DOI: <https://doi.org/10.15407/emodel.43.01.028>
UDC 004.052.32+681.518.5

D.V. Efanov, Doctor of Science (Tech.)

Federal State Autonomous Educational Institution of Higher Education
"Russian University of Transport"

Russian Federation, 127994, Moscow, Obraztsova str., build. 9/9,
contact phone number (+7) (911) 709 21 64, e-mail: TrES-4b@yandex.ru

V.V. Sapozhnikov, Doctor of Science (Tech.),

VI.V. Sapozhnikov, Doctor of Science (Tech.),

Federal State Budgetary Educational Institution of Higher Education

"Emperor Alexander I St. Petersburg State Transport University"

Russian Federation, 190031, St. Petersburg, Moskovsky ave., 9,

contact phone number (+7) (812) 457 85 79, e-mail: port.at.pgups@gmail.com

The Self-Checking Concurrent Error-Detection Systems Synthesis Based on the Boolean Complement to the Bose-Lin Codes with the Modulo Value $M = 4$

The presented paper is devoted to the development of the Boolean complement method for the organization of the self-checking concurrent error-detection (CED) systems for digital devices. The article considers the features of using the modular sum codes (Bose-Lin codes) for these purposes, especially the Bose-Lin code by the modulo $M = 4$. This code has two check bits and only four different check vectors, this makes it easier to use it in the organization of the self-checking CED system. The article presents the block diagrams of the organization of the CED system by the method of Boolean complement to the considered modular sum code. The examples of the CED system synthesis by the Boolean complement method are given. The article defines the restrictions imposed on the CED systems synthesis procedure, and also forms an algorithm for synthesizing a self-checking CED systems by the method of the Boolean complement to the Bose-Lin code by the modulo $M = 4$.

Key words: self-checking discrete device; self-checking concurrent error-detection systems; duplication method; Boolean complement method; Bose-Lin code; testability.

The use of the self-checking concurrent error-detection (CED) systems for automation and computer equipment makes it possible to timely identify the incorrectly functioning components [1, 2]. This, in turn, makes it possible to prevent the transmission of incorrect information to the output cascades of control systems, as well as to switch to backup sets (if available). As a result,

© Efanov D.V., Sapozhnikov V.V., Sapozhnikov VI.V., 2021

it is possible to avoid the occurrence of violations in the provided technological processes, to exclude failures of technical means of control (both protective and dangerous) [3].

There are several methods of the CED systems organization. The simplest method involves the use of duplication with a comparison of the values at the self-titled outputs of different sets of devices [4, 5]. However, this method of the CED systems organization has some disadvantages, the main of which are the high structural redundancy of the final diagnostic system, as well as the inability in some cases to form a complete set of test combinations for all components (the inability to ensure the property of complete self-checking of the structure) [6].

The method of check bits calculating is widely used in the CED systems organization. This method involves the check equipment synthesis in such a way that code words belonging to a pre-selected code are formed at the outputs of the checked object and the check block [7—9]. In this method, any uniform block codes can act as codes, but preference is given to codes that have low redundancy and are focused on detecting errors in codewords [10]. Such codes include parity codes, constant-weight codes and various sum codes [11]. The method of check bits calculating makes it possible to synthesize simpler CED systems, but at the same time this method excludes the possibility of minimizing the indicators of the complexity of the technical implementation of the device and does not make it possible to construct completely self-checking devices under any conditions. This is due to the fact that the method for the check bits calculating makes it possible to construct a single version of the control logic block for any diagnosis object. Codes focused on error detection do not always allow detecting any types of errors, but there are known methods for monitoring digital devices with the transformation of their structures or with the allocation of groups of checked outputs with full error-detection [2, 12, 13].

An alternative method is the Boolean complement method, which involves using a Boolean complement block for functions of the diagnosis object up to functions of a special type [14]. This provides control by a number of diagnostic features, for example, by the belonging of functions to the class of self-dual functions [15] or generated code vectors to a constant-weight code [16], or both properties at once [17]. The Boolean complement method in the CED systems synthesis in comparison with the traditional method of the check bits calculating gives the developer greater «flexibility» in the choice of their architecture and parameters [18]. The main advantages of the Boolean complement method are the ability to reduce the complexity of the technical implementation of the check circuit components and the ability to ensure their self-checking.

In the CED systems organizing by the Boolean complement method, a number of restrictions are imposed on the synthesis procedure, which is given in [19]. Methods of the completely self-checking digital devices synthesis by the Boolean complement method with the use of constant-weight codes [6, 14, 16, 20, 21] and the belonging of functions to the class of self-dual Boolean functions [22, 23] have been sufficiently studied. Much less attention is paid in publications to the use of separable codes (for example, sum codes) for these purposes. For example, one of the few publications is known [24] that covers studies of the use of classical Berger codes [25] in the CED systems synthesis using the Boolean complement method. The lack of research on this issue is probably due to the fact that in the CED systems organization using the Boolean complement method, in this case, it is necessary to take into account the limitations associated with the functions implemented by the diagnostic object.

The purpose of this publication is to present the results of research on the features of application of modular sum codes (for example, the sum code in the $M = 4$ modulo residue ring, or the Bose-Lin code with $M = 4$ modulo [26]) in the CED systems organization using the Boolean complement method.

Block diagrams of the implementation of self-checking devices using the Boolean complement method to Bose-Lin codes with the $M = 4$ modulo value. In the CED systems constructing based on Bose-Lin codes, the structure organized by the check bits calculating method is most often used (Fig. 1) [27]. In this structure, the diagnostic object is the $F(x)$ block, which calculates the system of operating functions $f_1, f_2, \dots, f_{m-1}, f_m$. The CED systems are constructed as follows. In it, the control logic block $G(x)$ calculates two check functions g_1 and g_2 such that any code word $\langle f_m f_{m-1} \dots f_2 f_1 g_2 g_1 \rangle$ belongs to the code with summation by modulo $M = 4$ (let's denote it as $S4(m, k)$ -code). The fact that the code word $\langle f_m f_{m-1} \dots f_2 f_1 g_2 g_1 \rangle$ corresponds the $S4(m, k)$ -code is checked by the $S4(m, k)$ -totally self-checking checker ($S4-TSC$), which is usually equipped with two outputs z^0 and z^1 [28, 29]. If the code word coming from $F(x)$ and $G(x)$ blocks belongs to a pre-selected $S4(m, k)$ -code, then a two-rail signal $\langle 01 \rangle$ or $\langle 10 \rangle$ is generated at the outputs of the $S4-TSC$. In the presence of inconsistencies in the data and check vectors, the two-rail is violated.

Malfunctions of the $F(x)$ block affect the values of the bits of data vectors, causing distortions in their bits. The fact of this discrepancy is fixed by the $S4-TSC$. The tester does not detect only such an error in the data vector, which causes its transformation into a data vector with a similar check vector. As identified in [30], $S4(m, k)$ -codes do not detect 100% symmetrical errors in data vectors, 100% unidirectional errors by the multiplicity $d = jM$, $j = 1, 2, \dots, \left\lfloor \frac{m}{M} \right\rfloor$, and a certain proportion of asymmetrical errors by the mul-

Fig. 1. The block diagram of the CED systems organization by the method of check bits calculating by the Bose-Lin code with the modulo value $M = 4$

tiplicity $d = M + 2j$, $j = 1, 2, \dots, \left\lfloor \frac{m-M}{2} \right\rfloor$. Other types of errors are detected

by the $S4(m, k)$ -codes.

Various approaches can be used to organize the CED systems with the detection of any errors at the outputs of the $F(x)$ block using the $S4(m, k)$ -code. One of them is the search for groups of unidirectionally and asymmetrically independent outputs (UAI-groups) with an acceptable multiplicity of unidirectional errors $d = 2$ and $d = 3$. This approach to the use of classical Berger codes is described in [31]. Another approach is a special transformation of the $F(x)$ block structure into the structure which is checkable by the $S4(m, k)$ -code. This transformation is similar to the one described in [13] for Berger codes.

The drawback of the block diagram Fig. 1 is that in the selected basis of functional elements, the $G(x)$ block in the CED systems has a single implementation. This eliminates the possibility of influencing the complexity of the technical implementation of the final device, as well as the possibility of providing the formation of a complete set of test combinations for the checking the $S4(m, k)$ -code checker.

An alternative way for the CED systems organization by the $S4(m, k)$ -codes is to use the Boolean complement method. Moreover, there are two possible ways to organize it (Fig. 2). The CED systems differs from the one used in the structure in Fig. 1 in that it also uses a Boolean complement block

Fig. 2. The block diagrams of the CED systems organization by the Boolean complement method to the Bose-Lin code with the modulo value $M = 4$ with the conversion of two functions (a) and with the conversion of one function (b)

Fig. 3. Combinational devices for which CED systems can be organized by the Boolean complement method to the Bose-Lin code with the modulo value $M = 4$ with the conversion of two functions (a) and with the conversion of one function (b)

formed by two (or one) adder by modulo two. In the first variant (Fig. 2, a), any two operating functions, for example, f_{m-1} and f_m , are converted using the functions of the complement g_1 and g_2 to the check functions h_1 and h_2 : $h_1 = f_1 \oplus g_1$, $h_2 = f_2 \oplus g_2$. Thus, the code word $\langle f_m^* f_{m-1}^* \dots f_2^* f_1^* h_2 h_1 \rangle$ is generated at the checker inputs. The $S4(m-2, 2)$ -code checker is used to check whether it belongs to the $S4(m, k)$ -code. The second variant (Fig. 2, b) involves converting only one operating function, for example, the f_m function using the g_2 function: $h_2 = f_2 \oplus g_2$; the low-order check bit function is calculated directly by the control logic block: $h_2 = g_2$. The $S4(m-1, 2)$ -code is used to check the code word $\langle f_m^* f_{m-1}^* \dots f_2^* f_1^* h_2 h_1 \rangle$.

This approach to the CED systems organization makes it possible to get a fairly large number of ways to implement them with different parameters. As noted above, according to the block diagram in Fig. 1, only one implementation variant is possible, and according to the block diagrams presented in Fig. 2,

the number of implementation variants is determined by the values:

$$N_{CED} = 2!C_m^2 = (m-1)m, \quad N_{CED} = 2!C_m^1 = 2m.$$

Thus, it is possible to search for a simpler (from the point of view of the complexity of the technical implementation) variant for constructing a self-checking device with the CED system, implemented by the Boolean complement method to Bose-Lin codes with the modulo value $M = 4$.

In contrast to the use of constant-weight codes for the CED systems organization by the Boolean complement method, in this variant it is required to exclude the possibility of simultaneous occurrence of errors in both data and check bits. This problem can be solved, for example, by searching for groups of converted and non-converted outputs that should not be dependent. The features of this search will be described below.

Let's give examples of using the ones shown in Fig. 2 block diagrams for organizing control of combinational automation devices (Fig. 3). The outputs of the devices shown in Fig. 3 correspond to the outputs of the devices shown in Fig. 2.

The examples of implementing self-checking devices using the Boolean complement method to the Bose-Lin code with the modulo value $M = 4$. In

Table 1. The description of the CED system operation implemented using the method of check bit calculating

No	x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4	f_5	f_6	g_1	g_2
0	0	0	0	0	0	1	1	0	0	1	1	1
1	0	0	0	1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1	0	0	1	0
3	0	0	1	1	0	1	1	0	0	0	1	0
4	0	1	0	0	0	0	0	0	1	0	0	1
5	0	1	0	1	0	1	1	0	0	0	1	0
6	0	1	1	0	0	1	0	0	1	0	1	0
7	0	1	1	1	1	0	0	0	1	0	1	0
8	1	0	0	0	0	1	1	1	0	1	0	0
9	1	0	0	1	0	1	1	0	1	0	1	1
10	1	0	1	0	1	1	1	0	0	1	0	0
11	1	0	1	1	1	0	0	1	1	1	0	0
12	1	1	0	0	0	0	0	1	1	1	1	1
13	1	1	0	1	0	1	1	0	0	1	1	1
14	1	1	1	0	0	0	1	1	0	1	1	1
15	1	1	1	1	0	0	0	0	0	1	0	1

Table 1, the combination device $F(x)$ is specified, and Table 2 and Table 3 show one variant for determining the values of the check functions g_1 and g_2 in accordance with the structures shown in Fig. 2. In the first case, the CED systems is constructed with control by the $S4(4, 2)$ -code, and in the second case – with checking by the $S4(5, 2)$ -code. The checker for the first code is easier than for the second code [30].

Minimization of functions $f_1 - f_6$ by the Karnaugh method gives the following result:

$$\begin{aligned} f_1 &= \overline{x_1}x_2x_3x_4 \vee x_1\overline{x_2}x_3\overline{x_4}; \\ f_2 &= \overline{x_1}x_2x_3\overline{x_4} \vee x_1\overline{x_2}x_3x_4 \vee \overline{x_1}x_2\overline{x_3}x_4 \vee \overline{x_1}x_2x_3x_4; \\ f_3 &= \overline{x_1}x_2x_3\overline{x_4} \vee x_1\overline{x_2}x_3\overline{x_4} \vee x_2x_3x_4 \vee \overline{x_2}x_3\overline{x_4}; \\ f_4 &= \overline{x_1}x_2x_3\overline{x_4} \vee x_1\overline{x_2}x_3\overline{x_4} \vee x_1\overline{x_2}x_3x_4 \vee \overline{x_1}x_2x_3x_4 \vee \overline{x_1}x_2\overline{x_3}x_4; \\ f_5 &= \overline{x_1}x_2x_3\overline{x_4} \vee x_2x_3\overline{x_4} \vee x_1x_2x_3; \\ f_6 &= \overline{x_2}x_3\overline{x_4} \vee x_1x_2 \vee x_1x_3. \end{aligned}$$

Let's estimate the complexity of the implementation of the $F(x)$ block by the number of inputs of logical elements (it directly depends on the number of operation signs (and letters) in the formula records of expressions describing the functions $f_1 - f_6$): $L_{F(x)} = 85$. Knowing the $L_{F(x)}$ value, it is easy to estimate the complexity of the technical implementation of a device with the CED systems, implemented using the duplication method:

$$L_D = 2L_{F(x)} + mL_{NOT} + (m-1)L_{TRC},$$

where $L_{NOT}=1$ is the complexity of implementing the signal inversion element (NOT), $L_{TRC}=12$ – complexity of implementing the two-rail signal compression element (TRC) [30]. For the considered example $L_D = 2 \cdot 85 + 6 \cdot 1 + 5 \cdot 12 = 236$.

The $S4(6, 2)$ -code is used for monitoring in Table 1. Minimization of g_1 and g_2 functions by the Karnaugh method makes it possible to obtain such logical expressions:

$$\begin{aligned} g_1 &= \overline{x_1}x_3 \vee \overline{x_1}x_2\overline{x_4} \vee x_1\overline{x_3}x_4 \vee \overline{x_1}x_2x_4 \vee x_1x_2\overline{x_4}; \\ g_2 &= x_1x_2 \vee \overline{x_1}x_3\overline{x_4} \vee \overline{x_2}x_3x_4. \end{aligned}$$

The complexity of implementing $L_{G(x)} = 30$. The complexity of implementing a device with the CED systems is determined by the formula:

$$L_{S4} = L_{F(x)} + L_{G(x)} + L_{G(f)} + (k-1)L_{TRC},$$

where $L_{G(f)}$ is the complexity of implementing the selected code's encoder.

Table 2. The description of the CED system operation

No	x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4	f_5	f_6	g_1	g_2	h_1	h_2	XOR^1	XOR^2
<i>With the conversion of two functions</i>																
0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	01	10
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	00	01
2	0	0	1	0	0	0	1	1	0	0	1	0	1	0	01	00
3	0	0	1	1	0	1	1	0	0	0	1	0	1	0	01	00
4	0	1	0	0	0	0	0	0	1	0	1	0	0	0	10	00
5	0	1	0	1	0	1	1	0	0	0	1	0	1	0	01	00
6	0	1	1	0	0	1	0	0	1	0	1	1	0	1	10	01
7	0	1	1	1	1	0	0	0	1	0	1	1	0	1	10	01
8	1	0	0	0	0	1	1	1	0	1	1	0	1	1	01	11
9	1	0	0	1	0	1	1	0	1	0	0	0	1	0	11	00
10	1	0	1	0	1	1	1	0	0	1	1	0	1	1	01	11
11	1	0	1	1	1	0	0	1	1	1	0	1	1	0	11	10
12	1	1	0	0	0	0	0	1	1	1	1	0	0	1	10	11
13	1	1	0	1	0	1	1	0	0	1	1	1	1	0	01	10
14	1	1	1	0	0	0	1	1	0	1	1	1	1	0	01	10
15	1	1	1	1	0	0	0	0	0	1	0	1	0	0	00	10
<i>With the conversion of one function</i>																
0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	–	11
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	–	01
2	0	0	1	0	0	0	1	1	0	0	1	0	1	0	–	00
3	0	0	1	1	0	1	1	0	0	0	1	0	1	0	–	00
4	0	1	0	0	0	0	0	0	1	0	0	1	0	1	–	01
5	0	1	0	1	0	1	1	0	0	0	1	0	1	0	–	00
6	0	1	1	0	0	1	0	0	1	0	1	0	1	0	–	00
7	0	1	1	1	1	0	0	0	1	0	1	0	1	0	–	00
8	1	0	0	0	0	1	1	1	0	1	1	0	1	1	–	10
9	1	0	0	1	0	1	1	0	1	0	1	1	1	1	–	01
10	1	0	1	0	1	1	1	0	0	1	1	0	1	1	–	10
11	1	0	1	1	1	0	0	1	1	1	1	0	1	1	–	10
12	1	1	0	0	0	0	0	1	1	1	1	1	1	0	–	11
13	1	1	0	1	0	1	1	0	0	1	1	1	1	0	–	11
14	1	1	1	0	0	0	1	1	0	1	1	1	1	0	–	11
15	1	1	1	1	0	0	0	0	0	1	0	1	0	0	–	11

Fig. 4. The Bose-Lin code encoders: $S_4(6, 2)$ -code (a), $S_4(5, 2)$ -code (b) and $S_4(4, 2)$ -code (c)

The complexity of the encoder depends on how many data and check bits the code has, as well as on the implementation method chosen for it. Fig. 4 shows the encoders for $S_4(6, 2)$, $S_4(5, 2)$ and $S_4(4, 2)$ codes implemented on standard adders, which are used in the examples of the implementation of the CED systems given here and below. The complexity of encoders is determined based on the complexity of implementing elementary adders included in their structures: $L_{FA} = 14$, $L_{HA} = 6$, $L_{XOR} = 6$. Thus, the complexity of the encoders $S_4(6, 2)$, $S_4(5, 2)$ and $S_4(4, 2)$ codes are equal, respectively: $L_{G(f)} = 46$, $L_{G(f)} = 38$, $L_{G(f)} = 30$. A device with the CED systems implemented using the method of check bits calculating has a complexity of implementation $L_{S_4} = 85 + 30 + 46 + 12 = 173$.

Table 2 describes devices with the CED systems implemented according to the ones shown in Fig. 2 structure. For them, the complexity of implementation is determined by the formula:

$$L_{S_4BC} = L_{F(x)} + L_{G(x)} + L_{BC} + L_{G(f)} + (k-1)L_{TRC},$$

where L_{BC} is the complexity of implementing the Boolean complement block.

Table 2 shows that:

$$g_1 = \overline{x_1}x_2 \vee \overline{x_1}x_3 \vee \overline{x_2}x_3 \vee \overline{x_4};$$

$$g_2 = \overline{x_1}\overline{x_2}\overline{x_3} \vee \overline{x_1}x_2x_4 \vee \overline{x_1}x_3x_4 \vee \overline{x_2}x_3 \vee \overline{x_2}x_3.$$

The complexity of implementing $L_{G(x)} = 29$. Thus, the complexity of implementing a device with the CED systems is equal to:

$$L_{S_4BC} = 85 + 29 + 2 \cdot 6 + 38 + 12 = 176.$$

Table 2 shows that:

$$g_1 = \overline{x_1} \overline{x_2} \vee \overline{x_1} x_3 \vee \overline{x_1} x_3 \vee \overline{x_2} \overline{x_4} \vee \overline{x_2} x_3 x_4;$$

$$g_2 = \overline{x_1} x_2 \vee \overline{x_2} \overline{x_3} x_4 \vee \overline{x_1} x_2 \overline{x_3} \overline{x_4}.$$

The complexity of the implementation can be estimated by the number of letters $L_{G(x)}=28$. The complexity of implementing a device with the CED systems is equal to: $L_{S4BC} = 85 + 28 + 6 + 30 + 12 = 161$.

The obtained numerical data for assessing the complexity of the implementation of devices are summarized in Table 3.

Table 3 shows that the self-checking device implemented according to the first variant (Fig. 2, a) is comparable in complexity to the device implemented according to the traditional block diagram (Fig. 1), and the second one is even simpler.

In contrast to the method of Boolean complement to constant-weight codes [32], the synthesis procedure for this method up to separable codes (in particular, up to $S4(m, k)$ -codes) does not require the selection and redefinition of the values of check functions – in this case, they are redefined «rigidly». This reduces the number of ways to determine values compared to using constant-weight codes for these purposes and simplifies the synthesis procedure itself.

The algorithm for the CED system synthesis. When implementing self-checking digital devices, the synthesis tasks may be different, including some restrictions that may be imposed. For example, the source device $F(x)$ may have an unchanged structure, configuration of elements, and relationships between them. In this case, when choosing the method of CED systems synthesis using the method of Boolean complement to the $S4(m, k)$ -code, the choice of transformed functions is required. It is necessary to prevent the propagation of internal errors of the device simultaneously to its converted and non-converted outputs. In other words, the converted outputs must form a group that is structurally or functionally independent of the non-converted outputs.

Definition 1. A group of outputs $(f_{i_1}, f_{i_2}, \dots, f_{i_r})$, $\{i_1, i_2, \dots, i_r\} \in \{1, 2, \dots, m\}$, is structurally independent of a group of outputs $(f_{j_1}, f_{j_2}, \dots, f_{j_p})$,

Table 3. The complexity of implementing a device with the CED system, that are implemented using various methods

L_D	L_{S4}	L_{S4BC} (Fig. 2, a)	L_{S4BC} (Fig. 2, b)
236	173	176	161
100	74.305	74.576	68.22
136.416	100	101.734	93.064

$\{j_1, j_2, \dots, j_p\} \in \{1, 2, \dots, m\} \setminus \{i_1, i_2, \dots, i_r\}$, if there is no element in the structure of a digital device whose paths lead simultaneously to at least one of the outputs $(f_{i_1}, f_{i_2}, \dots, f_{i_p})$ and at least one of the outputs of the group $(f_{j_1}, f_{j_2}, \dots, f_{j_p})$.

If there is a group of outputs $(f_{i_1}, f_{i_2}, \dots, f_{i_q})$, that is structurally independent of the output group $(f_{j_1}, f_{j_2}, \dots, f_{j_p})$, then the outputs of both the first and second groups can be selected as the converted outputs. Among these groups, the search is made for those outputs that should be converted taking into account the impact on the complexity of the technical implementation of the $G(x)$ block and the need to ensure the check ability of the selected code checker and elements of addition by modulo $M = 2$ (XOR's).

Definition 2. The group of outputs (f_{i_1}, f_{i_2}) , $i_1, i_2 \in \{1, 2, \dots, m\}$, is functionally independent of the group of outputs $\{f_1, f_2, \dots, f_m\} \setminus \{f_{i_1}, f_{i_2}\}$, if for any G_q element with the y_q output the following condition is satisfied:

$$\left(\frac{\partial f_{i_1}}{\partial y_q} \vee \frac{\partial f_{i_2}}{\partial y_q} \right) \frac{\partial f_j}{\partial y_q} = 0, \quad j \in \{f_1, f_2, \dots, f_m\} \setminus \{f_{i_1}, f_{i_2}\}. \quad (1)$$

The expression in brackets defines the distortion conditions for at least one output from the pair (f_{i_1}, f_{i_2}) , and the multiplier $\frac{\partial f_j}{\partial y_q}$ determines the distortion of any other output from the group $\{f_1, f_2, \dots, f_m\} \setminus \{f_{i_1}, f_{i_2}\}$.

Using the introduced concepts of structurally and functionally independent groups of device outputs, we will form an algorithm for the CED systems synthesis using the method of Boolean complement to the $S4(m, k)$ -code with conversion of two outputs for a device with a known structure.

The algorithm:

1. All pairs (f_{i_1}, f_{i_2}) , $i_1, i_2 \in \{1, 2, \dots, m\}$, of outputs $f_1, f_2, \dots, f_m$ of the device $F(x)$ are determined $\left(\text{in total } C_m^2 = \frac{m^2 - m}{2} \text{ pairs} \right)$.
2. Using the expression (1), we identify those pairs of outputs that are functionally independent of the other outputs.
3. The set Q of all pairs of outputs that are functionally independent of other outputs is formed and its power M is determined.
4. It is assumed $s = 1$.

5. A pair s is selected from the set Q – the outputs of the pair (f_{i_1}, f_{i_2}) . These outputs will be convertible.

6. For the remaining outputs, the smallest non-negative residue of the total number of one bit is calculated – the $r \pmod{4}$ number, this number is represented in binary form and written in the h_1 and h_2 bits.

7. The values of the complement functions g_1 and g_2 are uniquely determined:

$$g_1 = f_{i_1} \oplus h_1, \quad g_2 = f_{i_2} \oplus h_2.$$

8. The presence of all test combinations for the $S4(m, k)$ -code checker is checked: $T_{S4} = \{00, 01, 10, 11\}$.

9. The presence of all test combinations for the XOR_1 and XOR_2 conversion elements is checked: $T_{XOR_1} = T_{XOR_2} = \{00, 01, 10, 11\}$.

10. The outputs in the pair (f_{i_1}, f_{i_2}) are rearranged and steps 7 – 9 of the algorithm are performed for the new variant.

11. If a pair s from the set Q makes it possible to form all test combinations of sets T_{S4} , T_{XOR_1} and T_{XOR_2} for one of the variants for determining, then logical expressions describing the functions g_1 and g_2 and the complexity of their implementation in the selected basis of elements are determined for it. The obtained data is stored, and the pair itself is entered in the set P of acceptable variants for the CED system implementing; otherwise, the pair is included in the set R of parameters excluded from consideration.

12. It is assumed $s = s + 1$ and the condition $s \neq M$ is checked: if «yes», then steps 5 – 11 of the algorithm are performed, if "no", then the transition to the next step of the algorithm is carried out.

13. The P set of acceptable variants for the control circuit implementing is considered. If $P \equiv \emptyset$, the CED systems synthesis by the $S4(m, k)$ -code is not possible. If $P \neq \emptyset$, the variant of the $G(x)$ control logic block synthesis with minimal complexity of technical implementation is considered.

14. The complexity of a device implemented using the method of Boolean complement to the $S4(m, k)$ -code is compared with the complexity of a device implemented using the duplication method:

$$L_{S4BC} < L_D. \quad (2)$$

15. If the inequality (2) is satisfied, then the CED system is synthesized using the method of Boolean complement to the $S4(m, k)$ -code. Otherwise, the synthesis by the presented method does not allow synthesizing a simpler self-checking discrete device than by the duplication method.

The algorithm for the CED system synthesis according to the proposed method with conversion of one function is similar. In this case, not pairs of outputs are determined, but only single outputs, which are functionally independent relative to other outputs. For them, a simplified algorithm for the CED system synthesis is implemented. If we are dealing with the CED system synthesis for a device with a variable structure, the algorithm presented above can be simplified. In this case, the step of checking condition (1) in the algorithm is skipped (see clause 2 of the algorithm) and all $m(m-1)$ variants of the implementation of the CED system with two convertible outputs and all $2m$ variants of the implementation of the CED system with one convertible output are considered.

Conclusions

In contrast to the use of constant-weight codes in the CED systems organization using the Boolean complement method, when using separable codes, including $S4(m, k)$ -codes, additional restrictions are imposed on the synthesis procedure. These restrictions are related to the possibility of guaranteed fixing of distortions in both data and check bits of code words generated at the checker inputs. The possibility of the simultaneous error's occurrence in both data and check vectors arises due to the fact that the check bits are not generated by a separate device, but are obtained by converting the pre-selected operating functions of the diagnostic object. When CED systems organization using the Boolean complement method, this feature should be taken into account and the probability of such errors occurrence should be excluded by schematically, for example, by selecting groups of independent outputs. The algorithm proposed in the article makes it possible.

When using the Boolean complement method, it is possible to implement a large number of CED systems depending on the organization scheme (with one or two converted functions). At the same time, structures synthesized with the transformation of only one operating function are simpler in the sense of ensuring the check ability of the Boolean complement block. As a drawback, we can note the limitations of the approach to the CED systems organization with the analysis of the values of the operating functions of the diagnostic object: it is effective on a limited number of inputs of the diagnostic object, which is associated with the number of operations when analyzing all input effects (up to 30...35 even for the most powerful computing systems). The method, however, can also be used for decomposing the source object and monitoring its subcircuits, followed by combining the control outputs of each CED systems at the inputs of a self-checking comparator [33].

The use of modular sum codes in conjunction with the Boolean complement method is also promising from the point of view that a complete set of check vectors is always formed for them, which facilitates the synthesis of a totally self-checking digital device.

REFERENCES

1. Parkhomenko, P.P. and Sogomonyan, E.S. (1981), *Osnovy tekhnicheskoy diagnostiki (optimizatsiya algoritmov diagnostirovaniya, apparaturnye sredstva)* [Basics of technical diagnostics (optimization of diagnostic algorithms and equipment)], Energoatomizdat, Moscow, USSR.
2. Sogomonyan, E.S. and Slabakov, E.V. (1989), *Samoproverjaemye ustrojstva i otkazov-tojchivyye sistemy* [Self-checking devices and failover systems], Radio i Svjaz', Moscow, USSR.
3. Sapozhnikov, V.V., Sapozhnikov, V.I., Hristov, H.A. and Gavzov, D.V. (1995), *Metody postroeniya bezopasnyh mikroelektronnyh sistem zheleznodorozhnoj avtomatiki* [Methods for constructing safety microelectronic systems for railway automation], Transport, Moscow, Russia.
4. Goessel, M. and Graf, S. (1994), *Error Detection Circuits*, McGraw-Hill, London, UK.
5. Nicolaidis, M. and Zorian, Y. (1998), "On-Line Testing for VLSI – A Compendium of Approaches", *Journal of Electronic Testing: Theory and Applications*, Vol. 12, pp. 7-20.
6. Goessel, M., Morozov, A.V., Sapozhnikov, V.V. and Sapozhnikov, V.I. (2005), "Checking Combinational Circuits by the Method of Logic Complement", *Avtomatika i telemekhanika*, Vol. 8, pp. 161–172.
7. Mikoni, S.V. (1992), *Obshchie diagnosticheskie bazy znaniy vychislitel'nyh sistem* [General Diagnostic Knowledge Base of Computing Systems], SPIIRAN, St. Petersburg, Russia.
8. Mitra, S. and McCluskey, E.J. (2000), "Which Concurrent Error Detection Scheme to Choose?", *Proceedings of International Test Conference*, USA, Atlantic City, NJ, October 03-05, 2000, pp. 985-994.
9. Das, D., Touba, N.A., Seuring, M. and Gossel, M. (2000), "Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes", *Proceedings of IEEE 6th International On-Line Testing Workshop (IOLTW)*, Spain, Palma de Mallorca, July 3-5, 2000, pp. 171-176.
10. Piestrak, S.J. (1995), *Design of Self-Testing Checkers for Unidirectional Error Detecting Codes*, Oficyna Wydawnicza Politechniki Wrocławskiej, Warsaw, Poland.
11. Efanov, D., Sapozhnikov, V. and Sapozhnikov, V.I. (2017), "Generalized Algorithm of Building Summation Codes for the Tasks of Technical Diagnostics of Discrete Systems", *Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017)*, Novi Sad, Serbia, September 29 – October 2, 2017, pp. 365-371, DOI: 10.1109/EWDTS.2017.8110126.
12. Busaba, F.Y. and Lala, P.K. (1994), "Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors", *Journal of Electronic Testing: Theory and Applications*, Vol. 1, pp. 19-28, DOI: 10.1007/BF00971960.
13. Morosow, A., Saposhnikov, V.V., Saposhnikov, V.I. and Goessel, M. (1998), "Self-Checking Combinational Circuits with Unidirectionally Independent Outputs", *VLSI Design*, Vol. 5, № 4, pp. 333-345, DOI: 10.1155/1998/20389.

14. Goessel, M., Morozov, A.V., Sapozhnikov, V.V. and Sapozhnikov, VI.V. (2003), "Logic Complement, a New Method of Checking the Combinational Circuits", *Avtomatika i telemeckhanika*, Vol. 1, pp. 167-176.
15. Saposhnikov, VI.V., Dmitriev, A., Goessel, M. and Saposhnikov, V.V. (1996), "Self-Dual Parity Checking – a New Method for on Line Testing", *Proceedings of 14th IEEE VLSI Test Symposium*, USA, Princeton, 1996, pp. 162-168.
16. Göessel, M., Ocheretny, V., Sogomonyan, E. and Marienfeld, D. (2008), *New Methods of Concurrent Checking: Edition 1*, Springer Science+Business Media B.V., Dordrecht, Netherlands.
17. Efanov, D., Sapozhnikov, V., Sapozhnikov, VI., Osadchy, G. and Pivovarov, D. (2019), "Self-Dual Complement Method up to Constant-Weight Codes for Arrangement of Combinational Logical Circuits Concurrent Error-Detection Systems", *Proceedings of 17th IEEE East-West Design & Test Symposium (EWDTS'2019)*, Batumi, Georgia, September 13-16, 2019, pp. 136-143, DOI: 10.1109/EWDTS.2019.8884398.
18. Sapozhnikov, V.V., Sapozhnikov, VI.V., Efanov, D.V. and Pivovarov, D.V. (2017), "Boolean Complement Method Based on Constant-Weight Code «1-out-of-4» for Formation of Totally Self-Checking Concurrent Error Detection Systems", *Elektronnoe modelirovanie*, Vol. 39, № 2, pp. 15–34.
19. Efanov, D.V., Sapozhnikov, V.V., Sapozhnikov, VI.V. and Pivovarov, D.V. (2020), "The Synthesis Conditions of Completely Self-Testing Embedded-Control Circuits Based on the Boolean Complement Method to the «1-out-of-m» Constant-Weight Code", *Automatic Control and Computer Sciences*, Vol. 54, № 2, pp. 89-99, DOI: 10.3103/S0146411620020042.
20. Sen, S.K. (2010), "A Self-Checking Circuit for Concurrent Checking by 1-out-of-4 code with Design Optimization using Constraint Don't Cares", *National Conference on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEEERE 2010)*, Sikkim Manipal Institute of Technology, Sikkim, India, December 22-24, 2010.
21. Das, D.K., Roy, S.S., Dmitriev, A., Morozov, A. and Gössel, M. (2012), "Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes", *Proceedings of the 10th International Workshops on Boolean Problems*, Freiberg, Germany, September, 2012, pp. 33-40.
22. Sapozhnikov, V.V., Sapozhnikov, VI.V. and Goessel, M. (2001), *Samodvoystvennye diskretnye ustrojstva* [Self-Dual Discrete Devices], Energoatomizdat, St. Petersburg, Russia.
23. Lala, P.K. (2001), *Self-Checking and Fault-Tolerant Digital Design*, Morgan Kaufmann Publishers, San Francisco, USA.
24. Morozov, M., Saposhnikov, V.V., Saposhnikov, VI.V. and Goessel, M. (2000), "New Self-Checking Circuits by Use of Berger-codes", *Proceedings of 6th IEEE International On-Line Testing Workshop*, Palma De Mallorca, Spain, July 3-5, 2000, pp. 171-176.
25. Berger, J.M. (1961), "A Note on Error Detection Codes for Asymmetric Channels", *Information and Control*, Vol. 4, № 1, pp. 68-73, DOI: 10.1016/S0019-9958(61)80037-5.
26. Bose, B. and Lin, D.J. (1985), "Systematic Unidirectional Error-Detection Codes", *IEEE Transaction on Computers*, Vol. C-34, pp. 1026-1032.
27. Das, D. and Toubia, N.A. (1999), "Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes", *Journal of Electronic Testing: Theory and Applications*, Vol. 15, № 1-2, pp. 145-155, DOI: 10.1023/A:1008344603814.
28. Jha, N.K. (1991), "Totally Self-Checking Checker Designs for Bose-Lin, Bose and Blaum Codes", *IEEE Transaction on Computer-Aided Design*, Vol. 10, № 1, pp. 136-143, DOI: 10.1109/43.62799.

29. Nikolos, D. and Kavousianos, X. (1999), "Modular TSC Checkers for Bose-Lin and Bose Codes", *Proceedings of the 17th IEEE VLSI Test Symposium*, Dana Point, USA, April 25-29, 1999, pp. 354-360.
30. Sapozhnikov, V.V., Sapozhnikov, V.I. and Efanov, D.V. (2020), *Kody s summirovaniem dlya sistem tekhnicheskogo diagnostirovaniya. Tom 1: Klassicheskie kody Bergera i ih modifikatsii* [Sum Codes for Technical Diagnostics Systems. Volume 1: Classical Berger Codes and Their Modifications], Nauka, Moscow, Russia.
31. Efanov, D.V., Sapozhnikov, V.V. and Sapozhnikov, V.I. (2017), "Conditions for Detecting a Logical Element Fault in a Combination Device under Concurrent Checking Based on Berger's Code", *Avtomatika i telemekhanika*, Vol. 5, pp. 152-165.
32. Efanov, D.V., Sapozhnikov, V.V., Sapozhnikov, V.I. and Pivovarov, D.V. (2018), "The «2-out-of-4» constant-weight code application in the self-checking check circuits organization based on the Boolean complement method", *Informatika*, Vol. 15, № 4, pp. 71-85.
33. Nikolos, D. (1998), "Self-Testing Embedded Two-Rail Checkers", *Journal of Electronic Testing: Theory and Applications*, Vol. 12, № 1-2, pp. 69-79, DOI: 10.1023/A:1008281822966.

Received 25.07.20

Д.В. Єфанов, В.В. Сапожніков, В.І. Сапожніков

СИНТЕЗ САМОКОНТРОЛЬОВАНИХ СХЕМ ВБУДОВАНОГО
КОНТРОЛЮ МЕТОДОМ ЛОГІЧНОГО ДОПОВНЕННЯ
ДО КОДІВ БОУЗА—ЛІНА З МОДУЛЕМ $M = 4$

Розглянуто особливості використання в цифрових пристроях модульних кодів з підсумовуванням Боуза—Ліна, а саме коду з підсумовуванням по модулю $M = 4$, який має два контрольних розряди і чотири різних контрольних вектора, що полегшує його використання при створенні самоконтрольованої схеми вбудованого контролю. Описано структурні схеми організування контролю методом логічного доповнення до модульного коду з підсумовуванням, що розглядається. Наведено приклади синтезу схем контролю методом логічного доповнення. Визначено обмеження, які накладаються на процедуру синтезу схеми контролю, та сформовано алгоритм синтезу самоконтрольованої схеми вбудованого контролю методом логічного доповнення до коду з підсумовуванням по модулю $M = 4$.

К л ю ч о в і с л о в а: самоконтрольований дискретний пристрій, самоконтрольована схема вбудованого контролю, метод дублювання, метод логічного доповнення, код Боуза—Ліна, контролездатність

EFANOV Dmitry Viktorovich, D. Sc., Associate Professor, First Deputy General Director – Chief Engineer of «VEGA Group» LLC, Professor at Automation, Remote Control and Telecommunication on Railway Transport of Russian University of Transport, Professor at Higher School of Transport of the Institute of Mechanical Engineering, Materials and Transport of Peter the Great St. Petersburg Polytechnic University. In 2007 he graduated from St. Petersburg State Transport University. Research area – discrete mathematics, reliability and technical diagnostics of discrete systems.

SAPOZHNIKOV Valery Vladimirovich, D. Sc., Professor, Professor at Automation and Remote Control on Railways Department, Emperor Alexander I St. Petersburg State Transport University. In 1963 he graduated from the Leningrad Institute of Railway Engineers. Research area – reliable synthesis of discrete devices, synthesis of safety systems, synthesis of self-checking circuits, technical diagnostics of discrete systems.

SAPOZHNIKOV Vladimir Vladimirovich, D. Sc., Professor, Professor at Automation and Remote Control on Railways Department, Emperor Alexander I St. Petersburg State Transport University. In 1963 he graduated from the Leningrad Institute of Railway Engineers. Research area – reliable synthesis of discrete devices, synthesis of safety systems, synthesis of self-checking circuits, technical diagnostics of discrete systems.