DESIGN OF SELF-CHECKING DISCRETE DEVICES BASED ON BOOLEAN SIGNALS CORRECTION AND COMPOSITION OF CONSTANT-WEIGHT CODES OF THE "1-OUT-OF-4" AND "3-OUT-OF-4" TYPES. PART II: The Design Method with Conversion of Some Signals from the Object under Diagnosis

D. V. Efanov* and Y. I. Yelina**

*,**Peter the Great Saint Petersburg Polytechnic University, St. Petersburg, Russia
*Solomenko Institute of Transport Problems, Russian Academy of Sciences, St. Petersburg, Russia

*✉ TrES-4b@yandex.ru, **✉ eseniya-elina@mail.ru

Abstract. To reduce the implementation complexity of self-checking discrete devices in comparison with the design method described in part I of this study, the idea is to consider the composition properties of constant-weight codes of the "1-out-of-4" and "3-out-of-4" types. In this case, it is possible to decrease the number of signals converted in the Boolean signals correction block, fixing this number to $p = 4, 3, 2$, or 1. Part II presents modified structures for organizing concurrent error-detection (CED) circuits for objects under diagnosis and a circuit design algorithm considering the choice of p. As established below, decreasing p reduces the variability in the choice of codewords belonging to the composition of 1-out-of-4 and 3-out-of-4 codes; this may affect the total checking property of the checker. This drawback can be eliminated by possible permutations of outputs in controlled output subsets and among all outputs of the object under diagnosis. Another solution here is to increase p to 2, 3, or 4. The advantages of the developed method are shown experimentally, and its efficiency is demonstrated in comparison with duplication and other methods based on Boolean signals correction. When converting $p = 2$ or $p = 1$ signals in the selected output subset, the gain over the above methods is achieved in almost all benchmarks. According to the experimental results, the developed method has high potential for practical design of self-checking discrete devices.

Keywords: self-checking discrete device, concurrent error-detection (CED) circuit, Boolean signals correction, composition of constant-weight codes, 1-out-of-4 and 3-out-of-4 codes, structural redundancy, controllable structure.

INTRODUCTION

The design method for self-checking discrete devices—see part I of this study [1]—is remarkable for the conversion of all signals from a selected quadruple of outputs of an object under diagnosis. On the one hand, this feature provides high flexibility for the device designer: in a concurrent error-detection (CED) circuit, the values at the outputs of the signal correction block (SCB) can be set in eight different ways for each set of argument values. On the other hand, when converting all signals, it is necessary to compute the values of the corresponding number of correction functions by the block $G(X)$. The number of outputs significantly affects both the overall implementation complexity of a CED circuit and the structural redundancy of the self-checking discrete device. The other part of the CED circuit is standard. Nevertheless, according to the experimental results, the method described in part I is rather efficient compared to the classical duplication method [2] and, in several cases, is efficient compared to known methods based on Boolean signals correction (BSC) and the use of constant-weight codes [3].

In view of the above considerations, there exist two approaches to reducing the implementation complexity of a self-checking discrete device with BSC and the composition of constant-weight codes of the "1-out-of-4" and "3-out-of-4" types (further referred to as 1-out-of-4 and 3-out-of-4 codes, respectively). The first is to generate correction functions by the block $G(X)$ so that two or more functions coincide, or parts of subfunctions (cofactors) coincide. This can be done, e.g., by using *binary decision diagrams* (BDDs) and decomposing the Boolean functions computed by the block $G(X)$ into cofactors, followed by optimization of the resulting system of Boolean functions [4]. An alternative is to establish a relationship between the values computed by the object under diagnosis $F(X)$ and the block $G(X)$ for each set of argument values so that some correction functions turn out to be equivalent. This is possible among the large number of Boolean functions formed when using BSC. However, this problem remains unsettled. The second approach to reducing the implementation complexity of the block $G(X)$ is to decrease the number of its outputs while decreasing the number of elements converted in the SCB. For example, to build a CED circuit based on BSC and a 1-out-of-3 code, it suffices to convert only two of the three signals [5]. When solving the same problem with a 1-out-of-4 code, only three gates can be converted instead of four [6]. In the case of a 2-out-of-4 code, the number of gates converted can be reduced from four to two [7]. According to our previous research, the composition of 1-out-of-4 and 3-out-of-4 codes allows reducing the number of gates converted to one! Accordingly, efficient modifications are possible for the CED circuit structure based on BSC and the composition of 1-out-of-4 and 3-out-of-4 codes, originally presented in Fig. 1 of part I of the study [1]. Part II addresses these features of designing self-checking discrete devices.

1. STRUCTURAL MODIFICATION OF CED CIRCUITS BASED ON COMPOSITION OF 1-OUT-OF-4 AND 3-OUT-OF-4 CODES WITH CONVERSION OF SOME SIGNALS

Given the characteristics of the set of codewords in the composition of 1-out-of-4 and 3-out-of-4 codes, the CED circuit structure can be optimized by reducing the number of elements converted in the SCB.

Theorem 1. *The minimum number of XOR gates required to generate, from a codeword* $\langle f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)\rangle$, *a codeword* $\langle h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)\rangle$ *belonging to the composition of two 1-out-of-4 and 3-out-of-4 codes, does not depend on the type of functions implemented by the object under diagnosis and equals 1.*

P r o o f. Regardless of the type of Boolean functions implemented by the object under diagnosis, a codeword $\langle f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)\rangle$ with a weight $r \in \{0, 1, 2, 3, 4\}$ is formed when supplying each set of argument values to its inputs. If $r = 0$, then the codeword $\langle f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)\rangle$ can be converted into a codeword with weight $r = 1$, belonging to the composition of two 1-out-of-4 and 3-out-of-4 codes, by the correction of only one signal. In the case $r = 1$ or $r = 3$, then the codeword $\langle f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)\rangle$ does not need to be corrected: it belongs to the composition of two 1-out-of-4 and 3-out-of-4 codes. If $r = 2$, then the codeword $\langle f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)\rangle$ can be converted into a codeword with weight $r = 1$ or $r = 3$, belonging to the composition of two 1-out-of-4 and 3-out-of-4 codes, by the correction of only one signal. Finally, for $r = 4$, the codeword $\langle f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)\rangle$ can be converted into a codeword with weight $r = 3$, belonging to the composition of two 1-out-of-4 and 3-out-of-4 codes, by the correction of only one signal as well. The correction is unique for each set of argument values. Thus, all variants to convert the codewords $\langle f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)\rangle$ into those belonging to the composition of two 1-out-of-4 and 3-out-of-4 codes have been considered, and the proof of Theorem 1 is complete. ♦

In the original CED circuit structure (see Fig. 1 in part I of the study [1]), some outputs of the object under diagnosis may be left without conversion. In such a modified structure, $p \in \{1, 2, 3\}$ outputs from a quadruple selected on the object's output set are corrected, while $4 - p$ outputs are checked via a direct connection of the composition of two 1-out-of-4 and 3-out-of-4 codes to the checker's inputs.

The modified CED circuit structures for different values $p \in \{1, 2, 3\}$ are shown in Fig. 1. The number p significantly affects the implementation complexity of the block $G(X)$ and, consequently, much affects the structural redundancy of the resulting self-checking discrete device.

Decreasing the value of p affects how many outputs from the object's output quadruple will be directly connected to the *totally self-checking checker* (*TSC*). This, in turn, reduces the subset of codewords in the above composition of constant-weight codes when selecting the conversion method of the vector $\langle f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)\rangle$ into the vector $\langle h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)\rangle$.

Theorem 2. *For each set of argument values, when correcting* $p \in \{1, 2, 3, 4\}$ *outputs of the object under diagnosis, there are*

$$M_p = \left(2^{p-1}\right)^{2^t} \qquad (1)$$

ways to determine the values of the functions $g_i(X)$.

Really, 2^{p-1} (the base in formula (1)) is the number of codewords belonging to the composition of two 1-out-of-4 codes and 3-out-of-4 codes, into which

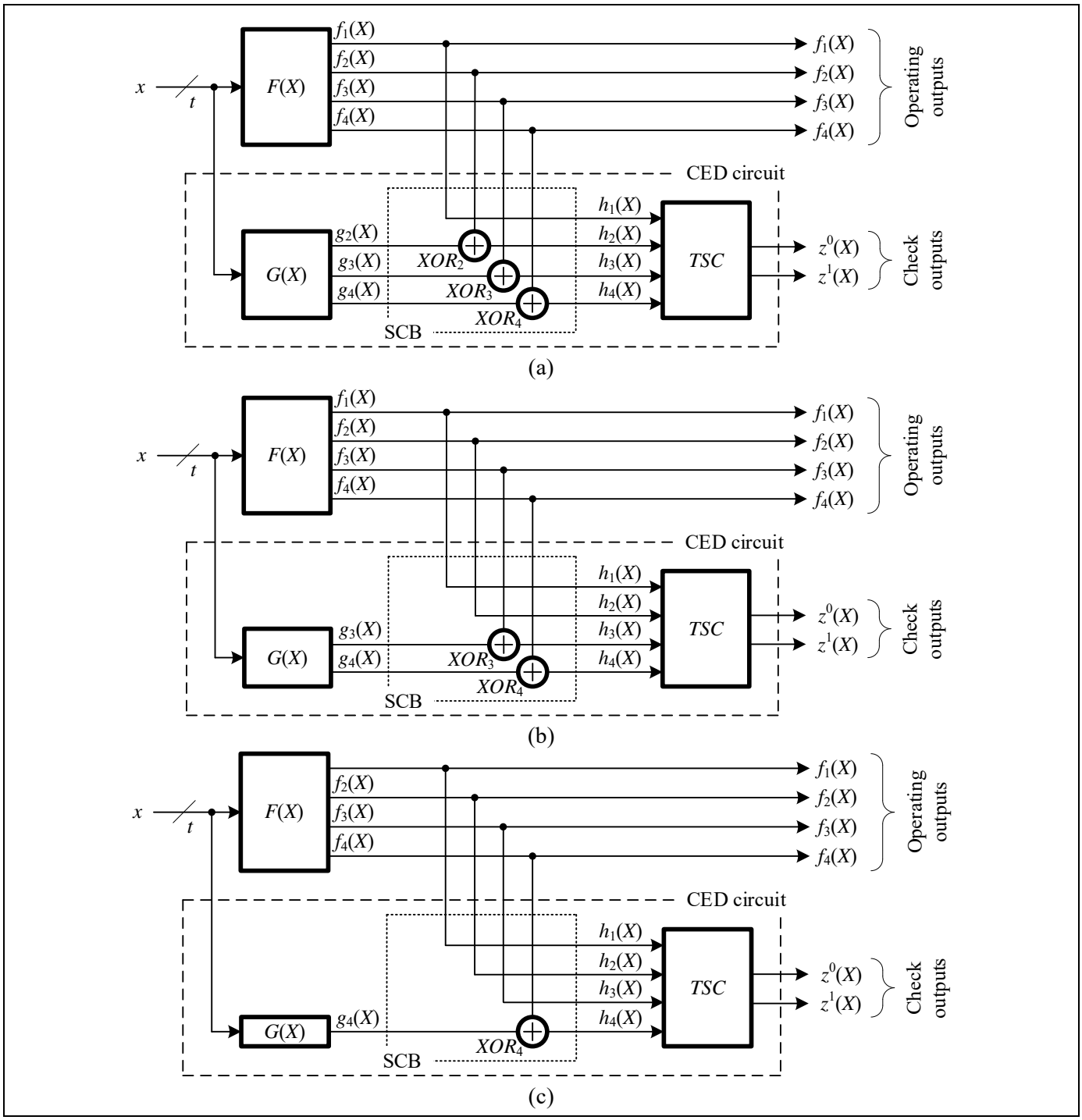

Fig. 1. Modified CED circuit structures based on BSC and the composition of 1-out-of-4 and 3-out-of-4 codes: (a) $p = 3$, (b) $p = 2$, and (c) $p = 1$.

the codewords $<f_4(X)\, f_3(X)\, f_2(X)\, f_1(X)>$ can be converted by correcting p symbols; and the exponent 2^t is the number of sets of argument values over which the values are obtained.

According to Theorem 2, the smaller the number p is, the less freedom the device designer will have when choosing the way of BSC in the SCB. But a smaller value of p also simplifies the realization of the complete self-checking property of the SCB itself.

2. ALGORITHM FOR DESIGNING CED CIRCUITS BASED ON THE COMPOSITION OF 1-OUT-OF-4 AND 3-OUT-OF-4

When designing a CED circuit based on BSC, the main task is the design of the block $G(X)$; for details, see part I of the study [1]. Let us show the design procedure of this block for each selected quadruple of the object's outputs under a different number of signals converted.

Algorithm 3.[1] *CED circuit design for a quadruple of object's outputs with a fixed number p of elements converted in the SCB:*

1. Select $p \in \{1, 2, 3, 4\}$, and identify p particular outputs to be converted and $4 - p$ outputs directly connected to the checker's inputs.

2. Order the outputs in the quadruple, placing first the $4 - p$ outputs directly connected to the checker's inputs and second the p outputs converted in the SCB.

3. Examine all sets of argument values, ranging from the set with a decimal equivalent of zero to that with a decimal equivalent of $2^t - 1$, sequentially in the lexicographic order. Analyze the values of the $(4 - p)$ functions implemented on the object's outputs directly connected to the checker's inputs. For the p outputs converted, on each set of argument values, determine a subset of codewords belonging to the composition of the constant-weight codes among which there is the codeword $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ representing the result of converting the vector $<f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)>$. Due to formula (1), for $p = 4$, we have eight codewords; for $p = 3$, four; for $p = 2$, two; and for $p = 1$, only one. In Table 1, the codewords of the composition of 1-out-of-4 and 3-out-of-4 codes are ordered by the changing values of the most significant bits; among them, $4 - p$ unchanging bits can be identified, and the appropriate codeword can be selected based on their values. When obtaining the values of the bits of the vector $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$, all codewords of the composition under consideration must be used at least once.

Table 1

The codewords of the composition of 1-out-of-4 and 3-out-of-4 codes, ordered by the changing values of the most significant bits

Codeword number	$<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$
1	0001
2	0010
3	0100
4	0111
5	1000
6	1011
7	1101
8	1110

4. Analyze the set of codewords $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ on the complete set of argument value. If at least one codeword from Table 1 is not formed, permute the outputs within the quadruple; if the result is the same, permute the outputs within the entire output set; if the result is still the same, increment the number p, thereby doubling the degree of freedom (i.e., the number of codewords selected).

5. Repeat Steps 3–6 of Algorithm 1 from part I of the study [1].

Note that at Step 3 of this algorithm, all eight codewords belonging to the given composition are used. However, by the expression (2) from part I of the study [1], this step can be modified to set the values of the bits of the vector $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ only to obtain four codewords of the given composition that will ensure complete checking by the checker. However, such a deviation from Algorithm 3 is possible only for $p = 2$, 3, or 4; for $p = 1$, the codeword $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ is uniquely determined.

For a device with $n > 4$ outputs, the CED circuit is designed using Algorithm 2; see part I of the study [1].

Unlike Algorithm 1 from part I, this algorithm may involve many iterations if it is impossible to generate the complete set of codewords for the composition of 1-out-of-4 and 3-out-of-4 codes. And the number of operations and the execution time of Algorithm 3 will both grow accordingly. In some cases, if the desired result (the values of the codewords $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$) is not obtained, the number of admissible permutations β for a given p should be specified. For example, for an object under diagnosis with many outputs, this number can be limited to β = 1000…10 000.

Example 3. Let us build a CED circuit for the discrete device from Example 1 (see part I of the study [1]) using Algorithm 3 with $p = 2$.

At Step 1 of Algorithm 3, we set $p = 2$. Assume that the outputs $f_4(X)$ and $f_3(X)$ are directly connected to the checker's inputs: $h_4(X) = f_4(X)$ and $h_3(X) = f_3(X)$. The signals from the outputs $f_2(X)$ and $f_1(X)$ are converted in the SCB: $h_2(X) = f_2(X) \oplus g_2(X)$ and $h_1(X) = f_1(X) \oplus g_1(X)$. According to Step 2 of Algorithm 3, we order the outputs as follows: $f_4(X)$, $f_3(X)$, $f_2(X)$, and $f_1(X)$. Using Step 3 of Algorithm 3 and Table 1, we set the values of the bits of the vector $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ for each set of argument values (Table 2). While filling in the rows, it is necessary to verify the formation of all codewords from Table 1; they are successfully formed. By repeating Steps 3–6 of Algorithm 1 from part I of the study [1], we find the values of the functions $g_2(X) = f_2(X) \oplus h_2(X)$ and $g_1(X) = f_1(X) \oplus h_1(X)$. We analyze the formation of check tests for XOR_2 and XOR_1 gates; these tests are formed. Next, based on the data of Table 2, the functions $g_2(X)$ and $g_1(X)$ are optimized, and the block $G(X)$ and the self-checking discrete device itself are designed in the selected element basis.

[1] The algorithms and examples are numbered sequentially, starting from part I of the study.

Table 2

The operation of a device with four outputs and the CED circuit for it

No.	The sets of argument values				The values at the outputs of the device $F(X)$				The values at the outputs of the SCB				The values at the outputs of the block $G(X)$		Test combinations of the SCB	
	x_4	x_3	x_2	x_1	$f_4(X)$	$f_3(X)$	$f_2(X)$	$f_1(X)$	$h_4(X)$	$h_3(X)$	$h_2(X)$	$h_1(X)$	$g_2(X)$	$g_1(X)$	XOR_2	XOR_1
0	0	0	0	0	1	1	0	0	1	1	1	0	1	0	01	00
1	0	0	0	1	0	0	1	0	0	0	0	1	1	1	11	01
2	0	0	1	0	1	0	1	0	1	0	1	1	1	0	11	00
3	0	0	1	1	0	0	0	0	0	0	0	1	1	0	01	00
4	0	1	0	0	0	1	0	0	0	1	1	1	1	1	01	01
5	0	1	0	1	0	0	1	0	0	0	1	0	0	0	10	00
6	0	1	1	0	0	1	0	0	0	1	0	0	1	1	01	01
7	0	1	1	1	0	0	0	1	0	0	1	0	0	0	00	10
8	1	0	0	0	1	0	1	0	1	0	0	0	0	1	10	01
9	1	0	0	1	1	0	0	0	1	0	0	0	1	0	01	00
10	1	0	1	0	0	1	1	0	0	1	0	0	1	0	11	00
11	1	0	1	1	0	0	0	1	0	0	0	1	0	0	00	10
12	1	1	0	0	1	0	1	1	1	0	0	0	1	1	11	11
13	1	1	0	1	1	0	1	0	1	0	1	1	0	1	10	01
14	1	1	1	0	1	1	1	1	1	1	0	1	1	0	11	10
15	1	1	1	1	0	1	0	1	0	1	1	1	1	0	01	10

The number of ways to select output quadruples from the complete set of object's outputs is determined as follows. Let n be the cardinality of the set of outputs of the object under diagnosis, and let the number of its subsets be $q = \left\lceil \frac{n}{4} \right\rceil$, among which $\left\lfloor \frac{n}{4} \right\rfloor$ are with non-repeating outputs and, possibly, one with some already used in the penultimate subset if $n(\text{mod}4) \neq 0$. (Here, $\lceil ... \rceil$ and $\lfloor ... \rfloor$ denote the ceiling and floor functions, respectively.) Then the first subset of outputs can be selected in C_n^4 ways; the second subset, in C_{n-4}^4 ways; …, the last subset with non-overlapping subsets of outputs, in $C^4_{n-4\left(\left\lfloor \frac{n}{4} \right\rfloor - 1\right)}$ ways. The remaining $n(\text{mod}4)$ outputs are selected uniquely. The total number of ways to select all subsets of outputs is given by

$$N = C_n^4 C_{n-4}^4 \cdot ... \cdot C^4_{n-4\left(\left\lfloor \frac{n}{4} \right\rfloor - 1\right)} = \prod_{i=1}^{i=\left\lfloor \frac{n}{4} \right\rfloor} C^4_{n-4(i-1)}. \qquad (2)$$

Within the quadruple of outputs, those converted in the SCB are selected in C_4^p ways. If all quadruples are controlled based on the same structure (Fig. 1), we have the factor

$$\gamma = \left(C_4^p\right)^{\left\lceil \frac{n}{4} \right\rceil}. \qquad (3)$$

Recall that 2^t sets of argument values are supplied to the object's inputs (see the expression (1)). Therefore, due to formulas (2) and (3), the CED circuit based on BSC and the composition of 1-out-of-4 and 3-out-of-4 codes with p outputs converted in the SCB is designed in

$$N^* = \left(2^{p-1}\right)^{2^t} \left(C_4^p\right)^{\left\lceil \frac{n}{4} \right\rceil} \prod_{i=1}^{i=\left\lfloor \frac{n}{4} \right\rfloor} C^4_{n-4(i-1)} \qquad (4)$$

possible ways.

For example, for $n = 17$, $t = 4$, and $p = 2$, we have $N^* = 546\,335\,889\,555\,456\,000$. In practice, this number of ways to design the CED circuit cannot be considered exhaustively. Therefore, the admissible number of iterations of Algorithm 3 should be limited, e.g., by the number $\beta = 1000…10\,000$ (see the discussion above). Next, it is necessary to fix a choice criterion (e.g., the minimum value of the structural redundancy index) and select the corresponding implementation method for the self-checking discrete device.

By increasing the value of p, one can always achieve the desired result, provided this is (in principle) possible for the object under diagnosis. The checkability conditions of SCB gates and the checker were provided in [8]. They are related to the presence

of ones and zeros for each of the functions implemented at the outputs of the object under diagnosis. Furthermore, we emphasize here that the above algorithm allows organizing computation control within a subset of four outputs and is in no way oriented toward finding controllable outputs among their complete set. This problem is not addressed in the paper; the interested reader can be recommended some publications on the search for controllable subsets of outputs with error detection using uniform divisible codes, on structural modifications of objects under diagnosis, and the controllable design of such objects [9–14]. Similar algorithms can also be formulated for the composition of 1-out-of-4 and 3-out-of-4 codes, considering the specifics of error detection in their codewords.

3. EXPERIMENTAL RESULTS

The primary goal of decreasing the number of object's signals converted in a CED circuit is to reduce the implementation complexity of self-checking discrete devices and to simplify the process of ensuring the complete self-checking property of a CED circuit. Therefore, we conducted experiments using the same test combinational circuits (benchmarks) from the MCNC Benchmarks [15, 16] as in part I of the study [1]. The experimental methodology remained the same—CED circuits were designed using the developed method. In addition, to compare the results, we used the previous data on the implementation complexity of self-checking discrete devices with CED circuits built by the duplication method and the two well-known methods based on BSC and the use of 1-out-of-4 and 2-out-of-4 codes.

The method based on BSC and the 1-out-of-4 code [17] converts three signals from the object under diagnosis into four ones by the formulas

$$\begin{cases} g_4(X)=0 \\ g_3(X)=\overline{f_4(X)}\,\overline{f_3(X)}f_2(X) \\ \quad \vee f_3(X)\big(f_4(X)\vee f_1(X)\big) \\ g_2(X)=\overline{f_4(X)}f_3(X)\overline{f_2(X)}f_1(X) \\ \quad \vee f_2(X)\Big(f_4(X)\vee \overline{f_3(X)}\vee \overline{f_1(X)}\Big) \\ g_1(X)=\overline{f_4(X)}\,\overline{f_3(X)}\,\overline{f_2(X)}\,\overline{f_1(X)} \\ \quad \vee f_1(X)\big(f_4(X)\vee f_3(X)\vee f_2(X)\big). \end{cases} \quad (5)$$

Note that the outputs in this paper are numbered differently from those in [17]: from left to right, following the numbering convention for bits in code vectors. Therefore, the numbering of all functions in system (5) was changed compared to the source. For complete matching with the work [17], the function indices should be replaced as follows: 4 → 1, 3 → 2, 2 → 3, and 1 → 4. Earlier methods, such as the one from [6], were not considered: the relationships between the output values of the object under diagnosis and the block $G(X)$ established therein do not allow for a complete check of one element converted in the SCB. It is necessary to analyze the values of the functions computed by the object under diagnosis for each set of argument values, with the selection of the codeword $\langle h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)\rangle$ obtained by the conversion.

The method based on BSC and the 2-out-of-4 code [7] converts two signals from the object under diagnosis into four ones by the formulas

$$\begin{cases} g_4(X)=0 \\ g_3(X)=0 \\ g_2(X)=f_1(X)\sim f_3(X) \\ g_1(X)=f_2(X)\sim f_4(X). \end{cases} \quad (6)$$

Tables 3–5 present the results of experiments using the set of benchmarks. As in part I of the study [1], the indicators $L_{F(X)}$ and $L_{G(X)}$ characterize the areas occupied by the device on a chip when designed in the *stdcell2_2.genlib* library; L_D, $L_{1/4+3/4}$, $L_{1/4*}$[2], and $L_{2/4}$ are the areas occupied on a chip by self-checking discrete devices with the CED circuit based on duplication, the novel method proposed in this article, and the methods from the works [17] and [7], respectively. The following relative indicators were calculated for comparing the three methods with each other:

$$\mu=\frac{L_{1/4+3/4}}{L_D}\cdot 100\%, \quad (7)$$

$$\theta=\frac{L_{1/4+3/4}}{L_{1/4*}}\cdot 100\%, \quad (8)$$

$$\zeta=\frac{L_{1/4+3/4}}{L_{2/4}}\cdot 100\%. \quad (9)$$

Also, for a convenient efficiency analysis of the novel methods, the following indicators (the relative gain in complexity) were calculated:

$$\Delta\mu=\left(1-\frac{L_{1/4+3/4}}{L_D}\right)\cdot 100\%, \quad (10)$$

$$\Delta\theta=\left(1-\frac{L_{1/4+3/4}}{L_{1/4*}}\right)\cdot 100\%, \quad (11)$$

$$\Delta\zeta=\left(1-\frac{L_{1/4+3/4}}{L_{2/4}}\right)\cdot 100\%. \quad (12)$$

[2] The asterisk (*) is used to indicate to the reader the difference between the methods for designing CED circuits based on BSC using the 1-out-of-4 code, which are compared in parts I and II of this study.

Table 3

Experimental results. The values of the effective area occupied by devices on the chip

Benchmark	n	q	$L_{F(X)}$	L_D	$p = 4$		$p = 3$		$p = 2$		$p = 1$		The method from [17]		The method from [17]	
					$L_{G(X)}$	$L_{1/4+3/4}$	$L_{G(X)}$	$L_{1/4+3/4}$	$L_{G(X)}$	$L_{1/4+3/4}$	$L_{G(X)}$	$L_{1/4+3/4}$	$L_{G(X)}$	$L_{1/4*}$	$L_{G(X)}$	$L_{2/4}$
rd84	4	1	4 912	10 464	4 056	9 240	2 528	7 672	1 592	6 696	1 576	6 640	4 072	9 320	4 416	9 600
sqrt8	4	1	1 160	2 960	1 280	2 712	1 056	2 448	1 112	2 464	752	2 064	1 176	2 672	1 088	2 520
sao2	4	1	2 992	6 624	3 752	7 016	2 024	5 248	1 520	4 704	968	4 112	2 056	5 424	1720	4 984
dist	5	2	6 968	14 784	8 080	15 784	7 584	15 208	5 344	12 888	3 880	11 344	5 080	12 616	4 336	11 768
newcwp	5	2	440	1 728	784	1 960	672	1 768	408	1 424	376	1 312	504	1 512	232	1 136
root	5	2	3 496	7 840	3 432	7 664	3 120	7 272	2 736	6 808	1 496	5 488	2 136	6 200	2 240	6 200
max512	6	2	9 632	20 320	7 944	18 312	7 168	17 456	5 776	15 984	3 560	13 688	6 648	17 040	6 136	16 424
dc1	7	2	976	3 216	680	2 392	600	2 232	464	2 016	280	1 752	592	2 520	552	2 376
dekoder	7	2	736	2 736	688	2 160	496	1 888	432	1 744	280	1 512	472	2 160	528	2 112
wim	7	2	712	2 688	1 000	2 448	640	2 008	552	1 840	336	1 544	408	2 072	544	2 104
newapla2	7	2	600	2 464	1 072	2 408	656	1 912	456	1 632	152	1 248	448	2 000	448	1 896
dc2	7	2	2 424	6 112	3 184	6 344	2 864	5 944	2 664	5 664	2 128	5 048	2 024	5 400	1 512	4 784
newbyte	8	2	592	2 656	808	2 136	504	1 752	376	1 544	96	1 184	264	2 040	216	1 544
mlp4	8	2	7 224	15 920	7 232	15 192	6 352	14 232	5 120	12 920	2 744	10 464	5 752	14 160	4 608	12 568
f51m	8	2	2 272	6 016	5 144	8 152	4 504	7 432	2 680	5 528	2 040	4 808	4 208	7 664	2 176	5 184
inc	9	3	2 376	6 432	2 480	6 056	1 992	5 448	1 928	5 264	1 112	4 328	1 584	5 336	1 440	4 744
dk27	9	3	528	2 736	2 256	3 984	1 240	2 848	920	2 408	960	2 328	720	2 624	544	2 000
newcpla2	10	3	1 896	5 680	3 800	6 896	1 616	4 592	1 424	4 280	568	3 304	1 408	4 872	1 192	4 208
sqr6	12	3	2 648	7 600	2 896	6 744	2 056	5 784	1 472	5 080	1 208	4 696	2 016	6 816	1 784	5 632
m1	12	3	3 064	8 432	1 912	6 176	1 104	5 248	736	4 760	440	4 344	1 144	6 440	960	5 200
p82	14	4	2 368	7 456	2 304	6 336	1 736	5 608	1 496	5 208	736	4 288	1 480	6 464	1 112	5 064
m2	16	4	10 096	23328	4 656	16 416	4 064	15 664	2 888	14 328	1 464	12 744	3 400	17 016	2 536	14 272
m3	16	4	13 464	30 064	5 952	21 080	4 624	19 592	3 168	17 976	1 616	16 264	4 160	21 144	3 280	18 384
m4	16	4	18 704	40 544	8 952	29 320	7 032	27 240	5 392	25 440	3 064	22 952	7 224	29 448	4 920	25 288
tms	16	4	6 784	16 704	5 096	13 544	3 416	11 704	2 680	10 808	1 752	9 720	3 272	13 552	2 696	11 144
newcpla1	16	4	2 520	8 176	4 888	9 072	2 376	6 400	2 144	6 008	1 680	5 384	3 024	9 064	2 208	6 392
newxcpla1	23	6	3 760	12 112	5 920	12 272	3 008	9 120	3 288	9 160	1 856	7 488	2 672	12 056	2 352	8 816
max128	24	6	20 192	45 184	9 232	32 016	2 216	24 760	1 752	24 056	888	22 952	1 640	28 648	1 352	24 136

Table 4

Experimental results. The values of the indicators μ, θ, and ζ

Benchmark	μ				θ				ζ			
	$p = 4$	$p = 3$	$p = 2$	$p = 1$	$p = 4$	$p = 3$	$p = 2$	$p = 1$	$p = 4$	$p = 3$	$p = 2$	$p = 1$
rd84	88.303	73.318	63.991	63.456	99.142	82.318	71.845	71.245	96.25	79.917	69.75	69.167
sqrt8	91.622	82.703	83.243	69.73	101.497	91.617	92.216	77.246	107.619	97.143	97.778	81.905
sao2	105.918	79.227	71.014	62.077	129.351	96.755	86.726	75.811	140.77	105.297	94.382	82.504
dist	106.764	102.868	87.175	76.732	125.111	120.545	102.156	89.918	134.126	129.232	109.517	96.397
newcwp	113.426	102.315	82.407	75.926	129.63	116.931	94.18	86.772	172.535	155.634	125.352	115.493
root	97.755	92.755	86.837	70	123.613	117.29	109.806	88.516	123.613	117.29	109.806	88.516
max512	90.118	85.906	78.661	67.362	107.465	102.441	93.803	80.329	111.495	106.283	97.321	83.341
dc1	74.378	69.403	62.687	54.478	94.921	88.571	80	69.524	100.673	93.939	84.848	73.737
dekoder	78.947	69.006	63.743	55.263	100	87.407	80.741	70	102.273	89.394	82.576	71.591
wim	91.071	74.702	68.452	57.44	118.147	96.911	88.803	74.517	116.35	95.437	87.452	73.384
newapla2	97.727	77.597	66.234	50.649	120.4	95.6	81.6	62.4	127.004	100.844	86.076	65.823
dc2	103.796	97.251	92.67	82.592	117.481	110.074	104.889	93.481	132.609	124.247	118.395	105.518
newbyte	80.422	65.964	58.133	44.578	104.706	85.882	75.686	58.039	138.342	113.472	100	76.684
mlp4	95.427	89.397	81.156	65.729	107.288	100.508	91.243	73.898	120.878	113.24	102.801	83.259
f51m	135.505	123.537	91.888	79.92	106.367	96.973	72.129	62.735	157.253	143.364	106.636	92.747
inc	94.154	84.701	81.841	67.289	113.493	102.099	98.651	81.109	127.656	114.84	110.961	91.231
dk27	145.614	104.094	88.012	85.088	151.829	108.537	91.768	88.72	199.2	142.4	120.4	116.4
newcpla2	121.408	80.845	75.352	58.169	141.544	94.253	87.849	67.816	163.878	109.125	101.711	78.517
sqr6	88.737	76.105	66.842	61.789	98.944	84.859	74.531	68.897	119.744	102.699	90.199	83.381
m1	73.245	62.239	56.452	51.518	95.901	81.491	73.913	67.453	118.769	100.923	91.538	83.538
p82	84.979	75.215	69.85	57.511	98.02	86.757	80.569	66.337	125.118	110.742	102.844	84.676
m2	70.37	67.147	61.42	54.63	96.474	92.055	84.203	74.894	115.022	109.753	100.392	89.294
m3	70.117	65.168	59.792	54.098	99.697	92.66	85.017	76.92	114.665	106.571	97.781	88.468
m4	72.316	67.186	62.747	56.61	99.565	92.502	86.39	77.941	115.944	107.719	100.601	90.762
tms	81.082	70.067	64.703	58.19	99.941	86.364	79.752	71.724	121.536	105.025	96.985	87.222
newcpla1	110.959	78.278	73.483	65.851	100.088	70.609	66.284	59.4	141.927	100.125	93.992	84.23
newxcpla1	101.321	75.297	75.627	61.823	101.792	75.647	75.979	62.11	139.201	103.448	103.902	84.936
max128	70.857	54.798	53.24	50.797	111.756	86.428	83.971	80.117	132.648	102.585	99.669	95.094

Table 5

Experimental results. The values of the indicators Δμ, Δθ, and Δζ

Benchmark	Δμ				Δθ				Δζ			
	$p = 4$	$p = 3$	$p = 2$	$p = 1$	$p = 4$	$p = 3$	$p = 2$	$p = 1$	$p = 4$	$p = 3$	$p = 2$	$p = 1$
rd84	11.697	26.682	36.009	36.544	0.858	17.682	28.155	28.755	3.75	20.083	30.25	30.833
sqrt8	8.378	17.297	16.757	30.27	–1.497	8.383	7.784	22.754	–7.619	2.857	2.222	18.095
sao2	–5.918	20.773	28.986	37.923	–29.351	3.245	13.274	24.189	–40.77	–5.297	5.618	17.496
dist	–6.764	–2.868	12.825	23.268	–25.111	–20.545	–2.156	10.082	–34.126	–29.232	–9.517	3.603
newcwp	–13.426	–2.315	17.593	24.074	–29.63	–16.931	5.82	13.228	–72.535	–55.634	–25.352	–15.493
root	2.245	7.245	13.163	30	–23.613	–17.29	–9.806	11.484	–23.613	–17.29	–9.806	11.484
max512	9.882	14.094	21.339	32.638	–7.465	–2.441	6.197	19.671	–11.495	–6.283	2.679	16.659
dc1	25.622	30.597	37.313	45.522	5.079	11.429	20	30.476	–0.673	6.061	15.152	26.263
dekoder	21.053	30.994	36.257	44.737	0	12.593	19.259	30	–2.273	10.606	17.424	28.409
wim	8.929	25.298	31.548	42.56	–18.147	3.089	11.197	25.483	–16.35	4.563	12.548	26.616
newapla2	2.273	22.403	33.766	49.351	–20.4	4.4	18.4	37.6	–27.004	–0.844	13.924	34.177
dc2	–3.796	2.749	7.33	17.408	–17.481	–10.074	–4.889	6.519	–32.609	–24.247	–18.395	–5.518
newbyte	19.578	34.036	41.867	55.422	–4.706	14.118	24.314	41.961	–38.342	–13.472	0	23.316
mlp4	4.573	10.603	18.844	34.271	–7.288	–0.508	8.757	26.102	–20.878	–13.24	–2.801	16.741
f51m	–35.505	–23.537	8.112	20.08	–6.367	3.027	27.871	37.265	–57.253	–43.364	–6.636	7.253
inc	5.846	15.299	18.159	32.711	–13.493	–2.099	1.349	18.891	–27.656	–14.84	–10.961	8.769
dk27	–45.614	–4.094	11.988	14.912	–51.829	–8.537	8.232	11.28	–99.2	–42.4	–20.4	–16.4
newcpla2	–21.408	19.155	24.648	41.831	–41.544	5.747	12.151	32.184	–63.878	–9.125	–1.711	21.483
sqr6	11.263	23.895	33.158	38.211	1.056	15.141	25.469	31.103	–19.744	–2.699	9.801	16.619
m1	26.755	37.761	43.548	48.482	4.099	18.509	26.087	32.547	–18.769	–0.923	8.462	16.462
p82	15.021	24.785	30.15	42.489	1.98	13.243	19.431	33.663	–25.118	–10.742	–2.844	15.324
m2	29.63	32.853	38.58	45.37	3.526	7.945	15.797	25.106	–15.022	–9.753	–0.392	10.706
m3	29.883	34.832	40.208	45.902	0.303	7.34	14.983	23.08	–14.665	–6.571	2.219	11.532
m4	27.684	32.814	37.253	43.39	0.435	7.498	13.61	22.059	–15.944	–7.719	–0.601	9.238
tms	18.918	29.933	35.297	41.81	0.059	13.636	20.248	28.276	–21.536	–5.025	3.015	12.778
newcpla1	–10.959	21.722	26.517	34.149	–0.088	29.391	33.716	40.6	–41.927	–0.125	6.008	15.77
newxcpla1	–1.321	24.703	24.373	38.177	–1.792	24.353	24.021	37.89	–39.201	–3.448	–3.902	15.064
max128	29.143	45.202	46.76	49.203	–11.756	13.572	16.029	19.883	–32.648	–2.585	0.331	4.906

The bar charts in Figs. 2–5 provide the values of the indicators Δμ, Δθ, and Δζ (10)–(12) computed for the novel method with different values of p.

As expected, for all the benchmarks considered, a smaller value of p increases the gain in terms of the implementation complexity of the device; see the bar chart in Fig. 2. For the vast majority of benchmarks, a gain is achieved compared to duplication for $p = 4$ and $p = 3$; in the cases $p = 2$ and $p = 1$, a gain is achieved for absolutely all benchmarks. The best results for the indicator Δμ are achieved for $p = 1$. However, even for $p = 2$, there is also a significant improvement compared to duplication.

The implementation complexity indicators seem logical to decrease as the number of corrected signals from the object under diagnosis decreases. Therefore, it is reasonable to compare the results not only with duplication but also with the methods involving fewer conversion devices than the total number of outputs in the quadruple. The methods described in [17] and [7] correct $p = 3$ and $p = 2$ signals, respectively.

The bar charts in Figs. 3–6 show the indicators Δμ, Δθ, and Δζ for different benchmarks and different values of p. They can be used to judge improvements, depending on p, compared to duplication and other methods under consideration.

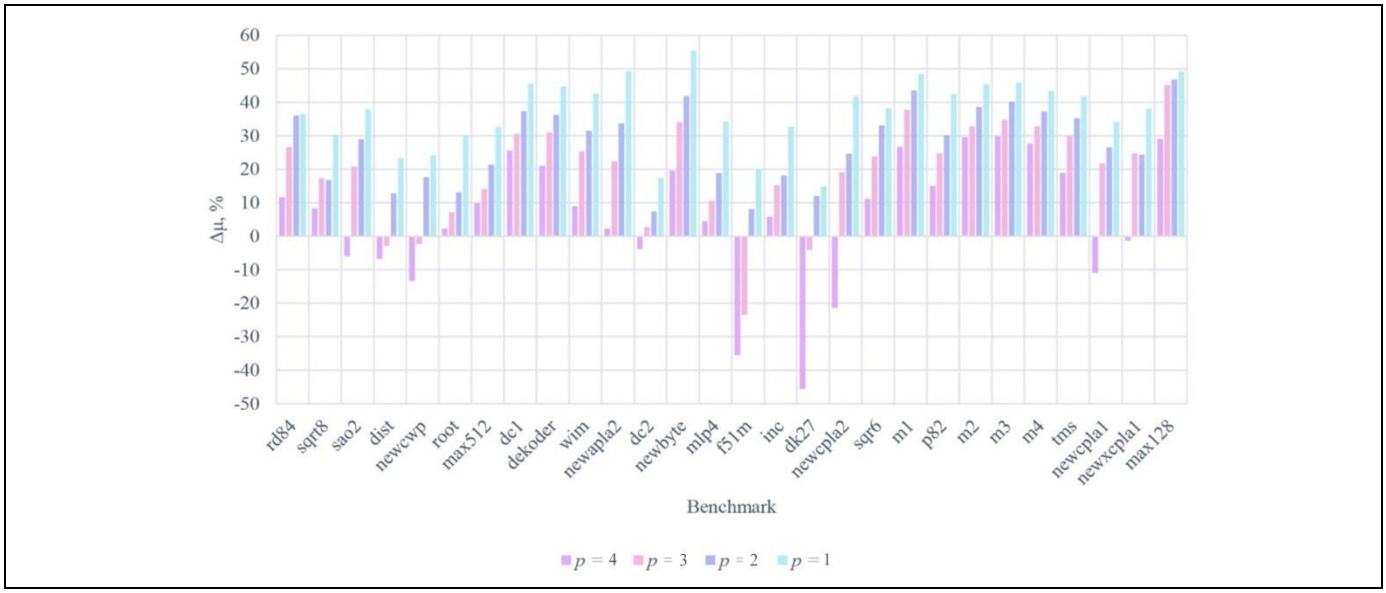

Fig. 2. The values of Δμ for different benchmarks.

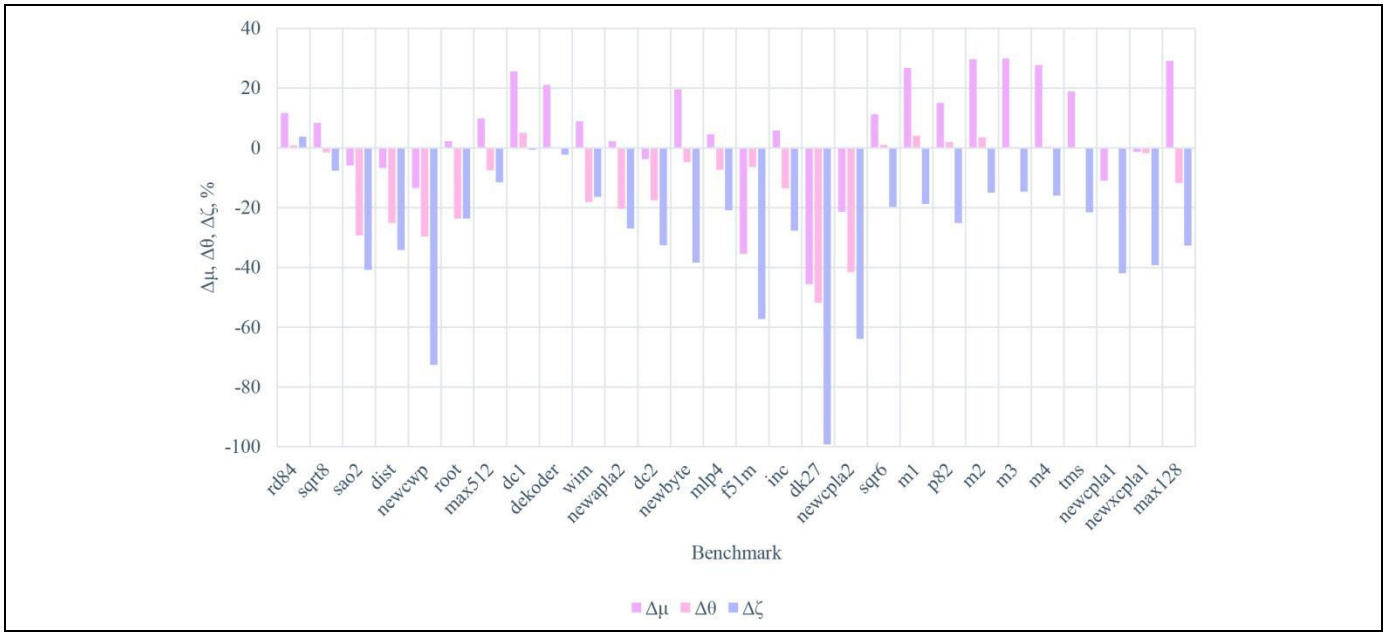

Fig. 3. The values of Δμ, Δθ, and Δζ for different benchmarks and $p = 4$.

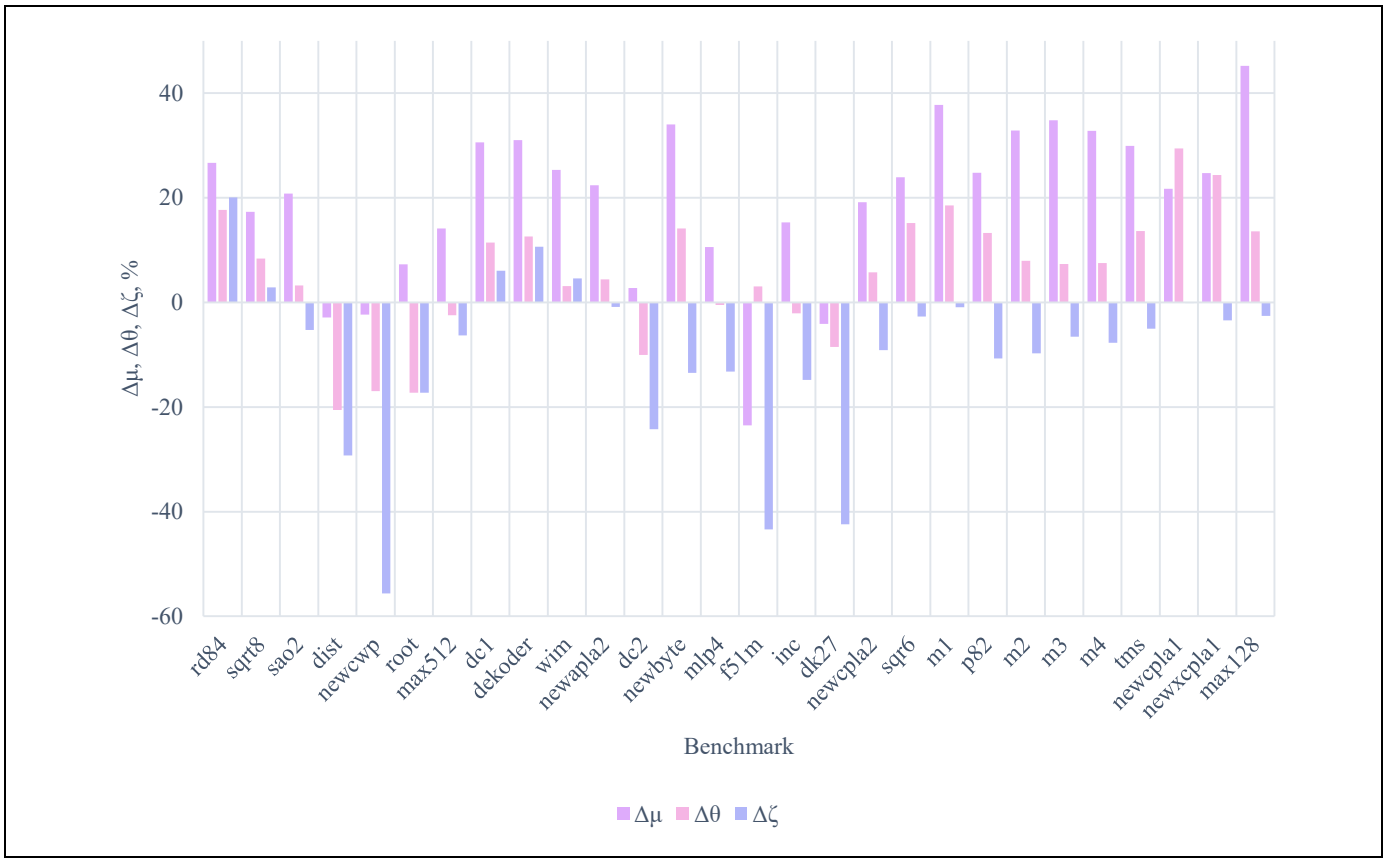

Fig. 4. The values of the parameters Δμ, Δθ, and Δζ for different benchmarks and $p = 3$.

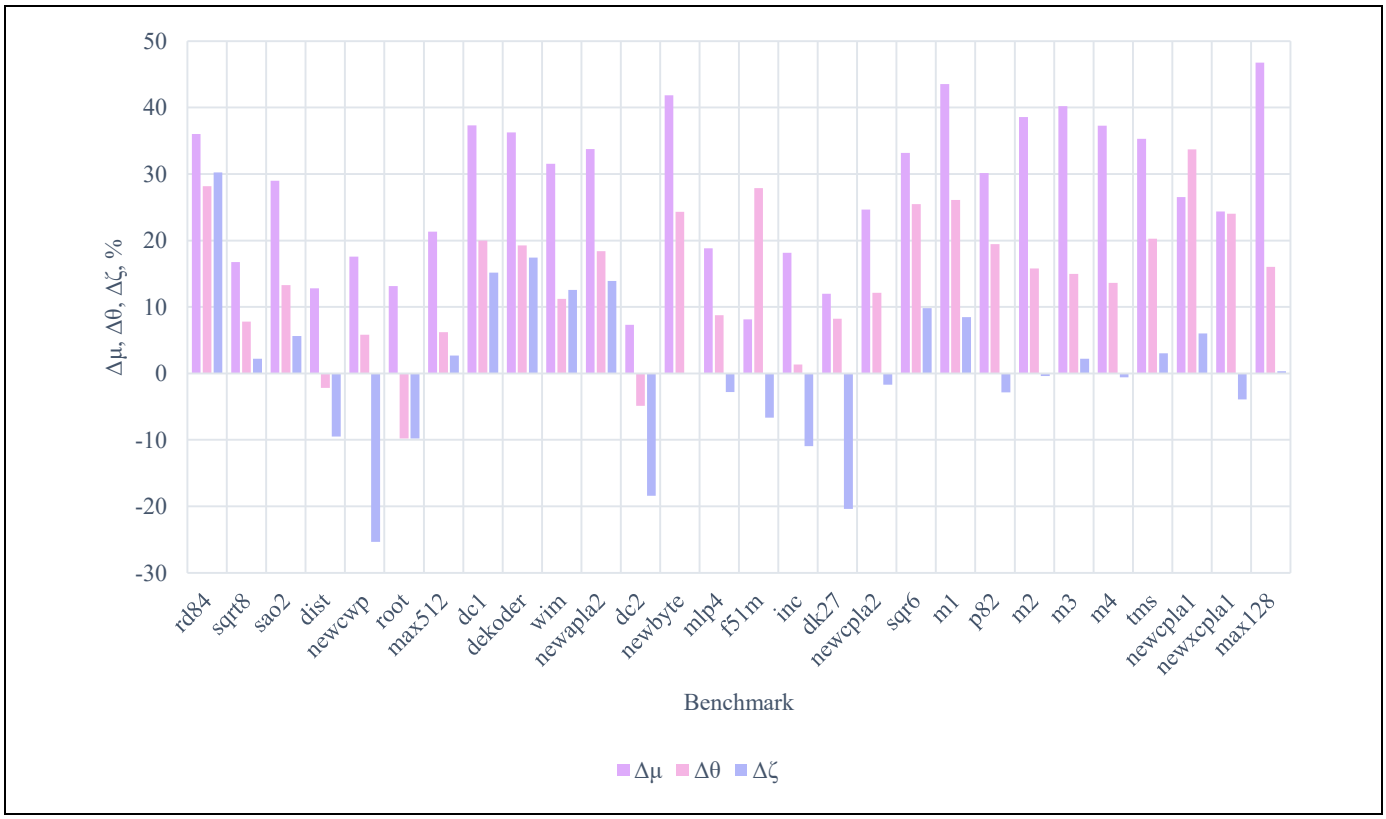

Fig. 5. The values of Δμ, Δθ, and Δζ for different benchmarks and $p = 2$.

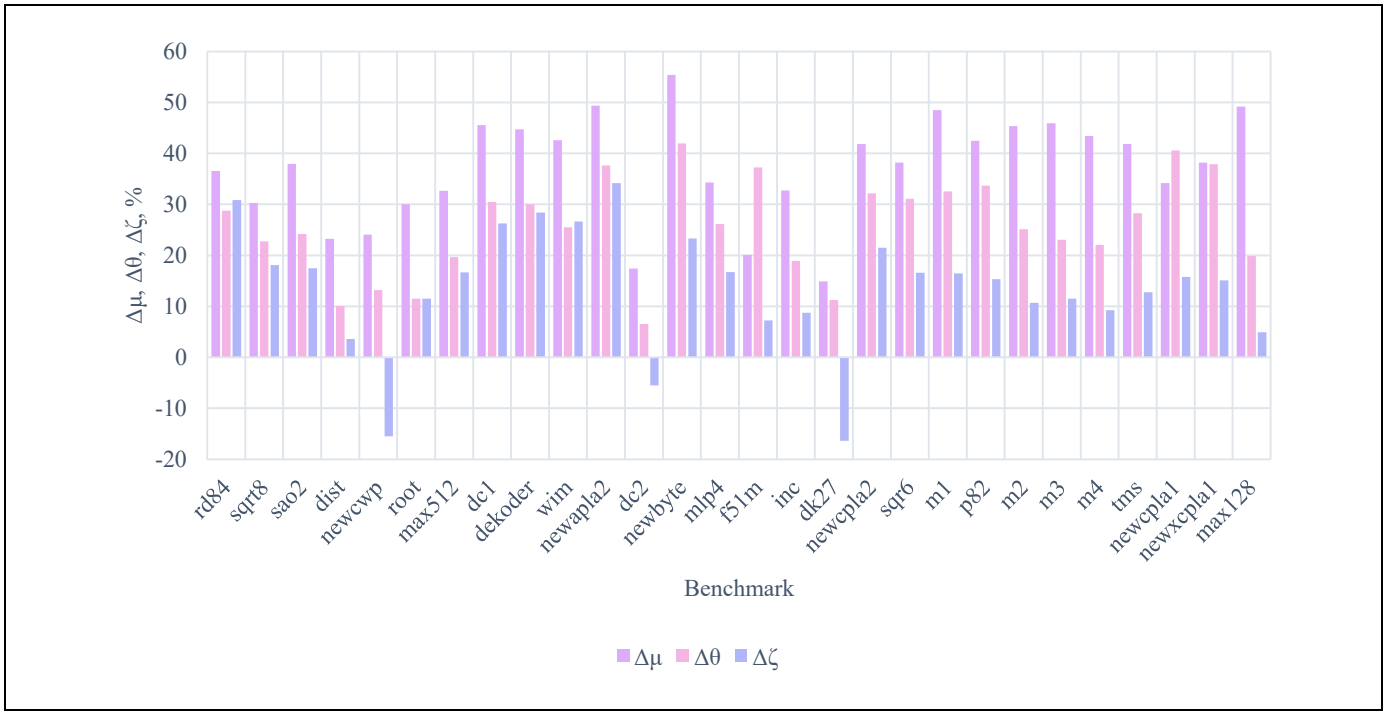

Fig. 6. The values of Δμ, Δθ, and Δζ for different benchmarks and $p = 1$.

Based on Fig. 3, for $p = 4$, the novel method is advantageous over duplication only. At the same time, for a small number of benchmarks, an improvement is achieved compared to the method described in [17].

The novel method yields slightly better results in the case $p = 3$ (Fig. 4). Here, for the overwhelming majority of benchmarks, an improvement is achieved both compared to duplication and compared to the method from [17]. However, the novel method performs much worse than the one from [7], resulting in an improvement only for a few benchmarks.

According to Figs. 5 and 6, the maximum gains over all methods considered are obtained for $p = 2$ and $p = 1$ (almost for all benchmarks). The results in the case $p = 1$ are, naturally, better than in the case $p = 2$.

The experiments clearly demonstrated the advantages and drawbacks of the novel method for designing self-checking discrete devices compared to the well-known methods for different values of p.

CONCLUSIONS

The methods for designing self-checking discrete devices based on BSC and the composition of 1-out-of-4 and 3-out-of-4 codes developed in this paper yield less redundant structures compared to the well-known counterparts. In addition, it is possible to solve optimization problems for a given object under diagnosis: the novel design methods imply numerous ways to implement CED circuits for the same object. As the number p is reduced, the number of ways to design the CED circuit for a selected quadruple of outputs decreases accordingly; however, given possible permutations of outputs within the quadruples and among all outputs of the object under diagnosis, the latter number remains considerable and significantly depends on the number t of object's inputs. For instance, the method involving a different number of signals converted can also be applied jointly for the same object under diagnosis with $n > 4$ outputs.

In contrast to the well-known design approaches, the composition of 1-out-of-4 and 3-out-of-4 codes allows building a CED circuit based on BSC by converting only one signal from the quadruple of the object's outputs. This cannot be achieved by applying individual constant-weight codes of the types 1-out-of-3, 1-out-of-4, or 2-out-of-4, which have been thoroughly described in the literature. Thus, the range of alternatives for designing a self-checking discrete device is significantly expanded compared to the existing methods.

The novel methods have the following advantage: by varying the value of p across a large number of ways to design CED circuits, one can affect the implementation complexity of self-checking discrete devices. As a drawback, we recall the statement from the Conclusions of part I: it is necessary to analyze the features of error propagation at the object's outputs to detect any faults when the values of functions computed by the object under diagnosis deviate across all sets of argument values where errors occur, or at least on one of them (depending on the task at hand).

Further research may focus on other methods for using the composition of 1-out-of-4 and 3-out-of-4 codes in the design of self-checking discrete devices, efficient algorithms for setting the values of functions computed by the block $G(X)$, the generalization of the obtained results to other compositions of constant-weight codes, and the practical implementation features of the novel methods for designing self-checking discrete devices on a different element base in various application fields.

Using the composition of 1-out-of-4 and 3-out-of-4 codes in the design of self-checking discrete devices based on BSC is a promising approach to the implementation of highly reliable and secure discrete devices.

REFERENCES

1. Efanov, D.V. and Yelina, Y.I., Design of Self-Checking Discrete Devices Based on Boolean Signals Correction and Composition of Constant-Weight Codes of the "1-Out-Of-4" and "3-Out-Of-4" Types. Part I: The Design Method with Conversion of All Signals from the Object under Diagnosis, *Control Sciences*, 2026, no. 1, pp. 30–40.
2. Goessel, M. and Graf, S., *Error Detection Circuits*, London: McGraw-Hill, 1994.
3. Efanov, D.V., Using the "1-Out-Of-4" Constant-Weight Code in the Synthesis of Self-Checking Concurrent Error-Detection Circuit Based on Boolean Signal Correction, *Tomsk State University Journal of Control and Computer Science*, 2025, no. 72, pp. 114–133. DOI: 10.17223/19988605/72/12 (In Russian.)
4. Bibilo, P.N., *Binarnye diagrammy reshenii v logicheskom proektirovanii* (Binary Decision Diagrams in Logical Design), Moscow: LENAND, 2024. (In Russian.)
5. Gessel, M., Morozov, A.V., Sapozhnikov, V.V., and Sapozhnikov, Vl.V., Logic Complement, a New Method of Checking the Combinational Circuits, *Automation and Remote Control*, 2003, vol. 64, no. 1, pp. 153–161.
6. Goessel, M., Morozov, A.V., Sapozhnikov, V.V., and Sapozhnikov, Vl.V., Checking Combinational Circuits by the Method of Logic Complement, *Automation and Remote Control*, 2005, vol. 66, no. 8, pp. 1336–1346.
7. Sapozhnikov, V.V., Sapozhnikov, Vl.V., and Efanov, D.V., Design of Self-Checking Concurrent Error Detection Systems Based on "2-Out-Of-4" Constant-Weight Code, *Control Sciences*, 2017, no. 1, pp. 57–64. (In Russian.)
8. Efanov, D.V. and Yelina, Y.I., Design of Self-Checking Digital Devices with Boolean Signals Correction Using Weight-Based Bose–Lin Codes, *Control Sciences*, 2024, no. 4, pp. 22–36. DOI: http://doi.org/10.25728/cs.2024.4.3 (In Russian.)
9. Slabakov, E.V. and Sogomonyan, E.S., Design of Totally Self-Checking Combinational Networks Using m-Out-Of-n Code, *Automation and Remote Control*, 1981, vol. 41, 1326–1333.
10. Sogomonyan, E.S. and Slabakov, E.V., *Samoproveryaemye ustroistva i otkazoustoichivye sistemy* (Self-Checking Devices and Fault-Tolerant Systems), Moscow: Radio i Svyaz', 1989. (In Russian.)
11. Goessel, M., Morozov, A.A., Sapozhnikov, V.V., and Sapozhnikov, Vl.V., Study of Combinational Self-Checking Devices with Independent and Monotonic Independent Outputs, *Automation and Remote Control*, 1997, vol. 58, no. 2, pt. 2, pp. 299–309.
12. Matrosova, A., Levin, I., and Ostanin, S.A., Self-Checking Synchronous FSM Network Design with Low Overhead, *VLSI Design*, 2000, vol. 11, no. 1, pp. 47–58. DOI: 10.1155/2000/46578
13. Efanov, D.V., Sapozhnikov, V.V., and Sapozhnikov, Vl.V., Organization of a Fully Self-Checking Structure of a Combinational Device Based on Searching for Groups of Symmetrically Independent Outputs, *Automatic Control and Computer Sciences*, 2020, vol. 54, no. 4, pp. 279–290. DOI: 10.3103/S0146411620040045
14. Efanov, D.V., Synthesis of Self-Checking Computing Devices Based on a Complete System of Special Groups of the Diagnostic Object Outputs, *Journal of Instrument Engineering*, 2023, vol. 66, no. 5, pp. 355–372. DOI: 10.17586/0021-3454-2023-66-5-355-372 (In Russian.)
15. McElvain, K., *IWLS'93 Benchmark Set. Version 4.0*. Distributed as a Part of IWLS'93 Benchmark Set, 1993.
16. *Collection of Digital Design Benchmarks*. URL: https://ddd.fit.cvut.cz/www/prj/Benchmarks/ (Accessed August 26, 2025.)
17. Sapozhnikov, V.V., Sapozhnikov, Vl.V., Efanov, D.V., and Pivovarov, D.V., Boolean Complement Method Based on Constant-Weight Code "1-Out-Of-4" for Formation of Totally Self-Checking Concurrent Error Detection Systems, *Electronic Modeling*, 2017, vol. 39, no. 2, pp. 15–34. (In Russian.)

This paper was recommended for publication by L.Yu. Filimonyuk, a member of the Editorial Board.

Received September 14, 2025, and revised November 27, 2025. Accepted November 27, 2025.

Author information

Efanov, Dmitry Viktorovich. Dr. Sci. (Eng.), Peter the Great Saint Petersburg Polytechnic University, St. Petersburg, Russia; Solomenko Institute of Transport Problems, Russian Academy of Sciences, St. Petersburg, Russia
✉ TrES-4b@yandex.ru
ORCID iD: https://orcid.org/0000-0002-4563-6411

Yelina, Yeseniya Igorevna. Postgraduate, Peter the Great Saint Petersburg Polytechnic University, St. Petersburg, Russia
✉ eseniya-elina@mail.ru
ORCID iD: https://orcid.org/0009-0004-4167-3591

Cite this paper

Efanov, D.V. and Yelina, Y.I., Design of Self-Checking Discrete Devices Based on Boolean Signals Correction and Composition of Constant-Weight Codes of the "1-Out-Of-4" and "3-Out-Of-4" Types. Part II: The Design Method with Conversion of Some Signals from the Object under Diagnosis. *Control Sciences* 2, 68–80 (2026).

Original Russian Text © Efanov, D.V., Yelina, Y.I., 2026, published in *Problemy Upravleniya*, 2026, no. 2, pp. 77–92.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com