplus \overline{0}_1 0_2 0_3 0_4 \oplus 0_1 \overline{0}_2\, \overline{0}_3\, \overline{0}_4$	1
1	0	0	0	1	0	$0_1 \oplus 0_2 \oplus \overline{0}_3 \oplus 1_4 \oplus \overline{0}_1 0_2 0_3 1_4 \oplus 0_1 \overline{0}_2\, \overline{0}_3\, \overline{1}_4$	0
2	0	0	1	0	0	$0_1 \oplus 0_2 \oplus \overline{1}_3 \oplus 0_4 \oplus \overline{0}_1 0_2 1_3 0_4 \oplus 0_1 \overline{0}_2\, \overline{1}_3\, \overline{0}_4$	0
3	0	0	1	1	1	$0_1 \oplus 0_2 \oplus \overline{1}_3 \oplus 1_4 \oplus \overline{0}_1 0_2 1_3 1_4 \oplus 0_1 \overline{0}_2\, \overline{1}_3\, \overline{1}_4$	1
4	0	1	0	0	0	$0_1 \oplus 1_2 \oplus \overline{0}_3 \oplus 0_4 \oplus \overline{0}_1 1_2 0_3 0_4 \oplus 0_1 \overline{1}_2\, \overline{0}_3\, \overline{0}_4$	0
5	0	1	0	1	1	$0_1 \oplus 1_2 \oplus \overline{0}_3 \oplus 1_4 \oplus \overline{0}_1 1_2 0_3 1_4 \oplus 0_1 \overline{1}_2\, \overline{0}_3\, \overline{1}_4$	1
6	0	1	1	0	1	$0_1 \oplus 1_2 \oplus \overline{1}_3 \oplus 0_4 \oplus \overline{0}_1 1_2 1_3 0_4 \oplus 0_1 \overline{1}_2\, \overline{1}_3\, \overline{0}_4$	1
7	0	1	1	1	1	$0_1 \oplus 1_2 \oplus \overline{1}_3 \oplus 1_4 \oplus \overline{0}_1 1_2 1_3 1_4 \oplus 0_1 \overline{1}_2\, \overline{1}_3\, \overline{1}_4$	1
8	1	0	0	0	1	$1_1 \oplus 0_2 \oplus \overline{0}_3 \oplus 0_4 \oplus \overline{1}_1 0_2 0_3 0_4 \oplus 1_1 \overline{0}_2\, \overline{0}_3\, \overline{0}_4$	1
9	1	0	0	1	1	$1_1 \oplus 0_2 \oplus \overline{0}_3 \oplus 1_4 \oplus \overline{1}_1 0_2 0_3 1_4 \oplus 1_1 \overline{0}_2\, \overline{0}_3\, \overline{1}_4$	1
10	1	0	1	0	1	$1_1 \oplus 0_2 \oplus \overline{1}_3 \oplus 0_4 \oplus \overline{1}_1 0_2 1_3 0_4 \oplus 1_1 \overline{0}_2\, \overline{1}_3\, \overline{0}_4$	1
11	1	0	1	1	0	$1_1 \oplus 0_2 \oplus \overline{1}_3 \oplus 1_4 \oplus \overline{1}_1 0_2 1_3 1_4 \oplus 1_1 \overline{0}_2\, \overline{1}_3\, \overline{1}_4$	0
12	1	1	0	0	1	$1_1 \oplus 1_2 \oplus \overline{0}_3 \oplus 0_4 \oplus \overline{1}_1 1_2 0_3 0_4 \oplus 1_1 \overline{1}_2\, \overline{0}_3\, \overline{0}_4$	1
13	1	1	0	1	0	$1_1 \oplus 1_2 \oplus \overline{0}_3 \oplus 1_4 \oplus \overline{1}_1 1_2 0_3 1_4 \oplus 1_1 \overline{1}_2\, \overline{0}_3\, \overline{1}_4$	0
14	1	1	1	0	0	$1_1 \oplus 1_2 \oplus \overline{1}_3 \oplus 0_4 \oplus \overline{1}_1 1_2 1_3 0_4 \oplus 1_1 \overline{1}_2\, \overline{1}_3\, \overline{0}_4$	0
15	1	1	1	1	1	$1_1 \oplus 1_2 \oplus \overline{1}_3 \oplus 1_4 \oplus \overline{1}_1 1_2 1_3 1_4 \oplus 1_1 \overline{1}_2\, \overline{1}_3\, \overline{1}_4$	1

Solution:

$$f=\begin{array}{r}0\\2\\4\\7\\9\\10\\12\\13\end{array}\left|\begin{array}{cccc}0&0&0&0\\0&0&1&0\\0&1&0&0\\0&1&1&1\\1&0&0&1\\1&0&1&0\\1&1&0&0\\1&1&0&1\end{array}\right|=\left|\begin{array}{cccc}0&0&0&0\\&0&1&0\\&1&0&0\\0&1&1&1\\1&&0&1\end{array}\right|=$$

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

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

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

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

$$=\left|\begin{array}{cccc}&&&0\\1&0&0&\\1&1&&\\1&1&&0\\&1&1&\end{array}\right|=\left|\begin{array}{cccc}&&&0\\1&0&0&\\1&1&&1\\&1&1&\end{array}\right|. \qquad (54)$$

MPNF of function $f(x_1,x_2,x_3,x_4)$ (53):

$$f_{\text{MPNF}}=x_1x_2x_4\oplus x_2x_3\oplus x_1\overline{x_2}\,\overline{x_3}\oplus\overline{x_4}. \qquad (55)$$

The cost of (55) implementation is

$$k_\theta/k_l/k_{in}=4/9/3, \qquad (56)$$

that matches [23], however, the function represented by the eleventh matrix in (54):

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

can be represented in a mixed basis:

$$f_{\text{MFMB}}=\overline{x_1}\oplus(x_4+x_1x_2)\oplus(x_1+x_2)x_3, \qquad (57)$$

which represents a 3-level logic as well, but at the following cost of implementation:

$$k_\theta/k_l/k_{in}=3/7/1,$$

which is better than the implementation cost (56) of minimum function (55).

The verification of the resulting MPNF of assigned function (53) in a mixed basis (57) is given in Table 10.

Table 10

Verification of MPNF in a mixed basis (57) – $\overline{x_1}\oplus(x_4+x_1x_2)\oplus(x_1+x_2)x_3$

No.	x_1	x_2	x_3	x_4	$f(x_1,x_2,x_3,x_4)$ (53)	$\overline{x_1}\oplus(x_4+x_1x_2)\oplus(x_1+x_2)x_3$	f_{MFMB}
0	0	0	0	0	1	$\overline{0_1}\oplus(0_4+0_10_2)\oplus(0_1+0_2)0_3$	1
1	0	0	0	1	0	$\overline{0_1}\oplus(1_4+0_10_2)\oplus(0_1+0_2)0_3$	0
2	0	0	1	0	1	$\overline{0_1}\oplus(0_4+0_10_2)\oplus(0_1+0_2)1_3$	1
3	0	0	1	1	0	$\overline{0_1}\oplus(1_4+0_10_2)\oplus(0_1+0_2)1_3$	0
4	0	1	0	0	1	$\overline{0_1}\oplus(0_4+0_11_2)\oplus(0_1+1_2)0_3$	1
5	0	1	0	1	0	$\overline{0_1}\oplus(1_4+0_11_2)\oplus(0_1+1_2)0_3$	0
6	0	1	1	0	0	$\overline{0_1}\oplus(0_4+0_11_2)\oplus(0_1+1_2)1_3$	0
7	0	1	1	1	1	$\overline{0_1}\oplus(1_4+0_11_2)\oplus(0_1+1_2)1_3$	1
8	1	0	0	0	0	$\overline{1_1}\oplus(0_4+1_10_2)\oplus(1_1+0_2)0_3$	0
9	1	0	0	1	1	$\overline{1_1}\oplus(1_4+1_10_2)\oplus(1_1+0_2)0_3$	1
10	1	0	1	0	1	$\overline{1_1}\oplus(0_4+1_10_2)\oplus(1_1+0_2)1_3$	1
11	1	0	1	1	0	$\overline{1_1}\oplus(1_4+1_10_2)\oplus(1_1+0_2)1_3$	0
12	1	1	0	0	1	$\overline{1_1}\oplus(0_4+1_11_2)\oplus(1_1+1_2)0_3$	1
13	1	1	0	1	1	$\overline{1_1}\oplus(1_4+1_11_2)\oplus(1_1+1_2)0_3$	1
14	1	1	1	0	0	$\overline{1_1}\oplus(0_4+1_11_2)\oplus(1_1+1_2)1_3$	0
15	1	1	1	1	0	$\overline{1_1}\oplus(1_4+1_11_2)\oplus(1_1+1_2)1_3$	0

Table 10 shows that MPNF in a mixed basis (57) $\overline{x_1}\oplus(x_4+x_1x_2)\oplus(x_1+x_2)x_3$ satisfies the assigned logic function $f(x_1, x_2, x_3, x_4)$ (53).

8. Discussion of results of minimizing Boolean functions in the Reed-Muller basis by a figurative transformation method

The mathematical apparatus of Boolean function minimization by a figurative transformation method is considered in works [24–26], and others. The technique of the method of figurative transformations is given in Table 11.

Table 11

Figurative transformation method technique

1	Binary combinatorial systems with repeated 2-(n, b)-design, 2-(n, x/b)-design
2	Verbal and figurative presentation of information
3	Logic operation of super-gluing the variables
4	Logical operation of incomplete super-gluing the variables
5	Hermeneutics of logic operations on binary equivalents of logic functions
6	Protocols of figurative transformations
7	Attribute of the minimum logic function
8	Minimization of Boolean functions on the complete truth table
9	Algorithm of analytical method and its automation
10	Extension of the analytical method to other logical bases
11	Algebra of equivalent transformation in the class of perfect normal forms of functions of Schaeffer algebra
12	Algebra of equivalent transformation in the class of perfect implicative normal forms
13	Relatively complex algorithms for the application of logic operations of absorption and super-gluing of variables
14	Stack of logic operations

New components in the minimization of Boolean functions by a figurative transformation method are given in Table 12.

Table 12

Components added to the minimization of Boolean functions by a figurative transformation method

1	Algorithms for simplifying a function with the procedure of inserting two identical ESOP conjuncterms with the following operation of super-gluing the variables
2	Singular function
3	Algebra of equivalent transformation in the class of polynomial normal forms of Boolean functions
4	Mixed basis

The algebra that we created in terms of rules for simplifying functions with an illustration of equivalent figurative transformations of logic procedures makes it possible to spread the method of figurative transformations to minimize Boolean functions on the Reed-Muller basis.

A special feature of the method of figurative transformations is that the method is based on the binary combinatorial systems with repeated 2-(n, b)-design, 2-(n, x/b)-design. For example, the truth table (Table 2) of the logic function $f(a, b, c)$ is a combinatorial system with repetition (chapter 6). This is a sufficient resource to minimize functions and makes it possible to do without auxiliary objects, such as Karnaugh maps, Weich diagrams, acyclic graph, non-directed graph, coverage tables, cubes, etc. The clarity of 2-dimensional binary matrices allows one to manually simplify Boolean functions (using a mathematical editor, such as MathType 7.4.0) within up to 64 input variables [9] for PESOP (DCNF) representation of the function.

The use of the method of figurative transformations to minimize functions in the Reed-Muller basis, to some extent, brings the problem of ESOP simplification to the level of a well-researched problem in the class of disjunctive-conjunctival normal forms (DCNF) of Boolean functions.

The algebra that we created for the equivalent transformation of functions in the Reed-Muller basis is represented by the following logic operations (Table 13):

Table 13

Logic operations on the Reed-Muller basis

No. of entry	Logic operation title	Reference to the text	Representation form
1	Gluing the variables	(22), (23)	ESOP
2	Semi-gluing the variables	(24), (25), (26), (27), (29)	ESOP
3	Super-gluing the variables	(31), (32), (33)	ESOP
4	Incomplete super-gluing the variables	(34), (35)	ESOP
5	Generalized gluing the variables	(36), (37), (38), (39)	ESOP
6	Variable absorption	(40), (41), (42), (43), (44), (45)	ESOP

When simplifying Boolean functions on the Reed-Muller basis, it is advisable to use a mixed basis.

Example 6: It is required, by using a figurative transformation method, to simplify the Boolean function $f(x_1, x_2, x_3, x_4))$ in ESOP, which is set in the canonical form (58) [23]:

$$f = (1,2,4,6,7,8,9,10,15), \tag{58}$$

and is represented by a truth table (Table 14).

Table 14

Truth table of the logic function $f(x_1, x_2, x_3, x_4)$ (63)

No.	x_1	x_2	x_3	x_4	f	No.	x_1	x_2	x_3	x_4	f
1	0	0	0	1	1	8	1	0	0	0	1
2	0	0	1	0	1	9	1	0	0	1	1
4	0	1	0	0	1	10	1	0	1	0	1
6	0	1	1	0	1	15	1	1	1	1	1
7	0	1	1	1	1	–	–	–	–	–	–

Solution:

Function (58) is singular. Simplification of function (58) is to be performed for PESOP and ESOP.

Minimization in PESOP.

Define the stack of logic operations of the first matrix of the function $f(x_1, x_2, x_3, x_4)$ as follows. We combine the sets of variables that contain unity in the far-right position of the set into a separate matrix. Such a matrix is presented in (59) first. In another separate matrix, we combine the sets of variables of function (58) which contain zeros in the extreme right position of the set. Such a matrix is given in (60) first. Simplification of function (58) in each matrix is performed separately.

$$f_{1,7,9,15}=\left|\begin{array}{c|cccc} 1 & 0 & 0 & 0 & 1 \\ 7 & 0 & 1 & 1 & 1 \\ 9 & 1 & 0 & 0 & 1 \\ 15 & 1 & 1 & 1 & 1 \end{array}\right|=\left|\begin{array}{cccc} & 0 & 0 & 1 \\ & 1 & 1 & 1 \end{array}\right|. \quad (59)$$

$$f_{2,4,6,8,10}=\left|\begin{array}{c|cccc} 2 & 0 & 0 & 1 & 0 \\ 4 & 0 & 1 & 0 & 0 \\ 6 & 0 & 1 & 1 & 0 \\ 8 & 1 & 0 & 0 & 0 \\ 10 & 1 & 0 & 1 & 0 \end{array}\right|=\left|\begin{array}{cccc} & 0 & 1 & 0 \\ 0 & 1 & & 0 \\ 1 & 0 & 0 & 0 \end{array}\right|=$$

$$=\left|\begin{array}{cccc} & 0 & 1 & 0 \\ 0 & 1 & & 0 \\ 1 & 0 & & 0 \end{array}\right|. \quad (60)$$

Combine the results of simplification of (59) and (60) in a total matrix:

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

Further simplifications of function (58) are no longer possible. MDNF of function $f\,(x_1, x_2, x_3, x_4)$ (58):

$$f_{\mathrm{MDNF}}=\overline{x_2}\,\overline{x_3}x_4+x_2x_3x_4+\overline{x_2}x_3\overline{x_4}+\overline{x_1}x_2\overline{x_4}+x_1\overline{x_2}\,\overline{x_4}. \quad (61)$$

The cost of (61) implementation is:

$$f_{\mathrm{MDNF}}=k_\theta/k_l/k_{in}=5/15/8. \quad (62)$$

The minimal function $f\,(x_1, x_2, x_3)$ (61) in a mixed basis:

$$f_{\mathrm{MFMB}}=x_2x_3x_4+\overline{x_2}\left(x_3\oplus x_4\right)+\left(x_1\oplus x_2\right)\overline{x_4}. \quad (63)$$

The cost of (63) implementation is:

$$k_\theta/k_l/k_{in}=3/9/2. \quad (64)$$

Minimization in PESOP:

$$f_{\mathrm{MPNF}}=\left|\begin{array}{c|cccc} 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 \\ 4 & 0 & 1 & 0 & 0 \\ 6 & 0 & 1 & 1 & 0 \\ 7 & 0 & 1 & 1 & 1 \\ 8 & 1 & 0 & 0 & 0 \\ 9 & 1 & 0 & 0 & 1 \\ 10 & 1 & 0 & 1 & 0 \\ 15 & 1 & 1 & 1 & 1 \end{array}\right|=\left|\begin{array}{cccc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{array}\right|=\left|\begin{array}{cccc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & & \\ 1 & 0 & & \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{array}\right|=$$

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

MPNF of function $f(x_1, x_2, x_3, x_4)$ (58):

$$f_{\mathrm{MPNF}}=x_2\oplus\overline{x_3}x_4\oplus x_1\overline{x_4}\oplus\overline{x_1}\,\overline{x_2}x_3\overline{x_4}. \quad (65)$$

When simplifying function (65) in ESOP, identity (2) is taken into consideration.

The result of minimization (65) coincides with the result of minimization in [23].

The cost of (65) implementation is:

$$k_\theta/k_l/k_{in}=4/9/5. \quad (66)$$

All minimum functions (61), (63), and (65) represent a 3-level logic. On a mixed basis, the minimum function (63) has better implementation indicators (64).

The method of figurative transformations provides simplification of an arbitrary ESOP function.

Example 7: It is required to find the minimum algebraic form of the Boolean function $f\,(a, b, c, d)$ (67) [18, 27]:

$$f\left(a,b,c,d\right)=\bar{a}\,\bar{c}\oplus\bar{a}b\bar{c}\,\bar{d}\oplus ab\oplus a\bar{c}d. \quad (67)$$

Solution:

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

MPNF of function $f(a, b, c, d)$ (67):

$$f_{\mathrm{MPNF}}=\bar{a}\,\bar{b}\,\bar{c}\,\bar{d}\oplus ab\oplus\bar{c}d. \quad (68)$$

The result of minimizing (68) the function (67) coincides with the result of minimization in [18, 27] but the procedure for minimizing when using figurative transformations is easier.

The selection of the stack of logic operations [9] for ESOP functions is demonstrated by the following example.

Example 8: It is required to choose the optimal stack of logic operations to simplify the Boolean function $f(x_1, x_2, x_3, x_4)$ in ESOP set in the canonical form (69) [18]:

$$f=\left(0,1,6,8,11,14,15\right). \quad (69)$$

Solution.

Function (69) is singular. To simplify (69), we choose Reed-Muller's algebra.

The procedure for splitting conjuncterms [18] and the corresponding stack of logic operations in the first matrix of function (69) produce the following result:

$$f_{\mathrm{MPNF}}=\left|\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 6 & 0 & 1 & 1 & 0 \\ 8 & 1 & 0 & 0 & 0 \\ 11 & 1 & 0 & 1 & 1 \\ 14 & 1 & 1 & 1 & 0 \\ 15 & 1 & 1 & 1 & 1 \end{array}\right|=\left|\begin{array}{cccc} 0 & 0 & 0 & \\ 0 & 1 & 1 & 0 \\ 1 & 0 & & 0 \\ 1 & 0 & 1 & \\ 1 & 1 & 1 & \end{array}\right|=\left|\begin{array}{cccc} 0 & 0 & 0 & \\ 0 & 1 & 1 & 0 \\ 1 & 0 & & 0 \\ 1 & & 1 & \end{array}\right|=$$

$$=\left(1-1-\right)\oplus\left(10-0\right)\oplus\left(\underline{000-}\right)\oplus\left(\underline{0110}\right). \quad (70)$$

Next, the underlined pair of terms in expression (70) is treated with the actual procedure of splitting conjuncterms [18]:

$$\begin{pmatrix} 000- \\ 0110 \end{pmatrix} \overset{\oplus}{\Rightarrow} \begin{pmatrix} 00-- \\ 0-1- \\ 0111 \end{pmatrix}.$$

Upon deriving the following expression:

$$f_{\mathrm{MPNF}} = (\underline{1-1-}) \oplus (\underline{\underline{10-0}}) \oplus \begin{pmatrix} \underline{\underline{00--}} \\ \underline{0-1-} \\ 0111 \end{pmatrix},$$

the underlined pairs are treated with the rules of simplification given in [18]:

$$\begin{pmatrix} 1-1- \\ 0-1- \end{pmatrix} \overset{\oplus}{\Rightarrow} (--1-) \text{ and } \begin{pmatrix} 10-0 \\ 00-- \end{pmatrix} \overset{\oplus}{\Rightarrow} \begin{pmatrix} -0-- \\ 10-1 \end{pmatrix}.$$

Next, the minimal PESOP function is derived [18]:

$$f_{\mathrm{MPNF}} = (--1-) \oplus (-0--) \oplus (10-1) \oplus (0111). \tag{71}$$

The algorithm of ESOP function simplification, which consists of the procedure for inserting two identical conjuncterms with the following operation of super-gluing the variables (chapter 7. 2), and the corresponding stack of logic operations in the first matrix of function (69), produce the following result:

$$f_{\mathrm{MPNF}} = \left| \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 6 & 0 & 1 & 1 & 0 \\ 8 & 1 & 0 & 0 & 0 \\ 11 & 1 & 0 & 1 & 1 \\ 14 & 1 & 1 & 1 & 0 \\ 15 & 1 & 1 & 1 & 1 \end{array} \right| = \left| \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{array} \right| =$$

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

MPNF of function $f(x_1, x_2, x_3, x_4)$ (69):

$$f_{\mathrm{MPNF}} = x_3 \oplus \overline{x_2} \oplus x_1 \overline{x_2} x_4 \oplus \overline{x_1} x_2 x_3 x_4. \tag{72}$$

The minimum ESOP functions (71) and (72) are the same but the simplification procedure based on the second stack is much simpler.

Limiting the use of the method of figurative transformations are cases when the switch function is represented on a mixed basis. In this case, the function must be represented by one logic basis.

The weak side of the method considered here is the small practical application of the method of figurative transformations to minimize Boolean functions in a polynomial format, followed by the design and manufacture of the corresponding computational components. The negative internal factors of the method are associated with additional time costs for establishing protocols for simplifying logic functions in the Reed-Muller basis, followed by the creation of a library of protocols that have an illustration of the corresponding figurative transformations.

The prospect of further research could be the search for new rules for the transformation of symmetrical logic functions and their minimization.

9. Conclusions

1. The perfect normal form of polynomial basis functions can be represented by equivalent binary sets (10) or an equivalent binary matrix (11), which, in this case, would give the conjuncterms of polynomial functions and the operation of adding modulo two for them. Such hermeneutics should be effectively used in simplifying logic functions and when deriving the result of logic operations in the binary matrix class of functions in the Reed-Muller basis.

2. Relatively complex algorithms for simplifying logic expressions involving the procedure of inserting two identical conjuncterms of polynomial functions with the following operation of super-gluing the variables (12), (14), (16) expand variants for their application, which increases the efficiency of the procedure for minimizing Boolean functions in ESOP by a figurative transformation method.

3. The apparatus of the method of figurative transformations effectively ensures the orthogonalization of logic functions and detects the singular function.

4. In order to properly simplify functions in the Reed-Muller basis by a figurative transformation method, Reed-Muller's algebra was refined in terms of the classical rules of equivalent transformation of ESOP and PESOP of Boolean functions of polynomial basis. The creation of polynomial algebra in terms of these rules largely solves the problem of minimizing functions on the Reed-Muller basis.

5. The effectiveness of the figurative transformation method to minimize Boolean functions in the Reed-Muller basis is demonstrated by the following examples:

– example 2 [20] – minimization of the 4-bit Boolean function;

– examples 3, 4 [18], examples 5, 6 [23], example 7 [18, 27] – minimization of 4-bit Boolean functions.

Based on the results of our comparison, it was established that the effectiveness of the method of figurative transformations to minimize Boolean functions in the Reed-Muller basis gives grounds for its application in the procedures for minimizing logic functions since the method of figurative transformations is capable of the following:

– to ensure the operational selection of the logic operations stack in the first binary matrix, which ultimately gives an optimal scenario for minimizing logic functions in the Reed-Muller basis;

– to improve the efficiency of the procedure for minimizing logic functions in the Reed-Muller basis by implementing relatively complex algorithms for simplifying logic expressions, which consist of the procedure for inserting two identical conjuncterms of ESOP functions followed by the operation of super-gluing the variables.

References

1. Rytsar, B. Ye. (2013). A Numeric Set-Theoretical Interpretation of Polynomial Zhegalkin. Upravlinnia systemamy i mashynamy, 1, 11–26. Available at: http://dspace.nbuv.gov.ua/handle/123456789/83125
2. Sasao, T. (1999). Switching Theory for Logic Synthesis. Springer US, 362. doi: https://doi.org/10.1007/978-1-4615-5139-3
3. Sasao, T. (1996). Representations of Logic Functions Using EXOR Operators. Representations of Discrete Functions, 29–54. doi: https://doi.org/10.1007/978-1-4613-1385-4_2
4. Sasao, T. (1997). Easily testable realizations for generalized Reed-Muller expressions. IEEE Transactions on Computers, 46 (6), 709–716. doi: https://doi.org/10.1109/12.600830
5. Zakrevskiy, A. D., Toporov, N. R. (2003). Polinomial'naya realizatsiya chastichnyh bulevyh funktsiy i sistem. Moscow: Editorial URSS, 200. Available at: https://www.libex.ru/detail/book14536.html
6. Zakrevskiy, A. D., Pottosin, Yu. V., Cheremisinova, L. D. (2007). Logicheskie osnovy proektirovaniya diskretnyh ustroystv. Moscow: Fizmatlit, 592. Available at: https://www.libex.ru/detail/book852648.html
7. Fujiwara, H. (1985). Logic testing and design for testability. Cambridge. doi: https://doi.org/10.7551/mitpress/4317.001.0001
8. Faraj, K. (2011). Design Error Detection and Correction System based on Reed-Muller Matrix for Memory Protection. International Journal of Computer Applications, 34 (8), 42–48. Available at: https://research.ijcaonline.org/volume34/number8/pxc3875929.pdf
9. Solomko, M. (2021). Developing an algorithm to minimize boolean functions for the visual-matrix form of the analytical method. Eastern-European Journal of Enterprise Technologies, 1 (4 (109)), 6–21. doi: https://doi.org/10.15587/1729-4061.2021.225325
10. Rytsar, B. (2015). The Minimization Method of Boolean Functionns in Polynomial Set-theoretical Format. Conference: Proc. 24th Inter. Workshop, CS@P'2015. Rzeszow, 130–146. Available at: https://www.researchgate.net/publication/298158364_The_Minimization_Method_of_Boolean_Functionns_in_Polynomial_Set-theoretical_Format
11. Sampson, M., Kalathas, M., Voudouris, D., Papakonstantinou, G. (2012). Exact ESOP expressions for incompletely specified functions. Integration, 45 (2), 197–204. doi: https://doi.org/10.1016/j.vlsi.2011.10.001
12. Knysh, D., Dubrova, E. (2011). Rule-based optimization of AND-XOR expressions. Facta Universitatis – Series: Electronics and Energetics, 24 (3), 437–449. doi: https://doi.org/10.2298/fuee1103437k
13. Bibilo, P. N., Lankevich, Y. Y. (2017). The Use of Zhegalkin Polynomials for Minimization of Multilevel Representations of Boolean Functions Based on Shannon Expansion. Programmnaya Ingeneria, 8 (8), 369–384. doi: https://doi.org/10.17587/prin.8.369-384
14. Egorova, E. K., Cheburakhin, I. F. (2013). On the minimization of complexity and automation of efficient representation of boolean functions in classes of formulas and circuits. Journal of Computer and Systems Sciences International, 52 (4), 618–627. doi: https://doi.org/10.1134/s1064230713030064
15. Frantseva, A. S. (2018). An Algorithm for Minimization of Boolean Functions in the Class of Toffoli Reversible Logic Circuits. The Bulletin of Irkutsk State University. Series Mathematics, 25, 144–158. doi: https://doi.org/10.26516/1997-7670.2018.25.144
16. He, Z., Xiao, L., Huo, Z., Wang, T., Wang, X. (2019). Fast Minimization of Fixed Polarity Reed-Muller Expressions. IEEE Access, 7, 24843–24851. doi: https://doi.org/10.1109/access.2019.2899035
17. Samofalov, K. G., Romlinkevich, A. M., Valuyskiy, V. N., Kanevskiy, Yu. S., Pinevich, M. M. (1987). Prikladnaya teoriya tsifrovyh avtomatov. Kyiv: Vischa shk. Golovnoe izd-vo, 375. Available at: http://stu.scask.ru/book_pta.php?id=62
18. Rytsar, B. Ye. (2015). New minimization method of logical functions in polynomial set-theoretical format. 1. Generalized rules of conjuncterms simplification. Upravlyayuschie sistemy i mashiny, 2, 39–57. Available at: http://dspace.nbuv.gov.ua/handle/123456789/87194
19. 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: https://doi.org/10.15587/2312-8372.2017.118336
20. Akinina, Ju. S., Podvalniy, S. L., Tyurin, S. V. (2016). The application of karnaugh maps for the polinomial transformation of boolean functions. Vestnik Voronezhskogo gosudarstvennogo tehnicheskogo universiteta, 12 (1), 4–7. Available at: https://cyberleninka.ru/article/n/primenenie-kart-karno-dlya-polinomialnogo-preobrazovaniya-bulevyh-funktsiy
21. Rytsar, B. Ye. (2013). A Numeric Set-Theoretical Interpretation of Reed-Muller Expressions with Fixed and Mixed Polarity. Upravlyayuschie sistemy i mashiny, 3, 30–44. Available at: http://dspace.nbuv.gov.ua/handle/123456789/83164
22. Tran, A. (1987). Graphical method for the conversion of minterms to Reed-Muller coefficients and the minimisation of exclusive-OR switching functions. IEE Proceedings E Computers and Digital Techniques, 134 (2), 93. doi: https://doi.org/10.1049/ip-e.1987.0016
23. Rytsar, B. Ye. (2015). A New Method of Minimization of Logical Functions in the Polynomial Set-theoretical Format. 2. Minimization of Complete and Incomplete Functions. Upravlyayuschie sistemy i mashiny, 4, 9–20. Available at: http://dspace.nbuv.gov.ua/handle/123456789/87235
24. 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: https://doi.org/10.15587/2312-8372.2018.140351
25. Riznyk, V., Solomko, M., Tadeyev, P., Nazaruk, V., Zubyk, L., Voloshyn, V. (2020). The algorithm for minimizing Boolean functions using a method of the optimal combination of the sequence of figurative transformations. Eastern-European Journal of Enterprise Technologies, 3 (4 (105)), 43–60. doi: https://doi.org/10.15587/1729-4061.2020.206308
26. Solomko, M., Khomiuk, N., Ivashchuk, Y., Nazaruk, V., Reinska, V., Zubyk, L., Popova, A. (2020). Implementation of the method of image transformations for minimizing the Sheffer functions. Eastern-European Journal of Enterprise Technologies, 5 (4 (107)), 19–34. doi: https://doi.org/10.15587/1729-4061.2020.214899
27. Mishchenko, A., Perkowski, M. (2001). Fast Heuristic Minimization of Exclusive-Sums-of-Products. Proc. Reed-Muller Inter. Workshop'01, 242–250. Available at: https://www.researchgate.net/publication/2367778_Fast_Heuristic_Minimization_of_Exclusive-Sums-of-Products