Optimum Sum Codes, that Effectively Detect the Errors of Low Multiplicities

Valery Sapozhnikov, Vladimir Sapozhnikov, Dmitry Efanov, Vyacheslav Dmitriev, Maria Cherepanova

Automation and Remote Control on Railways Department,
Petersburg State Transport University, Russian Federation

***Abstract* – The article provides the method of formation of the sum code with minimum total number of undetectable errors. The code, suggested by authors, has the same number of check bits, as a classic Berger code, but also has a better detection ability, particularly within the area of low multiplicity errors in data vectors. New sum code allows to develop concurrent error detection (CED) systems of logic units of automation and computer devices with low equipment redundancy and a high percentage of error detection in controlled blocks.**

I. Introduction

Sum codes are often used for data transmission and processing systems, as well as for design of reliable systems of automation and computer devices [1-9]. One of the examples of application of sum codes is a concurrent error detection (CED) system of arithmetical-logical units, that make a part of any modern systems of automation control [10-14].

The CED system structure is presented in Fig. 1. There, the initial arithmetical-logical unit $F(x)$, that realizes the system of operational Boolean functions $f_1(x)$, $f_2(x)$, …, $f_m(x)$, is equipped with the special control equipment. The control equipment includes the reference logic block $G(x)$, calculating the values of test functions $g_1(x)$, $g_2(x)$, …, $g_k(x)$, and the self-checking checker, that registers the conformance of the values of operational and test functions at any given time. This conformance is established at the CED system design stage and usually determined according to the rules of formation of preselected sum code. So the block $F(x)$ outputs are matched with the data vector of length m, and the block $G(x)$ outputs – with the check vector of length k. Those significant parameters of the CED system as detection ability and equipment redundancy are substantially depend on the rules of code formation, and the latter, in its turn, effects the power consumption, processing speed, testability and other characteristics of built discrete system [15,16].

The structure, shown in Fig. 1, in practice is built under condition of 100% detection of single faults [1,17]. By the reason of separate implementation of the blocks $F(x)$, $G(x)$ and checker at any given time the fault can occur only in one element of CED system. The reference logic block faults distort the values of test functions, that is registered by the checker. The checker, as the watchdog in CED system, is built as a self-checking device and thus detects its own faults at leas in one input set [1]. The CED system task is to provide the detection of single faults in block $F(x)$, when the internal configuration of links between logic elements within it can result in occurrence of distortions of different multiplicities in the values of output vector. Therefore, it is possible to consider the characteristics of sum code based on error detection in data vectors, studying by this the features of the CED system itself.

Different sum codes have different characteristics of error detection in data vectors and allow to build systems with various technical specifications. This paper is dedicated to the study of sum codes, that have minimum total number of undetectable errors in data vectors, as well as decreased number of double undetectable errors in comparison with classic sum codes.

II. Analysis of Berger codes and sum codes with weighted transitions properties for error detection in data vectors

Classic sum code, or Berger code [18], is built based on the following principle: data vector weight r (the sum of one data bits), and then the obtained value is presented in binary form and recorded into the bits of the check vector. The number of check bits in Berger code is calculated using the formula: $k = \lceil \log_2(m+1) \rceil$ (notation $\lceil ... \rceil$ is an integer, upward to the calculating value). Hereafter referred the Berger code to as $S(m,k)$-code.

In $S(m,k)$-code the same check vector corresponds to all data vectors with the same weight. For this value r is C_m^r of data vectors. With increasing of r the number of data vectors, corresponding to the same check vector, increases with its maximum at $\frac{m}{2}$ for the even m and at $\frac{m-1}{2}$ for the odd m, and then grows away. Such distribution of data vectors between the check vectors results in the fact, that $S(m,k)$-code does not detect significantly large amount of errors in data vectors. For example, $S(5,3)$-code does not detect 220 errors in data vectors, that forms 22.18% of all possible errors in data vectors (i.e. this code does not detect slightly less than one fourth of errors in data vectors). With low value of detection ability all $S(m,k)$-codes also have low efficiency of low multiplicities error detection – these codes do not detect 50% of double errors, 37.5% of quadruple errors, 31.25% six-fold errors, etc. [19].

Manuscript received January 18, 2015.

Valery Sapozhnikov is with the Petersburg State Transport University, Automation and Remote Control on Railways Department, Russian Federation.

Vladimir Sapozhnikov is with the Petersburg State Transport University, Automation and Remote Control on Railways Department, RF.

Dmitry Efanov is with the Petersburg State Transport University, Automation and Remote Control on Railways Department, RF (corresponding author to provide e-mail: mitriche@yandex.ru).

Vyacheslav Dmitriev is with the Petersburg State Transport University, Automation and Remote Control on Railways Department, RF.

Maria Cherepanova is with the Petersburg State Transport University, Automation and Remote Control on Railways Department, RF.

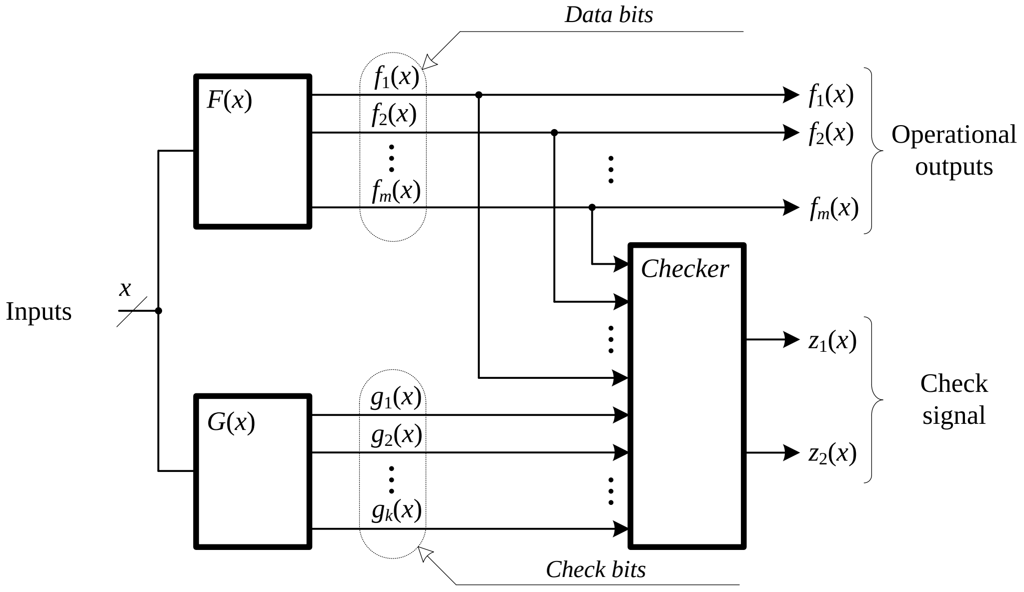

Fig. 1. CED system structure

So the problem of increasing the detection abiliity arises, which includes the detection ability within the area of low multiplicity errors, on condition of keeping the number of check bits in the code. Practically speaking, for example, in CED systems the solution of this problem will allow to detect the bigger number of single faults of controlled logic unit without increasing the equipment redundancy.

In [20] the term of an optimum separable code by the criterion of minimum number of undetectable errors for the specified values of *m* and *k* was introduced. The optimum code has a uniform distribution of all 2^m data vectors among 2^k check vectors and does not detect

$$N_{m,k}^{\min} = 2^m\left(2^{m-k}-1\right), \qquad (1)$$

errors in data vectors. For instance, the optimum code, where *m*=5 and *k*=3, has $N_{5,3}^{\min} = 2^5\left(2^{5-3}-1\right) = 32 \cdot 3 = 96$ undetectable errors, i.e. 2.29 times less than *S*(5,3)-code. The same paper suggests the method for improving the *S*(*m*,*k*)-codes characteristics for error detection in data vectors without sacrificing the number of check bits. This method is based on transformation of each Berger code word in its check part using the specific algorithm. This algorithm implicates the determination of the vector weight to the modulus $M = 2^{\lceil \log_2(m+1) \rceil - 1}$ and using the correction factor, equals to modulo two sum of values of some data vector bits. Modified Berger codes, at an average, detect twice as many errors in data vectors as classic *S*(*m*,*k*)-codes, however, they are not the optimum codes. In [21-23] there is a suggestion of the line of modified Berger codes, that are obtained by establishing the modulus for data vector weight determination, selected from the set $M \in \left\{2;4;...;2^{\lceil \log_2(m+1) \rceil - 2}\right\}$. These codes have decreased number of check bits in comparison with Berger codes. Modular-modified Berger codes with its *m* and *k* values are also not an optimum (excluding the code, where *M*=2, that has 2 check bits).

Optimum sum code with the same number of check bits as Berger code, can be built by alteration of the algorithm of formation of the sum code with weighted transitions [24-26]:

Algorithm 1. *Check vectors obtaining for sum code with weighted transitions:*

1. Data vector bits are considered, that have the adjacent positions.
2. Each transition from the bit to the bit it is assigned the weight ratio from the positive integers (1, 2, …, *m*–1), starting from the lower order bit.
3. The sum of so called active transitions (transition between the bit is called active, if the modulo two sum of the values of adjacent bits equals 1) is calculated.
4. Obtained value is presented in binary form and recorded in check vector.

Let us set the sum code with weighted transitions as *WT*(*m*,*k*)-code.

WT(*m*,*k*)-code detects bigger number of errors in data vectors, than *S*(*m*,*k*)-code, and however it has the increased number of check bits. This number is equals $k = \left\lceil \log_2 \frac{m(m-1)}{2} \right\rceil$. The increased number of check bits in *WT*(*m*,*k*)-code in comparison with *S*(*m*,*k*)-code results in increase of complexity of control equipment in CED system (see Fig. 1). Besides, the check bits of *WT*(*m*,*k*)-code are used ineffectively – *WT*(*m*,*k*)-codes are not optimum codes [27]. Optimum code are obtained by using the following algorithm of formation.

Algorithm 2. *Check vectors obtaining for modified sum code with weighted transitions:*

1. The steps 1 – 3 of Algorithm 1 are carried out.
2. The modulus $M = 2^{\lceil \log_2(m+1) \rceil}$ value is established (this is the modulus of Berger code).
3. The sum *V* of so called active transitions is calculated.
4. The least non-negative residue of *V* value for established modulus is calculated: $W = (V) \bmod M$.
5. The number *W* is presented in binary form and recorded in check vector.

To demonstrate the operation of algorithm of formation of modified sum code with weighted transitions or *WTM*(*m*,*k*)-code, let us use the example of *WT*8(5,3)-code formation. Table I presents the weight ratios of each transition between the bits of data vector, and Table 2 presents all code words of *WT*8(5,3)-code. For instance, in <00101> data vector transitions are considered active between the bits x_1 and x_2, x_2 and x_3, x_3 and x_4. To obtain *V* value it is necessary to sum up the weight ratios of these transitions: $V=w_{1,2}+w_{2,3}+w_{3,4}=1+2+3=6$. The least non-negative residue of *V* equals to *W*=(6)mod8=6, that in binary form is presented as <110>.

TABLE I
WEIGHT RATIOS OF TRANSITIONS IN DATA VECTOR

$w_{4,5}$	$w_{3,4}$	$w_{2,3}$	$w_{1,2}$
4	3	2	1

By analyzing Table II, it is not too difficult to notice that *V* values for data vectors, equidistant form the middle of the table (these vectors have opposite values of similar bits), are equal. At that the number of repeating *V* values is unequal (Table III). This results in the irregularity in the distribution of data vectors of *WT*(*m*,*k*)-code among the check vectors (within the test groups). Implementing the modification of *WT*(*m*,*k*)-code under Algorithm 2, this disadvantage is eliminated, because the calculation is made up to the value *M*–1=7: (8)mod8=0, (9)mod8=1 и (10)mod8=2.

TABLE II
CODE WORDS OF *WT*8(5,3)-CODE

№	Data vector bits					*V*	*W*=(*V*)mod8	Check vector bits		
	x_5	x_4	x_3	x_2	x_1			y_3	y_2	y_1
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	0	0	1
2	0	0	0	1	0	3	3	0	1	1
3	0	0	0	1	1	2	2	0	1	0
4	0	0	1	0	0	5	5	1	0	1
5	0	0	1	0	1	6	6	1	1	0
6	0	0	1	1	0	4	4	1	0	0
7	0	0	1	1	1	3	3	0	1	1
8	0	1	0	0	0	7	7	1	1	1
9	0	1	0	0	1	8	0	0	0	0
10	0	1	0	1	0	10	2	0	1	0
11	0	1	0	1	1	9	1	0	0	1
12	0	1	1	0	0	6	6	1	1	0
13	0	1	1	0	1	7	7	1	1	1
14	0	1	1	1	0	5	5	1	0	1
15	0	1	1	1	1	4	4	1	0	0
16	1	0	0	0	0	4	4	1	0	0
17	1	0	0	0	1	5	5	1	0	1
18	1	0	0	1	0	7	7	1	1	1
19	1	0	0	1	1	6	6	1	1	0
20	1	0	1	0	0	9	1	0	0	1
21	1	0	1	0	1	10	2	0	1	0
22	1	0	1	1	0	8	0	0	0	0
23	1	0	1	1	1	7	7	1	1	1
24	1	1	0	0	0	3	3	0	1	1
25	1	1	0	0	1	4	4	1	0	0
26	1	1	0	1	0	6	6	1	1	0
27	1	1	0	1	1	5	5	1	0	1
28	1	1	1	0	0	2	2	0	1	0
29	1	1	1	0	1	3	3	0	1	1
30	1	1	1	1	0	1	1	0	0	1
31	1	1	1	1	1	0	0	0	0	0

TABLE III
DISTRIBUTION OF V VALUES FOR WT8(5,3)-CODE FORMATION

V										
0	1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9	10
			3	4	5	6	7			
			3	4	5	6	7			

Table IV provides the distribution of data vectors within the test groups for the *WT*8(5,3)-code of interest. Within each test group there are 4 data vectors. What is more, in virtue of the equality of *V* values for equidistant from the middle of Table II data vectors within each test group there are always two data vectors minimum, that differ from each other in all bits. This induces the undetectable errors of maximum multiplicity *d*=*m* in *WT*8(5,3)-code. Undetectable errors of lower multiplicities appear in *WT*8(5,3)-codes in case of distortions, that transfer data vectors of one test group into data vectors, not-equidistant from the middle of Table II. This type of distribution of data vectors among the test groups remains unchanged and for all *WTM*(*m*,*k*)-codes. The only exclusion is *WTM*(*m*,*k*)-codes with the number of data bits $m = 2^t$ (*t*=1, 2, …). For these codes there is no filling the last test group, because the number of data bits is that the value $W = 2^k - 1$ does not form. Actually, where *m*=4, for example, maximum data vector weight is $W=w_{1,2}+w_{2,3}+w_{3,4}=1+2+3=6$.

Apropos *WTM*(*m*,*k*)-codes with data vectors lengths $m \neq 2^t$ (*t*=1, 2, …), it should be noticed that within each test group there are transitions for them similar by multiplicity. These transitions determine the number of distortions with different multiplicity within each test group. For example, Fig. 2 shows the graph of all possible transitions within <111> test group of *WT*8(5,3)-code.

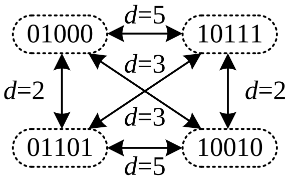

Fig. 2. Transitions within <111> test group of *WT*8(5,3)-code

The total number of undetectable errors in *WTM*(*m*,*k*)-codes is obtained by multiplying the number of errors within one test group by *M*.

Table V5 provides the distribution of undetectable errors in data vectors of *WTM*(*m*,*k*)-codes for several *m* values. Analysis of such table allowed to establish the features of a new class of sum codes based on error detection in data vectors:

- any *WTM*(*m*,*k*)-code detects 100% of single distortions (is a fault-tolerant code);
- *WTM*(*m*,*k*)-codes with data vector lengths $m \neq 2^t$ (*t*=1, 2, …) are optimum;
- *WTM*(*m*,*k*)-codes with data vector lengths $m = 2^t$ (*t*=1, 2, …) are close to optimum, but not such codes;
- *WTM*(*m*,*k*)-codes with even values of *m* detect 100% of errors of odd multiplicities in data vectors (any Berger codes have this feature);
- *WTM*(*m*,*k*)-codes detect 100% of errors of multiplicity *d*=*m*–1 in data vectors;
- *WTM*(*m*,*k*)-codes do not detect 100% of errors of maximum multiplicity *d*=*m* in data vectors;
- the number of undetectable errors of each multiplicity (as well as the total number of undetectable errors) is multiple of modulus *M*.

The properties of *WTM*(m,k)-codes are explained by the rules of its formation. For example, fault-tolerance is explained by the fact, that for weight ratios the sequence of positive integers from 1 to m–1 is used, and modulus for residue determination is always bigger than the maximum transition weight in data vector: $w_{m-1,m}=m-1$, $M = 2^{\lceil \log_2 (m+1) \rceil}$ and $M>m-1$. If modulus M will be equal to the weight of transition between the highest bits $w_{m,m-1}$, so the residue of this value would be 0: $(w_{m-1,m})\mathrm{mod}M=(M)\mathrm{mod}M=0$. And this would mean that even with activation of transition the weight values in total sum W would always be 0, and the bit would be uncontrolled.

TABLE IV
DISTRIBUTION OF DATA VECTORS OF *WT*8(5,3)-CODE WITHIN TEST GROUPS

W=(*V*)mod8							
0	1	2	3	4	5	6	7
Check vector							
000	**001**	**010**	**011**	**100**	**101**	**110**	**111**
00000	00001	00011	00010	00110	00100	00101	01000
01001	01011	01010	00111	01111	01110	01100	01101
10110	10100	10101	11000	10000	10001	10011	10010
11111	11110	11000	11101	11001	11011	11010	10111

TABLE V
DISTRIBUTION OF UNDETECTABLE ERRORS IN DATA VECTORS OF SOME *WTM*(*M*,*K*)-CODES

m	*k*	*M*	Distribution of undetectable errors by multiplicities, $N_{m,d}$												N_m
			1	2	3	4	5	6	7	8	9	10	11	12	
2	2	4	0	4											4
3	2	4	0	0	8										8
4	3	8	0	8	0	16									24
5	3	8	0	32	32	0	32								96
6	3	8	0	192	0	192	0	64							448
7	3	8	0	448	448	448	448	0	128						1920
8	4	16	0	832	0	1936	0	832	0	256					3856
9	4	16	0	2304	1280	4096	4096	1280	2304	0	512				15872
10	4	16	0	7680	0	24064	0	24064	0	7680	0	1024			64512
11	4	16	0	17408	7424	58496	45696	45696	58496	7424	17408	0	2048		260096
12	4	16	0	44032	0	242688	0	466944	0	242688	0	44032	0	4096	1044480

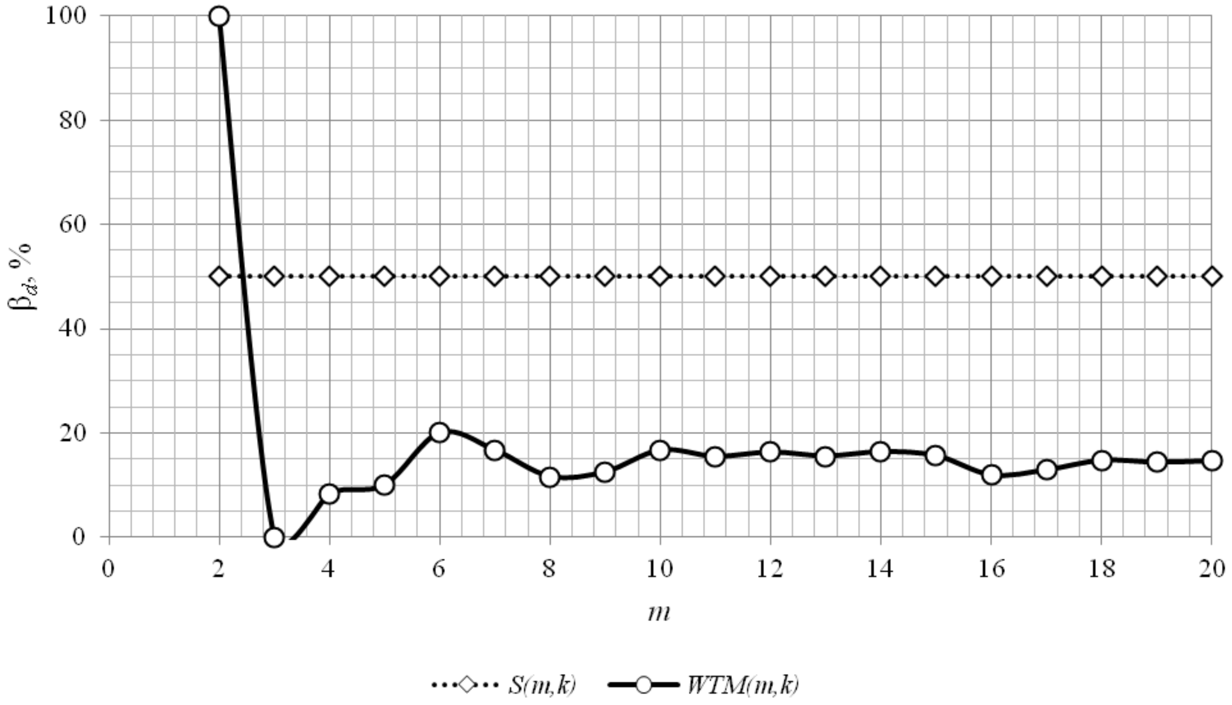

Fig. 3. Part of double undetectable errors in total number of double errors in *S*(*m*,*k*) and *WTM*(*m*,*k*)-codes

Fig. 3 gives the comparison of *WTM*(*m*,*k*)-codes and *S*(*m*,*k*)-codes by its ability of detection of double errors in data vectors – there is a function of the number of double undetectable errors in data vectors to the total number of double errors in data vectors (β_d) by the data vector length. Based on Fig. 3 it follows that *WTM*(*m*,*k*)-codes more than twice as effective in double errors detection, as *S*(*m*,*k*)-codes. Such improvement is true and for errors of any even multiplicities (Table VI). Diagrams of Fig. 4 show the function of η_m, which equals to the number of undetectable errors of given multiplicity in Berger codes and in weight-based codes of interest, to the data vector length *m*. The value η_m demonstrates what fold decrease the number of undetectable errors of given multiplicity in *WTM*(*m*,*k*)-codes in comparison with *S*(*m*,*k*)-codes with the specified *m* value.

For odd *m* values the efficiency of detection of errors of even multiplicities in *WTM*(*m*,*k*)-codes, in comparison with *m*±1, increases, that happens due to the presence of errors of odd multiplicities within the class of undetectable errors in codes with odd *m* values.

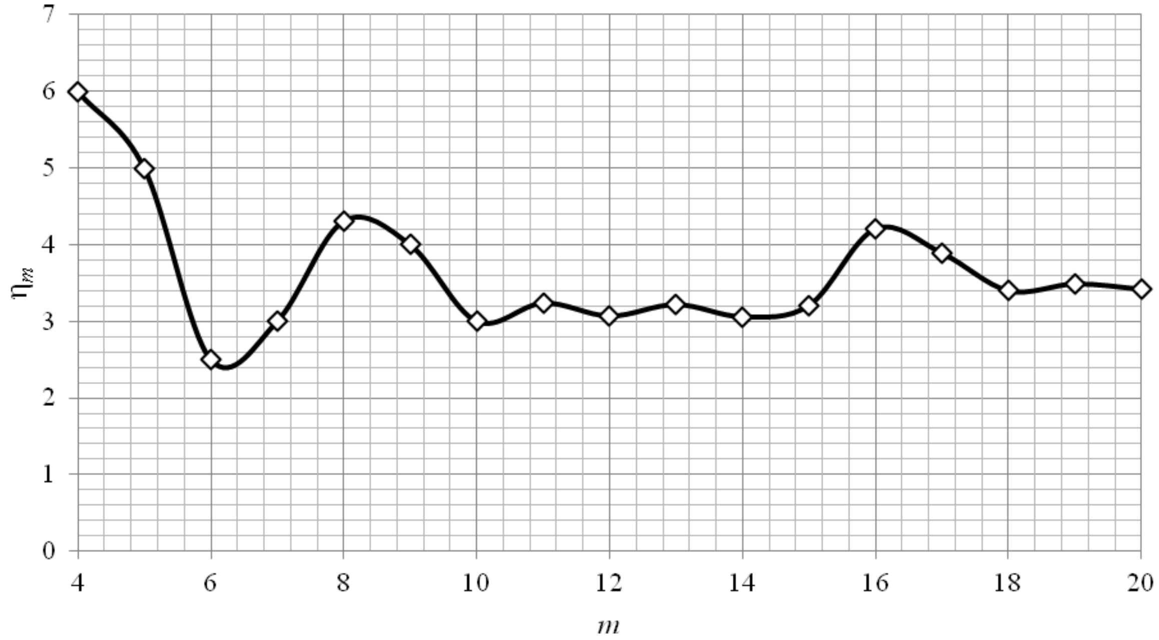

Fig. 4. Improving the characteristic of double errors detection of *WTM*(*m*,*k*)-codes in comparison with *S*(*m*,*k*)-codes

TABLE VI

VALUE BD FOR EACH MULTIPLICITY FOR CLASSIC AND WEIGHT-BASED SUM CODES

m	Part of undetectable errors of multiplicity *d* in total number of errors of given multiplicity, %																			
	1	**2**	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
WTM(*m*,*k*)-code																				
2	0	**100**																		
3	0	**0**	100																	
4	0	**8.33**	0	100																
5	0	**10**	10	0	100															
6	0	**20**	0	20	0	100														
7	0	**16.67**	10	10	16.67	0	100													
8	0	**11.61**	0	10.8	0	11.61	0	100												
9	0	**12.5**	2.98	6.35	6.35	2.98	12.5	0	100											
10	0	**16.67**	0	11.19	0	11.19	0	16.67	0	100										
11	0	**15.45**	2.2	8.66	4.83	4.83	8.66	2.2	15.45	0	100									
12	0	**16.29**	0	11.97	0	12.34	0	11.97	0	16.29	0	100								
13	0	**15.54**	2.05	8.89	4.75	6.54	6.54	4.75	8.89	2.05	15.54	0	100							
14	0	**16.35**	0	12.76	0	12.27	0	12.27	0	12.76	0	16.35	0	100						
15	0	**15.6**	3.1	9.23	4.26	7.32	5.79	5.79	7.32	4.26	9.23	3.1	15.6	0	100					
16	0	**11.88**	0	6.61	0	6.15	0	6.15	0	6.15	0	6.61	0	11.88	0	100				
17	0	**12.87**	0.81	5.65	1.7	4.09	2.49	3.27	3.27	2.49	4.09	1.7	5.65	0.81	12.87	0	100			
18	0	**14.71**	0	7.25	0	6.26	0	6.15	0	6.15	0	6.26	0	7.25	0	14.71	0	100		
19	0	**14.33**	0.68	6.47	1.46	4.5	2.2	3.61	2.9	2.9	3.61	2.2	4.5	1.46	6.47	0.68	14.33	0	100	
20	0	**14.61**	0	7.47	0	6.43	0	6.2	0	6.16	0	6.2	0	6.43	0	7.47	0	14.61	0	100
S(*m*,*k*)-code																				
2÷20	0	**50**	0	37.5	0	31.25	0	27.34	0	24.61	0	22.56	0	20.95	0	19.64	0	18.55	0	17.62

III. Conclusion

This paper suggests the method for formation of a sum code with minimum total number of undetectable errors in data vectors. Moreover, this new sum code more effectively detects the errors of low multiplicities in data vectors, than the classic Berger code. These advantages of $WTM(m,k)$-code can be considered while CED systems design with 100% detection of single faults in controlled logic units.

References

[1] E.S. Sogomonyan, and E.V. Slabakov "Self-Checking Devices and Fault-Tolerant Systems" (in Russ.), Moscow: Radio & Communication, 1989, 208 p.

[2] S.J. Piestrak "Design of Self-Testing Checkers for Unidirectional Error Detecting Codes", Wrocław: Oficyna Wydawnicza Politechniki Wrocłavskiej, 1995, 111 p.

[3] Y.-Y. Guo, J.-C. Lo, and C. Metra "Fast and Area-Time Efficient Berger Code Checkers", *Workshop on Defect and Fault-Tolerance in VLSI Systems*, 1997, October 20-22, pp. 110-118.

[4] A.Yu. Matrosova, I. Levin, and S.A. Ostanin "Self-Checking Synchronous FSM Network Design with Low Overhead", *VLSI Design*. – 2000. – Vol. 11. – Issue 1. – Pp. 47-58.

[5] A. Matrosova, I. Levin, and S. Ostanin "Survivable Self-Checking Sequential Circuits", *Proc. of 2001 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems* (DFT 2001), Oct. 24-26, San Francisco, CA, 2001, 395-402 pp.

[6] L-T. Wang, C.E. Stroud, and N.A. Touba "System-on-Chip Test Architectures: Nanometer Design for Testability", Morgan Kaufmann Publishers, 2008, 856 p.

[7] R. Ubar, J. Raik, and H.-T. Vierhaus "Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source)", Information Science Reference, Hershey – New York, IGI Global, 2011, 578 p.

[8] A.H. Abdulhadi, and A.H. Maamar "Self Checking Register File Using Berger Code", *6th WSEAS International Conference on Circuits, Systems, Control & Signal Processing*, 2007, Cairo, Egypt, December 29-31, pp. 62-68.

[9] P. Srihari "Sum Codes: A Binary Channel Coding Scheme", *International Journal of Computer Science And Technology*, 2014, Vol. 5, Issue 1, pp. 60-64.

[10] N.K. Jha, and S. Wang "Design and Synthesis of Self-Checking VLSI Circuits", *IEEE Trans. Computer-Aided Design*, 1993, Vol. 12, Issue 6, pp. 878-887.

[11] S.S. Gorshe, and B. Bose "A Self-Checking ALU Design with Efficient Codes", *Proceedings of 14th VLSI Test Symposium*, Princeton, NJ, USA, 1996, pp. 157-161.

[12] M. Nicolaidis, and Y. Zorian "On-Line Testing for VLSI – A Compendium of Approaches", *Journal of Electronic Testing: Theory and Applications*, 1998, vol. 12, issue 1-2, pp. 7-20.

[13] O. Potin, Ch. Dufaza, and Ch. Landrault "A New Scheme for Off-Line and On-Line Testing With ABC And Berger Encoding", *Proc. 4th IEEE Int. On-line Testing Workshop*, Italia, Capri, 1998, 71-75.

[14] D. Das, and N.A. Touba "Weight-Based Codes and Their Application to Concurrent Error Detection of Multilevel Circuits", *Proc. 17th IEEE Test Symposium*, USA, California, 1999, pp. 370-376.

[15] V.V. Sapozhnikov, and Vl.V. Sapozhnikov "Self-Checking Discrete Devices" (in Russ.), St. Petersburg: Energoatomizdat, 1992, pp. 224.

[16] L.D. Cheremisinova "Logical Synthesis of Combinational KMOP Circuits Considering the Power Dissipation" (in Russ.), *Bulletin of Tomsk State University. Management, Computer Engineering and Informatics*, 2014, Issue 3, pp. 89-98.

[17] P.P. Parkhomenko, and E.S. Sogomonyan "Basics of Technical Diagnostics (Optimization of Diagnostic Algorithms and Equipment)" (in Russ.), Moscow: Energoatomizdat, 1981, pp. 320.

[18] J.M. Berger "A Note on Error Detection Codes for Asymmetric Channels", *Information and Control*, 1961, vol. 4, issue 1, pp. 68-73.

[19] D.V. Efanov, V.V. Sapozhnikov, and Vl.V. Sapozhnikov "On Sum Code Properties in Functional Control Circuits", *Automation and Remote Control*, 2010, vol. 71, issue 6, pp. 1117-1123.

[20] A.A. Blyudov, D.V. Efanov, V.V. Sapozhnikov, and Vl.V. Sapozhnikov "Formation of the Berger Modified Code with Minimum Number of Undetectable Errors of Informational Bits" (in Russ.)", *Electronic Modeling*, 2012, vol. 34, issue 6, pp. 17-29.

[21] A.A. Blyudov, D.V. Efanov, V.V. Sapozhnikov, and Vl.V. Sapozhnikov "Sum codes for organization of control of combinational circuits", *Automation and Remote Control*, 2013, vol. 74, issue 6, pp. 1020-1028.

[22] D. Efanov, V. Sapozhnikov, Vl. Sapozhnikov, and A. Blyudov "On the Problem of Selection of Sum Code for Combinational Circuit Test Organization", Proceedings of 11th IEEE East-West Design & Test Symposium (EWDTS`2013), Rostov-on-Don, Russia, September 27-30, 2013, pp. 261-266.

[23] A.A. Blyudov, D.V. Efanov, V.V. Sapozhnikov, and Vl.V. Sapozhnikov "On Codes with Summation of Unit Bits In Concurrent Error Detection Systems", *Automation and Remote Control*, 2014, vol. 75, issue 8, pp. 1460-1470.

[24] V.V. Saposhnikov, and Vl.V. Saposhnikov "New Code for Fault Detection in Logic Circuits", *Proc. 4th Int. Conf. on Unconventional Electromechanical and Electrical Systems*, St. Petersburg, Russia, June 21-24, 1999, pp. 693-696.

[25] V. Mehov, V. Saposhnikov, Vl. Sapozhnikov, and D. Urganskov "Concurrent Error Detection Based on New Code with Modulo Weighted Transitions between Information Bits", *Proc. of 7th IEEE East-West Design&Test Workshop (EWDTW`2007)*, Erevan, Armenia, September 25-30, 2007, pp. 21-26.

[26] V.B. Mekhov, V.V. Saposhnikov, and Vl.V. Saposhnikov "Checking of Combinational Circuits Basing on Modification Sum Codes", *Automation and Remote Control*, 2008, vol. 69, issue 8, pp. 1411-1422.

[27] V.V. Sapozhnikov, Vl.V. Sapozhnikov, D.V. Efanov, and V.V. Dmitriev "Properties Of Sum Codes With Weighted Transitions With Direct Sequence Of Weight Ratios (in Russ.)", *Computer Science and Control Systems*, 2014, issue 4, pp. 77-88.