

Article

Novel Methods for Synthesizing Self-Checking Combinational Circuits by Means of Boolean Signal Correction and Polynomial Codes

Dmitry V. Efanov ^{1,*} , Ruslan B. Abdullaev ² , Dmitry G. Plotnikov ^{1,*}, Marina V. Bolsunovskaya ¹, Alexey S. Odoevsky ¹ and Georgy S. Vasilyanov ¹

¹ Laboratory “Industrial Stream Data Processing Systems”, Higher School of Mechanical Engineering, Material and Transport Institute, Peter the Great St. Petersburg Polytechnic University, St. Petersburg 195251, Russia; bolsun_mv@spbstu.ru (M.V.B.); odoevskij_as@spbstu.ru (A.S.O.); vasil_gs@spbstu.ru (G.S.V.)

² “Automation and Remote Control” Department, Tashkent State Transport University, Tashkent 100167, Uzbekistan; ruslan_0507@mail.ru

* Correspondence: efanov_dv@spbstu.ru (D.V.E.); plotnikov_dg@spbstu.ru (D.G.P.); Tel.: +7-(911)-709-2164 (D.V.E.)

Abstract: This paper proposes the use of a polynomial code for synthesizing self-checking digital devices. The code is chosen for its error detection characteristics in data symbols and is used for Boolean signals correction in embedded control circuits. In practice, it is possible to equip the device with the ability to detect faults. In contrast to the approaches found in the world literature to solve this problem, this proposal suggests identifying groups of structurally independent outputs to distinguish between convertible and non-convertible outputs of the diagnosed block in the embedded control circuit. The only outputs that can be converted are those that are used as checking symbols for the polynomial code in the embedded control circuit. The other functions remain unchanged. The polynomial codes are used to select them. The authors present algorithms for synthesizing fault detection devices using the proposed approach.

Keywords: circuit synthesis; embedded control circuit; concurrent error-detection; self-checking devices; Boolean signal correction; polynomial codes



Citation: Efanov, D.V.; Abdullaev, R.B.; Plotnikov, D.G.; Bolsunovskaya, M.V.; Odoevsky, A.S.; Vasilyanov, G.S. Novel Methods for Synthesizing Self-Checking Combinational Circuits by Means of Boolean Signal Correction and Polynomial Codes.

Computation **2024**, *12*, 135. <https://doi.org/10.3390/computation12070135>

Academic Editor: Demos T. Tsahalidis

Received: 3 June 2024

Revised: 26 June 2024

Accepted: 27 June 2024

Published: 1 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

When synthesizing control systems and their components, such as blocks, nodes, converters, and devices, for transport-related technological processes in critical application systems, developers must focus on achieving controllability, self-checking, and fault tolerance of subsystems’ blocks and components [1–3]. Simultaneously, it is necessary to implement hardware, data, and time redundancy in the source objects to ensure prompt detection and correction of calculation errors and to pinpoint the source of malfunction. One effective method involves using embedded control circuits (ECCs) in control devices. This feature enables the detection of calculation errors during operation without the need to disconnect the devices from the controlled objects. Research suggests that the Boolean Signals Correction (BSC) method, also known as the Logical Complement method, is effective in ECC synthesis [4–6].

The paper proposes a novel approach to the ECC system in digital devices. It applies the use of the BSC method in conjunction with polynomial codes, which have broad applications in data transmission and calculation control [7,8]. The features of error detection through polynomial codes in the data symbols of code words are used. The BSC method is employed in conjunction with polynomial codes to facilitate self-checking of ECC elements in practice. Furthermore, the paper will demonstrate later that the method can always be selected to fully cover errors at the outputs of diagnostic objects. Compared to

the established and employed techniques for calculation duplication and control without signal correction in the ECC, the suggested method for constructing devices with fault detection ensures consistent testability of the device components. In addition, compared to calculation duplication and control without signal correction in the ECC, the technical implementation of self-checking digital devices can be made simpler.

The approach outlined below is recommended to employ when creating digital devices with fault detection features during operation.

2. The Investigated Approach to Creating Devices with Fault Detection

Let us outline the methodology for creating devices with fault detection. The current state of scientific endeavors will be presented, and our research’s position within the overall theory of the synthesis of self-checking digital devices will be illustrated.

Employing the BSC method implies converting the operational vector $\langle f_n f_{n-1} \dots f_2 f_1 \rangle$ produced at the outputs of the controlled device $F(x)$ (diagnostic object) into a code word $\langle h_n h_{n-1} \dots h_2 h_1 \rangle$ belonging to a pre-selected binary block uniform code [9,10]. The conversion is executed using q ($q \leq n$) two-input XOR gates following the established rule: $h_i = f_i \oplus g_i, i = \overline{1, q}$, where the function $g_i, i = \overline{1, q}$ is a control function calculated by a specialized control logic block $G(x)$. All or part of the functions $f_i, i = \overline{1, n}$ can be converted. The need to convert the value of the operational function is determined by the diagnostic parameter selected to control the calculations. Figure 1 portrays the structure for implementing the ECC through the BSC method.

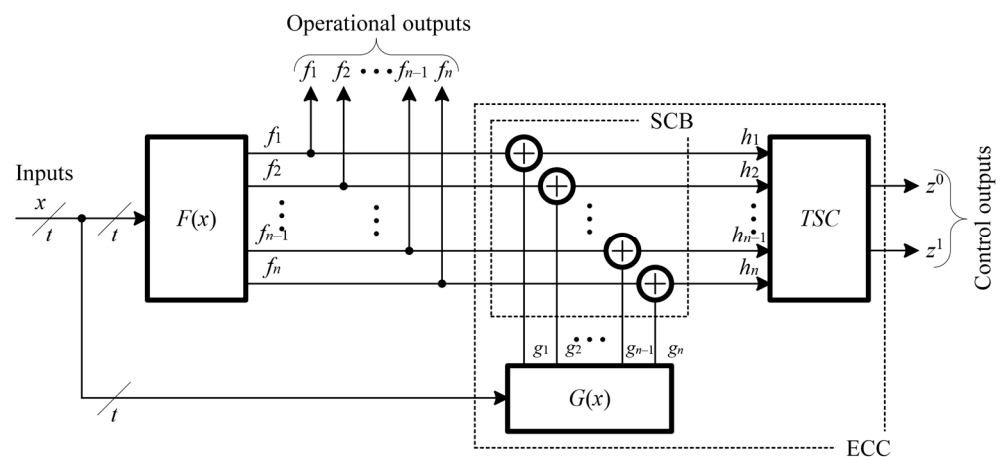


Figure 1. General structure of the ECC implemented through the BSC method.

In the BSC-implemented ECC, three functional blocks are assigned: the block for calculating the control functions of the complement $G(x)$, the signal correction block (SCB), and the totally self-checking checker (TSC). The SCB is formed by a cascade of two-input XOR gates that convert the values of operational functions. One of the inputs for each conversion gate is connected to a specific output of the diagnostic object, whereas the $G(x)$ block output is linked to the other input. The SCB output forms a code vector $\langle h_n h_{n-1} \dots h_2 h_1 \rangle$. When designing the ECC, a diagnostic feature is selected to monitor calculations in the diagnostic object. The instance is the affiliation of the generated code vector $\langle h_n h_{n-1} \dots h_2 h_1 \rangle$ to a pre-selected binary block uniform code. At each input combination, a clear and unambiguous correspondence is established between the vectors $\langle f_n f_{n-1} \dots f_2 f_1 \rangle$ and $\langle h_n h_{n-1} \dots h_2 h_1 \rangle$, which is checked during operation of the device with the checker’s ECC. Since the checker serves as the “final safeguard” in the ECC, it operates with complete self-checking and features two control outputs, z^0 and z^1 [10]. The two-rail signal at both outputs z^0 and z^1 indicates the accuracy of the calculations performed by block $F(x)$. A disorder in the two-rail signal at outputs z^0 and z^1 indicates an error in the calculations within one of the ECC blocks or within the diagnostic object itself.

In contrast to the conventional method for synthesizing ECC, which involves calculating checking symbols, the BSC method involves converting the values of the diagnostic object's operational functions into ECC instead of concatenating them with control function values [4,5]. This aspect of the BSC method enables a vast array of ECC implementations by offering diverse options to convert signals into the ECC. Additionally, it provides a comprehensive set of test combinations for XOR gates to both the SCB and the TSC, which is significantly less challenging. As illustrated in numerous studies, including [5], the use of the BSC method enables the development of fully self-checking structures for computing systems with less redundancy than duplication. Furthermore, it simplifies the process of selecting the least redundant among them.

To establish the foundation of the ECC organization, any uniform block code can be selected. This may include an indivisible code, such as the constant-weight code [11] or the Borden code [12], or one of the divisible codes, such as the Berger code [13] or any of its modifications [14].

In scientific literature, the most extensively researched branch is the ECC synthesis theory that uses the BSC method. This method involves the use of indivisible codes [15–18]. Divisible codes are often not considered for these purposes [19]. This feature arises because a malfunction in block $F(x)$ may cause distortion at any $f_i, i = \overline{1, n}$ output. This distortion may be propagated as a single malfunction in the diagnostic object to SCB outputs, causing the corresponding digits in the code vector $\langle h_n h_{n-1} \dots h_2 h_1 \rangle$ to be distorted. When considering indivisible codes, the error detection properties within the code words will determine the method of error correction required for calculations performed by the tester's circuits. If a divisible code is used in the synthesis of the ECC, distortion can occur in both data and checking symbols. Therefore, when designing a fully self-checking device, the use of divisible codes with the BSC approach must prevent undetectable errors in the data symbols and concurrent errors in the checking and data symbols. This requirement aims to streamline the process of synthesizing a completely self-checking ECC [20].

The presence of SCB also precludes the use of well-established techniques for synthesizing ECC using properties of error manifestation monotonicity in digits (also referred to as 'unidirectional implementations' of the initial $F(x)$ devices or control of the device by groups of unidirectionally independent outputs) [21–25]. Despite implementing the block $F(x)$ monotonously or allocating groups of unidirectionally independent outputs, there is still a possibility of disrupting the monotonicity of functions in the ECC. However, alternative circuitry approaches are available to arrange a fully self-checking ECC in this scenario.

This paper presents the features of utilizing polynomial or algebraic codes for the ECC synthesis through the BSC method [26]. These codes are commonly used to regulate calculations in modern microelectronic and microprocessor systems [27,28]. The objective of this paper is to develop the ECC organization methods with the BSC approach, considering the characteristics of error detection through polynomial codes.

3. Issue Formulation and Statement of the Problem to Be Addressed

When operating the system as per the block diagram shown in Figure 1, malfunctions may occur, leading to crashes and sustainability failures. The most likely cause is a single malfunction in one of the functional blocks. The ECC is designed to be completely self-checking, resulting in protective combinations at the z^0 and z^1 outputs on at least one of the input combinations if any malfunctions occur from the selected model. The self-checking circuit should be able to self-test, with each fault having its own test combination to ensure the transmission of the circuit outputs. Furthermore, the circuit should be protected from malfunctions that either do not distort the outputs or set them to protective states. A malfunction occurring in the diagnostic object $F(x)$ can lead to distortion at several outputs at once. In other words, the output (data) vector can be affected by an error with multiplicity $d \in \{1, 2, \dots, n\}$. To ensure a fully self-checking structure, the ECC must also be completely self-checking, and the following prerequisites must be met:

1. the $F(x)$ circuit must be testable, meaning that there must be at least one input combination where the malfunction appears as an error in the data vector;
2. the error at the outputs of the $F(x)$ block should not be compensated at the inputs of the ECC, i.e., it should not be perceived in the ECC as an acceptable (operational) combination.

The first problem is solved by synthesizing the controllable structure of the $F(x)$ block. The second problem is addressed through schematic representation and selection of the output vector encoding method.

Consider the following problem. Suppose there is an $F(x)$ device with a predetermined structure that has outputs $f_1, f_2, \dots, f_{n-1}, f_n$. It is required to develop a self-checking device for a single constant fault model with an embedded control circuit synthesized using the BSC method and a designated polynomial code.

The problem will be solved using discrete mathematics, technical diagnostics, Boolean algebra, Boolean differentiation, coding theory, probability theory, and logic circuit optimization theory.

4. Structures for Synthesizing the ECC through the BSC Method Utilizing Polynomial Codes

Figure 2 illustrates a standard configuration of the ECC for the $F(x)$ device with the BSC method employed to a uniform block code that can be divided arbitrarily. In this case, the diagnostic object's m outputs are not converted and are directly fed to the TSC inputs. The remaining $k = n - m$ outputs' values are adjusted in SCB before being sent to the TSC inputs. At the checker's inputs, a code word from the (m, k) block code is formed, where m is the number of data symbols and k is the number of checking symbols.

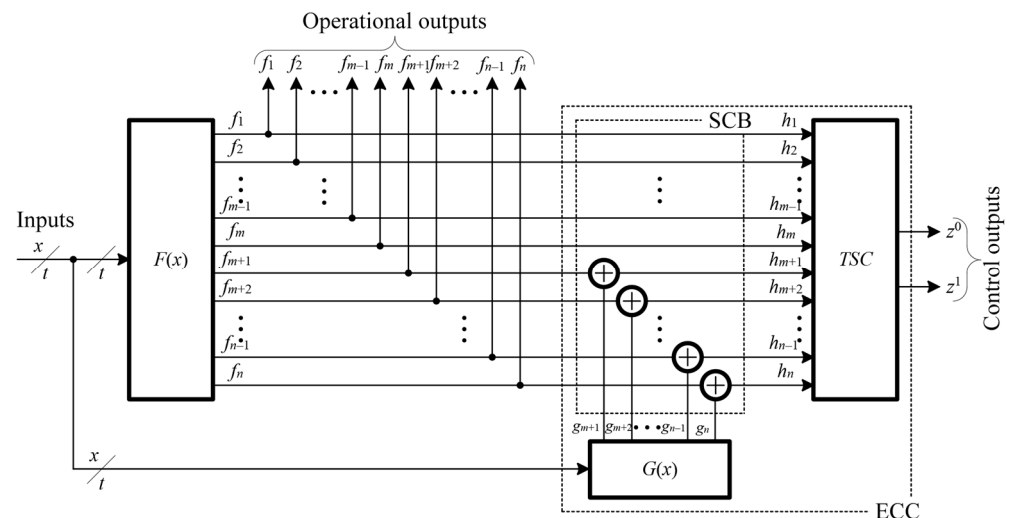


Figure 2. General structure of the ECC implemented through the BSC method, employing divisible uniform block codes.

Figure 2 shows a general representation of the conversion of k functions in SCB using k two-input XOR gates. It is worth mentioning that as the number of data elements in the SCB structure increases, it becomes increasingly difficult to provide a comprehensive set of test combinations for their inputs. Solutions to this problem using the minimum necessary conversion gates in the SCB will be explored in this text. This is achieved if part of the $f_1, f_2, \dots, f_{n-1}, f_n$ outputs will form the values of data symbols, while the remaining outputs will be corrected to form checking symbols. Figure 3 shows the structures of the ECC organization when utilizing a polynomial code with $k = 2$ and $k = 3$ checking symbols.

Figure 3 illustrates the structures where the SCB consists of two and three XOR gates, respectively. This allows for the conversion of any vector $\langle f_n f_{n-1} \dots f_2 f_1 \rangle$ into a code word $\langle h_n h_{n-1} \dots h_2 h_1 \rangle$ that belongs to a divisible code with two and three checking symbols. It is worth noting that the number of conversion gates can be reduced. For instance, in the SCB

structure shown in Figure 3a, only the f_n output can be converted into the checking symbol h_n of the polynomial code. The other checking symbol h_{n+1} can be directly calculated by the $G(x)$ block without any conversion. In this case, the number of TSC inputs increases by one, while the number of XOR gates decreases. If the structure shown in Figure 3a does not use conversions at all, then the SCB is actually degenerate, and the TSC inputs are directly connected to the outputs of the $G(x)$ block. They form the checking symbols h_{n+1} and h_{n+2} of the polynomial code. This circuit adheres fully to the conventional structure of the ECC [6]. The same applies to the structure presented in Figure 3b. The structure that utilizes both the XOR gates and the direct calculations of the checking symbols by the $G(x)$ block is hereinafter referred to as the hybrid structure.

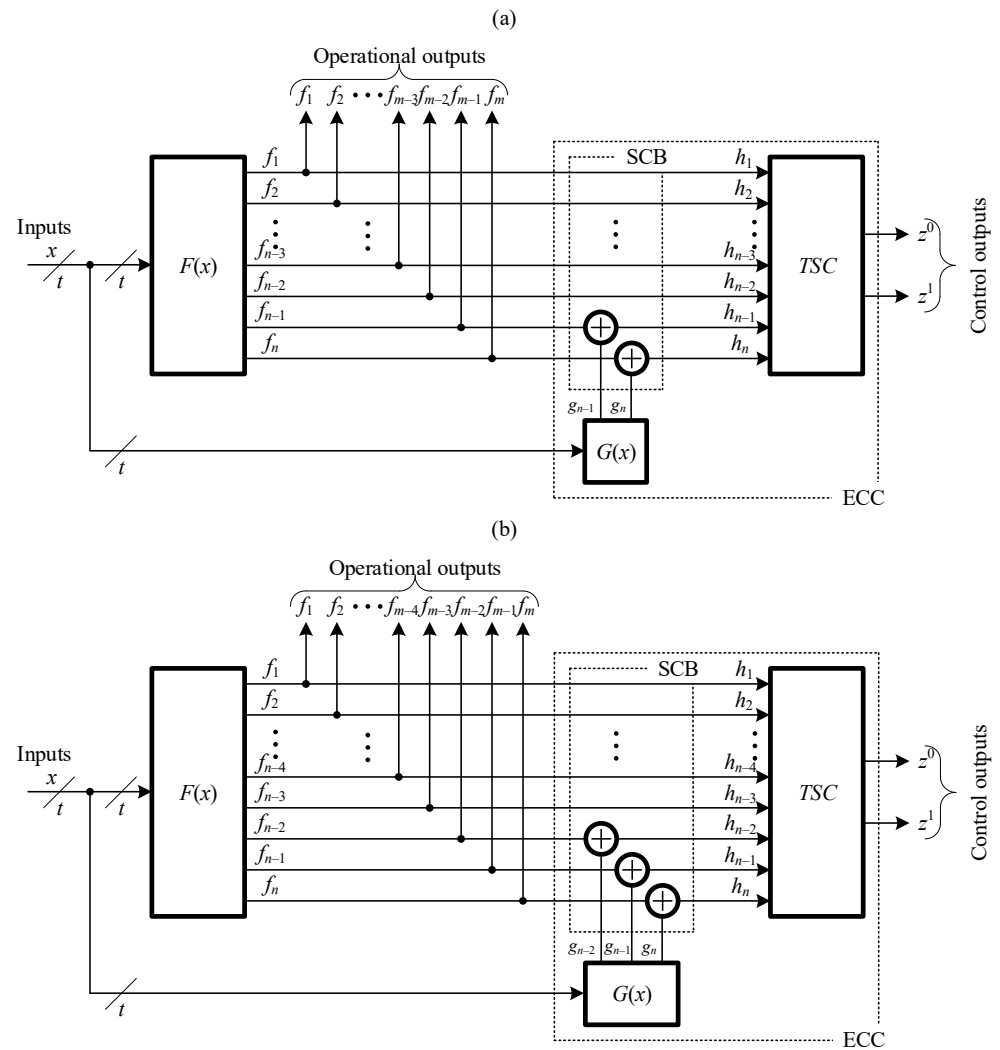


Figure 3. Structures for organizing the ECC through the BSC method utilizing polynomial codes: (a) with two checking symbols; (b) with three checking symbols.

Let us focus solely on the structures presented in Figure 3.

5. Allocation Rules for Convertible and Non-Convertible Outputs

The authors thoroughly researched error detection through polynomial codes within the conventional ECC structure and investigated the impact of controlled device malfunctions on computing data symbol functions [26,29]. In fact, the error detection characteristics are known at those outputs of the structures appearing in Figure 3 that were not converted in SCB. This condition applies to the ECC organization.

Statement 1. *The outputs of the $F(x)$ block after conversion should not be structurally or functionally related to any of its other outputs.*

Proof. If Statement 1 is fulfilled, any single malfunction will result in an error in either the non-convertible output group or the convertible output group. In the first case, the error will be detected, as it violates the correspondence between the values of data and checking symbols. This causes the control vector to not correspond to the data vector, since the error is always sent to the XOR gate’s output. In the second case, it is imperative to eliminate any undetectable errors that may occur at the output and are not converted. In other words, one must configure error propagation properties to the non-convertible outputs of the diagnostic object and execute one of the following actions:

1. Select a polynomial code that can detect all possible types of errors at the outputs;
2. Identify groups of controllable outputs using the selected polynomial code;
3. Convert the diagnostic object’s structure into a structure with controllable groups of outputs (up to a single group containing all the outputs).

Thus, Statement 1 is proved. \square

It follows directly from the above that the characteristics of error detection via polynomial codes in data symbols, established in [26,29], are critical.

In [20], the concept of structurally and functionally independent groups of outputs is introduced.

Definition 1. *A group of outputs $(f_{i_1}, f_{i_2}, \dots, f_{i_s}), \{i_1, i_2, \dots, i_s\} \in \{1, 2, \dots, n\}$ is structurally independent of a group of outputs $(f_{j_1}, f_{j_2}, \dots, f_{j_l}), \{j_1, j_2, \dots, j_l\} \in \{1, 2, \dots, n\} \setminus \{i_1, i_2, \dots, i_s\}$ if there is no such element in the digital device’s structure, the paths from which lead simultaneously to at least one of the $(f_{i_1}, f_{i_2}, \dots, f_{i_s})$ outputs and at least one of the group outputs.*

If there is a group of outputs $(f_{i_1}, f_{i_2}, \dots, f_{i_s})$ that is structurally independent of another group of outputs $(f_{j_1}, f_{j_2}, \dots, f_{j_l})$, then both groups’ outputs can be selected as convertible outputs. Among such groups, the focus is on identifying outputs that require conversion while considering the impact on the technical implementation of the $G(x)$ block complexity and the need to maintain controllability of the polynomial code TSC and XOR gates.

The concept of functionally independent sets of outputs can be extended to express the following significant principles.

Statement 2. *A group of outputs $(f_a, f_b), a, b \in \{1, 2, \dots, n\}$ is functionally independent of a group of outputs $\{f_{i_1}, f_{i_2}, \dots, f_{i_v}\} = \{f_1, f_2, \dots, f_n\} \setminus \{f_a, f_b\}$, if the condition is met for any G_q gate with y_q output:*

$$\left(\frac{\partial f_a}{\partial y_q} \vee \frac{\partial f_b}{\partial y_q}\right) \left(\frac{\partial f_{i_1}}{\partial y_q} \vee \frac{\partial f_{i_2}}{\partial y_q} \vee \dots \vee \frac{\partial f_{i_v}}{\partial y_q}\right) = 0. \tag{1}$$

Statement 3. *A group of outputs $(f_a, f_b, f_c), a, b, c \in \{1, 2, \dots, n\}$ is functionally independent of a group of outputs $\{f_{i_1}, f_{i_2}, \dots, f_{i_v}\} = \{f_1, f_2, \dots, f_n\} \setminus \{f_a, f_b, f_c\}$, if the condition is met for any G_q gate with y_q output:*

$$\left(\frac{\partial f_a}{\partial y_q} \vee \frac{\partial f_b}{\partial y_q} \vee \frac{\partial f_c}{\partial y_q}\right) \left(\frac{\partial f_{i_1}}{\partial y_q} \vee \frac{\partial f_{i_2}}{\partial y_q} \vee \dots \vee \frac{\partial f_{i_v}}{\partial y_q}\right) = 0. \tag{2}$$

Proof. The structures of Formulas (1) and (2) are identical. Each formula shows the result of multiplying two Boolean expressions to the left. For the selected set of two or three outputs, the disjunction of Boolean derivatives is displayed in the first parentheses. The second parentheses display the disjunction of Boolean derivatives for the remaining outputs. The output of f_i will only display an error resulting from a malfunction of element

G_q , if $\frac{\partial f_i}{\partial y_q} \neq 0$. In this case, the Boolean derivative equals a specified Boolean function $\frac{\partial f_i}{\partial y_q} \neq g_i(x)$. For those input combinations where $g_i(x) = 1$, the values at the output f_i will be distorted. The error resulting from a malfunction of element G_q will manifest at the outputs f_a and f_b if $\frac{\partial f_a}{\partial y_q} \neq 0$ and $\frac{\partial f_b}{\partial y_q} \neq 0$. $\frac{\partial f_a}{\partial y_q} \vee \frac{\partial f_b}{\partial y_q} \neq 0$. These outputs will be dependent in this scenario. In the event that any element malfunctions, only the outputs f_a and f_b will be distorted. According to definition 1, this group of outputs will be independent of the group of outputs $\{f_{i_1}, f_{i_2}, \dots, f_{i_v}\} = \{f_1, f_2, \dots, f_n\} \setminus \{f_a, f_b\}$. Therefore, there should not be any circumstances in which the Boolean derivative of any variable for any other output does not equal zero: $\frac{\partial f_i}{\partial y_q} = 0, i \in \{i_1, i_2, \dots, i_v\}$. Then the expression is valid for all outputs from the group: $\frac{\partial f_{i_1}}{\partial y_q} \vee \frac{\partial f_{i_2}}{\partial y_q} \vee \dots \vee \frac{\partial f_{i_v}}{\partial y_q} = 0$. It follows that distortions may appear in either of the selected output groups, $\{f_a, f_b\}$ or $\{f_{i_1}, f_{i_2}, \dots, f_{i_v}\} = \{f_1, f_2, \dots, f_n\} \setminus \{f_a, f_b\}$, if $\forall y_q, q \in Q : \left(\frac{\partial f_a}{\partial y_q} \vee \frac{\partial f_b}{\partial y_q}\right) \left(\frac{\partial f_{i_1}}{\partial y_q} \vee \frac{\partial f_{i_2}}{\partial y_q} \vee \dots \vee \frac{\partial f_{i_v}}{\partial y_q}\right) = 0$. In this example, Q represents the collection of indices for q elements in the device's structure.

In a similar vein, the scenario involving three outputs is examined. Statements 2 and 3 are proved. \square

When considering k allocated transformable outputs, Statements 2 and 3 can be applied more broadly.

Statement 4. A group of outputs $(f_a, f_b, \dots, f_k), a, b, \dots, k \in \{1, 2, \dots, n\}$ is functionally independent of a group of outputs $\{f_{i_1}, f_{i_2}, \dots, f_{i_v}\} = \{f_1, f_2, \dots, f_n\} \setminus \{f_a, f_b, \dots, f_k\}$, if the condition is met for any G_q gate with y_q output:

$$\left(\frac{\partial f_a}{\partial y_q} \vee \frac{\partial f_b}{\partial y_q} \vee \dots \vee \frac{\partial f_k}{\partial y_q}\right) \left(\frac{\partial f_{i_1}}{\partial y_q} \vee \frac{\partial f_{i_2}}{\partial y_q} \vee \dots \vee \frac{\partial f_{i_v}}{\partial y_q}\right) = 0. \tag{3}$$

Statement 4 is proven using mathematical induction, which directly follows from the proofs of Statements 2 and 3. The proof is not provided here due to its simplicity.

Similarly, requirements are established for groups of functionally independent outputs that have a significant quantity of outputs (as shown in Figure 2).

Therefore, to fulfil the initial requirement in polynomial codes of excluding simultaneous errors in the data and checking symbols, structurally and functionally independent outputs should be grouped. In this instance, the converted outputs belong to one of the independent groups comprising a limited number of device outputs, while the non-convertible outputs belong to the other group within the ECC. As mentioned above, any combination of distortions in the group of converted outputs will be detected.

In order to develop a fully self-checking ECC, it is imperative to eliminate any undetectable errors in a group of non-convertible outputs. This can be achieved by allocating and controlling r -independent groups of outputs (I^r -groups) [30] through polynomial codes, ensuring effective outcomes.

Definition 2. A group of outputs of a digital device $\{f_{i_1}, f_{i_2}, \dots, f_{i_s}\}, i_1, i_2, \dots, i_s \in \{1, 2, \dots, n\}$ is an r -independent group of outputs (I^r -group) if a malfunction of the output of any G_q logic gate of its structure distorts the values of no more than r outputs of the group.

The following statement is proved in the same source [30].

Statement 5. The set of outputs of a digital device $\{f_{i_1}, f_{i_2}, \dots, f_{i_s}\}, s \geq r + 1$ forms an I^r -group if it satisfies a condition for every subset of C_q^{r+1} $r + 1$ of its outputs and for every G_q gate of its internal structure:

$$\frac{\partial f_{i_1}}{\partial y_q} \frac{\partial f_{i_2}}{\partial y_q} \dots \frac{\partial f_{i_{r+1}}}{\partial y_q} = 0. \tag{4}$$

Proof. If the malfunctioning element G_q results in an error at the output f_i , then $\frac{\partial f_i}{\partial y_q} \neq 0$. The derivative of another output will not be zero if it is distorted. The conditions for the absence of simultaneous distortion of the $s + 1$ output, when the element G_q malfunctions, are defined by the expression $\frac{\partial f_{i_1}}{\partial y_q} \frac{\partial f_{i_2}}{\partial y_q} \dots \frac{\partial f_{s+1}}{\partial y_q} = 0$. The outputs of the device will only function independently if and when the predetermined requirement of $\forall y_q, q \in Q$ is satisfied.

Statement 5 is proved. \square

The I' -group can be controlled by a polynomial code that detects any errors with multiplicities $d \in \{1, 2, \dots, r\}$.

6. Specific Classes of Polynomial Codes with Low Redundancy

Studies of error detection with various types and multiplicities via polynomial codes [26,29] have revealed several specific classes of codes capable of detecting errors with varying multiplicities $d \in \{1, 2, \dots, r\}$ at different r values. Table 1 summarizes the main generator polynomials that can be used effectively when allocating I' -groups to a set of controlled device outputs. The polynomial code is briefly denoted by the Latin letter P followed by a decimal number representing the polynomial in binary form. The PN -code will then be utilized. For instance, the binary number 1011 corresponds to the polynomial $x^3 + x + x^0$. Each digit in the binary number implies the presence (1) or absence (0) of a corresponding term in the polynomial. The decimal number corresponding to the binary number 11 represents this polynomial. The number of data symbols (the length of the data vector) is denoted by m . The value $k = \lceil \log_2(N + 1) \rceil - 1$ determines how many checking symbols are in the PN code word.

Table 1. Polynomial code classes with multiplicities-containing error detection $d \in \{1, 2, \dots, r\}$.

Generator Polynomial	Designation	Number of Checking Symbols	Constraints	Code Rate
Codes with detection of any errors with multiplicities $d \leq 2$				
$x^2 + x + x^0$	$P7$	2	$m \leq 3$	$R \leq 0.6$
$x^3 + x + x^0, x^3 + x^2 + x^0$	$P11, P13$	3	$m \leq 7$	$R \leq 0.7$
$x^4 + x + x^0, x^4 + x^3 + x^0$	$P19, P25$	4	$m \leq 15$	$R \leq 0.789$
$x^5 + x^1 + x^0, x^5 + x^4 + x^0$	$P35, P49$	5	$m \leq 31$	$R \leq 0.861$
Codes with detection of any errors with multiplicities $d \leq 3$				
$x^3 + x^2 + x^1 + x^0$	$P15$	3	$m \leq 4$	$R \leq 0.571$
$x^4 + x^2 + x^1 + x^0, x^4 + x^3 + x^2 + x^0$	$P23, P29$	4	$m \leq 7$	$R \leq 0.636$
$x^5 + x^2 + x^1 + x^0, x^5 + x^4 + x^3 + x^0$	$P39, P57$	5	$m \leq 14$	$R \leq 0.737$
$x^5 + x^3 + x^1 + x^0, x^5 + x^4 + x^2 + x^0$	$P43, P53$	5	$m \leq 15$	$R \leq 0.75$
Codes with detection of any errors with multiplicities $d \leq 4$				
$x^4 + x^3 + x^2 + x^1 + x^0$	$P31$	4	$m \leq 5$	$R \leq 0.556$
Codes with detection of any errors with multiplicities $d \leq 5$				
$x^5 + x^4 + x^3 + x^2 + x^1 + x^0$	$P63$	5	$m \leq 6$	$R \leq 0.545$

Table 1 shows the R rate limit for codes. It is important to note that, for many of the polynomial codes, the value of the code rate R is close to 0.5, which is typical for repetition codes [31]. Nevertheless, it is possible to select a polynomial code with a higher code rate value than that of repetition codes. For best results, it is recommended to utilize polynomial codes with the highest code rate. This determines their application features in organizing the ECC. For instance, if the controlled device comprises a single I^2 -group of $n = 14$ outputs, it would be optimal to apply polynomial codes $P19$ and $P25$. These codes possess four checking symbols each, which facilitates effective control of the device. An alternative option to consider would be the selection of $P11$ and $P13$ codes, each consisting of three checking symbols. This would require the allocation of two groups of seven outputs.

However, when selecting a code, it is important to consider both the quantity of checking symbols and the complexity indicators of the encoders and control logic block functions. This should be taken into account in each specific case separately, for example, during experimentation or modeling.

7. Algorithms for Synthesizing the ECC through the BSC Method Utilizing Polynomial Codes

By following the previously outlined principles, it is possible to devise algorithms for synthesizing the ECC through the BSC method utilizing polynomial codes. The most straightforward approach is to create an ECC with a random allocation of the device outputs into two subsets. However, this approach will result in the largest number of undetectable errors. Selecting functionally independent output groups in the device based on Formulas (1)–(3) can eliminate errors associated with simultaneous distortions and data and checking symbols at the TSC’s inputs. However, errors related to distortion of non-convertible outputs may be partially undetectable. The characteristics of error detection are determined by the features of the code selected for the organization of the ECC.

The following algorithm enables the construction of an ECC, excluding concurrent distortions in data and checking symbols, while also providing error coverage for the data vector (see Figure 4).

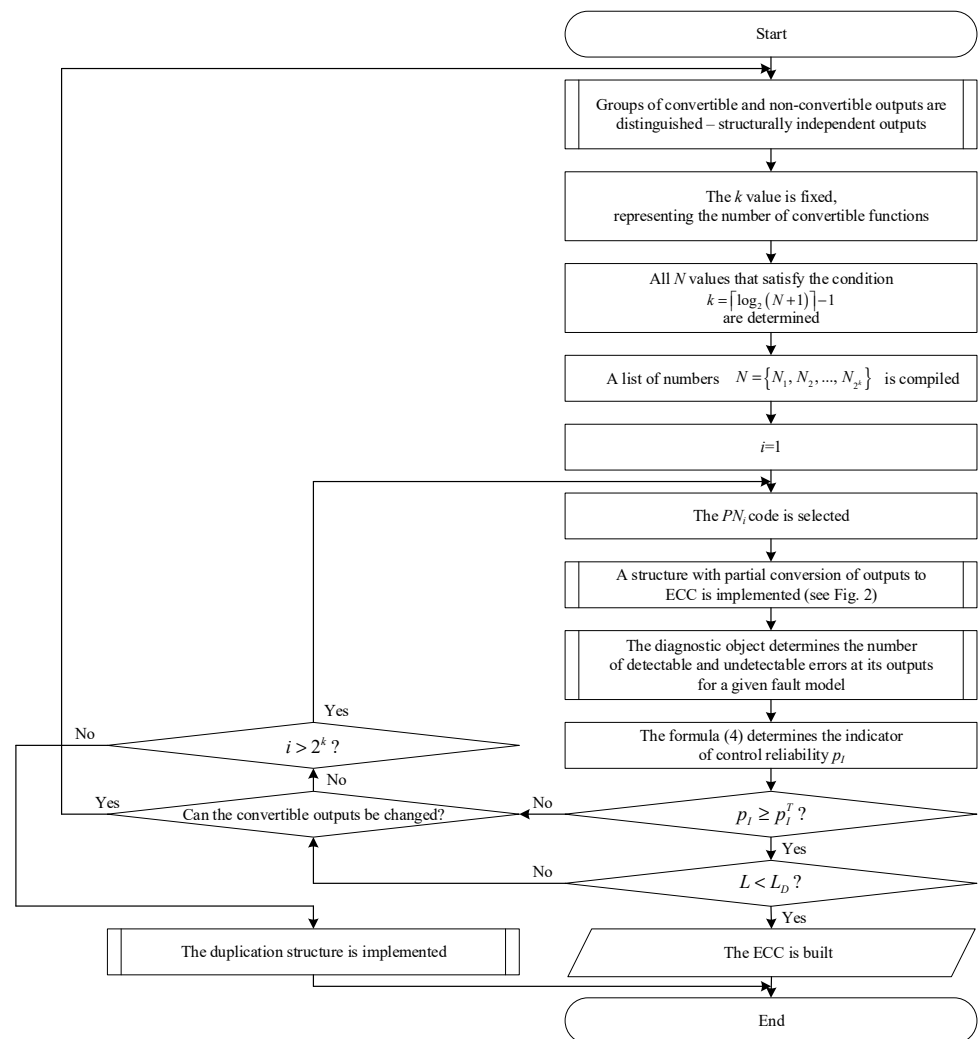


Figure 4. Algorithm for synthesizing the ECC through the BSC method utilizing polynomial codes and one control group of outputs with incomplete error coverage.

The algorithm shown in Figure 4 is based on the initial selection of the diagnostic object’s number of transformed functions, k . Figure 2 depicts the construction of the structure. In this scenario, the value N is selected, which enables the creation of a PN code with the number of checking symbols equal to k . Since $k = \lceil \log_2(N + 1) \rceil - 1$, this expression satisfies 2^k values of N , which make up the list $N = \{N_1, N_2, \dots, N_{2^k}\}$. The initial step is to select the first number from the list and construct an ECC for it based on the structure illustrated in Figure 2. The method for assessing the reliability of the control is as follows:

$$p_I = \frac{M - M'}{M}, \tag{5}$$

where M is the total number of errors occurring at the device’s outputs, and M' is the total number of errors not detected using the selected polynomial error code.

If the specified indicator meets the requirements of the ECC designer, i.e., $p_I \geq p_I^T$, and if p_I^T , the preset value of the reliability indicator, is in line with expectations, then proceed to the next step. The process involves determining the complexity indicator of the ECC implementation (denoted as L) and then comparing it with the complexity of the ECC implementation using the standard duplication method (denoted as L_D). If $L < L_D$, ECC is constructed according to the proposed method. Otherwise, the system checks whether outputs can be replaced for conversion to ECC and selects the number N .

Figure 5 shows an example of a combinational logic device.

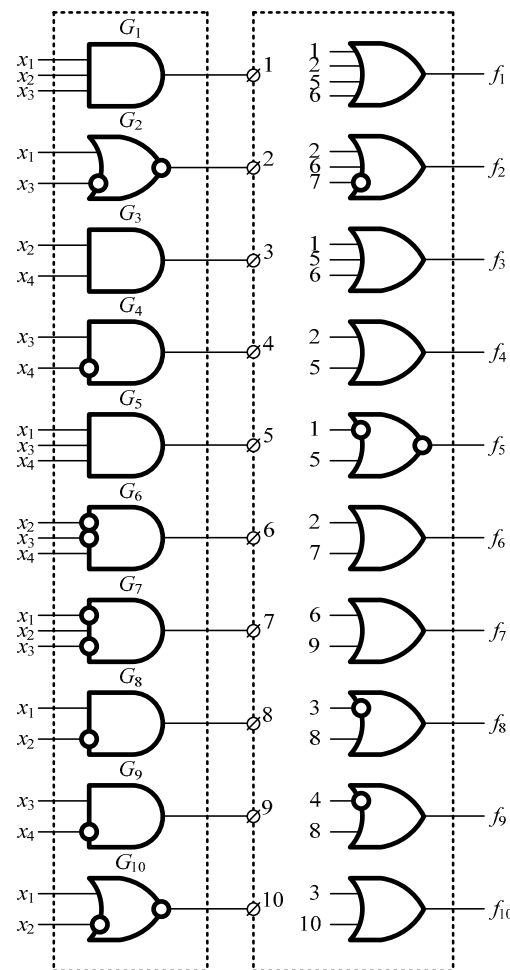


Figure 5. Two-level combinational device.

Tables 2 and 3 describe the output dependencies. Based on the analysis, it can be concluded that a subset of the outputs $\{f_8, f_9, f_{10}\}$ is independent of the second subset $\{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$.

Table 2. Matrix of the dependencies of outputs and logic gates in the first cascade.

First Cascade Gates	Device Outputs									
	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
G_1	×		×		×					
G_2	×	×		×		×				
G_3								×		×
G_4									×	
G_5	×		×	×	×					
G_6	×	×	×							
G_7		×					×			
G_8								×	×	
G_9							×			
G_{10}										×

Note. The symbol «×» is positioned at the intersection of the row and column if the logic gate $G_i, i = \overline{1, 10}$ is connected to the output $f_j, j = \overline{1, 10}$, which corresponds to the column.

Table 3. Matrix of the dependencies of the combinational device’s outputs.

Device Outputs	Device Outputs									
	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
f_1	–	–	–	–	–	–	–	–	–	–
f_2	D	–	–	–	–	–	–	–	–	–
f_3	D	D	–	–	–	–	–	–	–	–
f_4	D	D	D	–	–	–	–	–	–	–
f_5	D	I	D	D	–	–	–	–	–	–
f_6	D	D	I	D	I	–	–	–	–	–
f_7	D	D	D	I	I	I	–	–	–	–
f_8	I	I	I	I	I	I	I	–	–	–
f_9	I	I	I	I	I	I	I	D	–	–
f_{10}	I	I	I	I	I	I	I	D	I	–

Notes. 1. The letter ‘D’ in the table denotes dependent outputs, while the letter ‘I’ denotes independent outputs of the controlled device. 2. The symbol ‘–’ is used to indicate a case where comparison is not required.

Table 1 shows that the device can be controlled using either P_{11} or P_{13} codes. These codes will have $k = 3$ checking symbols and $m = 7$ data symbols for a given device. The structure shown in Figure 3b will be arranged accordingly. With the specified number of data symbols, both code P_{11} and code P_{13} will detect any one- and two-fold errors at the circuit outputs.

If we consider a model of single constant faults of logic gates of the internal structure (stuck-at faults), then the total number of errors at the device’s outputs will be determined by the value: $M = 2^t(2N_G)$. The first multiplier determines the total number of input combinations, and the second one determines the total number of single constant faults of the device. In this case, $M = 2^4 \cdot (2 \cdot 20) = 640$. According to Table 2, malfunctions of $G_1, G_2, G_5,$ and G_6 may go undetected as they are connected to device outputs through multiple paths. The analysis of single constant faults for the designated elements in the structure of the combinational device revealed that only one three-fold monotone error caused by a stuck-at-0 fault of the G_5 gate in the input set 1011 (0.156% of the total number of output device errors) remained undetected when controlled by the P_{13} code. The P_{11} code does not detect significantly more errors, comprising 13 errors (2.032% of total output errors). These errors include seven four-fold unidirectional errors caused by a stuck-at-1 failure of the G_5 gate in the input sets 0000, 0100, 0101, 1000, 1010, 1100, 1101, one three-fold

unidirectional error caused by a stuck-at-0 failure of the G_6 gate in the input set 0001, and five three-fold unidirectional errors caused by the stuck-at-1 failure of the G_6 gate in the input sets 0001, 0101, 1000, 1100, and 1101.

Thus, the reliability factor for error control at the outputs of the device using the $P13$ code is $p_I = (640 - 1)/640 \approx 0.998$. When using the $P11$ code, the reliability factor is equal to $p_I = (640 - 13)/640 \approx 0.98$. It is recommended to use both codes for organizing the ECC in a specific device, provided that the threshold value of $p_I < 0.98$ is met. Priority should be given to the $P13$ code, which identifies all errors except one.

It is feasible to enhance the quantity of errors identified by the $P11$ and $P13$ codes at the device's outputs by implementing redundancy in the first stage of its structure. For example, to exclude a single undetectable error caused by the $P13$ code, it is necessary to reserve the G_5 gate and connect the inputs of the second cascade's gates to the outputs of the device f_1 and f_3 , using the output of one of the copies. Additionally, the inputs of the second cascade's gates must connect to the f_4 and f_5 outputs of the device through the other copy. Similarly, by separating the outputs into pairs f_1, f_3 and f_2, f_5 and reserving the G_6 gate, any undetectable errors under the $P11$ code can be excluded.

In order to increase the number of detected errors, diverse circuit design methods can be employed, encompassing:

- error control at the outputs of the device by selecting groups of outputs and using polynomial codes with fewer data symbols;
- error control at the outputs of the device by selecting I^2 -groups of outputs and special polynomial codes that identify such distortions;
- conversion of the source device's structure into a structure whose outputs form an I^2 -group.

Observations of polynomial codes' characteristics [26,29] show that increasing the number of data symbols results in twice as many undetectable errors, given a constant number of check symbols. Consequently, the number of undetectable errors increases in the outputs of controlled devices. To address this issue, it is recommended to control the outputs by groups. Figure 6 illustrates the algorithm for synthesizing ECC in this scenario.

Figure 7 illustrates the ECC's organizational structure with the allocation of two groups of controlled outputs by polynomial codes. For simplicity, the system's operational outputs are not displayed. In this scenario, two independent control circuits are employed for each subgroup of outputs. This leads to the allocation of subsets of convertible and non-convertible outputs. It is important to note that only the groups of convertible and non-convertible outputs within each independently controlled group of outputs of the device $F(x)$ should be independent. There are no other restrictions on outputs. In other words, there are two subgroups of convertible outputs and two subgroups of non-convertible outputs that may be dependent. All the ECC's gates conform to standard specifications, with the exception of the control logic blocks $G_1(x)$ and $G_2(x)$, which are synthesized individually for each subgroup of outputs of the device $F(x)$. It is possible to optimize and implement their joint uniform block $G(x)$. The first group of outputs is controlled based on the selected polynomial code using the TSC_1 , while the second group of outputs is controlled based on another selected polynomial code using the TSC_2 . The results obtained from the checkers are merged via the TRC two-rail signal compression component. This component serves as both the diagnostic system's outputs and the combined outputs of the checkers.

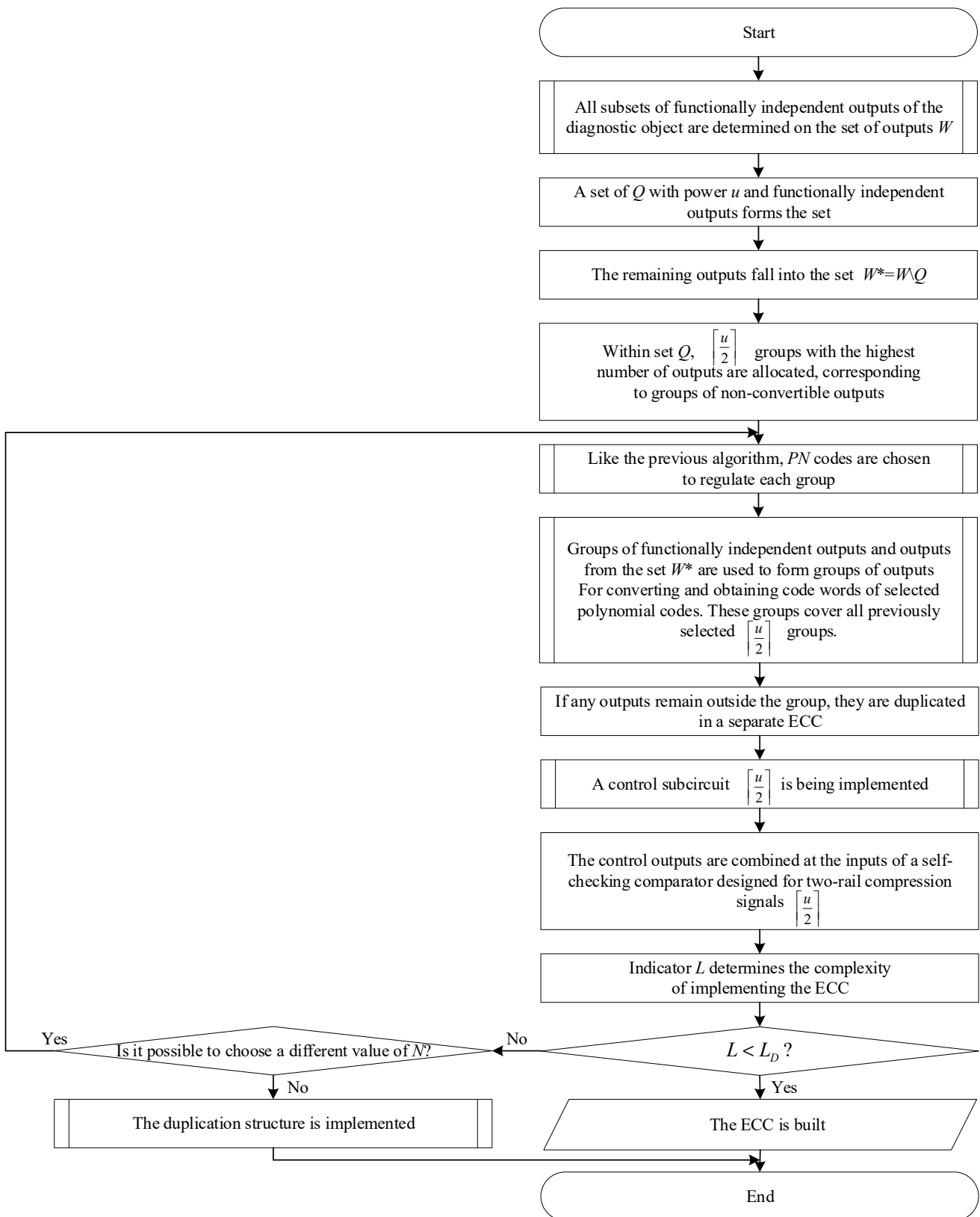


Figure 6. Algorithm for synthesizing the ECC through the BSC method utilizing polynomial codes and several control groups of outputs with incomplete error coverage.

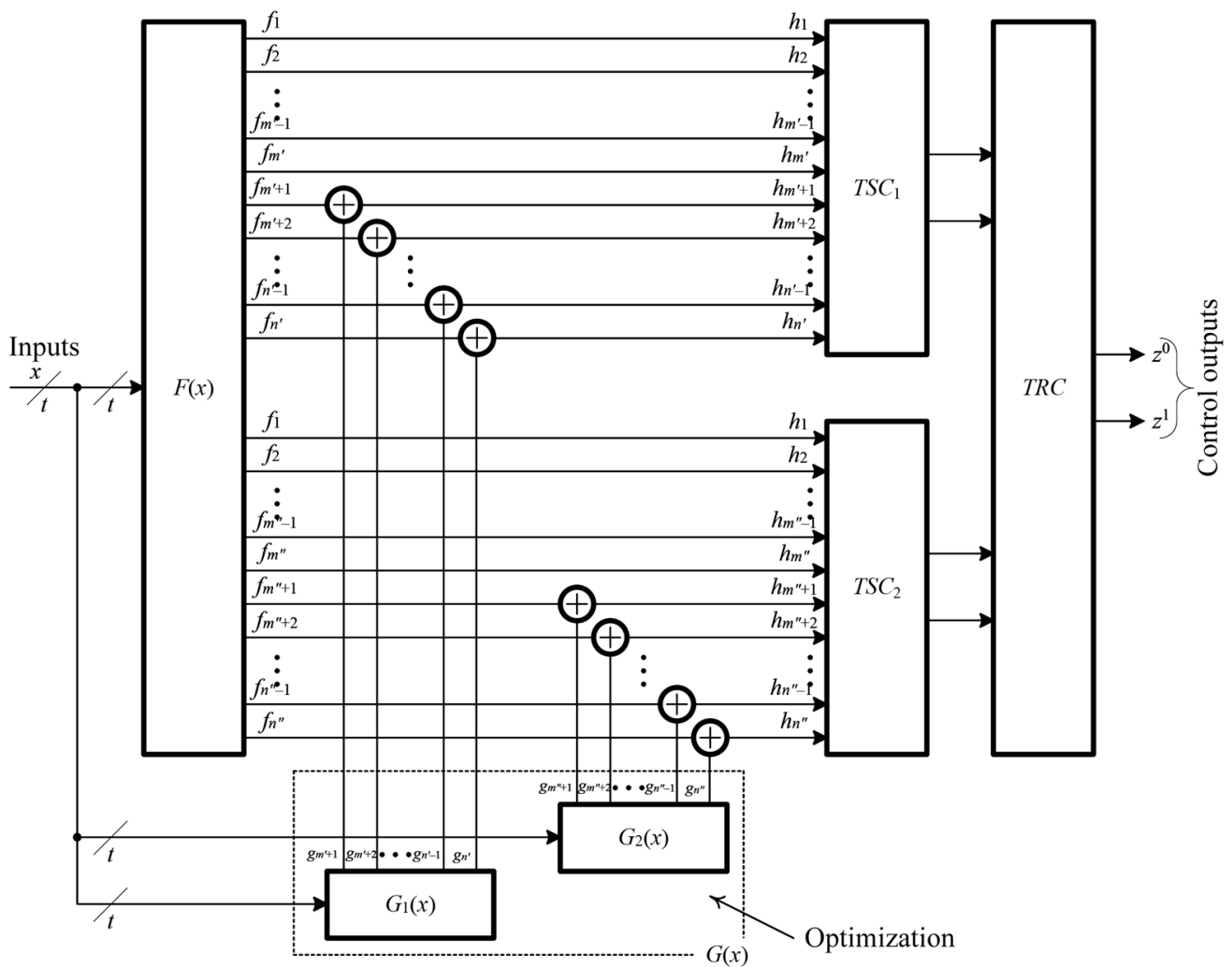


Figure 7. Structures for organizing ECC through the BSC method utilizing polynomial codes with group output control.

In general, $\lceil \frac{t}{2} \rceil$ groups of outputs can be allocated. The structure of the ECC’s organization is similar to that of two output groups.

It should be emphasized that the ECC, when implemented using the first and second algorithms, is not entirely self-checking. This is because it may not identify all errors in the diagnostic object’s outputs. To ensure comprehensive self-checking, it is necessary to refer to the concept of r -independent groups of outputs.

By utilizing the polynomials outlined in Table 1, it is possible to create polynomial codes with specific constraints on the number of data symbols. This allows for the development of an algorithm for the ECC synthesis that allocates I^r -groups of outputs to the controlled device’s output set. An example of an algorithm for searching for I^2 -groups of outputs is shown in Figure 8.

The search for I^2 groups of outputs enables proceeding to the organization of the ECC with full error coverage (see Figure 9).

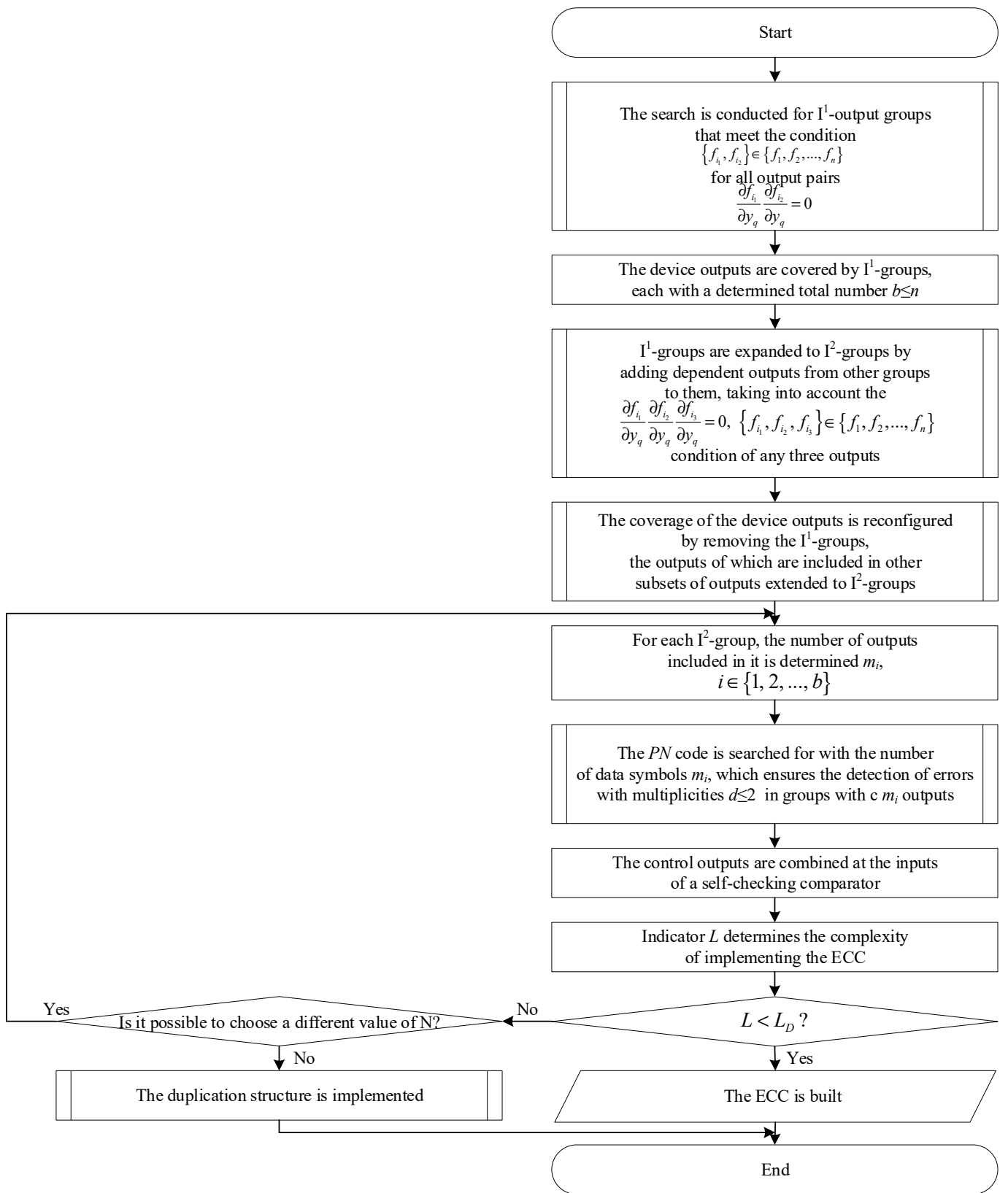


Figure 8. Algorithm for sequential search for I²-groups of outputs with their control by the selected polynomial code.

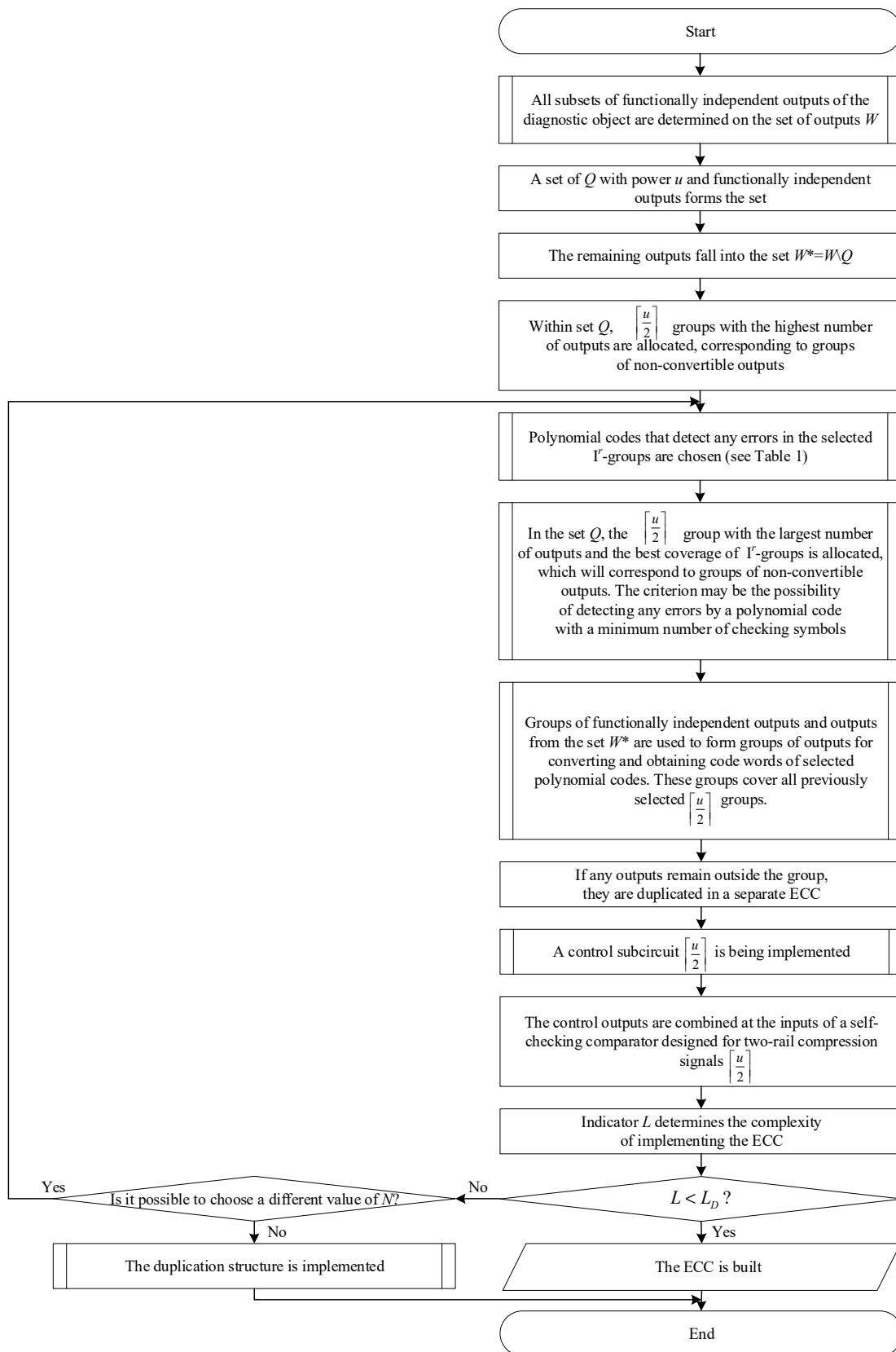


Figure 9. Algorithm for synthesizing the ECC through the BSC method utilizing polynomial codes and allocating several control groups of outputs with complete error coverage.

8. Experimental Results

To confirm the effectiveness of the proposed approaches to organizing ECC, the BSC method with a control by polynomial codes was used [32]. Codes with $k = 2$ and $k = 3$ were selected. Experiments were conducted using the Multisim electronic circuit modeling environment.

Two ECCs are modeled (see Figures 10 and 11) in the Multisim environment, based on the structures of the ECC organization proposed in Figure 3.

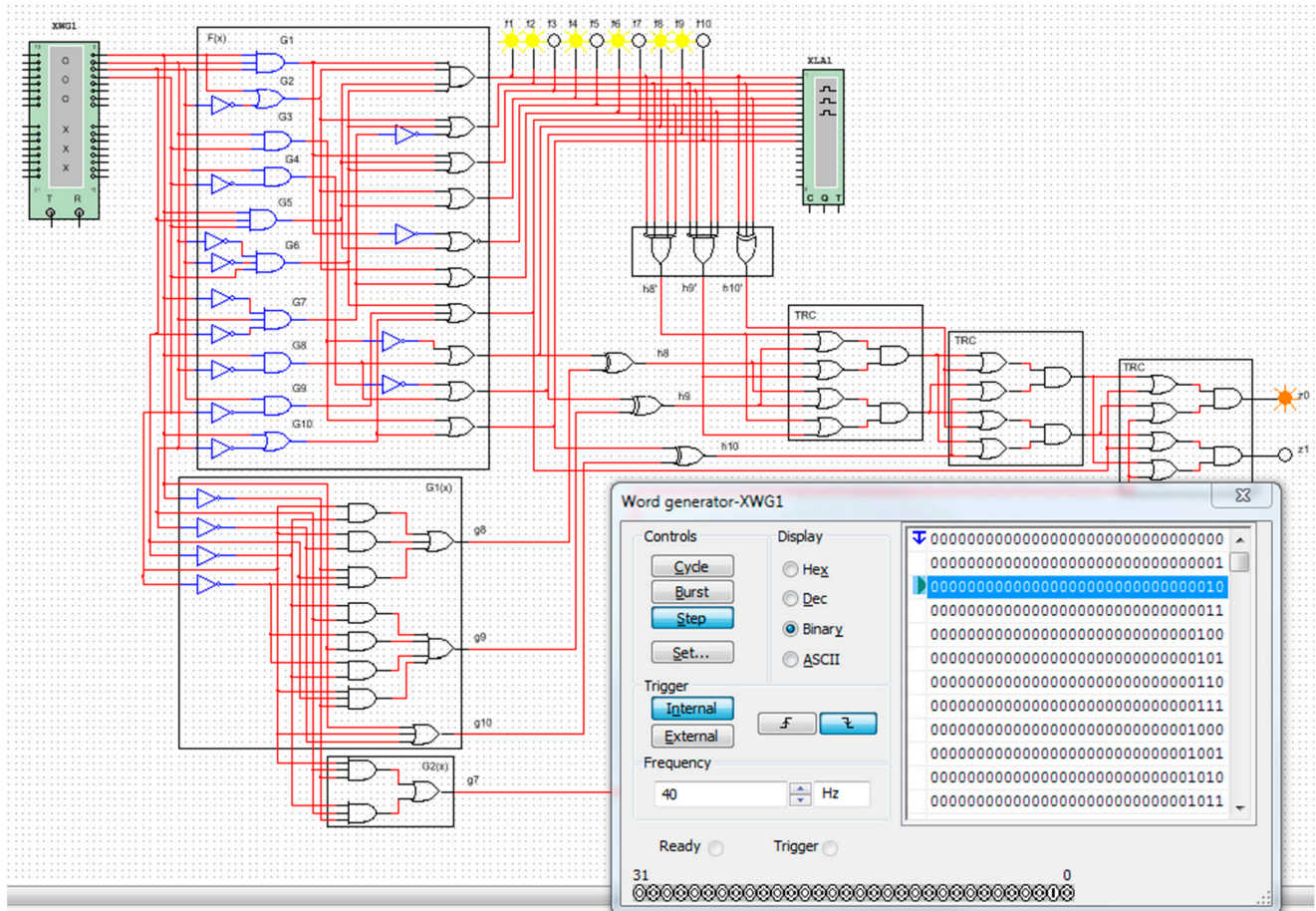


Figure 10. The Multisim environment was applied to model the ECC using the BSC method. This involved allocating groups of outputs and controlling them using P13 code.

The initial ECC model (Figure 10) utilizes the P13 code as the base code of the system. Table 1 demonstrates that this enables the identification of all errors with a multiplicity of $d \leq 2$ at $m \leq 7$.

In the second model of the ECC (see Figure 11), the P7 code was used in both testers. This ensured the complete detection of all double errors at $m \leq 3$.

During the experiment, stuck-at faults, specifically stuck-at-0/1 (gate U69 in Figure 11), were sequentially installed at the outputs of gates whose paths are connected to three or more outputs of the diagnosed circuit. These gates are the logic gates of the first circuit level, namely $G_1, G_2, G_5,$ and G_6 . The system received a sequence of input vectors, and the resulting values of the output working vectors were recorded. If a single internal fault is transmitted to the outputs of the diagnosed circuit and distorts the bits of the output vector $\{f_1, f_2, \dots, f_{10}\}$, the values of the vector $\{z^0, z^1\}$ are analyzed. If the values of the vector bits $\{z^0, z^1\}$ remain unchanged when the vector bits $\{f_1, f_2, \dots, f_{10}\}$ are distorted, it is considered that this ECC structure does not detect this malfunction. Otherwise, it does. This analysis investigates internal faults at the outputs of elements $G_1, G_2, G_5,$ and G_6 for ECC structures based on the structures proposed by the authors.

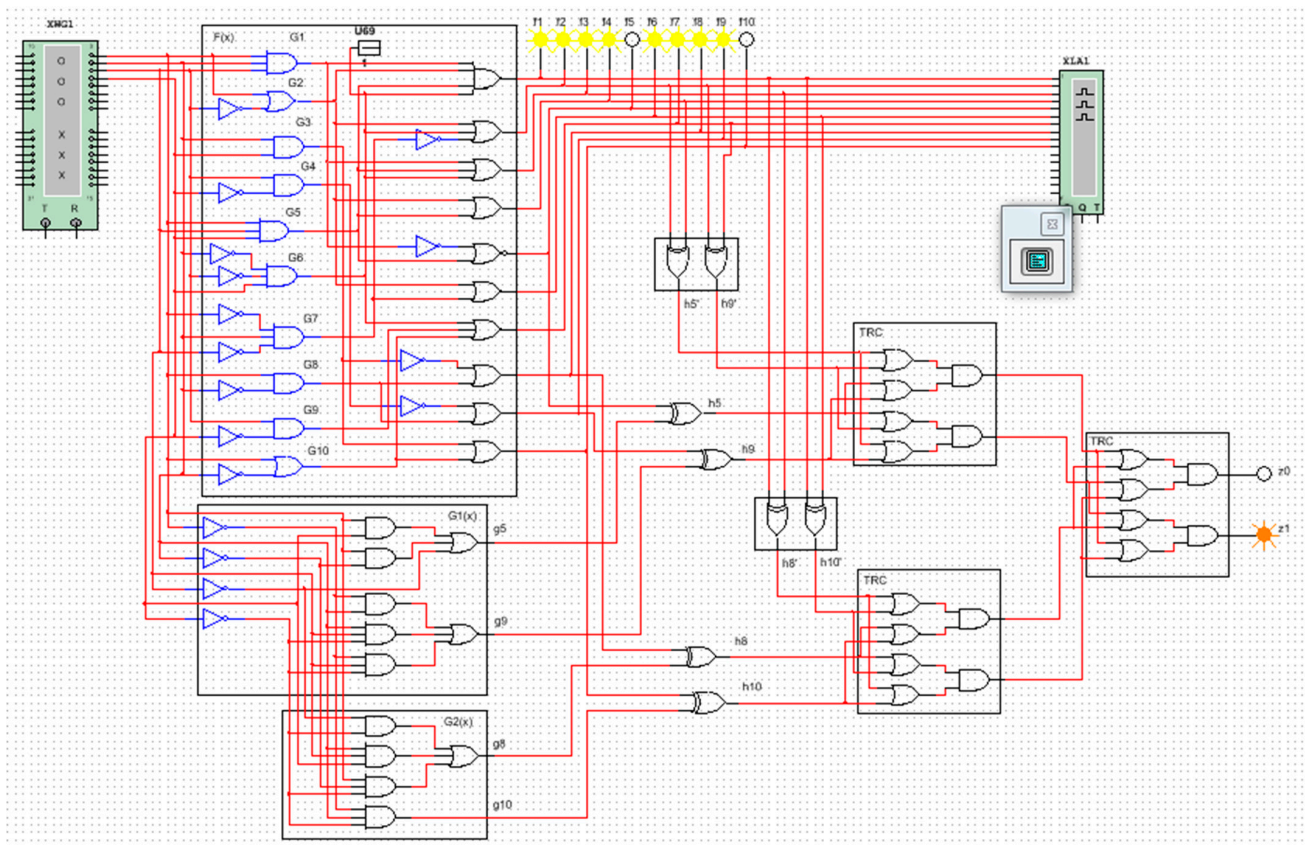


Figure 11. The ECC is modeled using the BSC method, with I^2 -output groups allocated and controlled by $P7$ code in the Multisim environment.

The experimental results indicate that the ECC, organized using the BSC method based on the $P13$ code, only fails to detect four three-fold unidirectional errors out of all possible errors. Meanwhile, the ECC constructed using the BSC method with the $P7$ code detects errors at the outputs of the diagnosed circuit.

9. Discussion

The conducted experiment leads to the conclusion that the use of the ECC is effective in detecting errors at the outputs of diagnosed circuits. To achieve this, the BSC method based on the $P13$ code ($P11$ code) should be employed, with a maximum error multiplicity on the working vectors of $d \leq 2$. Otherwise, some errors may go undetected. If the design satisfies the specified error coverage indicator based on the selected method, the ECC can be implemented in this manner. Otherwise, it may be necessary to implement an ECC based on the allocation of controllable groups of outputs (convertible and non-convertible). Alternatively, polynomial codes with a large value of k could be used.

The algorithms presented in the previous section should be considered for multi-output combinational devices. Further research is required to examine the characteristics of the ECC synthesis for these devices. Additionally, conducting experiments using popular benchmarks such as MCNC Benchmarks [33,34] would be beneficial in the future.

Furthermore, additional research areas may be linked to specific circuit approaches designed to address two objectives. The first objective is to increase error detection to full coverage. To accomplish this task, allocate controllable output groups and convert the device's structure into one with controllable outputs. The second task is to simplify the technical implementation of the ECC. The issue can be addressed through different methods, including pre-compressing signals from the diagnostic object, using a pulse mode of operation, and implementing additional control of the self-duality of functions calculated

in the ECC [35,36]. Further research could explore the detection of errors, including their types and multiplicities, in the entire code word through polynomial codes.

10. Conclusions

Polynomial codes are effective for organizing ECC using the BSC method. The selected polynomial codes should have the ability to detect errors with small multiplicities, including one- and two-fold errors, and occasionally three-fold and higher multiplicities.

Distortions can occur at any controlled output of the ECC structure, leading to simultaneous distortions in both the data and checking symbols of the polynomial code. Therefore, it is crucial to consider the error detection properties of various types and multiples in the entire code word of the polynomial code. However, circuit engineering techniques can be used to limit the spread of internal errors in the diagnostic object and categorize functionally autonomous outputs when organizing the ECC. It is recommended to allocate data and checking symbols from groups of functionally independent outputs. Errors can only occur in the data or checking symbols. In the checking symbols, any violation of the correspondence between the data and checking symbols will be detected upon manifestation. Errors in data symbols may not always be detected. In such cases, the error detection properties of the polynomial codes in data symbols, as studied in [26,29], should be applied. The paper's ECC synthesis algorithms enable the creation of structures that can detect errors in diagnostic object outputs with a certain probability of coverage, as well as fully self-checking structures.

The experiment involving modeling devices with the ECC implemented using the algorithms presented in this paper revealed that a small proportion of errors caused by malfunctions in the diagnostic object's structure may not be detected. However, increasing the number of checking symbols in code words always improves the detection capability. Complete error coverage at the outputs of diagnostic objects is always achievable.

The advantage of polynomial codes in implementing the ECC through the BSC method lies in their ability to ensure uniform distribution of data vectors amid control vectors. This feature enhances the ease of checking and testing conversion gates within the structure. One potential drawback is the need to locate convertible and non-convertible outputs. This task becomes increasingly complex as the total number of device outputs, the number of internal components, and the number of inputs increases. Furthermore, it is possible to identify one additional benefit of the application of polynomial codes in the synthesis of embedded control circuits based on Boolean signal correction in comparison with the use of constant-weight codes [4,5,16–18]. In the case of a large number of controlled device outputs, it will be considerably simpler to provide test generation for polynomial code testers than for constant-weight code testers. Concurrently, the structures of polynomial code testers are notably more straightforward than those of equilibrium code testers, which are implemented on the basis of standard XOR gates in the form of combinational circuits. It should be highlighted that the application of the methods developed in this paper to the synthesis of embedded control circuits in practice always results in the construction of self-checking devices. In many instances, such devices will be less redundant than other methods, including the standard duplication method.

It should be acknowledged that the application of the algorithms developed by the authors for the synthesis of embedded control circuits is constrained by their computational complexity. Given that there is a selection of groups of controllable outputs from a set of power n , the time complexity is asymptotically evaluated by the value $2^{O(n)}$. This implies that the problem is solved in exponential time with a linear exponent. In practice, the aforementioned limitation of modern computing power permits the implementation of the developed algorithms in devices with a maximum of 30–35 outputs. Nevertheless, methods of device decomposition and synthesis of cascaded embedded control circuits for groups of allocated outputs can be applied. It can be reasonably concluded that the developed algorithms have the potential for widespread application in the synthesis of self-checking digital devices and computing systems.

The use of polynomial codes when synthesizing the ECC through the BSC method is an approach that has practical efficacy in the deployment of fault detection devices with low hardware redundancies and high rates of error detection in calculations.

Author Contributions: Conceptualization, D.V.E.; methodology, D.V.E.; validation, D.V.E.; formal analysis, R.B.A. and D.V.E.; investigation, D.G.P. and D.V.E.; writing—original draft preparation, R.B.A. and D.V.E.; writing—review and editing, D.V.E. and M.V.B.; visualization, D.G.P.; supervision, D.V.E.; project administration, A.S.O. and G.S.V. All authors have read and agreed to the published version of the manuscript.

Funding: The research is funded by the Ministry of Science and Higher Education of the Russian Federation as part of the World-Class Research Center program: Advanced Digital Technologies (contract No. 075-15-2022-311 dated 20 April 2022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Drozd, A.; Kharchenko, V.; Antoshchuk, S.; Sulima, J.; Drozd, M. Checkability of the Digital Components in Safety-Critical Systems: Problems and Solutions. In Proceedings of the 9th IEEE East-West Design & Test Symposium (EWDTS'2011), Sevastopol, Ukraine, 9–12 December 2011; pp. 411–416. [\[CrossRef\]](#)
2. Drozd, O.; Perebeinos, I.; Martynyuk, O.; Zashcholkin, K.; Ivanova, O.; Drozd, M. Hidden Fault Analysis of FPGA Projects for Critical Applications. In Proceedings of the IEEE International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavsko, Ukraine, 25–29 February 2020. paper 142. [\[CrossRef\]](#)
3. Kharchenko, V.; Tyurin, S.; Fesenko, H.; Goncharovskij, O. The Fault Tolerant Černý Finite State Machine: A Concept and VHDL Models. In Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2021), Cracow, Poland, 22–25 September 2021; Volume 2, pp. 1163–1169. [\[CrossRef\]](#)
4. Goessel, M.; Morozov, A.V.; Sapozhnikov, V.V.; Sapozhnikov, V.V. Logic Complement, a New Method of Checking the Combinational Circuits. *Autom. Remote Control* **2003**, *64*, 153–161. [\[CrossRef\]](#)
5. Goessel, M.; Morozov, A.V.; Sapozhnikov, V.V.; Sapozhnikov, V.V. Checking Combinational Circuits by the Method of Logic Complement. *Autom. Remote Control* **2005**, *66*, 1336–1346. [\[CrossRef\]](#)
6. Sogomonyan, E.S.; Slabakov, E.V. *Self-Checking Devices and Fault-Tolerant Systems*; Radio & Communication; Morgan Kaufmann: Moscow, Russia, 1989; 208p. (In Russian)
7. Sellers, F.F.; Hsiao, M.-Y.; Bearnson, L.W. *Error Detecting Logic for Digital Computers*; McGraw-Hill: New York, NY, USA, 1968; 295p.
8. Berlekamp, E. *Algebraic Coding Theory*; Revised Edition; World Scientific Publishing Co Pte Ltd.: Singapore, 2015; 501p.
9. Piestrak, S.J. *Design of Self-Testing Checkers for Unidirectional Error Detecting Codes*; Oficyna Wydawnicza Politechniki Wrocławskiej; Wrocław, Poland, 1995; 111p.
10. Saposhnikov, V.V.; Saposhnikov, V.V. *Self-Checking Digital Devices*; Energoatomizdat: St. Petersburg, Russia, 1992; 224p. (In Russian)
11. Freiman, C.V. Optimal Error Detection Codes for Completely Asymmetric Binary Channels. *Data Control* **1962**, *5*, 64–71. [\[CrossRef\]](#)
12. Borden, J.M. Optimal Asymmetric Error Detecting Codes. *Informational Control* **1982**, *53*, 66–73. [\[CrossRef\]](#)
13. Berger, J.M. A Note on Error Detection Codes for Asymmetric Channels. *Informational Control* **1961**, *4*, 68–73. [\[CrossRef\]](#)
14. Efanov, D.; Sapozhnikov, V.; Sapozhnikov, V. Generalized Algorithm of Building Summation Codes for the Tasks of Technical Diagnostics of Discrete Systems. In Proceedings of the 15th IEEE East-West Design & Test Symposium (EWDTS'2017), Novi Sad, Serbia, 29 September–2 October 2017; pp. 365–371. [\[CrossRef\]](#)
15. Saposhnikov, V.V.; Saposhnikov, V.V.; Morozov, A.; Goessel, M.; Osadchy, G. Design of Totally Self-Checking Combinational Circuits by Use of Complementary Circuits. In Proceedings of the 2th IEEE East-West Design & Test Symposium (EWDTS'2004), Crimea, Ukraine, 15–17 September 2004; pp. 83–87.
16. Göessel, M.; Ocheretny, V.; Sogomonyan, E.; Marienfeld, D. *New Methods of Concurrent Checking*, 1st ed.; Springer Science+Business Media B.V.: Dordrecht, The Netherlands, 2008; 184p.
17. Sen, S.K. A Self-Checking Circuit for Concurrent Checking by 1-out-of-4 code with Design Optimization using Constraint Don't Cares. In Proceedings of the National Conference on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEERE 2010), Sikkim Manipal Institute of Technology, Majitar, Sikkim, India, 22–24 December 2010.

18. Das, D.K.; Roy, S.S.; Dmitriev, A.; Morozov, A.; Gössel, M. Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes. In Proceedings of the 10th International Workshops on Boolean Problems, Freiberg, Germany, 19–21 September 2012; pp. 33–40.
19. Morozov, M.; Saposhnikov, V.V.; Saposhnikov, V.V.; Goessel, M. New Self-Checking Circuits by Use of Berger-Codes. In Proceedings of the 6th IEEE International On-Line Testing Workshop, Palma de Mallorca, Spain, 3–5 July 2000; pp. 171–176.
20. Sogomonyan, E.S.; Gössel, M. Design of Self-Testing and On-Line Fault Detection Combinational Circuits with Weakly Independent Outputs. *J. Electron. Test. Theory Appl.* **1993**, *4*, 267–281. [[CrossRef](#)]
21. Busaba, F.Y.; Lala, P.K. Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors. *J. Electron. Test. Theory Appl.* **1994**, *5*, 19–28. [[CrossRef](#)]
22. Saposhnikov, V.V.; Morosov, A.; Saposhnikov, V.V.; Gössel, M. A New Design Method for Self-Checking Unidirectional Combinational Circuits. *J. Electron. Test. Theory Appl.* **1998**, *12*, 41–53. [[CrossRef](#)]
23. Morosow, A.; Saposhnikov, V.V.; Saposhnikov, V.V.; Goessel, M. Self-Checking Combinational Circuits with Unidirectionally Independent Outputs. *VLSI Des.* **1998**, *5*, 333–345. [[CrossRef](#)]
24. Matrosova, A.Y.; Levin, I.; Ostanin, S.A. Self-Checking Synchronous FSM Network Design with Low Overhead. *VLSI Des.* **2000**, *11*, 47–58. [[CrossRef](#)]
25. Ryan, W.E.; Lin, S. *Channel Codes: Classical and Modern*; Cambridge University Press: Cambridge, UK, 2009; 692p.
26. Efanov, D.; Plotnikov, D.; Sapozhnikov, V.; Sapozhnikov, V.; Abdullaev, R. Experimental Studies of Polynomial Codes in Concurrent Error Detection Systems of Combinational Logical Circuits. In Proceedings of the 16th IEEE East-West Design & Test Symposium (EWDTS'2018), Kazan, Russia, 14–17 September 2018; pp. 184–190. [[CrossRef](#)]
27. Borchert, C.; Schirmeier, H.; Spinczyk, O. Generic Soft-Error Detection and Correction for Concurrent Data Structures. *IEEE Trans. Dependable Secur. Comput.* **2017**, *14*, 22–36. [[CrossRef](#)]
28. Li, J.; Liu, S.; Reviriego, P.; Xiao, L.; Lombardi, F. Scheme for Periodical Concurrent Fault Detection in Parallel CRC Circuits. *IET Comput. Digit. Tech.* **2020**, *14*, 80–85. [[CrossRef](#)]
29. Abdullaev, R.; Efanov, D. Polynomial Codes Properties Application in Concurrent Error-Detection Systems of Combinational Logic Devices. In Proceedings of the 19th IEEE East-West Design & Test Symposium (EWDTS'2021), Batumi, Georgia, 10–13 September 2021; pp. 40–46. [[CrossRef](#)]
30. Efanov, D.; Sapozhnikov, V.; Sapozhnikov, V. The Synthesis of Self-Checking Combinational Devices Based on Properties of Codes with Summation of Weighted Transitions and Selection of Testable Outputs Groups. In Proceedings of the 15th IEEE East-West Design & Test Symposium (EWDTS'2017), Novi Sad, Serbia, 29 September–2 October 2017; pp. 671–677. [[CrossRef](#)]
31. Berezyuk, N.T.; Andrushchenko, A.G.; Moshchitsky, S.S.; Glushkov, V.I.; Benesha, M.M.; Gavrilov, V.A. *Data Coding (Binary Codes)*; Berezyuk, N.T., Ed.; Vishcha Shkola Publishing Association: Kharkov, Ukraine, 1978; 252p. (In Russian)
32. Efanov, D.V.; Abdullaev, R.B. Boolean Complement Method to Polynomial Codes for Combinational Circuits Testing. In Proceedings of the IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, Russia, 25–28 January 2022; pp. 139–144. [[CrossRef](#)]
33. Sentovich, E.M.; Singh, K.J.; Moon, C.; Savoj, H.; Brayton, R.K.; Sangiovanni-Vincentelli, A. Sequential Circuit Design Using Synthesis and Optimization. In Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers & Processors, Cambridge, MA, USA, 11–14 October 1992; pp. 328–333. [[CrossRef](#)]
34. Sentovich, E.M.; Singh, K.J.; Lavagno, L.; Moon, C.; Murgai, R.; Saldanha, A.; Savoj, H.; Stephan, P.R.; Brayton, R.K.; Sangiovanni-Vincentelli, A. *SIS: A System for Sequential Circuit Synthesis*; Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California: Berkeley, CA, USA, 4 May 1992; 45p.
35. Saposhnikov, V.I.V.; Dmitriev, A.; Goessel, M.; Saposhnikov, V.V. Self-Dual Parity Checking—A New Method for on Line Testing. In Proceedings of the 14th IEEE VLSI Test Symposium, Princeton, NJ, USA; 1996; pp. 162–168.
36. Lala, P.K. *Self-Checking and Fault-Tolerant Digital Design*; Morgan Kaufmann Publishers: San Francisco, CA, USA, 2001; 216p.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.