DOI: <https://doi.org/10.15407/emodel.43.06.061>
UDC 004.052.32+681.518.5

D.V. Efanov, D. Sc. (Tech.)

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

Russian Federation, 127994, Moscow, Obratsova str., build. 9/9
contact phone number (+7) (911) 7092164, e-mail: TrES-4b@yandex.ru

Transitions weight-based sum code for the digital computing devices synthesis

A method for constructing a family of sum codes is described based on weighting the transitions between groups of bits in the data vector. In this case, weights are used that are powers of the number 2. This makes it possible to obtain a code with check bits described by linear functions. The proposed weight-based sum code makes it possible to synthesize self-checking devices based on the standard elements and optimization methods of the logical device's structures. A standard structure of a device with a concurrent error-detection (CED) circuit based on transitions weight-based sum codes between bits groups in the data vector is presented. The standard structure advantage lies in the possibility of synthesizing CED circuits with technical implementation reduced complexity by using codes with check bits numbers that are much smaller than the data bits numbers. Self-checking devices synthesized using the described codes in some cases may turn out to be less redundant than when using the standard duplication structure. The structure disadvantage is the need to consider the restrictions on the multiplicity of errors arising at the outputs of the diagnostic objects. This limitation increases with a decrease in the number of check bits. Despite this, in many cases, the use of a standard structure based on transitions weight-based sum code between groups of digits in the data vector makes it possible to synthesize self-checking digital devices. Using a standard structure for organizing a CED circuit allows going to implement of fault-tolerant digital devices according to standard structures, one of which is given in this article.

Key words: self-checking built-in control circuit, checking of calculations, transitions weight-based sum code, detection of errors in the data vector, standard CED circuit structure, standard fault-tolerant device structure.

Modern automation and computer technology devices are developed and designed considering the possibilities of monitoring and localizing the faults and errors in the calculation results that occur in their structures [1-7]. In this case, an important component is the presence of CED circuits for blocks and sub-systems [8, 9]. As a rule, such circuits are implemented as self-checking and

© Efanov D.V., 2021

allow identifying not only errors in calculations by the diagnostic object itself but also its faults and errors at the outputs of internal elements [10].

The simplest way to synthesize CED circuits is to implement them based on the duplication principle. This implies the use of a copy of the diagnostic object and a comparison of the results of calculations at the outputs of the same name of both devices [11]. This approach has its merits. Firstly, the structure of the CED circuit is basic, and simulation of the operation of the diagnostic object is not required. This makes it easy to synthesize. Secondly, any combinations of faults are detected in the diagnostic object. However, a significant disadvantage is that the complexity of the technical implementation of a device with a CED circuit implemented by the duplication method is significant: the structural redundancy is 300-400% of the original device [12].

To reduce the complexity of the devices with a CED circuit technical implementation code methods are used [13, 14]. In this case, less redundant devices are synthesized since block uniform binary codes with low redundancy are selected as the basis for the CED circuit. As a rule, codes with a high code rate $R > 0,5$ are used. These can be classical Hamming codes and their various modifications oriented towards error correction [15], as well as Berger codes and their various modifications oriented towards error detection [16, 17]. The use of different codes implies the CED circuit with different characteristics synthesis. At the same time, various methods are used to achieve the detection of any errors combinations at the diagnosed objects outputs, taking into account the peculiarities of their internal structures and the selected fault model, concerning which the self-checking device is implemented [10].

When constructing a CED circuit, a different standard structure is often used [18]. It is based on the use of parity codes. At the same time, to build a self-checking computing device, it is required to use circuitry methods and checking groups of independent outputs or the initial synthesis of a device with a structure suitable for parity [3, 8]. This feature is associated with the properties of error detection by parity codes – they don't detect any error in the data vector with even multiplicity [16].

In [19], a standard structure is described, organized using a transitions weight-based sum code between bits occupying adjacent positions in data vectors. It possesses "good" detecting properties, however, it has a high structural redundancy of the final device. For example, for several devices, the implementation complexity is reduced by 20-25% compared to duplication.

Let's consider an approach to the transitions weight-based sum code construction that creates the possibility of synthesizing CED circuits using standard structures. It makes it possible to reduce the implementation complexity not only in comparison with the duplication method but also in comparison

with the structure mentioned above. In this case, the detecting properties deteriorate. However, in some cases, depending on the structure of the object being diagnosed and the selected model of malfunctions, this is not essential.

A **transitions weight-based sum code** was proposed in [20]. The method of its construction is based on weighing the transitions between the bits occupying adjacent positions in the data vector. This class of codes, as well as its modifications, are described in detail in [17], where it is indicated that a special way of constructing a code makes it possible to build a standard structure for organizing CED circuits for automation and computer equipment. This structure is described in [19].

Algorithm 1. Rules for determining the check bits values in a codeword of a transitions weight-based sum code:

1. Consider the bits of the data vector that occupy adjacent positions: f_i and f_{i+1} , $i = \overline{1, m-1}$ bits.

2. It is necessary to assign the weight coefficients $w_{i,i+1}$ from a series of increasing powers of the number 2 to the transitions between the bits of the data vector, starting with the least significant bit:

$$[w_{i,i+1}] = [w_{m-1,m}, w_{m-2,m-1}, \dots, w_{2,3}, w_{1,2}] = [2^{m-2}, 2^{m-3}, \dots, 2^1, 2^0].$$

3. Determine the transition activation functions values:

$$p_{i,i+1} = f_i \oplus f_{i+1}, \quad i = \overline{1, m-1}. \quad (1)$$

4. Calculating the total weight of active transitions:

$$W = \sum_{i=1}^{m-1} w_{i,i+1} p_{i,i+1}. \quad (2)$$

5. The resulting number is represented in binary and written in the bits of the check vector.

The code obtained by Algorithm 1 is denoted by the T_m -code, where m is the length of the data vector, and T is the type of the code (transitions weight-based code). T_m -code codewords have $k = m - 1$ check bit.

The weighting factors used in constructing the T_m -code are non-repeating powers of 2. Therefore, the values of the activation function (1) are written sequentially into the bits of the check vector:

$$\begin{aligned} g_1 &= p_1 = f_1 \oplus f_2; \\ g_2 &= p_2 = f_2 \oplus f_3; \\ &\dots\dots\dots \\ g_{m-1} &= p_{m-1} = f_{m-1} \oplus f_m. \end{aligned} \quad (3)$$

Fig. 1. T_6 -code encoder

The data vector is converted into a check vector:

$$\langle f_m f_{m-1} \dots f_2 f_1 \rangle \rightarrow \langle g_{m-1} g_{m-2} \dots g_2 g_1 \rangle.$$

Consider the process of obtaining the values of the T_6 -code check vector bits for the data vector $\langle 101001 \rangle$:

$$p_{1,2} = f_1 \oplus f_2 = 1 \oplus 0 = 1,$$

$$p_{2,3} = f_2 \oplus f_3 = 0 \oplus 0 = 0,$$

$$p_{3,4} = f_3 \oplus f_4 = 0 \oplus 1 = 1, \quad p_{4,5} = f_4 \oplus f_5 = 1 \oplus 0 = 1,$$

$$p_{5,6} = f_5 \oplus f_6 = 0 \oplus 1 = 1; \quad \langle g_5 g_4 g_3 g_2 g_1 \rangle = \langle 11101 \rangle.$$

When constructing a T_m -code, you don't need to use points 4 and 5 of Algorithm 1 and real weighting. It is a linear systematic code whose check vector bits can be defined in matrix form. For the considered T_6 -code, the matrix looks like this:

$$M_{T_6} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

The matrix columns correspond to the data vector bits (from left to right from the least significant bit to the most significant). The matrix rows correspond to the check vector bits (from top to bottom from the least significant bit to the most significant). Units indicate the data bits added modulo $M = 2$ (XOR gate) for each check bit. Any T_m -code encoder has a standard implementation. For the T_6 -code, it is shown in Fig. 1. The T_m -code has an important feature, which determines the possibility of its effective use in the synthesis problems of the completely self-checking CED circuit: T_m -codes detect any errors in data vectors, except for errors with multiplicities $d = m$ [19].

To prove this property, consider an error in the data vector with multiplicity $d = m$. In this case, all bits of the data vector are inverted. The encoder calculates the following functions (see (2)):

$$g_1 = p_1 = \overline{f_1} \oplus \overline{f_2} = f_1 \oplus f_2;$$

$$g_2 = p_2 = \overline{f_2} \oplus \overline{f_3} = f_2 \oplus f_3;$$

.....

$$g_{m-1} = p_{m-1} = \overline{f_{m-1}} \oplus \overline{f_m} = f_{m-1} \oplus f_m.$$

(4)

System (4) is proof of the fact that the values of all check functions will not change when $d = m$ bits are distorted. If $d = a$ ($a < m$) bits are distorted, then $m - a$ bits are not inverted. In this case, at least one check bit is distorted. An entire codeword with a distorted check vector will not match a T_m -code word. The error will be detected.

The number of errors with multiplicity $d = m$ is equal to 2^m , and they are all undetectable T_m -codes. For example, if talking about any stuck-at faults in some combinational device, then distortions with multiplicities $d = m$ will not occur at its outputs if there are no paths from any internal gate to all outputs of the device. The condition for the absence of simultaneous distortions of all outputs are written in the form:

$$\frac{\partial f_1}{\partial y_l} \frac{\partial f_2}{\partial y_l} \dots \frac{\partial f_m}{\partial y_l} = 0, \quad (5)$$

where y_l is the function implemented at the output of the gate G_l , connected by paths with all outputs of the device. If the expression on the left side of (5) is equal 0, then there is not a single input combination on which errors are transmitted to all outputs of the device.

For real devices, there may be no internal connections not only with all external outputs but also with a smaller number of them. Thus, the idea of modifying the T_m -code arises.

The modified transitions weight-bits sum code is constructed according to the following rules.

Algorithm 2. Rules for determining the check bits values in the codeword of the modified transitions weight-bits sum code:

1. Consider q bits of the data vector occupying consecutively adjacent positions: bits $f_i, f_{i+1}, \dots, f_{i+q}, i = \overline{1, m-q+1}$.

2. Assign the weight coefficients $w_{i,i+1}$ from a series of increasing powers of the number 2 to the transitions between the bits of the data vector, starting with the least significant bit:

$$\begin{aligned} [w_{i-(i+q)}] &= [w_{(m-q+1)-m}, w_{(m-q)-(m-1)}, \dots, w_{2-(q+1)}, w_{1-q}] = \\ &= [2^{m-q}, 2^{m-q-1}, \dots, 2^1, 2^0]. \end{aligned}$$

3. Using the formula (1), determine the values of the transition activation functions.

4. Using the formula (2), calculate the total weight of active transitions.

5. The resulting number is represented in binary and written in the bits of the check vector.

Items 4 and 5 of Algorithm 2 can be replaced with one item: the obtained values of the transition activation functions are written into the code check vec-

Fig. 2. T_6^3 -code encoder (a) and T_6^4 -code encoder (b)

tor bits. The code obtained by Algorithm 2 will be denoted by the T_m^q -code. Its codewords will have $k = m - q + 1$ check bits.

The T_m -code considered above is a special case of a modified transitions weight-bits sum code for $q = 2$, i.e., T_m^2 -code.

The code vectors bits values for T_m^3 -codes are obtained by the formulas:

$$\begin{aligned} g_1 = p_1 &= f_1 \oplus f_2 \oplus f_3; \\ g_2 = p_2 &= f_2 \oplus f_3 \oplus f_4; \\ &\dots\dots\dots \\ g_{m-2} = p_{m-2} &= f_{m-2} \oplus f_{m-1} \oplus f_m. \end{aligned} \quad (6)$$

System (6) can also be represented in matrix form. For example, for a T_6^3 -code, have the following matrix:

$$M_{T_6^3} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \quad (7)$$

The T_6^3 -code encoder is shown in Fig. 2, a. The bit values of the code vectors for T_m^4 -codes are obtained by the formulas:

$$\begin{aligned} g_1 = p_1 &= f_1 \oplus f_2 \oplus f_3 \oplus f_4; \\ g_2 = p_2 &= f_2 \oplus f_3 \oplus f_4 \oplus f_5; \\ &\dots\dots\dots \\ g_{m-3} = p_{m-3} &= f_{m-3} \oplus f_{m-2} \oplus f_{m-1} \oplus f_m. \end{aligned} \quad (8)$$

$$\mathbf{M}_{T_6^4} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \quad (9)$$

$$\begin{aligned} g_1 &= p_1 = f_1 \oplus f_2 \oplus \dots \oplus f_{q-1} \oplus f_q; \\ g_2 &= p_2 = f_2 \oplus f_3 \oplus \dots \oplus f_q \oplus f_{q+1}; \\ &\vdots \\ g_{m-a+1} &= p_{m-a+1} = f_{m-a+1} \oplus f_{m-a+2} \oplus \dots \oplus f_{m-1} \oplus f_m. \end{aligned} \tag{10}$$

$$\mathbf{M}_{T^q_m} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & \dots & 1 & 1 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 & 1 & 1 & \dots & 1 \end{pmatrix}.$$

noted that the case $q = 2$ is special since it allows constructing codes with the detection of any errors with multiplicities $d \leq m - 1$. Nevertheless, T_m^q -codes for $q > 2$ can also be applied in practice. In this case, a choice of code is required taking into account the characteristics of the automation and computer equipment for which the CED circuit will be built.

The generalized standard CED circuit structure. The use of the T_m^q -code allows us to propose a standard CED circuit structure for automation and computing devices [19]. The CED circuit has three functional blocks.

The first block, $G(x)$, implements the check functions system $g_1, g_2, \dots, g_{m-q}, g_{m-q+1}$ based on the original object $F(x)$ input combinations values.

The second block, $G(f)$, implements the check functions system $g'_1, g'_2, \dots, g'_{m-q}, g'_{m-q+1}$ based on the object $F(x)$ output combinations values.

Characteristics of error detection in data vectors by T_m^3 - and T_m^4 -codes

m	k	Distribution of undetectable errors by a multiplicity							Total undetectable errors
		1	2	3	4	5	6	7	
T^3_m -codes									
4	2	$\frac{0}{0}$	$\frac{16}{16,667}$	$\frac{32}{50}$	$\frac{0}{0}$	—	—	—	$\frac{48}{20}$
5	3	$\frac{0}{0}$	$\frac{0}{0}$	$\frac{64}{20}$	$\frac{32}{20}$	$\frac{0}{0}$	—	—	$\frac{96}{9,677}$
6	4	$\frac{0}{0}$	$\frac{0}{0}$	$\frac{0}{0}$	$\frac{192}{20}$	$\frac{0}{0}$	$\frac{0}{0}$	—	$\frac{192}{4,762}$
7	5	$\frac{0}{0}$	$\frac{0}{0}$	$\frac{0}{0}$	$\frac{128}{2,857}$	$\frac{256}{9,524}$	$\frac{0}{0}$	$\frac{0}{0}$	$\frac{384}{2,362}$
T^4_m -codes									
4	2	$\frac{0}{0}$	$\frac{96}{100}$	$\frac{0}{0}$	$\frac{16}{100}$	—	—	—	$\frac{112}{46,667}$
5	3	$\frac{0}{0}$	$\frac{96}{30}$	$\frac{96}{30}$	$\frac{0}{0}$	$\frac{32}{100}$	—	—	$\frac{224}{22,581}$
6	4	$\frac{0}{0}$	$\frac{64}{6,667}$	$\frac{256}{20}$	$\frac{64}{6,667}$	$\frac{0}{0}$	$\frac{64}{100}$	—	$\frac{448}{11,111}$
7	5	$\frac{0}{0}$	$\frac{0}{0}$	$\frac{384}{8,571}$	$\frac{384}{8,571}$	$\frac{0}{0}$	$\frac{0}{0}$	$\frac{128}{100}$	$\frac{896}{5,512}$

Note: above the line, the absolute number of undetectable errors is indicated, below the line – the percentage (%) of undetectable errors from the total number.

 Fig. 3. The CED circuit structure based on the T_m^q -code

The third block, TRC , is a comparator and is designed to compare signals from the same-named outputs $G(x)$ and $G(f)$ blocks. It is implemented on the standard modules cascade connection for compression of two-rail signals (two-rail checkers) [21, 22]; therefore, the values from the outputs of the $G(x)$ block are pre-inverted. The comparator is equipped with two check outputs, z^0 and z^1 , at which, in the absence of faults and errors in calculations, a two-rail signal is realized. Two-rail violation indicates the presence of malfunctions in the operation of the $F(x)$ device or the CED circuit.

All blocks are shown in Fig. 3 CED circuits based on them are standard. Block $G(f)$ is implemented based on rules (10) using standard XOR 's. The TRC block is implemented using standard two-rail compression elements. Block $G(x)$ is implemented in practice by optimizing the structure shown in Fig. 4 by standard methods [23-25].

Thus, when implementing a self-checking computing device according to the structure shown in Fig. 3, only the synthesis of the $G(x)$ block is required. The rest of the blocks are typical for a device with m outputs and the corresponding T_m^q -code.

Fig. 4. The $G(x)$ block structure in the CED circuit

The generalized standard correction circuit structure. The use of the T_m^q -code allows us to propose a generalized standard correction circuits structure for automation and computer devices (Fig. 5). A similar structure, when used to checking computations of parity codes, is described in [26], and T_m^2 -codes (T_m -codes) — in [27, 28].

In the structure shown in Fig. 5, to detect errors at the original object $F(x)$ outputs, an additional device $F^*(x)$, is used, which is an original device copy (or a device that calculates similar functions but has a different architecture), and an XOR gates $a_1, a_2, \dots, a_{m-1}, a_m$ to detect outputs with errors. If an error is detected at any of the outputs, then a single value will be present at the corresponding output of the comparison gate. Such circuit arrangement allows detecting any manifestations of single faults in the form of distortions of values at the object $F(x)$ outputs.

To determine the device at the outputs of which an error has occurred, the CED circuit using the T_m^q -code is used. In this case, checking calculations is carried out at the diagnostic object $F(x)$ outputs. Unlike the structure considered in [26], where the built-in control circuit is synthesized by the parity method [18], in the generalized structure, the code control allows detecting a much larger number of errors at the outputs of the $F(x)$ block.

The CED circuit outputs generate a signal of no error $z = z^0 \oplus z^1$. Therefore, it is inverted at the signal correction block inputs.

The error correction block includes gates for deciding on correction (AND gates) and signals correction gates (XOR gates $b_1, b_2, \dots, b_{m-1}, b_m$). The gates for deciding on correction have two inputs. The first inputs receive signals $e_1, e_2, \dots, e_{m-1}, e_m$ from the comparison gates $a_1, a_2, \dots, a_{m-1}, a_m$ and determine those outputs at which it is necessary to correct the signals. The second outputs receive signals $u_1, u_2, \dots, u_{m-1}, u_m$ from the built-in control circuit: $u_1 = u_2 = \dots = u_{m-1} = u_m = z$. Correction is carried out if $z = 0$. In this case, the

Fig. 5. The correction circuit structure based on the T_m^q -code

values at the CED circuit two-rail outputs coincide, indicating errors in the calculations.

The structure shown in Fig. 5, is not protected from faults in the signal correction gates unit; however, the traditional structure with majority signal correction has the same drawback [26]. In practice, this drawback is levelled using highly reliable gates in the signal correction unit synthesis with low failure rates and internal redundancy and implemented in completely self-checking devices.

When organizing a CED circuit using a T_m^q -code, the correction structure has a high correcting ability at the initial diagnostic object outputs, and the number of the detected errors in the CED circuit is determined by the properties of the diagnostic object structure and the selected T_m^q -code.

The use of T_m^q -codes is also possible in the fault-tolerant structures synthesis by the Boolean complement method, where the additional device $F^*(x)$ is replaced by a simpler block for fixing distorted signals. A fault-tolerant device implementation variant with any Boolean complement calculations check block in the block for fixing the distorted signals is possible. Such structures are described in detail in [29]. They use the least redundant parity code to

check the calculations. Replacing the parity code with a T_m^q -code increases the detecting ability of the built-in control and somewhat increases the final device's structural redundancy. This circumstance should be considered when implementing digital computing systems, considering the specifics of the original object.

Conclusions

In the well-known works [12, 17, 19, 20], it is proposed to build sum codes based on weighting the principle of the transitions between bits occupying adjacent positions in data vectors. In fact, about establishing a certain weighting factor for a pair of bits for which the function $p_{i,i+1} = f_i \oplus f_{i+1}$, $i = \overline{1, m-1}$ is calculated. Any pairs can be considered (not necessarily for bits with adjacent positions in the data vector) and the function $p_{i_1, i_2} = f_{i_1} \oplus f_{i_2}$, $i_1, i_2 \in \{1, 2, \dots, m\}$ is calculated. In this case, all data vector bits must be covered. When generalizing this method, it should be considered that the discharges can be repeated in pairs and the transition can be weighed not between a pair of discharges, but between a group (not necessarily arranged in a row in the data vector) discharges.

The transitions weight-based sum code when establishing weight coefficients from the set $\{2^0, 2^1, \dots, 2^\gamma\}$ (γ is a natural number) become classical linear codes and can be specified in matrix form. The proposed method for constructing T_m^q -codes allows one to construct codes with standard encoders, implemented based on *XOR*'s, which can detect any errors in the region of small multiplicity d (including two- and three-fold errors) for certain lengths of data vectors.

These two features of T_m^q -codes indicate the possibility of their use in the CED circuit synthesis. In this case, such a CED circuit structure will be standard since all its blocks will be standard, and the check logic block will be executed based on standard optimization methods for combinational circuits. This is the advantage of using T_m^q -codes when synthesizing CED circuits. A disadvantage can be considered the need to search for the best T_m^q -code when organizing CED circuit for a specific device $F(x)$, for which it will be necessary to analyze the internal structure and determine the types and multiplicity of errors that occur.

This procedure can turn out to be very laborious and is determined by the diagnostic object inputs number and the considered model of faults. For example, for stuck-at faults in the diagnostic object internal structure gates outputs, the analysis process complexity is determined by the number of gates in

the structure N_y and the number of inputs t . The complexity of the analysis process can be estimated by an expression of form $2N_y 2^t = 2^{t+1} N_y$. With the number of inputs $t > 20$ analyzing the diagnostic object structure using computing means takes a long time. The use of decomposition methods is required.

The considered advantage T_m^q -codes and structures of CED circuits based on them is that the implementation of devices with fault detection is possible on any gate base and does not require binding to the fault model. The detecting ability of CED circuit and the possibility of its implementation in the form of a self-checking structure must be evaluated in each specific case. Devices with signal correction circuits can be implemented based on T_m^q -codes. In this case, the CED circuit will be included in their structures, and the correcting ability will be determined by the properties of the structure of the diagnosed object and the T_m^q -code selected to checking the calculations.

REFERENCES

1. Matrosova, A.Yu. (1990), *Algoritmicheskie metody sinteza testov* [Algorithmic methods for the synthesis of tests], Publishing house of Tomsk University, Tomsk, USSR.
2. Wang, L.-T., Wu, C.-W. and Wen, X. (2006), *VLSI Test Principles and Architectures: Design for Testability*, Morgan Kaufmann Publishers, San Francisco, USA.
3. Göessel, M., Ocheretny, V., Sogomonyan, E. and Marienfeld, D. (2008), *New Methods of Concurrent Checking: Edition 1*, Springer Science+Business Media B.V., Dordrecht, Netherlands.
4. Navabi, Z. (2011), *Digital System Test and Testable Design: Using HDL Models and Architectures*, Springer Science+Business Media.
5. Drozd, A.V., Kharchenko, V.S. and Antoshchuk, S.G. (2012), *Rabochee diagnostirovanie bezopasnykh informatsionno-upravljayustchikh sistem* [Objects and Methods of On-Line Testing for Safe Instrumentation and Control Systems], National Aerospace University "KhAI", Kharkov, Ukraine.
6. Hahanov, V. (2018), *Cyber Physical Computing for IoT-driven Services*, Springer International Publishing AG, New York, USA, DOI: 10.1007/978-3-319-54825-8.
7. Mikoni, S.V., Sokolov, B.V. and Yusupov, R.M. (2018), *Kvalimetriya modelej i polimodel'nyh kompleksov* [Qualimetry of models and polymodel complexes], Academy of Science, Moscow, Russia.
8. Sogomonyan, E.S. and Slabakov, E.V. (1989), *Samoproverjaemyje ustrojstva i otkazoustojchivyye sistemy* [Self-checking devices and failover systems], Radio i Svjaz, Moscow, USSR.
9. Mikoni, S.V. (1992), *Obshchie diagnosticheskie bazy znaniy vychislitel'nyh sistem* [General Diagnostic Knowledge Base of Computing Systems], SPIIRAN, St. Petersburg, Russia.
10. Parkhomenko, P.P. and Sogomonyan, E.S. (1981), *Osnovy tekhnicheskoy diagnostiki (optimizatsiya algoritmov diagnostirovaniya, apparaturnyye sredstva)* [Basics of technical diagnostics (optimization of diagnostic algorithms and equipment)], Energoatomizdat, Moscow, USSR.
11. Goessel, M. and Graf, S. (1994), *Error Detection Circuits*, McGraw-Hill, London, UK.

12. Sapozhnikov, V.V., Sapozhnikov, V.I. and Efanov, D.V. (2019), *Osnovy teorii nadezhnosti i tekhnicheskoy diagnostiki* [Fundamentals of the theory of reliability and technical diagnostics], Publishing House "Lan", St. Petersburg, Russia.
13. Nicolaidis, M. and Zorian, Y. (1998), "On-Line Testing for VLSI – A Compendium of Approaches", *Journal of Electronic Testing: Theory and Application*, Vol. 12, no. 1-2, pp. 7-20, DOI: 10.1023/A:1008244815697.
14. Mitra, S. and McCluskey, E.J. (2000), "Which Concurrent Error Detection Scheme to Choose?", *Proceedings of International Test Conference*, Atlantic City, NJ, USA, October 03-05, 2000, pp. 985-994.
15. Sapozhnikov, V.V., Sapozhnikov, V.I. and Efanov, D.V. (2018), *Kody Khemminga v sistemakh funktsionalnogo kontrolja* [Hamming Codes in Concurrent Error Detection Systems of Logic Devices], Nauka, St. Petersburg, Russia.
16. Sapozhnikov, V.V., Sapozhnikov, V.I. and Efanov, D.V. (2020), *Kody s summirovaniem dlya sistem tekhnicheskogo diagnostirovaniya. Tom 1: Klassicheskie kody Bergera i ih modifikacii* [Sum Codes for Technical Diagnostics Systems. Volume 1: Classical Berger Codes and Their Modifications], Nauka, Moscow, Russia.
17. Sapozhnikov, V.V., Sapozhnikov, V.I. and Efanov, D.V. (2021), *Kody s summirovaniem dlya sistem tekhnicheskogo diagnostirovaniya. Tom 2: Vzveshennyye kody s summirovaniem* [Sum Codes for Technical Diagnostics Systems. Volume 2: Weight-Based Sum Codes], Nauka, Moscow, Russia.
18. Aksjonova, G.P. (1973), "Method of Synthesizing Built-in Monitoring Arrangements for Automata with Memory", *Avtomatika i telemekhanika*, 1973, Vol. 34, no. 2, pp. 267-273.
19. Sapozhnikov, V.V., Sapozhnikov, V.I., Efanov, D.V. and Dmitriev, V.V. (2017), "New Structures of the Concurrent Error Detection Systems for Logic Circuits", *Avtomatika i telemekhanika*, Vol. 78, no. 2, pp. 300-312, DOI: 10.1134/S0005117917020096.
20. Sapozhnikov, V. and Sapozhnikov, V.I. (1999), "New Code for Fault Detection in Logic Circuits", *Trudy 4-y konferentsii po netradsitsionnym elektromekhanicheskim i elektricheskim sistemam* [Proceedings of 4th International Conference on Unconventional Electromechanical and Electrical Systems], St. Petersburg, Russia, June 21-24, 1999, pp. 693-696.
21. Nikolos, D. (1998), "Self-Testing Embedded Two-Rail Checkers", *On-Line Testing for VLSI*, Ch. 7, pp. 69-79, DOI 10.1007/978-1-4757-60-69-9_7.
22. Lala, P.K. (2001), *Self-Checking and Fault-Tolerant Digital Design*, Morgan Kaufmann Publishers, San Francisco, USA.
23. Zakrevskij, A., Pottosin, Yu. and Cheremisinova, L. (2009), *Optimization in Boolean Space*, TUT Press, Tallinn, Estonia.
24. Sentovich, E.M., Singh, K.J., Moon, C., Savoj, H., Brayton, R.K. and Sangiovanni-Vincentelli, A. (1992), "Sequential Circuit Design Using Synthesis and Optimization", *Proceedings IEEE International Conference on Computer Design: VLSI in Computers & Processors*, October 11-14, 1992, Cambridge, MA, USA, pp. 328-333, DOI: 10.1109/ICCD.1992.276282.
25. Sentovich, E. M., Singh, K. J., Lavagno L. et al (1992), *SIS: A System for Sequential Circuit Synthesis*, Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkeley, USA.
26. Sapozhnikov, V.V., Sapozhnikov, V.I. and Efanov, D.V. (2020), "Signal Correction Circuit for Combinational Automation Devices on the Basis of Boolean Complement with Control of Calculations by Parity", *Informatica*, Vol. 17, no. 2, pp. 71-85, available at: <https://doi.org/10.37661/1816-0301-2020-17-2-71-85>.

27. Sapozhnikov, V., Sapozhnikov, V.I. and Efanov, D. (2020), "Typical Signal Correction Structures Based on Duplication with the Integrated Control Circuit", Proceedings of 18th IEEE East-West Design & Test Symposium (EWDTS'2020), Varna, Bulgaria, September 4-7, 2020, pp. 78-87, DOI: 10.1109/EWDTS50664.2020.9224649.
28. Efanov, D.V., Sapozhnikov, V.V. and Sapozhnikov, V.I. (2020), "Typical Structure of a Duplicate Error Correction Scheme with Code Control with Summation of Weighted Transitions", *Elektronne modelyuvannya*, Vol. 42, no. 5, pp. 38-50, DOI: 10.15407/emodel.42.05.038.
29. Efanov, D.V., Sapozhnikov, V.V. and Sapozhnikov, V.I. (2021), "Boolean-Complement Based Fault-Tolerant Electronic Device Architectures", *Avtomatika i telemekhanika*, Vol. 82, no. 8, pp. 1403-1417, DOI: 10.1134/S0005117921080075.

Received 26.07.21

Д.В. Єфанов

КОДИ З ПІДСУМОВУВАННЯМ ВАГОВИХ
КОЕФІЦІЄНТІВ ГРУП РОЗРЯДІВ ІНФОРМАЦІЙНОГО ВЕКТОРА
ДЛЯ СИНТЕЗУ ЦИФРОВИХ ОБЧИСЛЮВАЛЬНИХ ПРИСТРОЇВ

Описано спосіб побудови сімейства кодів з підсумовуванням, оснований на зважуванні переходів між групами розрядів у інформаційному векторі. При цьому використано вагові коефіцієнти, що є степенями числа 2. Це дозволяє отримати код з контрольними розрядами, які описуються лінійними функціями. Застосування запропонованих зважених кодів з підсумовуванням дає змогу синтезувати схеми вбудованого контролю (СВК) на основі стандартних елементів і стандартних методів оптимізації структур логічних пристроїв. Наведено стандартну структуру СВК, основу на використанні кодів з підсумовуванням зважених переходів між групою розрядів інформаційного вектора. Перевага стандартної структури полягає в можливості синтезу СВК із зменшеною складністю технічної реалізації за допомогою використання кодів з меншим числом контрольних розрядів, ніж при використанні дублювання. Недоліком структури є необхідність враховувати обмеження на кратності помилок, які виникають на виходах об'єктів діагностування, що посилюються при зменшенні числа контрольних розрядів. Незважаючи на це в багатьох випадках застосування стандартної структури на основі кодів з підсумовуванням зважених переходів між групами розрядів у інформаційному векторі дозволяє синтезувати самоперевіряємі цифрові пристрої. Використання стандартної структури для організації СВК дає також змогу реалізувати відмовостійкі цифрові пристрої по стандартним структурам.

К л ю ч о в і с л о в а: самоперевіряєма схема вбудованого контролю, контроль обчислень, код з підсумовуванням вагових переходів, виявлення помилок в інформаційному векторі, стандартна структура схеми вбудованого контролю, стандартна структура відмовостійкого пристрою.

DMITRY V. Efanov, Doctor of Technical Sciences, Associate professor, Deputy General Director for Research and Development of LLC STC "Integrated Monitoring Systems", Professor at Higher School of Transport of Institute of Machinery, Materials, and Transport of St. Petersburg and Peter the Great St. Petersburg Polytechnic University, Professor at Automation, Remote Control and Telecommunication on Railway Transport of Russian University of Transport. Graduation: Emperor Alexander I St. Petersburg State Transport University, 2007. The specialist in the area of discrete mathematics, reliability and technical diagnostics of discrete systems, synthesis of fault-tolerant and safety control systems, methods of monitoring transport systems.