UDC 681.325
DOI: 10.15587/1729-4061.2020.214899

IMPLEMENTATION OF THE METHOD OF IMAGE TRANSFORMATIONS FOR MINIMIZING THE SHEFFER FUNCTIONS

M. Solomko
PhD, Associate Professor*
E-mail: doctrinas@ukr.net
N. Khomiuk
PhD
Department of International Economic Relations and Project Management
Lesya Ukrainka Eastern European National University
Voli ave., 13, Lutsk, Ukraine, 43025
E-mail: nataljabilous@gmail.com
Y. Ivashchuk
PhD
Department of Higher Mathematics**
E-mail: zrsd@i.ua
V. Nazaruk
PhD*
E-mail: vitalij.nazaruk@gmail.com
V. Reinska
PhD, Associate Professor*
E-mail: vikrein@ukr.net
L. Zubyk
PhD, Associate Professor
Department of Software Systems and Technologies
Taras Shevchenko National University of Kyiv
Volodymyrska str., 60, Kyiv, Ukraine, 01033
E-mail: labrob@ukr.net
A. Popova
PhD
Department of Accounting, Taxation and International Economic Relations
Kharkiv National Automobile and Highway University
Yaroslava Mudroho str., 25, Kharkiv, Ukraine, 61002
E-mail: angelikapoppova@meta.ua
*Department of Computer Engineering**
**National University of Water and Environmental Engineering
Soborna str., 11, Rivne, Ukraine, 33028

The studies have established the possibility of reducing computational complexity, higher productivity of minimization of the Boolean functions in the class of expanded normal forms of the Sheffer algebra functions by the method of image transformations.

Expansion of the method of image transformations to the minimization of functions of the Sheffer algebra makes it possible to identify new algebraic rules of logical transformations. Simplification of the Sheffer functions on binary structures of the 2-(n, b)-designs features exceptional situations. They are used both when deriving the result of simplification of functions from a binary matrix and introducing the Sheffer function to the matrix.

It was shown that the expanded normal form of the n-digit Sheffer function can be represented by binary sets or a matrix. Logical operations with the matrix structure provide the result of simplification of the Sheffer functions. This makes it possible to concentrate the principle of minimization within the truth table of a given function and do without auxiliary objects, such as Karnaugh map, Weich diagrams, coverage tables, etc.

Compared with the analogs of minimizing the Sheffer algebra functions, the method under the study makes the following to be possible:

– reduce algorithmic complexity of minimizing expanded normal forms of the Sheffer functions (ENSF-1 and ENSF-2);

– increase the productivity of minimizing the Sheffer algebra functions by 100–150 %;

– demonstrate clarity of the process of minimizing the ENSF-1 or ENSF-2;

– ensure self-sufficiency of the method of image transformations to minimize the Sheffer algebra functions by introducing the tag of minimum function and minimization in the complete truth table of the ENSF-1 and ENSF-2.

There are reasons to assert that application of the method of image transformations to the minimization of the Sheffer algebra functions brings the problem of minimization of the ENSF-1 and ENSF-2 to the level of a well-studied problem in the class of disjunctive-conjunctive normal forms (DCNF) of Boolean functions

Keywords: method of image transformations, minimization of the Sheffer functions, Sheffer stroke, Sheffer term, ENSF-1, ENSF-2

Received date 14.08.2020
Accepted date 12.10.2020
Published date 30.10.2020

1. Introduction

The entire history of digital circuits is based on the choice of logical basis and optimization of functions on this basis. Webb and Sheffer bases attract attention because they consist of one function, that is, they are monofunctional and easily implemented by transistor circuits. The Sheffer basis uses NAND logic elements.

Technological advantage of the NOR and NAND logic elements follows from two theses. First, they provide a functionally complete basis: each Boolean function can be implemented by combining NOR or NAND elements (Fig. 1).

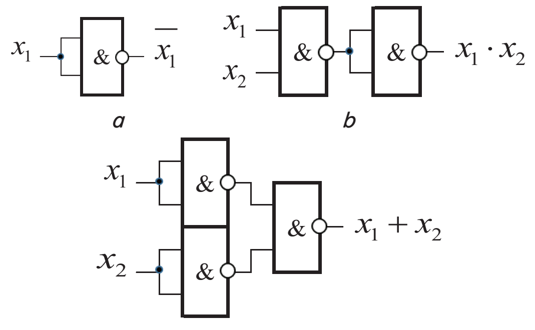


Fig. 1. Implementation of the elements of the basic basis {NOT, AND, OR} using AND-NOT elements: *a* – inverter; *b* – conjunction; *c* – disjunction

The second thesis implies that these logic elements require fewer transistors (for example, NAND gate in the NMOS logic is simpler than the AND or OR logic element) [1–3]. However, developers naturally use the concepts based on the logical basis {AND, OR, NOT}, and not based on {NOR, NAND}. In addition, almost all known methods of minimizing logic circuits from Karnaugh maps to Espresso algorithms, give results based on {AND, OR, NOT} as well [4, 5]. Only after such minimization, special algorithms replace elements of the {AND, OR, NOT} basis with elements of the {NOR, NAND} basis. Such mapping is trivial in a case of two-level minimization in a form of DNF or CNF. However, FPGA circuits feature a multilevel implementation of Boolean functions [6–9]. Optimal conversion of a multilevel circuit from the {AND, OR, NOT} basis to the {NOR, NAND} basis is an uneasy task.

Sheffer stroke is a 2-digit 2-NAND logical operation (Fig. 2).

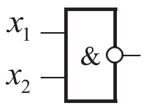


Fig. 2. 2-NAND logic element

The symbols /, | or ↑ are commonly used to denote the «Sheffer stroke» operation. The «Sheffer stroke» operation is a negation of conjunction $f=(x_1/x_2)=\overline{x_1 \cdot x_2}=\overline{x_1}+\overline{x_2}$, so the «false» value of the «Sheffer stroke» operation is only obtained when both arguments x_1 and x_2 take the «true» value (Table 1).

Table 1

Truth table of the «Sheffer stroke» operation

x_1	x_2	$f=x_1/x_2=\overline{x_1 \cdot x_2}=\overline{x_1}+\overline{x_2}$
0	0	1
0	1	1
1	0	1
1	1	0

Since the Sheffer basis belongs to the field of optimization of logical functions [10], the studies aimed at improvement of such factors as:

– methods of simplification of the Boolean functions in the Sheffer basis;

– minimization of logic circuits based on the Sheffer functions;

– reliability of the optimal result of the function minimization;

– cost of the process of minimizing the logical functions remain topical.

2. Literature review and problem statement

The rapid transformation of the multilevel {AND, OR, NOT} basis logic into a functionally equivalent circuit of the {NOR, NAND} basis was considered in [11]. It was shown that the problem can be solved by replacing the AND and OR logic elements with NAND or NOR elements. However, this requires an introduction of additional inverters or distribution of logic elements in some cases. The authors proposed algorithms for fast circuit conversion from the {AND, OR, NOT} basis to the {NOR, NAND} basis while minimizing the number of inverters. The presented algorithms make it possible to convert any multilevel circuit into a circuit that is a combination of the NOR, NAND logic elements, or both types of universal logic elements.

The implementation of logic functions using NAND or NOR gates was considered in [12]. Replacement of the NOT, OR, and AND logic elements in the circuit with the NAND logic elements was demonstrated. Unambiguity of replacing logic elements from the {AND, OR, NOT} basis with elements from the {NOR, NAND} basis was provided by means of de Morgan transformations. The procedure of replacing logic elements is ended with optimization of the circuit consisting in replacement of a combination of an even number of inversions in a form of the NOT function with a conductor segment.

An algorithm of implementation of the Boolean value of functions using only NAND logic elements or only NOR logic elements was presented in [13]. The algorithm of converting the logical structure of the Boolean basis into a mono base structure was performed in stages. Initially, the Boolean function represented by {AND, OR, NOT} logic elements was transformed using de Morgan laws in various forms so that only the NAND elements or only the NOR elements were used in the circuit. Next, excess inverters were removed. In the case where there are two serial inverters (when an inverted output goes directly to an inverted input), both inverters are removed because they cancel each other. At the last stage, the inverters that remained are replaced with equivalent NAND or NOR elements.

Implementation of logic circuits using only the NAND and NOR elements was considered in [14]. It was noted there that the NAND and NOR elements are universal logic elements that can be used to implement any logic function without the need to use any other type of logic elements. This is advantageous in practice because the NAND and NOR elements are economical, easier to manufacture, and are the main elements used in all families of digital logic chips. In fact, the AND elements are usually implemented as a NAND element followed by inverters and not vice versa. Similarly, the OR gate is usually implemented as a NOR gate followed by the inverter and not vice versa. To implement the function using only the NAND elements, it must first be simplified to the sum of productions (DNF). To implement the function using only the NOR elements, it must first be simplified to the product of sums (CNF). It was shown in [14] that the DNF function can be implemented only using the NAND elements while the CNF function can be implemented using only the NOR elements.

The study [15] proposes the implementation of fully optical universal logic gates based on a photonic crystal NAND and NOR with basic logic gates, namely NOT, AND, and OR in a two-level logic with the use of the law of dualization. According to the law, the NAND or NOR elements can be designed with two NOT elements at the first level, cascading to the second OR level or the AND element, respectively. Initially, the basic logic elements were designed and optimized so that their implementation in the design of NAND and NOR gateways cannot degrade the threshold value of logic levels. The length of the input waveguides in each design is kept common to have a change in the interference of the input signals with a phase difference of 0° and 180°. The proposed basic logic elements were obtained with a coefficient of contrast of 11.04, 8.24, and 5.18 dB, respectively while the calculated values of the coefficient of contrast of the NAND and NOR elements are 5.81 and 4.02 dB, respectively. Moreover, the data transfer rate of the proposed designs exceeds 7.485 which indicates their inevitable use in high-speed data networks.

Optimization of combined logic circuits using the NAND elements and genetic programming was considered in [16]. Optimization of the logic circuit that implements the Boolean function can be performed according to various criteria. This can be an optimization of the circuit complexity, the number of logic levels, the number of semiconductor devices in the circuit, and so on. The authors of [16] have presented an approach that uses genetic programming to optimize a given Boolean function concerning the above criteria. Instead of a set of {AND, OR, NOT, XOR} logic elements, universal NAND elements were used which provided better performance and compactness of the circuit. Conventional methods of minimizing logical structures give simplified expressions in two standard forms: the sum of products (SOP) or product of sums (POS). The SOP form can be converted to a NAND expression using a procedure but the conversion results in an optimized scheme neither in terms of the number of logic elements nor in terms of the number of logical levels. The results of experimental studies have shown that the method of genetic programming presented in [16] gives better results compared to the conversion of the SOP form into a NAND expression in terms of the number of gates, logic levels, and semiconductor devices in the circuit.

A general view of synthetic biology as a standard engineering field and synthetic computational structures was considered in [17]. Among the most common logical units, the functions NOT and AND are among the most important elements. An example of synthetic implementation of the NAND logic based on genetic regulatory elements was considered. More complex circuits or chips with combinations of such elements are provided by the planar technology. It was noted that the NOR or NAND logic elements can be used to construct any possible circuit but this extrapolation was not successful in its application to synthetic biological structures.

It follows from a review of [11–17] that the method of genetic programming presented in [16] gives better results in minimizing the logical function using the NAND universal logic elements compared with the transformation of the SOP form into the NAND structure in terms of the number of gates, logic levels, and circuit transistors. The result was determined by the use of multi-purpose optimization of the circuits using genetic programming. Thus, the process of the evolutionary design of a logical path is compared with forms of the conventional design process based on knowledge of design and experience in analytical methods.

The peculiarity of minimizing the Boolean functions by the method of image transformations is in the use of binary matrices with an established list of rules for reducing the Boolean functions. Some combinatorial system, metadata that can explain other data, e.g. determine the minimum Boolean function for other basis is the result of reducing terms of the binary matrix. In addition to this property of binary matrices, they also contain combinatorial images. Semantic capacity, the ability to transfer large amounts of data using a small number of characters, and, as a result, the implication of a considerable part of these data is a characteristic feature of images.

Almost all known methods of minimizing logical functions from Karnaugh maps to Espresso algorithms give results in the Boolean basis. Only after such minimization, special algorithms replace elements of the {AND, OR, NOT} basis with elements of the {NOR, NAND} basis [4, 5, 11–14].

The method of image transformations provides the minimization of logical functions directly in monobases, in particular, the Sheffer basis. Thus, the considered algorithms and methods of minimization of the switching functions [4, 5, 11–14] and the method of image transformations have different approaches and therefore they see different prospects of technological capability of minimizing monofunctional functions. In particular, one of the prospects is the application of new algebraic rules of equivalent transformation of logical functions for the {AND-NOT} monobasis which will expand the capabilities of the analytical method.

In this regard, there is a reason to believe that the procedure of minimizing the switching functions represented by algorithms and methods of minimization [11–15] is insufficient for the theoretical study of optimal minimization of the monofunctional functions. This necessitates studies in equivalent image transformations of the monofunctional functions, in particular, for the {AND-NOT} monobasis. From the application point of view, this approach makes it possible to expand the capabilities of the technology of designing digital components for the {AND-NOT} monobasis.

3. The aim and objectives of the study

The study objective is to extend the method of image transformations to the minimization of the Boolean functions in the class of perfect normal forms of the Sheffer functions (ENSF-1 and ENSF-2). This makes it possible to simplify, improve the productivity of minimizing the Sheffer functions by extending algebraic rules of logical transformations.

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

– establish the adequacy of application of the method of image transformations to minimize the Boolean functions in the Sheffer basis, in particular, determine the hermeneutics of logical operations with binary structures;

– analyze and provide rules for simplifying the Sheffer functions in exceptional situations;

– expand the Sheffer algebra in the part of minimizing logical functions;

– establish a tag of the minimum Sheffer function;

– analyze the feasibility of using image transformations to simplify two normal forms ENSF-1 and ENSF-2 in the complete table of truth of a given Sheffer function.

4. The Sheffer functions

Any logical function in the Sheffer algebra can be represented in a canonical form: AND-NOT/AND-NOT (NAND/NAND). The form AND-NOT/AND-NOT is a normal (two-level), both internally and externally, is the AND-NOT function.

The canonical form of the AND-NOT/AND-NOT logical function of n variables. It consists of the Sheffer terms of the n-th rank united by the AND-NOT operation.

The Sheffer term of the n-th rank takes the following generalized form [18]:

$$\overline{x_n^{\sigma_n}\cdot x_{n-1}^{\sigma_{n-1}}\cdot\ldots\cdot x_n^{\sigma_1}} \text{ or } x_n^{\sigma_n}/x_{n-1}^{\sigma_{n-1}}/\ldots/x_n^{\sigma_1}, \qquad (1)$$

where

$$x_i^{\sigma_i}=\begin{cases} x_i, \text{ if } \sigma_i=1 \\ \overline{x_i}, \text{ if } \sigma_i=0, \end{cases},$$

$i=2,3,\ldots,n,\ \ <\sigma_n\sigma_{n-1}\ldots\sigma_1>$ – binary set.

The Sheffer term $\overline{x_n^{\sigma_n}\cdot x_{n-1}^{\sigma_{n-1}}\cdot\ldots\cdot x_n^{\sigma_1}}$ or the Sheffer function $x_n^{\sigma_n}/x_{n-1}^{\sigma_{n-1}}/\ldots/x_n^{\sigma_1}$ corresponds to any binary set and, conversely, a binary set (tuple) corresponds to the Sheffer term $x_n^{\sigma_n}\cdot x_{n-1}^{\sigma_{n-1}}\cdot\ldots\cdot x_n^{\sigma_1}$ or the Sheffer function $x_n^{\sigma_n}/x_{n-1}^{\sigma_{n-1}}/\ldots/x_n^{\sigma_1}$. For example, the Sheffer term $x_1x_2\bar{x}_3\bar{x}_4$ or the Sheffer function $x_1/x_2/\bar{x}_3/\bar{x}_4$, corresponds to the set <1100> and the set <01001> corresponds to the Sheffer term $\bar{x}_1x_2\bar{x}_3\bar{x}_4x_5$ or the Sheffer function $\bar{x}_1/x_2/\bar{x}_3/\bar{x}_4/x_5$.

The rules for performing the «Sheffer stroke» operation have the following interrelations (Table 2) verified using the truth tables.

The transition from the Boolean basis to the Sheffer basis is performed using the equations:

$$x_1/x_2=\overline{x_1\cdot x_2}=\overline{x_1}+\overline{x_2};$$

$$x_1\cdot x_2=\overline{\overline{x_1}+\overline{x_2}}=\overline{x_1/x_2}=(x_1/x_2)/(x_1/x_2);$$

$$\overline{x_1}=x_1/x_2;$$

$$x_1+x_2=\overline{\overline{x_1}\cdot\overline{x_2}}=\overline{x_1}/\overline{x_2}=(x_1/x_1)/(x_2/x_2).$$

Only the interchange law is valid in the Sheffer algebra:

$$x_1/x_2=x_2/x_1;\quad x_1/x_2/x_3=x_2/x_3/x_1.$$

Table 2

Rules of performing the «Sheffer stroke» operation and conversion formulas

No.	Sheffer stroke, 2-NAND
1	$x/x=\overline{x}$
2	$x/\overline{x}=1$
3	$x/1=\overline{x}$
4	$x/0=1$
5	$\overline{x}/0=1$
6	$\overline{x}/1=x$
7	$x_1/x_2=\overline{\overline{x_1}\downarrow\overline{x_2}}$
8	$\overline{x_1}/\overline{x_2}=\overline{x_1\downarrow x_2}$
9	$\overline{x_1/x_2}=\overline{x_1}\downarrow\overline{x_2}$
10	$\overline{x_1/x_2}=x_1x_2$

All definitions for the functions in the algebra of logic in the {AND, OR, NOT} basis have also their analogs with the Sheffer {AND-NOT} basis (Table 3). Replacement of the {AND, OR, NOT} basis with the {AND-NOT} basis is possible based on de Morgan formulas:

$$x_1\cdot x_2=\overline{\overline{x_1}+\overline{x_2}};\quad x_1+x_2=\overline{\overline{x_1}\cdot\overline{x_2}}.$$

To obtain the ENSF-1, the sets of variables at which the Sheffer function returns a value of 1 are marked in the truth table. The marked sets are written as terms in which the «Sheffer stroke» is used as an operation. The terms are also combined with each other by the «Sheffer stroke» operation (2):

$$f(x_1,x_2,\ldots,x_n)=\underset{\alpha}{/}f(\alpha_1,\alpha_2,\ldots,\alpha_n)x_1^{\alpha_1}/x_2^{\alpha_2}/\ldots/x_n^{\alpha_n}, \qquad (2)$$

where

$$\alpha_i\in\{0,1\} \text{ and } x_i^{\alpha_i}=\begin{cases} x, \text{ if } \alpha_i=1 \\ \overline{x}, \text{ if } \alpha_i=0 \end{cases}.$$

If the variable for a given term takes a value of 1, it is taken in a direct code, otherwise in an inverse code.

Table 3

Thesauri of logical bases

No.	Thesaurus of the {AND, OR, NOT} basis	Thesaurus of the {AND-NOT} basis
1	Implicant	Inversant
2	Simple implicant	Simple inversant
3	Expanded disjunctive normal form (EDNF)	Expanded normal Sheffer form 1 (ENSF-1)
4	Expanded conjunctive normal form (ECNF)	Expanded normal Sheffer form 2 (ENSF-2)
5	Minimal disjunctive normal form (MDNF)	Minimal normal Sheffer form 1 (MNSF-1)
6	Minimal conjunctive normal form (MCNF)	Minimal normal Sheffer form 2 (MNSF-2)

Example 1. Present the ENSF-1 for the $f(x_1, x_2, x_3)$ function (Table 4).

Table 4

The truth table of the $f(x_1, x_2, x_3)$ logical function

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1	1	0	0	1
0	0	1	1	1	0	1	0
0	1	0	0	1	1	0	1
0	1	1	1	1	1	1	0

The following is obtained according to the algorithm:

$$f(x_1,x_2,x_3)=\left(\overline{x_1}/\overline{x_2}/\overline{x_3}\right)/\left(\overline{x_1}/\overline{x_2}/x_3\right)/$$
$$/\left(\overline{x_1}/x_2/x_3\right)/\left(x_1/\overline{x_2}/\overline{x_3}\right)/\left(x_1/x_2/\overline{x_3}\right).$$

To obtain the ENSF-2, the variables in each Sheffer term are combined by the 'Sheffer stroke' operation on which it depends. If the variable for a given term takes zero value, it is taken in a direct code, otherwise in an inverse code.

Example 2. Present the ENSF-2 for the $f(x_1, x_2, x_3)$ function (Table 4).

The following is obtained in accordance with the algorithm:

$$f(x_1,x_2,x_3)=\left(x_1/\overline{x_2}/x_3\right)/\left(\overline{x_1}/x_2/\overline{x_3}\right)/\left(\overline{x_1}/\overline{x_2}/\overline{x_3}\right).$$

By analogy with the two-digit functions, the n-digit Sheffer functions (S_n) are also considered (Table 5):

$$S_n = x_1/x_2/x_3/.../x_n.$$

When $n=2$, a two-digit Sheffer function is obtained from Table 5, and when $n=1$, the Sheffer function degenerates in a negation function. Therefore, the expression x/x can be represented as $\bar{x}$:

$$x/x=\bar{x}.$$

The following relations are valid for the n-digit Sheffer functions:

$$x/x/x/.../x=\bar{x};$$

$$x/x/.../x/\bar{x}/.../\bar{x}=1;$$

$$x_1/.../x_l/1/.../1=x_1/.../x_l;$$

$$x_1/.../x_l/0/.../0=1;$$

$$x_1/x_2/.../x_n=\overline{x_1x_2...x_n}=\bar{x}_1\vee\bar{x}_2\vee...\vee\bar{x}_n.$$

Table 5

The truth table of the n-digit Sheffer function (S_n)

x_1	x_2	x_3	.	.	.	x_{n-1}	x_n	S_n
0	0	0	–	–	–	0	0	1
0	0	0	–	–	–	0	1	1
0	0	1	–	–	–	1	0	0
–	–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–	–
1	0	1	–	–	–	0	1	1
1	1	0	–	–	–	1	0	0
1	1	1	–	–	–	1	1	1

5. The results of minimizing the Sheffer functions by the method of image transformations

Equivalent image transformations give the following result when minimizing the Sheffer functions:

– hermeneutics of logical operations is determined on binary structures;

– rules of simplifying the Sheffer functions are provided in exceptional situations;

– the Sheffer algebra is extended in the part of minimizing the logical functions;

– a tag of minimum Sheffer function is set;

– minimization of the Sheffer functions for the complete truth table is simplified.

5. 1. Hermeneutics of logical operations with binary structures

In the conjunctive normal form (CNF) of the Boolean function:

$$F=\left(\overline{x_1}+\overline{x_2}+\overline{x_3}\right)\left(\overline{x_1}+\overline{x_2}+x_3\right). \quad (3)$$

Replace the variables with inversion $\bar{x}_n$ with 0_n and replace the variables without inversion x_n with 1_n where n is a numerical index determining the bit size of the symbol-variable («1» or «0») in the function maxterm (3) [19]. As a result, the binary equivalent of expression of the logical function in the CNF is obtained [19]:

$$F_{\text{CNF}}=\left(0_1+0_2+0_3\right)\left(0_1+0_2+1_3\right) \quad (4)$$

or the matrix:

$$F_{\text{CNF}}=\begin{vmatrix}0&0&0\\0&0&1\end{vmatrix}. \quad (5)$$

The disjunctive normal form (DNF) of the Boolean function:

$$F_{\text{DNF}}=\overline{x_1}\,\overline{x_2}\,\overline{x_3}+\overline{x_1}\,\overline{x_2}x_3 \quad (6)$$

can be represented by binary codes:

$$F=0_10_20_3+0_10_21_3 \quad (7)$$

or the matrix:

$$F_{\text{DNF}}=\begin{vmatrix}0&0&0\\0&0&1\end{vmatrix}. \quad (8)$$

When observing matrices (5) and (8), it can be seen that the CNF and DNF of the logical functions are represented by matrices with the same combinatorial structures [19]. The difference between them in terms of equivalent transformations lies in the hermeneutics of the logical operations. The matrix representing CNF of the Boolean function (5) presents maxterms of the function and the conjunction operation for them. The matrix representing DNF of the Boolean function (8) presents minterms of the function and the disjunction operation for them.

Combinatorial structure-images and equivalent transformations with them can be extended to the Sheffer basis. For example, if the variables with inversion $\bar{x}$ are replaced with «0», and the variables without inversion x are replaced with «1» for the recording of variable degeneracy in the Sheffer basis,

$$x / x = \overline{x}, \qquad (9)$$

then the binary equivalent of the degeneracy of variables in the Sheffer basis is obtained:

$$1/1 = 0, \qquad (10)$$

or the matrix:

$$\begin{vmatrix} 1 \\ 1 \end{vmatrix} = |\,1\,| = \overline{x}. \qquad (11)$$

Since the original matrix (11) consists of one column, and, therefore, there are no conditions for the occurrence of the third exceptional situation (par. 5.2), the first or second exceptional situation (par. 5.2) can be applied when deriving the result in (11).

The record of the degeneracy of variables in the Sheffer basis (10) using the matrix (11) gives the one-digit terms of the Sheffer function and the «Sheffer stroke» operation for them.

In other words, the degeneracy of variables (10) in the Sheffer basis has an illustration of an image (11).

If operation (9) is extended to the *n*-digit Sheffer terms, a logical operation of idempotency of variables for the Sheffer basis is obtained:

$$(x_1 / x_2) / (x_1 / x_2) = \overline{x_1 / x_2}. \qquad (12)$$

Transformation of expression (12) can be continued:

$$\overline{x_1 / x_2} = \overline{\overline{x_1} + \overline{x_2}} = x_1 x_2. \qquad (13)$$

When examining the expression (13), it can be seen that the idempotence operation for the two-digit Sheffer terms corresponds to the logical product of variables x_1 and x_2.

The operation of idempotence (12) in the Sheffer basis has an illustration of image (14):

$$\begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix} = |\,1 \quad 1\,| = \overline{x_1 / x_2};$$

$$\begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix} = |\,1 \quad 0\,| = \overline{x_2 / \overline{x_2}}. \qquad (14)$$

When deriving the result (14), the second exceptional situation was applied (par. 5.2).

The idempotence operation for the three-digit Schaefer terms corresponds to the logical product of variables x_1, x_2, and x_3:

$$(x_1 / x_2 / x_3) / (x_1 / x_2 / x_3) = \overline{x_1 / x_2 / x_3}. \qquad (15)$$

$$\overline{x_1 / x_2 / x_3} = \overline{\overline{x_1} + \overline{x_2} + \overline{x_3}} = x_1 x_2 x_3.$$

The proof of idempotency of variables x_1, x_2, and x_3 in (15) is based on the following transformations:

$$(x_1 / x_2 / x_3) / (x_1 / x_2 / x_3) = \overline{x_1 x_2 x_3} / \overline{x_1 x_2 x_3} =$$
$$= \left(\overline{x_1} + \overline{x_2} + \overline{x_3}\right) / \left(\overline{x_1} + \overline{x_2} + \overline{x_3}\right) =$$
$$= \overline{\left(\overline{x_1} + \overline{x_2} + \overline{x_3}\right) \cdot \left(\overline{x_1} + \overline{x_2} + \overline{x_3}\right)} = \overline{\overline{x_1} + \overline{x_2} + \overline{x_3}} =$$
$$= x_1 x_2 x_3 = \overline{x_1 / x_2 / x_3}.$$

The operation of idempotence (15) in the Sheffer basis has an illustration of image (16):

$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix} = |\,1 \quad 1 \quad 1\,| = \overline{x_1 / x_2 / x_3};$$

$$\begin{vmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \end{vmatrix} = |\,1 \quad 0 \quad 1\,| = \overline{x_1 / \overline{x_2} / x_3}. \qquad (16)$$

When deriving the result (16), the second exceptional situation was used (par. 5.2).

Oher variants of idempotence of variables for the *n*-digit Sheffer terms have the following form:

$$\overline{(x_1 / x_2 / x_3)} / \overline{(x_1 / x_2 / x_3)} = x_1 / x_2 / x_3. \qquad (17)$$

Proof of the result of operation (17) is based on the following transformations:

$$\overline{(x_1 / x_2 / x_3)} / \overline{(x_1 / x_2 / x_3)} = (x_1 x_2 x_3) / (x_1 x_2 x_3) =$$
$$= \overline{(x_1 x_2 x_3) \cdot (x_1 x_2 x_3)} = \overline{x_1 x_2 x_3} = x_1 / x_2 / x_3.$$

Let us prove the transformation:

$$(x_1 / x_2 / x_3) / \overline{(x_1 / x_2 / x_3)} = 1. \qquad (18)$$

Proof:

$$(x_1 / x_2 / x_3) / \overline{(x_1 / x_2 / x_3)} =$$
$$= \left(\left(\overline{x_1} + \overline{x_2}\right) / x_3\right) / \overline{(x_1 / x_2 / x_3)} =$$
$$= \left(x_1 x_2 + \overline{x_3}\right) / \overline{(x_1 / x_2 / x_3)} =$$
$$= \left(x_1 x_2 + \overline{x_3}\right) / \overline{\left(\left(\overline{x_1} + \overline{x_2}\right) / x_3\right)} =$$
$$= \left(x_1 x_2 + \overline{x_3}\right) / \overline{\left(x_1 x_2 + \overline{x_3}\right)} = \left(x_1 x_2 + \overline{x_3}\right) / \left(\left(\overline{x_1} + \overline{x_2}\right) x_3\right) =$$
$$= \overline{\left(x_1 x_2 + \overline{x_3}\right) \cdot \left(\left(\overline{x_1} + \overline{x_2}\right) x_3\right)} = \left(\overline{x_1 x_2 + \overline{x_3}}\right) + \overline{\left(\left(\overline{x_1} + \overline{x_2}\right) x_3\right)} =$$
$$= \left(\left(\overline{x_1} + \overline{x_2}\right) \cdot x_3\right) + \left(\left(\overline{\overline{x_1} + \overline{x_2}}\right) + \overline{x_3}\right) =$$
$$= \left(\left(\overline{x_1} + \overline{x_2}\right) \cdot x_3\right) + \left(x_1 x_2 + \overline{x_3}\right) = \left(\overline{x_1} + \overline{x_2}\right) \cdot x_3 + x_1 x_2 + \overline{x_3} =$$
$$= \overline{x_1} + \overline{x_2} + x_1 x_2 + \overline{x_3} = \overline{x_1} + \overline{x_2} + x_2 + \overline{x_3} = \overline{x_1} + 1 + \overline{x_3} = 1.$$

Transformation (18) is proved.

To represent the expanded normal form of the *n*-digit Sheffer function (ENSF) by binary equivalent or matrix, variables with inversion $\overline{x}_n$ should be replaced by 0_n and variables without inversion x_n by 1_n [19] where *n* is the numerical index that determines bitness of the symbol-variable «1» or «0» in terms of the Sheffer function.

The perfect normal form of the 3-digit Sheffer function:

$$F = \left(\overline{x_1} / \overline{x_2} / \overline{x_3}\right) / \left(\overline{x_1} / \overline{x_2} / x_3\right), \qquad (19)$$

can be represented by binary sets:

$$F = (0_1 / 0_2 / 0_3) / (0_1 / 0_2 / 1_3), \qquad (20)$$

or the matrix:

$$F = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix}. \qquad (21)$$

In this case, the matrix (21) will be an instance of the class of binary matrices of the Sheffer functions.

When considering the matrix (21), it can be seen that its combinatorial structure coincides with structures of the (5) and (8) matrices. The difference between matrices (21) and (5) and (8) with respect to equipotential transformations lies in the hermeneutics of logical operations: the matrix (21) presents terms of the Sheffer function and the «Sheffer stroke» operation for them. It is expedient to apply these hermeneutics when deriving the result of logical operations in the class of binary matrices of the Sheffer functions.

Example 3. Find the ENSF-1 and ENSF-2 for the $f(x_1, x_2, x_3)$ function (Table 6), find the MNSF-1, MNSF-2 by the method of image transformations.

Table 6

The truth table of the $f(x_1, x_2, x_3)$ function

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	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
1	1	1	1

The ENSF-1 of the $f(x_1, x_2, x_3)$ function takes the following form:

$$F_{\text{ENSF-1}} = \left(\overline{x_1} / \overline{x_2} / x_3\right) / \left(\overline{x_1} / x_2 / x_3\right) / \\ / \left(x_1 / x_2 / \overline{x_3}\right) / \left(x_1 / x_2 / x_3\right).$$

$$F_{\text{MNSF-1}} = \begin{vmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 0 & & 1 \\ 1 & 1 & \end{vmatrix} = \\ = \overline{x_1} x_3 + x_1 x_2 = \left(\overline{x_1} / x_3\right) / \left(x_1 / x_2\right)$$

or

$$F_{\text{MNSF-1}} = \begin{vmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix} = \\ = \begin{vmatrix} 0 & & 1 \\ 1 & 1 & \end{vmatrix} = \left(\overline{x_1} / x_3\right) / \left(x_1 / x_2\right).$$

The rules of gluing the variables (30), (32) were used to search for $F_{\text{MNSF-1}}$.

The ENSF-2 of the $f(x_1, x_2, x_3)$ function takes the form:

$$f\left(x_1, x_2, x_3\right) = \left(x_1 / x_2 / x_3\right) / \left(x_1 / \overline{x_2} / x_3\right) / \\ / \left(\overline{x_1} / x_2 / x_3\right) / \left(\overline{x_1} / x_2 / \overline{x_3}\right).$$

$$F_{\text{MNSF-2}} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 1 & & 1 \\ 0 & 1 & \end{vmatrix} = \\ = \left(x_1 + x_3\right)\left(\overline{x_1} + x_2\right) = \overline{\left(\overline{x_1}\,\overline{x_3}\right)}\,\overline{\left(x_1 \overline{x_2}\right)} = \overline{\left(\overline{x_1} / \overline{x_3}\right) / \left(x_1 / \overline{x_2}\right)}.$$

or

$$F_{\text{MNSF-2}} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix} = \\ = \begin{vmatrix} 1 & & 1 \\ 0 & 1 & \end{vmatrix} = \overline{\left(\overline{x_1} / \overline{x_3}\right) / \left(x_1 / \overline{x_2}\right)}. \tag{22}$$

The rules of gluing variables (30), (32) were used to search for $F_{\text{MNSF-2}}$.

Thus, when deriving $F_{\text{MNSF-2}}$ (22) containing more than one term, the entire derived expression $F_{\text{MNSF-2}}$ and the variables of the final binary matrix of the Sheffer function are inverted.

5. 2. Exceptional situations in simplification of the Sheffer functions on binary structures

When simplifying the Sheffer functions on binary structures, it is necessary to take into account the exceptional situations.

The first exceptional situation. If there are terms in the simplified Sheffer function which consist of one variable (that is the term has the maximum rank), then the variable describing the term is taken in the inverse code.

One literal is given in the Sheffer basis in the form:

$$x / x = \overline{x} \text{ or } \overline{x} / \overline{x} = x,$$

to which logic elements correspond (Fig. 3).

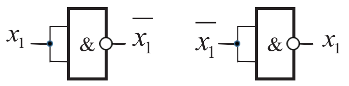


Fig. 3. Logic elements for one literal in the Sheffer basis

In the case when the final binary matrix of the minimal Sheffer function consists of several terms including a term (terms) consisting of one literal, then the literal must be inverted.

Example 4. Find the MNSF-1 for the function:

$$f\left(x_1, x_2, x_3, x_4\right) = \Sigma\left(0,4,5,6,7,8,12,13,14,15\right). \tag{23}$$

Σ in expression (23) defines minterms at which the function returns back «1» at the output.

An expanded normal form of the ENSF-1 for the $f(x_1, x_2, x_3, x_4)$ function (23) is as follows:

$$F_{\text{ENSF-1}} = \left(\overline{x_1} / \overline{x_2} / \overline{x_3} / \overline{x_4}\right) / \left(\overline{x_1} / x_2 / \overline{x_3} / \overline{x_4}\right) / \\ / \left(\overline{x_1} / x_2 / \overline{x_3} / x_4\right) / \left(\overline{x_1} / x_2 / x_3 / \overline{x_4}\right) / \\ / \left(\overline{x_1} / x_2 / x_3 / x_4\right) / \left(x_1 / \overline{x_2} / \overline{x_3} / \overline{x_4}\right) / \\ / \left(x_1 / x_2 / \overline{x_3} / \overline{x_4}\right) / \left(x_1 / x_2 / \overline{x_3} / x_4\right) / \\ / \left(x_1 / x_2 / x_3 / \overline{x_4}\right) / \left(x_1 / x_2 / x_3 / x_4\right),$$

$$F_{\text{MNSF-1}}=\left|\begin{array}{c|cccc}0&0&0&0&0\\4&0&1&0&0\\5&0&1&0&1\\6&0&1&1&0\\7&0&1&1&1\\8&1&0&0&0\\12&1&1&0&0\\13&1&1&0&1\\14&1&1&1&0\\15&1&1&1&1\end{array}\right|=\left|\begin{array}{cccc} &0&0&0\\ &1& & \end{array}\right|=$$

$$=\left|\begin{array}{cccc} & &0&0\\ &1& & \end{array}\right|=\left(\overline{x_3}/\overline{x_4}\right)/\overline{x_2}.$$

In the submatrix of blocks 4–7 and 12–15 (highlighted in red) which contains the combinatorial system 2-(3, 8)-design, the operation of supergluing of variables was used [20]. The simple gluing of variables is highlighted in black. Since the final binary matrix of the minimal Sheffer function contains a term with one literal, the latter is inverted when deriving $F_{\text{MNSF-1}}$. As a result, the minimum normal form of the Sheffer function will be obtained:

$$F_{\text{MNSF-1}}=\left(\overline{x_3}/\overline{x_4}\right)/\overline{x_2}.$$

A similar result can be obtained in the Sheffer algebra after the semi-gluing operation:

$$\overline{x_2}/\left(\overline{x_2}/\overline{x_3}/\overline{x_4}\right)=\overline{x_2}/\overline{\overline{x_2}\,\overline{x_3}\,\overline{x_4}}=$$

$$=\overline{x_2}/x_2+x_3+x_4=\overline{\overline{x_2}\left(x_2+x_3+x_4\right)}=$$

$$=x_2+\overline{\left(x_2+x_3+x_4\right)}=x_2+\overline{x_2}\,\overline{x_3}\,\overline{x_4}=$$

$$=x_2+\overline{x_3}\,\overline{x_4}=x_2+\overline{\overline{x_3}/\overline{x_4}}=$$

$$=\overline{\overline{x_2}\cdot\left(\overline{x_3}/\overline{x_4}\right)}=\overline{x_2}/\left(\overline{x_3}/\overline{x_4}\right).$$

Example 5. Find the MNSF-1 for the function:

$$f\left(x_1,x_2,x_3,x_4\right)=\Sigma\left(0,4,5,6,7,8,12,13,14,15\right). \tag{24}$$

Solution:

$$F_{\text{MNSF-1}}=\left|\begin{array}{c|cccc}0&0&0&0&0\\4&0&0&0&1\\5&0&0&1&0\\6&0&0&1&1\\7&0&1&0&0\\8&1&0&0&0\\12&1&0&0&1\\13&1&0&1&0\\14&1&0&1&1\\15&1&1&0&0\end{array}\right|=$$

$$=\left|\begin{array}{cccc} &1&0&0\\ &0& & \end{array}\right|=\left|\begin{array}{cccc} & &0&0\\ &0& & \end{array}\right|=\left(\overline{x_3}/\overline{x_4}\right)/x_2.$$

A similar result can be obtained in the Sheffer algebra after operation of semi-gluing:

$$x_2/\left(x_2/\overline{x_3}/\overline{x_4}\right)=x_2/\overline{x_2\overline{x_3}\,\overline{x_4}}=$$

$$=x_2/\overline{x_2}+x_3+x_4=\overline{x_2\left(\overline{x_2}+x_3+x_4\right)}=$$

$$=\overline{x_2}+\overline{\left(\overline{x_2}+x_3+x_4\right)}=\overline{x_2}+x_2\overline{x_3}\,\overline{x_4}=\overline{x_2}+\overline{x_3}\,\overline{x_4}=$$

$$=\overline{x_2\cdot\left(\overline{\overline{x_3}}\,\overline{\overline{x_4}}\right)}=x_2/\left(\overline{x_3}/\overline{x_4}\right).$$

Example 6. Simplify the function set in the ENSF-1.

$$F_{\text{ENSF-1}}=\left(\overline{x_1}/\overline{x_2}/\overline{x_3}\right)/\left(\overline{x_1}/\overline{x_2}/x_3\right)/\left(\overline{x_1}/x_2/x_3\right)/$$
$$/\left(x_1/\overline{x_2}/\overline{x_3}\right)/\left(x_1/\overline{x_2}/x_3\right)/\left(x_1/x_2/x_3\right). \tag{25}$$

Proof:

$$F_{\text{MNSF-1}}=\left|\begin{array}{ccc}0&0&0\\0&0&1\\0&1&1\\1&0&0\\1&0&1\\1&1&1\end{array}\right|=\left|\begin{array}{ccc} &0& \\ &1&1\end{array}\right|=$$

$$=\left|\begin{array}{ccc} &0& \\ & &1\end{array}\right|=x_2/\overline{x_3}. \tag{26}$$

The two terms in the final binary matrix of the Sheffer function (26) contain one variable each which are inverted during derivation of $F_{\text{MNSF-1}}$.

The verification of the obtained MNSF-1 (26) is presented in Table 7.

Table 7

Verification of the MNFS-1: $x_2/\overline{x_3}$

No.	x_1	x_2	x_3	$F_{\text{ENSF-1}}$	$x_2/\overline{x_3}$	F_{MNSF}
0	0	0	0	1	$0/\bar{0}$	1
1	0	0	1	1	$0/\bar{1}$	1
3	0	1	1	1	$1/\bar{1}$	1
4	1	0	0	1	$0/\bar{0}$	1
5	1	0	1	1	$0/\bar{1}$	1
7	1	1	1	1	$1/\bar{1}$	1
2	0	1	0	0	$1/\bar{0}$	0
6	1	1	0	0	$1/\bar{0}$	0

Taking into account Table 7, it can be seen that the MNSF-1, $x_2/\bar{x}_3$, satisfies the set logical function (25).

The second exceptional situation. If the result of simplification of the Sheffer function is only one Sheffer term containing several literals, then the general inversion over all literals is taken (Fig. 4).

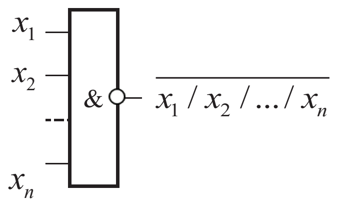


Fig. 4. Logic element n-NAND in the Sheffer basis

Example 7. Simplify the function set in the ENSF-1.

$$F_{\text{ENSF-1}}=\left(\overline{x_1}/\overline{x_2}/\overline{x_3}\right)/\left(\overline{x_1}/\overline{x_2}/x_3\right).$$

Proof:

$$F_{\text{MNSF-1}}=\left|\begin{array}{ccc}0&0&0\\0&0&1\end{array}\right|=\left|\begin{array}{ccc}0&0&\;\end{array}\right|=\overline{\overline{x_2}/\overline{x_2}}.$$

Example 8. Simplify the FNSF-2 for the function from example 6.

$$F_{\text{ESNF-2}}=\left(x_1/\overline{x_2}/x_3\right)/\left(\overline{x_1}/\overline{x_2}/x_3\right).$$

Proof:

$$F_{\text{MNSF-2}}=\left|\begin{array}{ccc}1&0&1\\0&0&1\end{array}\right|=\left|\begin{array}{ccc}\;&0&1\end{array}\right|=$$

$$=\overline{x_2}+x_3=\overline{x_2\overline{x_3}}=x_2/\overline{x_3}$$

or

$$F_{\text{MNSF-2}}=\left|\begin{array}{ccc}1&0&1\\0&0&1\end{array}\right|=\left|\begin{array}{ccc}\;&0&1\end{array}\right|=x_2/\overline{x_3}. \qquad (27)$$

Thus, when deriving $F_{\text{MNSF-2}}$ (27) containing one term, variables of the final binary matrix of the Sheffer function are inverted and the term itself is not inverted. Note that the expressions $F_{\text{MNSF-1}}$ (26) and $F_{\text{MNSF-2}}$ (27) coincide.

Example 9. Minimize the $f(x_1, x_2, x_3, x_4)$ logical function given in the truth table (Table 8):

Table 8

The truth table of the $f(x_1, x_2, x_3, x_4)$ function

No.	x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
2	0	0	1	0	0
6	0	1	1	0	0
7	0	1	1	1	0
9	1	0	0	1	0
11	1	0	1	1	0

Proof:

$$F_{\text{MNSF-2}}=\left|\begin{array}{c|cccc}2&1&1&0&1\\6&1&0&0&1\\7&1&0&0&0\\9&0&1&1&0\\11&0&1&0&0\end{array}\right|=$$

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

$$=\left(x_1+\overline{x_3}+x_4\right)\left(x_1+\overline{x_2}+\overline{x_3}\right)\left(\overline{x_1}+x_2+\overline{x_4}\right)=$$

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

$$=\overline{\left(\overline{x_1}/x_3/\overline{x_4}\right)/\left(\overline{x_1}/x_2/x_3\right)/\left(x_1/\overline{x_2}/x_4\right)}.$$

The MNSF-2 of the function given in the truth table (Table 8):

$$F_{\text{MNSF-2}}=\overline{\left(\overline{x_1}/x_3/\overline{x_4}\right)/\left(\overline{x_1}/x_2/x_3\right)/\left(x_1/\overline{x_2}/x_4\right)}. \qquad (28)$$

Let us recall that when deriving $F_{\text{MNSF-2}}$ which contains more than one term, the entire derived expression $F_{\text{MNSF-2}}$ and variables of the final Sheffer binary matrix are inverted.

The verification of the MNSF-2 (28) is presented in Table 9.

Examination of Table 9 shows that MNSF-2:

$$\overline{\left(\overline{x_1}/x_3/\overline{x_4}\right)/\left(\overline{x_1}/x_2/x_3\right)/\left(x_1/\overline{x_2}/x_4\right)}$$

satisfies the logical function given in Table 8.

The third exceptional situation. If simplification of the function results in only one term containing only one literal, then the MNSF will look like:

$$f_{\text{MNSF}}=\overline{\overline{x_n}}=x_n.$$

Thus, the result of simplifying the Sheffer function does not change in the third exceptional situation.

In the Sheffer basis, the logic element corresponds to the literal with a double inversion (Fig. 5).

Table 9

Verification of MNSF-2: $\overline{\left(\overline{x_1}/x_3/\overline{x_4}\right)/\left(\overline{x_1}/x_2/x_3\right)/\left(x_1/\overline{x_2}/x_4\right)}$

No.	x_1	x_2	x_3	x_4	$F_{\text{ENSF-2}}$	$\overline{\left(\overline{x_1}/x_3/\overline{x_4}\right)/\left(\overline{x_1}/x_2/x_3\right)/\left(x_1/\overline{x_2}/x_4\right)}$	$F_{\text{MNSF-2}}$
2	0	0	1	0	0	$\overline{\left(\bar{0}/1/\bar{0}\right)/\left(\bar{0}/0/1\right)/\left(0/\bar{0}/0\right)}$	0
6	0	1	1	0	0	$\overline{\left(\bar{0}/1/\bar{0}\right)/\left(\bar{0}/1/1\right)/\left(0/\bar{1}/0\right)}$	0
7	0	1	1	1	0	$\overline{\left(\bar{0}/1/\bar{1}\right)/\left(\bar{0}/1/1\right)/\left(0/\bar{1}/1\right)}$	0
9	1	0	0	1	0	$\overline{\left(\bar{1}/0/\bar{1}\right)/\left(\bar{1}/0/0\right)/\left(1/\bar{0}/1\right)}$	0
11	1	0	1	1	0	$\overline{\left(\bar{1}/1/\bar{1}\right)/\left(\bar{1}/0/1\right)/\left(1/\bar{0}/1\right)}$	0

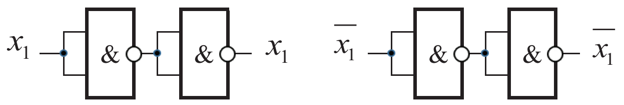


Fig. 5. Logic elements for a literal with a double inversion in the Sheffer basis

Rules (30), (38), (45) can serve as an example of simplifying the Sheffer functions in the event of the occurrence of the third exceptional situation.

5. 3. Equipotential transformations of the Boolean functions in the Sheffer basis

In a general case, when minimizing the Boolean functions in the Sheffer basis by the method of visual transformations, the following rules of the logical algebra are possible.

Variables of two-digit terms of ENSF-1 can be glued by means of the transformation:

$$\left(x_1 / x_2\right) / \left(\overline{x}_1 / x_2\right) = x_2. \qquad (29)$$

Proof:

$$\left(x_1 / x_2\right) / \left(\overline{x_1} / x_2\right) = \overline{\left(x_1 x_2\right)} / \overline{\left(\overline{x_1} x_2\right)} =$$

$$= \left(\overline{x_1} + \overline{x_2}\right) / \left(x_1 + \overline{x_2}\right) = \overline{\left(\overline{x_1} + \overline{x_2}\right) \cdot \left(x_1 + \overline{x_2}\right)} =$$

$$= \overline{\overline{\overline{x_1}\,\overline{x_2} + x_1 \overline{x_2} + \overline{x_2}}} = \overline{\overline{x_2}\left(\overline{x_1} + x_1 + 1\right)} = \overline{\overline{x_2}} = x_2.$$

Equipotential transformations for the rule of gluing variables of the two-digit ENSF terms (29) have an illustration of combinatorial image (30):

$$\begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix} = \left| \quad 1 \right| = x_2. \qquad (30)$$

The third exceptional situation was taken into account when deriving the result of gluing variables of the two-digit terms.

The variables in the three-digit terms of the ENSF-1 can be glued by means of the transformation:

$$\left(x_1 / x_2 / x_3\right) / \left(x_1 / x_2 / \overline{x_3}\right) = \overline{x_1 / x_2}. \qquad (31)$$

Proof:

$$\left(x_1 / x_2 / x_3\right) / \left(x_1 / x_2 / \overline{x_3}\right) = \overline{x_1 x_2 x_3} / \overline{x_1 x_2 \overline{x_3}} =$$

$$= \left(\overline{x_1} + \overline{x_2} + \overline{x_3}\right) / \left(\overline{x_1} + \overline{x_2} + x_3\right) =$$

$$= \overline{\left(\overline{x_1} + \overline{x_2} + \overline{x_3}\right) \cdot \left(\overline{x_1} + \overline{x_2} + x_3\right)} =$$

$$= \overline{\left(\overline{x_1} + \overline{x_2} + \overline{x_3}\right)} + \overline{\left(\overline{x_1} + \overline{x_2} + x_3\right)} =$$

$$= x_1 x_2 x_3 + x_1 x_2 \overline{x_3} = x_1 x_2 \left(x_3 + \overline{x_3}\right) =$$

$$= x_1 x_2 = \overline{x_1 / x_2}.$$

Equipotential transformations for the rule of gluing variables of three-digit terms of the ENSF-1 (31) have an illustration of image (32):

$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \left| 1 \quad 1 \quad \right| = \overline{x_1 / x_2}. \qquad (32)$$

The second exceptional situation was taken into account when deriving the result of gluing variables of the three-digit terms of ENSF-1.

Bonding of variables of three-digit ENSF-2 terms

Let us recall that to obtain the ENSF-2, variables in each Sheffer term are combined by the «Sheffer stroke» operation which depends on them. If the variable for the given term takes a zero value, it is taken in the direct code, otherwise in the inverse code [21].

For example, binary sets of variables:

$$\left(0_1 / 0_2 / 0_3\right) / \left(0_1 / 0_2 / 1_3\right) \qquad (33)$$

correspond to the three-digit ENSF-2 terms:

$$\left(x_1 / x_2 / x_3\right) / \left(x_1 / x_2 / \overline{x_3}\right). \qquad (34)$$

Before performing the operation of gluing variables for the ENSF-2, values of binary variables in the terms are inverted.

$$\begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \left| 1 \quad 1 \quad \right| =$$

$$= \left(x_1 + x_2\right) = \overline{\overline{x_1}}\,\overline{\overline{x_2}} = \overline{x_1} / \overline{x_2}$$

or

$$\begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \left| 1 \quad 1 \quad \right| = \overline{x_1} / \overline{x_2}. \qquad (35)$$

The verification of the result of gluing variables for the ENSF-2 (35) is presented in Table 10.

Table 10

Verification of the expression $\overline{x_1} / \overline{x_2}$

x_1	x_2	x_3	ENSF-2 value	$\overline{x_1} / \overline{x_2}$	Verification
0	0	0	0	$\overline{0} / \overline{0}$	0
0	0	1	0	$\overline{0} / \overline{0}$	0

It is seen from Table 10 that the result of gluing variables $\overline{x}_1 / \overline{x}_2$ satisfies the given terms of the ENSF-2 (33).

Note that the expressions in the results of gluing the variables of the three-digit terms of the ENSF-1 (32) and ENSF-2 (35) differ. This feature applies also to MNSF-1 and MNSF-2 with a larger number of terms which makes it possible to choose the optimal structure of the minimum function.

Variables of four-digit terms of the ENSF-1 can be glued by means of the transformation:

$$\left(x_1 / x_2 / x_3 / x_4\right) / \left(\overline{x_1} / x_2 / x_3 / x_4\right) = \overline{x_2 / x_3 / x_4}. \qquad (36)$$

Proof:

$$\left(x_1 / x_2 / x_3 / x_4\right) / \left(\overline{x_1} / x_2 / x_3 / x_4\right) =$$

$$= \overline{\left(x_1 x_2 x_3 x_4\right)} / \overline{\left(\overline{x_1} x_2 x_3 x_4\right)} =$$

$$= \left(\overline{x_1} + \overline{x_2} + \overline{x_3} + \overline{x_4}\right) / \left(x_1 + \overline{x_2} + \overline{x_3} + \overline{x_4}\right) =$$

$$= \overline{\left(\overline{x_1} + \overline{x_2} + \overline{x_3} + \overline{x_4}\right) \cdot \left(x_1 + \overline{x_2} + \overline{x_3} + \overline{x_4}\right)} =$$

$$= \overline{\overline{x_2} + \overline{x_3} + \overline{x_4}} = x_2 x_3 x_4 = \overline{x_2 / x_3 / x_4}.$$

Equipotential transformations for the rule of gluing variables of the four-digit ENSF-1 terms (36) have an illustration of image (37):

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} & 1 & 1 & 1 \end{vmatrix} = \overline{x_2 / x_3 / x_4}. \qquad (37)$$

The rule of supergluing variables

Combinatorial properties of the block diagram with repetition provide the rule of supergluing the variables [20] in the Sheffer basis.

Example 10. The rule of using super-gluing the variables for three-digit terms of the ENSF-1 can be presented as follows:

$$\begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{vmatrix} = x_2. \qquad (38)$$

Proof:

$$(x_1/x_2/x_3)/(\overline{x_1}/x_2/x_3)/(x_1/x_2/\overline{x_3})/(\overline{x_1}/x_2/\overline{x_3})=$$
$$=\overline{x_1x_2x_3}/\overline{\overline{x_1}x_2x_3}/\overline{x_1x_2\overline{x_3}}/\overline{\overline{x_1}x_2\overline{x_3}}=$$
$$=(\overline{x_1}+\overline{x_2}+\overline{x_3})/(x_1+\overline{x_2}+\overline{x_3})/(\overline{x_1}+\overline{x_2}+x_3)/(x_1+\overline{x_2}+x_3)=$$
$$=\overline{(\overline{x_1}+\overline{x_2}+\overline{x_3})\cdot(x_1+\overline{x_2}+\overline{x_3})\cdot(\overline{x_1}+\overline{x_2}+x_3)\cdot(x_1+\overline{x_2}+x_3)}=$$
$$=\overline{(\overline{x_1}+\overline{x_2}+\overline{x_3})}+\overline{(x_1+\overline{x_2}+\overline{x_3})}+\overline{(\overline{x_1}+\overline{x_2}+x_3)}+\overline{(x_1+\overline{x_2}+x_3)}=$$
$$=x_1x_2x_3+\overline{x_1}x_2x_3+x_1x_2\overline{x_3}+\overline{x_1}x_2\overline{x_3}=$$
$$=x_2(x_1x_3+\overline{x_1}x_3+x_1\overline{x_3}+\overline{x_1}\,\overline{x_3})=$$
$$=x_2(x_3(x_1+\overline{x_1})+\overline{x_3}(x_1+\overline{x_1}))=x_2(x_3+\overline{x_3})=x_2.$$

The 2-(2, 4)-design [19] is used in the rule (38).

Example 11. For example, the rule of supergluing variables for four-digit terms of the ENSF-1 can be as follows:

$$\begin{vmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{vmatrix} = \overline{x_1/\overline{x_3}}. \qquad (39)$$

Proof:

$$(x_1/x_2/\overline{x_3}/x_4)/(x_1/x_2/\overline{x_3}/\overline{x_4})\downarrow(x_1/\overline{x_2}/\overline{x_3}/x_4)\downarrow(x_1/\overline{x_2}/\overline{x_3}/\overline{x_4})=$$
$$=\overline{x_1x_2\overline{x_3}x_4}/\overline{x_1x_2\overline{x_3}\,\overline{x_4}}/\overline{x_1\overline{x_2}\,\overline{x_3}x_4}/\overline{x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}}=$$
$$=(\overline{x_1}+\overline{x_2}+x_3+\overline{x_4})/(\overline{x_1}+\overline{x_2}+x_3+x_4)/(\overline{x_1}+x_2+x_3+\overline{x_4})/(\overline{x_1}+x_2+x_3+x_4)=$$
$$=\overline{(\overline{x_1}+\overline{x_2}+x_3+\overline{x_4})\cdot(\overline{x_1}+\overline{x_2}+x_3+x_4)\cdot(\overline{x_1}+x_2+x_3+\overline{x_4})\cdot(\overline{x_1}+x_2+x_3+x_4)}=$$
$$=\overline{(\overline{x_1}+\overline{x_2}+x_3+\overline{x_4})}+\overline{(\overline{x_1}+\overline{x_2}+x_3+x_4)}+\overline{(\overline{x_1}+x_2+x_3+\overline{x_4})}+\overline{(\overline{x_1}+x_2+x_3+x_4)}=$$
$$=(x_1x_2\overline{x_3}x_4)+(x_1x_2\overline{x_3}\,\overline{x_4})+(x_1\overline{x_2}\,\overline{x_3}x_4)+(x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4})=$$
$$=x_1\overline{x_3}(x_2x_4+x_2\overline{x_4}+\overline{x_2}x_4+\overline{x_2}\,\overline{x_4})=x_1\overline{x_3}(x_2(x_4+\overline{x_4})+\overline{x_2}(x_4+\overline{x_4}))=$$
$$=x_1\overline{x_3}(x_2+\overline{x_2})=x_1\cdot\overline{x_3}=\overline{x_1/\overline{x_3}}.$$

The 2-(2, 4)-design is used in the rule (39). Variables x_1, x_3 can occupy any digit (position) in the ENSF-1 term.

The rule of incomplete supergluing variables

Combinatorial properties of the incomplete combinatorial system with repetition of the 2-(n, x/b)-design [20] provide the rule of incomplete supergluing variables in the Sheffer basis.

Example 12. For example, the rule of incomplete supergluing variables for two-digit and three-digit terms of the ENSF-1 can be as follows:

$$\begin{vmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} & 0 \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} & 0 \\ 0 & \end{vmatrix} = x_1/x_2. \qquad (40)$$

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

$$= \begin{vmatrix} 0 & & \\ 1 & 0 & \\ 1 & & 0 \end{vmatrix} = \begin{vmatrix} 0 & & \\ & 0 & \\ & & 0 \end{vmatrix} = x_1/x_2/x_3. \qquad (41)$$

Proof of the result of (40):

$$(\overline{x_1}/\overline{x_2})/(x_1/\overline{x_2})/(\overline{x_1}/x_2)=$$
$$=\overline{\overline{x_1}\,\overline{x_2}}/\overline{x_1\overline{x_2}}/\overline{\overline{x_1}x_2}=$$
$$=(x_1+x_2)/(\overline{x_1}+x_2)/(x_1+\overline{x_2})=$$
$$=\overline{(x_1+x_2)\cdot(\overline{x_1}+x_2)\cdot(x_1+\overline{x_2})}=$$
$$=(\overline{x_1+x_2})+\overline{(\overline{x_1}+x_2)}+\overline{(x_1+\overline{x_2})}=$$
$$=\overline{x_1}\,\overline{x_2}+x_1\overline{x_2}+\overline{x_1}x_2=\overline{x_2}(\overline{x_1}+x_1)+\overline{x_1}x_2=$$
$$=\overline{x_2}+\overline{x_1}x_2=\overline{x_1}+\overline{x_2}=x_1/x_2.$$

The 2-(2, 3/4)-design [20] is used in the rule (40). When deriving the result of the operation of incomplete supergluing of variables of two-digit ENSF-1 terms (40), the first exceptional situation was taken into account.

The result of (41) is similarly proved. The 2-(3, 7/8)-design [20] is used in the rule (41). When deriving the result of the operation of incomplete supergluing of variables of three-digit ENSF-1 terms (41), the first exceptional situation was taken into account.

Generalized gluing of variables can be done by means of the transformation:

$$(x_1/x_2)/(x_1/x_3)/(x_2/\overline{x_3})=$$
$$=(x_1/x_3)/(x_2/\overline{x_3}). \qquad (42)$$

Proof:

$$(x_1/x_2)/(x_1/x_3)/(x_2/\overline{x_3})=$$
$$=\overline{(x_1/x_2/x_3)/(x_1/x_2/\overline{x_3})}/(x_1/x_3)/(x_2/\overline{x_3})=$$
$$=\overline{\overline{x_1x_2x_3}/\overline{x_1x_2\overline{\overline{x_3}}}}/\overline{x_1x_3}/\overline{x_2\overline{\overline{x_3}}}=$$
$$=\overline{(\overline{x_1}+\overline{x_2}+\overline{x_3})/(\overline{x_1}+\overline{x_2}+x_3)}/\overline{x_1}+\overline{x_3}/\overline{x_2}+x_3=$$
$$=\overline{\overline{(\overline{x_1}+\overline{x_2}+\overline{x_3})\cdot(\overline{x_1}+\overline{x_2}+x_3)}}/\overline{x_1}+\overline{x_3}/\overline{x_2}+x_3=$$
$$=\overline{\overline{(\overline{x_1}+\overline{x_2}+\overline{x_3})\cdot(\overline{x_1}+\overline{x_2}+x_3)}}/\overline{x_1}+\overline{x_3}/\overline{x_2}+x_3=$$
$$=\overline{(x_1x_2x_3+x_1x_2\overline{x_3})}/\overline{x_1}+\overline{x_3}/\overline{x_2}+x_3=$$
$$=\overline{(x_1x_2x_3+x_1x_2\overline{x_3})}/\overline{(\overline{x_1}+\overline{x_3})\cdot(\overline{x_2}+x_3)}=$$
$$=\overline{\left(\overline{(x_1x_2x_3+x_1x_2\overline{x_3})}\right)\cdot\left(\overline{(\overline{x_1}+\overline{x_3})\cdot(\overline{x_2}+x_3)}\right)}=$$
$$=(x_1x_2x_3+x_1x_2\overline{x_3})+\left((\overline{x_1}+\overline{x_3})\cdot(\overline{x_2}+x_3)\right)=$$
$$=x_1x_2x_3+x_1x_2\overline{x_3}+x_1x_3+x_2\overline{x_3}=$$
$$=\left|\begin{array}{ccc} x_1 & x_2 & x_3 \\ x_1 & x_2 & \overline{x_3} \\ x_1 & & x_3 \\ & x_2 & \overline{x_3} \end{array}\right|=\left|\begin{array}{ccc} x_1 & & x_3 \\ & x_2 & \overline{x_3} \end{array}\right|=$$
$$=x_1x_3+x_2\overline{x_3}=\overline{x_1x_3}\cdot\overline{x_2\overline{x_3}}=(x_1/x_3)/(x_2/\overline{x_3}).$$

Thus, $(x_1/x_2)/(x_1/x_3)/(x_2/\overline{x_3})=(x_1/x_3)/(x_2/\overline{x_3})$, what was necessary to prove.

Equipotential transformations for the rule of generalized gluing of variables (42) in the Sheffer basis have an illustration of image (43):

$$\left|\begin{array}{ccc} 1 & 1 & \\ 1 & & 1 \\ & 1 & 0 \end{array}\right|=\left|\begin{array}{ccc} 1 & & 1 \\ & 1 & 0 \end{array}\right|=(x_1/x_3)/(x_2/\overline{x_3}). \tag{43}$$

Another variant of the rule of generalized gluing of variables for the ENSF-1:

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

The rule of absorption of variables is reduced to the transformations:

1. $x_1/(\overline{x_1}/x_2)=\overline{x_1}.$ (44)

Proof:

$$x_1/(\overline{x_1}/x_2)=x_1/\overline{\overline{x_1}x_2}=$$
$$=x_1/(x_1+\overline{x_2})=\overline{x_1\cdot(x_1+\overline{x_2})}=\overline{x_1}.$$

Since the left-hand side of expression (44) is the initial Sheffer function, then, to introduce it in the matrix, one variable representing the term must be inverted (the first exceptional situation):

$$x_1/(\overline{x_1}/x_2)=\left|\begin{array}{cc} 0 & \\ 0 & 1 \end{array}\right|=\left|\begin{array}{cc} 0 & \end{array}\right|=\overline{x_1}. \tag{45}$$

When deriving the result (45), the third exceptional situation was applied (final matrix of the Sheffer function contains only one term with one variable: the result of function simplification does not change).

2. $\overline{x_1}/(x_1/x_2)=x_1.$

Proof:

$$\overline{x_1}/(x_1/x_2)=\overline{x_1}/\overline{x_1x_2}=$$
$$=\overline{x_1}/(\overline{x_1}+\overline{x_2})=\overline{\overline{x_1}\cdot(\overline{x_1}+\overline{x_2})}=\overline{\overline{x_1}}=x_1.$$

3. $(x_1/x_2)/(x_1/x_2/x_3)=\overline{x_1/x_2}.$

Proof:

$$(x_1/x_2)/(x_1/x_2/x_3)=\overline{x_1x_2}\downarrow\overline{x_1x_2x_3}=$$
$$=(\overline{x_1}+\overline{x_2})\downarrow(\overline{x_1}+\overline{x_2}+\overline{x_3})=$$
$$=\overline{(\overline{x_1}+\overline{x_2})\cdot(\overline{x_1}+\overline{x_2}+\overline{x_3})}=$$
$$=\overline{(\overline{x_1}+\overline{x_2})}+\overline{(\overline{x_1}+\overline{x_2}+\overline{x_3})}=$$
$$=x_1x_2+x_1x_2x_3=x_1x_2(1+x_3)=x_1x_2=\overline{x_1/x_2}.$$
$$\left|\begin{array}{ccc} 1 & 1 & \\ 1 & 1 & 1 \end{array}\right|=\left|\begin{array}{ccc} 1 & 1 & \end{array}\right|=\overline{x_1/x_2}.$$

4. $(x_1/x_2)/(x_1/x_2/x_3/x_4)=\overline{x_1/x_2}.$

Proof:

$$(x_1/x_2)/(x_1/x_2/x_3/x_4)=\overline{x_1x_2}/\overline{x_1x_2x_3x_4}=$$
$$=(\overline{x_1}+\overline{x_2})/(\overline{x_1}+\overline{x_2}+\overline{x_3}+\overline{x_4})=$$
$$=\overline{(\overline{x_1}+\overline{x_2})\cdot(\overline{x_1}+\overline{x_2}+\overline{x_3}+\overline{x_4})}=$$
$$=\overline{(\overline{x_1}+\overline{x_2})}+\overline{(\overline{x_1}+\overline{x_2}+\overline{x_3}+\overline{x_4})}=$$
$$=x_1x_2+x_1x_2x_3x_4=x_1x_2(1+x_3x_4)=x_1x_2=\overline{x_1/x_2}.$$
$$\left|\begin{array}{cccc} 1 & 1 & & \\ 1 & 1 & 1 & 1 \end{array}\right|=\left|\begin{array}{cccc} 1 & 1 & & \end{array}\right|=\overline{x_1/x_2}.$$

The rule of semi-gluing variables can be realized by means of the following transformations:

$$(\overline{x_1}/x_2)/(x_1/x_2/x_3)=\overline{x_2/(x_1/\overline{x_3})}, \tag{46}$$
$$(\overline{x_1}/x_2)/(x_1/x_2/x_3)=\overline{\overline{x_1}x_2}/\overline{(x_2x_3)}. \tag{47}$$

Proof of the result (46):

$$(\overline{x_1}/x_2)/(x_1/x_2/x_3)=\overline{\overline{x_1}x_2}/\overline{x_1x_2x_3}=$$
$$=(x_1+\overline{x_2})/(\overline{x_1}+\overline{x_2}+\overline{x_3})=\overline{(x_1+\overline{x_2})\cdot(\overline{x_1}+\overline{x_2}+\overline{x_3})}=$$
$$=\overline{(x_1+\overline{x_2})}+\overline{(\overline{x_1}+\overline{x_2}+\overline{x_3})}=\overline{x_1}x_2+x_1x_2x_3=x_2(\overline{x_1}+x_1x_3)=$$
$$=x_2(\overline{x_1}+x_3)=x_2\overline{(x_1\overline{x_3})}=\overline{x_2/(x_1/\overline{x_3})}.$$

The result (47) is similarly proved.

The rule of semi-gluing the variables (46) has an illustration of the image:

$$\left|\begin{array}{ccc}0 & 1 & \\ 1 & 1 & 1\end{array}\right| = \left|\begin{array}{ccc}0 & 1 & \\ & 1 & 1\end{array}\right| = \overline{x}_1 x_2 + x_2 x_3 =$$
$$= x_2\left(\overline{x}_1 + x_3\right) = x_2\overline{\left(x_1\overline{x}_3\right)} = \overline{x_2 / \left(x_1 / \overline{x}_3\right)}. \qquad (48)$$

The rule of semi-gluing the variables (47) has an illustration of the image:

$$\left|\begin{array}{ccc}0 & 1 & \\ 1 & 1 & 1\end{array}\right| = \left|\begin{array}{ccc}0 & 1 & \\ & 1 & 1\end{array}\right| = \overline{x}_1 x_2 + x_2 x_3 = \overline{\overline{x}_1 x_2} / \overline{\left(x_2 x_3\right)}.$$

The rule:

$$x_1 / \left(x_1 / x_2\right) = x_1 / \overline{x}_2$$

is proved by means of the following transformations:

$$x_1 / \left(x_1 / x_2\right) = x_1 / \overline{x_1 x_2} = x_1 / \left(\overline{x}_1 + \overline{x}_2\right) =$$
$$= \overline{x_1 \cdot \left(\overline{x}_1 + \overline{x}_2\right)} = \overline{x_1 \overline{x}_2} = x_1 / \overline{x}_2.$$

$$x_1 / \left(x_1 / x_2\right) = \left|\begin{array}{cc}0 & \\ 1 & 1\end{array}\right| = \left|\begin{array}{cc}0 & \\ & 1\end{array}\right| = x_1 / \overline{x}_2. \qquad (49)$$

When deriving the result (49), the first exceptional situation was applied.

Since the combinatorial structure of the truth tables of logical functions provides more information about orthogonality, contiguity, the unambiguity of the truth table blocks [22], the use of combinatorial images to find objects of equipotential transformation is effective when simplifying the Sheffer functions.

5. 4. Establishing sign of the minimum Sheffer function

A tag of the minimum Sheffer function is established by performing successive procedures to reduce the function in sets of the truth table at which the function returns «1» and «0» at the output. In the error-free reduction of a given function in these two procedures and the algebraic transformation of the second result of function reduction to the first result, such an algebraic transformation will coincide with the first result of the function reduction.

Example 13. Simplify the logical function $f(x_1, x_2, x_3)$ by image transformations given in the truth table (Table 11) and establish the minimum function tag.

Table 11

The truth table of the $f(x_1, x_2, x_3)$ function

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Proof.

The ENSF-1 of the given $f(x_1, x_2, x_3)$ function has the form:

$$F_{\text{ENSF-1}} = \left(\overline{x}_1 / x_2 / x_3\right) / \left(x_1 / x_2 / \overline{x}_3\right) / \left(x_1 / x_2 / x_3\right).$$

Simplification of ENSF-1 of the $f(x_1, x_2, x_3)$ function by equipotential image transformations:

$$F_{\text{MNSF-1}} = \left|\begin{array}{ccc}0 & 1 & 1\\ 1 & 1 & 0\\ 1 & 1 & 1\end{array}\right| = \left|\begin{array}{ccc}0 & 1 & 1\\ 1 & 1 & \end{array}\right| =$$
$$= \left|\begin{array}{ccc} & 1 & 1\\ 1 & 1 & \end{array}\right| = \left(x_1 / x_2\right) / \left(x_2 / x_3\right). \qquad (50)$$

The ENSF-2 of the given $f(x_1, x_2, x_3)$ function has the form:

$$F_{\text{ENSF-2}} = \left(x_1 / x_2 / x_3\right) / \left(x_1 / x_2 / \overline{x}_3\right) /$$
$$/ \left(x_1 / \overline{x}_2 / x_3\right) / \left(\overline{x}_1 / x_2 / x_3\right) / \left(\overline{x}_1 / x_2 / \overline{x}_3\right).$$

Simplification of the ENSF-2 of the $f(x_1, x_2, x_3)$ function by equipoentialt inage transformations:

$$F_{\text{MNSF-2}} = \left|\begin{array}{ccc}1 & 1 & 1\\ 1 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 1\\ 0 & 1 & 0\end{array}\right| = \left|\begin{array}{ccc} & 1 & \\ 1 & 0 & 1\end{array}\right| =$$
$$= \left|\begin{array}{ccc} & 1 & \\ 1 & & 1\end{array}\right| = \overline{\left(\overline{x}_1 / \overline{x}_3\right) / x_2}. \qquad (51)$$

Algebraic transformation of expression (51) to expression (50):

$$\overline{\left(\overline{x}_1 / \overline{x}_3\right) / x_2} = \overline{\overline{\left(\overline{x}_1 / \overline{x}_3\right) \cdot x_2}} = \left(\overline{x}_1 / \overline{x}_3\right) \cdot x_2 =$$
$$= \overline{\left(\overline{x}_1 \cdot \overline{x}_3\right)} \cdot x_2 = \left(x_1 + x_3\right) x_2 = x_1 x_2 + x_2 x_3 =$$
$$= \overline{\overline{\overline{\left(x_1 x_2\right)} \cdot \overline{\left(x_2 x_3\right)}}} = \overline{\left(x_1 x_2\right)} / \overline{\left(x_2 x_3\right)} =$$
$$= \left(x_1 / x_2\right) / \left(x_2 / x_3\right). \qquad (52)$$

Algebraic expressions (50) and (52) coincide which, on the basis of the minimum function, indicates obtaining the minimum normal form of the Sheffer function by the simplification procedure.

5. 5. Minimization of the Sheffer functions in the complete truth table

Example 14. Minimize the logical function $f(x_1, x_2, x_3, x_4)$ given in a canonical form [23] in the complete truth table by image transformations in two expanded normal forms ENSF-1 and ENSF-2:

$$F\left(x_1, x_2, x_3, x_4\right) = \Sigma\left(0, 1, 6, 8, 11, 14, 15\right). \qquad (53)$$

Σ in expression (53) determines minterms at which the $f(x_1, x_2, x_3, x_4)$ function returns «1» at the output.

Choose the minimum function based on the results of the simplification of two expanded normal forms ENSF-1 and ENSF-2.

Minimization of the ENSF-1 (53) is illustrated by image transformations:

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

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

The MNSF-1 of the $f(x_1, x_2, x_3, x_4)$ function:

$$F_{\text{MNSF-1}}=\left(\overline{x_1}/\overline{x_2}/\overline{x_3}\right)/\left(\overline{x_2}/\overline{x_3}/\overline{x_4}\right)/$$
$$/\left(x_2/x_3/\overline{x_4}\right)/\left(x_1/x_3/x_4\right). \quad (54)$$

Minimization of the ENSF-2 of the given function (53):

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

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

MNSF-2 of the $f(x_1, x_2, x_3, x_4)$ function:

$$F_{\text{MNSF-2}}=$$
$$=\overline{\left(\overline{x_2}/x_3/\overline{x_4}\right)/\left(\overline{x_1}/x_3/x_4\right)/\left(x_2/\overline{x_3}\right)/\left(x_1/\overline{x_3}/x_4\right)}. \quad (55)$$

The MNSF-2 (55) contains fewer literals compared to the MNSF-1 (54). Thus, at the same functionality of expressions (54) and (55) (Table 12), the latter corresponds to a simpler structure (Fig. 6, *a*).

It can be seen from Fig. 6 that implementation by means of the combinational circuit of the MNSF-2 (Fig. 6, *a*) is simpler because it contains a two-input logic element AND-NOT which is absent in the scheme that implements the MNSF-1 (Fig. 6, *b*).

Table 12 presents the functionality of the MNSF-2 and MNSF-1 for the function given in a canonical form (53).

It can be seen from Table 12 that the $F_{\text{MNSF-2}}$ and $F_{\text{MNSF-1}}$ have the same functionality, however, the $F_{\text{MNSF-2}}$ has one literal less.

According to the results of the minimization of two normal forms ENSF-1 and ENSF-2, choose MNSF-2 (55) as the minimum one.

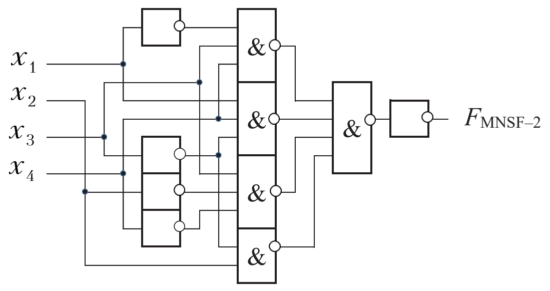
Table 12

The truth table of functions

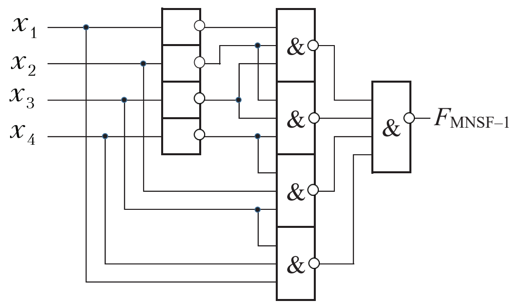
$$F_{\text{MNSF-2}}=\overline{\left(\overline{x_1}/x_3/x_4\right)/\left(x_1/\overline{x_3}/x_4\right)/\left(\overline{x_2}/x_3/\overline{x_4}\right)/\left(x_2/\overline{x_3}\right)};$$

$$F_{\text{MNSF-1}}=\left(\overline{x_1}/\overline{x_2}/\overline{x_3}\right)/\left(\overline{x_2}/\overline{x_3}/\overline{x_4}\right)/\left(x_2/x_3/\overline{x_4}\right)/\left(x_1/x_3/x_4\right)$$

No.	x_1	x_2	x_3	x_4	$F_{\text{MNSF-2}}$	$F_{\text{MNSF-1}}$
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.	x_1	x_2	x_3	x_4	$F_{\text{MNSF-2}}$	$F_{\text{MNSF-1}}$
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



a



b

Fig. 6. Implementation of the minimum logical $f(x_1, x_2, x_3, x_4)$ function by means of a combinational circuit: *a* – MNSF-2; *b* – MNSF-1

6. Discussion of the results obtained in minimization of the Sheffer functions by the method of image transformations

The mathematical apparatus of image transformations was considered in [10, 19, 20, 22, 24]. Hermeneutics of logic operations with binary structures provides sufficient didactics of simplification of the Boolean functions including the class of expanded normal forms of functions of the Sheffer algebra.

The main task of minimizing the ENSF-1 and ENSF-2 implies finding terms suitable for a particular algebraic operation. However, with an increasing number of variables of the algebraic expression, such a search can become quite difficult. When simplifying logical formulas, it is not always obvious which of the laws of logical algebra should be applied at a given step. In turn, image transformations, due to their inherent clarity and unification of original procedures, make it possible to solve this problem to some extent. In some cases, the apparatus of image transformations is the only way to continue the optimal simplification of the logical expression.

Binary structures with repetition which are actually the truth tables of given functions were the object of solving the problem of minimizing Boolean functions in the Sheffer basis by the method of image transformations. This has allowed us to do without auxiliary objects, such as Karnaugh map, Weich diagrams, acyclic graph, coverage tables, etc.

For example, improvement of the first stage of Quine's method proposed by McCluskey can be applied to the Sheffer basis.

Let us find the MNSF-1 of a 4-bit Boolean function by the Quine-McCluskey method for the sets of literals at which function (56) returns the value of 1.

$$F=\begin{Bmatrix}0000,\ 0010,\ 0100,\ 0110,\\ 1000,\ 1010,\ 1100,\ 1101,\ 1110\end{Bmatrix}. \qquad (56)$$

Break the sets of variables into groups depending on the number of ones in them and glue the variables in neighboring groups (Fig. 7).

Persistent (PNSF-1) of function (56) which is simultaneously the MNSF-1 (57) was obtained:

$$F_{\text{MNSF}-1}=x_4/\left(x_1/x_2/\overline{x_3}\right). \qquad (57)$$

Finding the MNSF-1 by the method of image transformations is reduced to the following procedure:

$$F_{\text{MNSF}-1}=\left|\begin{array}{r|cccc}0&0&0&0&0\\2&0&0&1&0\\4&0&1&0&0\\6&0&1&1&0\\8&1&0&0&0\\10&1&0&1&0\\12&1&1&0&0\\13&1&1&0&1\\14&1&1&1&0\end{array}\right|=$$

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

$$=x_4/\left(x_1/x_2/\overline{x_3}\right). \qquad (58)$$

The result of finding the MNSF-1 by two methods is the same, however, the method of image transformations is much simpler.

The considered method of simplification of logical expression in the ENSF-1 and ENSF-2 features the use of exceptional situations. They are effectively used both in deriving the simplification result from the binary matrix and introducing the Sheffer function into the matrix.

The visuality of image transformations makes it possible to apply a manual method of minimization of the Sheffer functions (using the mathematical editor, e.g. Math Type v. 7.0) approximately within ten input variables.

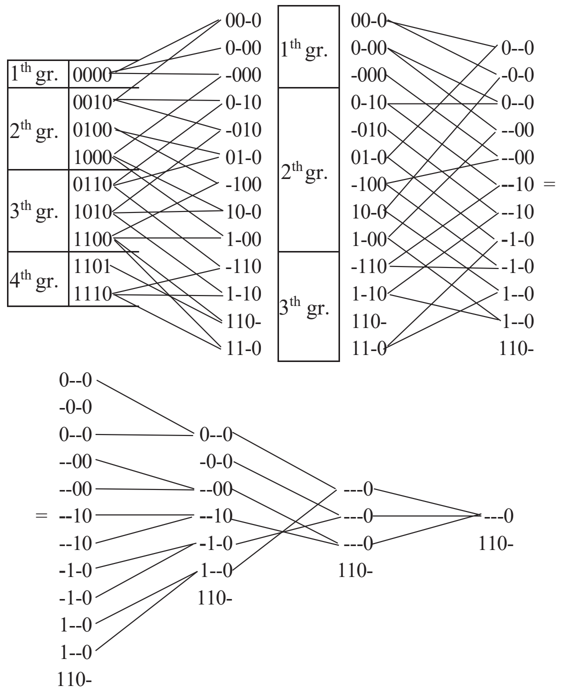


Fig. 7. Breaking sets of the function variables (56) into groups followed by procedures of gluing, absorption, and idempotency

Application of the method of image transformations to minimize functions of the Sheffer algebra brings the problem of simplification of the ENSF-1 and ENSF-2 to the level of a well-researched problem in the class of disjunctive-conjunctive normal forms (DCNF) of the Boolean functions.

Limitations of the method application include the cases when the switching function contains logical operations from several bases. In this case, the function must be represented by a single logical basis.

The weakness of this method is in a narrow practical application of equivalent image transformations to the process of minimizing the Sheffer functions with subsequent manufacture of appropriate computational components. Negative internal factors of the method are associated with additional time spent on the establishment of protocols for simplifying the Sheffer functions with the subsequent creation of a library of rules of logical algebra which illustrate the corresponding image transformations. Since image transformations are a universal apparatus for minimizing the Boolean functions, the prospect of further studies may consist, e.g. in the application of the method of minimizing the Boolean functions in the class of expanded normal forms of the Pierce-Webb algebra functions.

7. Conclusions

1. It was established that the minimization of the Boolean functions in the Sheffer basis by the method of image transformations is based on a block diagram with repetition. It is actually a truth table of the given function. This makes it possible to concentrate the principle of minimization within the truth table of the function and thus do without auxiliary objects, such as Karnaugh map, Weich diagrams, acyclic graph, cubic representation, etc.

The expanded normal form of the n-digit Sheffer function can be represented by binary sets (20) or a matrix (21) which in this case will represent terms of the Sheffer function and the «Sheffer stroke» operation for them. Such hermeneutics should be used effectively in the reduction of a logical function and deriving the result of logical operations in the class of binary matrices of the Sheffer functions.

2. When simplifying the Sheffer functions on binary structures, it is necessary to take into account exceptional situations. In total, three exceptional situations have been identified. They have an effective application both when deriving the minimization result from the binary matrix (45) and introducing the Sheffer function to the matrix (49).

3. For a proper minimization of the Sheffer functions by the method of image transformations, the rules of supergluing (38), (39), incomplete supergluing the variables (40), (41), generalized gluing the variables (43) were developed and absorption rules and semi-gluing the variables (48) were made more clear.

4. It was found that image transformations simplify the procedure of establishing the criterion of the minimum Sheffer function (example 13) which guarantees an optimal reduction of the number of the logical function variables without losing its functionality.

5. It was found that the best result of minimizing the Sheffer functions can be achieved in both ENSF-1 and ENSF-2 (example 14). It follows that minimization of a given function should be carried out in two expanded normal forms: ENSF-1 and ENSF-2 using a complete truth table. The optimal function should be chosen based on the results of minimizing two normal forms: ENSF-1 and ENSF-2.

References

1. Pucknell, D. A. (1990). Fundamentals of Digital Logic Design: With VLSI Circuit applications. Prentice Hall, 486.
2. Mano, M. M., Kime, C. (2003). Logic and Computer Design Fundamentals. Prentice Hall, 650.
3. Baranov, S. (2008). Logic and System Design of Digital Systems. Tallinn: TUT Press.
4. De Micheli, G. (1994). Synthesis and Optimization of Digital Circuits. McGraw-Hill, 597.
5. Zakrevskij, A., Pottosin, Yu., Cheremisinova, L. (2009). Optimization in Boolean Space. Tallinn: TUT Press. Available at: http://www.ester.ee/record=b2461762*est
6. Luba, T. (2000). Synteza układ'ow logicznych. Warszawa: WSISiZ.
7. Rawski, M., Łuba, T., Jachna, Z., Tomaszewicz, P. (2005). The Influence of Functional Decomposition on Modern Digital Design Process. Design of Embedded Control Systems, 193–204. doi: https://doi.org/10.1007/0-387-28327-7_17
8. Bibilo, N. (2009). Decomposition of Boolean Functions by Means of Solving Logical Equations. Minsk: Belaruskaya Navuka.
9. Borowik, G., Łabiak, G., Bukowiec, A. (2015). FSM-Based Logic Controller Synthesis in Programmable Devices with Embedded Memory Blocks. Topics in Intelligent Engineering and Informatics, 123–151. doi: https://doi.org/10.1007/978-3-319-12652-4_8
10. 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
11. Baranov, S., Karatkevich, A. (2018). On Transformation of a Logical Circuit to a Circuit with NAND and NOR Gates Only. INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 64 (3), 373–378. doi: http://doi.org/10.24425/123535
12. Maxfield, M. (2018). Implementing Logic Functions Using Only NAND or NOR Gates. Available at: https://www.eeweb.com/implementing-logic-functions-using-only-nand-or-nor-gates/
13. An algorithm to implement a boolean function using only NAND's or only NOR's. Available at: https://cnx.org/contents/vJcXn_C0@4.9:vLMEHoQ0@6/An-algorithm-to-implement-a-boolean-function-using-only-NAND-s-or-only-NOR-s
14. Kana, A. F. (2008). Implimenting logical circuit using NAND and NOR gate only. Digital Logic Design, 47–54. Available at: http://american.cs.ucdavis.edu/academic/ecs154a.sum14/postscript/cosc205.pdf
15. Shaik, E. haq, Rangaswamy, N. (2017). Realization of all-optical NAND and NOR logic functions with photonic crystal based NOT, OR and AND gates using De Morgan's theorem. Journal of Optics, 47 (1), 8–21. doi: https://doi.org/10.1007/s12596-017-0441-y
16. Rajaei, A., Houshmand, M., Rouhani, M. (2011). Optimization of Combinational Logic Circuits Using NAND Gates and Genetic Programming. Soft Computing in Industrial Applications, 405–414. doi: https://doi.org/10.1007/978-3-642-20505-7_36
17. Macia, J., Sole, R. (2014). How to Make a Synthetic Multicellular Computer. PLoS ONE, 9 (2), e81248. doi: https://doi.org/10.1371/journal.pone.0081248
18. Dychka, I. A., Tarasenko, V. P., Onai, M. V. (2019). Osnovy prykladnoi teoriyi tsyfrovykh avtomativ. Kyiv: KPI im. Ihoria Sikorskoho, 508.
19. Riznyk, V., Solomko, M. (2018). Minimization of conjunctive normal forms of boolean functions by combinatorial method. Technology Audit and Production Reserves, 5 (2 (43)), 42–55. doi: https://doi.org/10.15587/2312-8372.2018.146312
20. 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
21. Havrylenko, S. Yu., Klymenko, A. M., Noskov, V. I. (2014). Lohika dyskretnykh avtomativ. Kharkiv, 129. Available at: http://web.kpi.kharkov.ua/otp/wp-content/uploads/sites/152/2016/05/Kompyuterna_logika_2sem_praktikum.pdf
22. Riznyk, V., Solomko, M. (2017). Minimization of Boolean functions by combinatorial method. Technology Audit and Production Reserves, 4 (2 (36)), 49–64. doi: https://doi.org/10.15587/2312-8372.2017.108532
23. Rytsar, B. Ye. (2015). New minimization method of logical functions in polynomial set-theoretical format. 1. Generalized rules of conjuncterms simplification. Upravlyayushchie sistemy i mashiny, 2, 39–57.
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