Properties of code with summation for logical circuit test organization

Anton Blyudov, Dmitry Efanov, Valery Sapozhnikov, Vladimir Sapozhnikov
Petersburg State Transport University
"Automation and Remote Control on Railways" department
E-mail:mitriche@yandex.ru

Abstract

In this paper we consider binary codes with summation used at designing test systems of combinational logical circuits. We offer new codes, determine their properties and compare these codes with each other.

1. Introduction

The general structure of a functional control system of the combinational circuit is shown in Fig 1, this control is based on the usage of a code with summation [1]. The block of the basic logic $f(x)$ realizes the system of the Boolean functions $f_1(x)$, $f_2(x)$,…, $f_m(x)$. The block of an additional logic $g(x)$ realizes such functions $g_1(x)$, $g_2(x)$, …, $g_k(x)$, so the operative output vectors (f_1, f_2, …, f_m g_1, g_2, …, g_k) are the code words of some previously chosen code. The check bits of code words are calculated at the outputs g_1, g_2, …, g_k, the generator G also calculates them by the values of the functions $f_1(x)$, $f_2(x)$, …, $f_m(x)$. The module MC compares the values of the same name signals at the outputs of the blocks $g(x)$ and G.

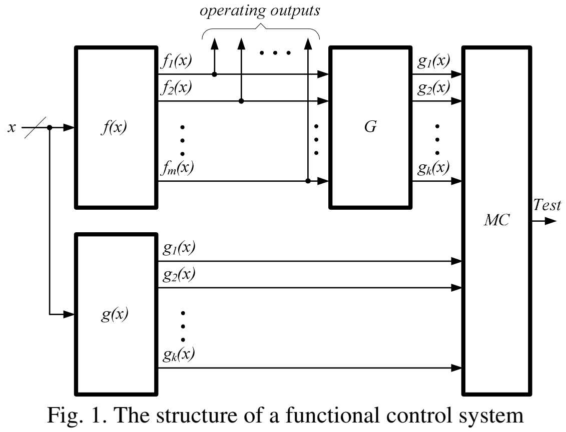

Fig. 1. The structure of a functional control system

Let's designate a classic code with summation (also known as the Berger code [2]) as $S(n,m)$-code (m is the number of informational bits, $n=m+k$ is the total number of bits, k is the number of check bits of code vectors). The check vector in the code word is a binary number equal to the number of "ones" among the informational bits. The number of check bits is $k = \left]\log_2(m+1)\right[$. The $S(n,m)$-code can be presented as an aggregate of isolated groups of code words, whose number is $m+1$. Any group contains vectors with the same number of "ones". The same check word matches to all the vectors of an isolated group. The $S(12,8)$-code is given in Tab. 1. Each isolated group is determined by one representative.

Table 1. Codes with summation

Informational words are representatives of isolated groups								check words					
								$S(12,8)$				$S4(10,8)$	
x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	y_4	y_3	y_2	y_1	y_2	y_1
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1	0	1
0	0	0	0	0	0	1	1	0	0	1	0	1	0
0	0	0	0	0	1	1	1	0	0	1	1	1	1
0	0	0	0	1	1	1	1	0	1	0	0	0	0
0	0	0	1	1	1	1	1	0	1	0	1	0	1
0	0	1	1	1	1	1	1	0	1	1	0	1	0
0	1	1	1	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	0	0	0	0	0

During the analysis of the circuit properties to find out errors we admit a possibility of failing of only one structure block. The failures of the block $g(x)$ cause distortions of the check bits and they are always detected. The failures of the block $f(x)$ lead to the distortions of informational bits and they might not be detected. We don't consider the possibility of accumulating errors which appears because of an undetectable error in the block $f(x)$.

The properties of detecting faults of the functional control circuit are considerably determined by the properties of a code with summation which determines

978-1-4799-2096-9/13/$31.00 ©2013 IEEE

errors of informational bits. The appearance of errors of any multiplicity is possible at the outputs of the block *f(x)*. The errors of an even multiplicity, which have the number of distortions of the informational bits of the type 1→0 equal to the distortions 0→1, cannot be detected in the *S(n,m)*-code. Let's denote the ratio (in percent) of the number of undetectable errors of informational bits of the multiplicity *t* to the whole number of such possible errors by β_t.

As shown in [3], the *S(n,m)*-code has the following property: the share of undetectable errors β_t doesn't depend on the number of informational bits and it is constant

$$\beta_t = \frac{C_t^{\frac{t}{2}}}{2^t}.$$

Undetectable errors appear when the informational vector transfers to another one, belonged to the same isolated group. The false transfer between informational vectors belonging to different isolated groups is detected.

However, the Berger code contains a plenty enough large number of check bits, therefore, in Fig. 1 the blocks *g(x)* and *G* have a large number of outputs and the block *MC* has a large number of inputs. This determines an essential complexity of control equipment. Besides, the Berger code has a great number of undetectable errors. Particularly, it doesn't detect 50% of distortions of multiplicity 2.

Therefore, from a practical point of view, it is advisable to use some modifications of the Berger code with better characteristics at the control circuit designing. The effect of using such codes may be reached if we consider an important characteristic of the block *f(x)*, namely the maximum multiplicity t_{max} of the error at the outputs $f_1(x), f_2(x), \ldots, f_m(x)$ caused by a single failure in the block [4], [5].

2. Modulo codes with summation

Decreasing a number of check bits in the code with summation could be reached if the calculation of "ones" in the informational vector is carried out by some modulo *M* ($M = 2^i$, *i*=1,2,…), which is less than the modulo $M = 2^k$, used at the forming the *S(n,m)*-code. Let's call such codes "modulo codes" and denote them as *SM(n,m)*. There are $]\log_2(m+1)[-1$ modulo codes, besides the *S(n,m)*-code, for any value of *m*. For example, three modulo codes *S2(9,8)*, *S4(10,8)* and *S8(11,8)* could be formed for *m=8*. The *S4(10,8)*-code is shown in Tab. 1.

The computer program, which realizes the algorithm of calculating of the characteristics β_t for any code with summation, was developed to determine the properties of modulo codes. The algorithm forms isolated groups of informational vectors. The code distance between each two vectors of one group is determined. Then we count the total number of undetectable errors of each multiplicity. The results received during the research of modulo codes are shown in Tab. 2. They determine the properties of modulo codes:

1. All the errors of information bits of the odd multiplicity are detected in the modulo codes, but the errors of the even multiplicity can't be detected.
2. The share of the undetectable errors β_t for a modulo code doesn't depend on the number of informational bits and it is a constant value.
3. Any *S2(n,m)*-code (a parity code) doesn't detect 100% of errors of informational bits of any even multiplicity.
4. Any *S4(n,m)*-code doesn't detect 50% of errors of informational bits of any even multiplicity.
5. The *SM(n,m)*-code has the same number of errors of multiplicity $t<M$ like the *S(n,m)*-code.
6. Any *SM(n,m)*-code with $M\geq 4$ doesn't detect 50% of errors of informational bits of multiplicity 2.
7. The *SM(n,m)*-code with $M\geq 8$ has more errors of multiplicity $t\geq M$ than the *S(n,m)*-code.
8. The *SM(n,m)*-code has the value of characteristic β_t equal or less than the same characteristic for the *SM'(n,m)*-code *(M'>M)* for every *t*.

Table 2. Values of the characteristic β_t

Code	Multiplicity of an error *t*							
	2	4	6	8	10	12	14	16
S(n,m)	50	37,5	31,25	27,34	24,61	22,56	20,95	19,638
S2(n,m)	100	100	100	100	100	100	100	100
S4(n,m)	50	50	50	50	50	50	50	50
S8(n,m)	50	37,5	31,25	28,13	26,56	25,78	25,39	25,2
S16(n,m)	50	37,5	31,25	27,34	24,61	22,56	20,95	19,641

The efficiency of using the modulo codes is shown in the following example. Let the control scheme of the block *f(x)* with the number of outputs *m=100* be designed. In this case we can use both the code *S(107,100)* ("ones" are counted by the modulo *M=128*) and the modulo codes with *M=2,4,…,64*. Let's also suppose that the experiment with the block *f(x)* determined the maximum multiplicity of the error as $t_{max}=8$. As follows from Tab. 2 and the properties 6 – 8, the *S(107,100)*-code and the modulo codes with *M=8,16,32,64* have the identical characteristics β_t for errors of the multiplicity t≤8. Therefore in this case it is advisable to use the *S8(103,100)*-code, which has a minimum number of check bits. In comparison with the usage of the classical *S(107,100)*-code we achieve a significant decreasing the complexity of control equipment, because the number of outputs of the blocks *f(x)* and *G* are reduced from 7 to 3 and the number of inputs of the *MC* block is reduced from 14 to 6.

3. Modified modulo codes with summation

The method of forming a modified code *RS(n,m)* with summation by converting every word of the *S(n,m)*-code to the corresponding word of a new code with the help of the special rules is offered in [6]. Herewith the number of check bits is kept (the complexity of control equipment isn't increased), also the main property of a code with summation (detecting errors of odd multiplicity) is kept too. The advantage of the *RS(n,m)*-code is that it has considerably better characteristics of detecting errors than the *S(n,m)*-code. For example, in [6] the considered *RS(9,6)*-code has undetectable double errors by 2,5 times less in comparison with the *S(9,6)*-code and the fourfold errors by 1,25 times less.

The simultaneous decreasing of the number of check bits and the increasing of the detection possibility of the code can be reached by forming a modulo modified code with summation (let's note them as the *RSM(n,m)*-code). At forming a *RSM(n,m)*-code for converting words of the *S(n,m)*-code we use (unlike the *RS(n,m)*-code, [6]) the fixed value of a modulo *M* (independent on the value of *m*). The *RSM*-codes with values of $M=2,4,8,\ldots,2^b$ ($b =]\log_2(m+1)[-2$) are possible for the current *m*. For example, there are *RSM*-codes with *M=2, 4, 8* at *m=16*.

The method of forming a *RSM(n,m)*-code is: the *S(n,m)*-code is formed for the current *m*. Then every word of the obtained code is converted to the word of the *RSM(n,m)*-code by the rules illustrated for the *RS2(8,6)*-code in Tab. 3. The modulo *M* (in this case *M=2*) is fixed. For the current informational word the number of "ones" *q* is counted, which is converted to the number *W=q(modM)*. The special coefficient *α* is determined. If $x_m \oplus x_{m-1} \oplus \ldots \oplus x_p = 0$ ($p =]\log_2 M[+2$), then *α=0*, otherwise *α=1*. Then the resultant weight of an informational word $V = W + M\alpha$ is counted. The check word is a binary notation of *V*.

For example, in Tab. 3 for the informational word 111111 we have *q=6*, *W=6(mod2)=0*, $p =]\log_2 2[+2=3$, *α=0* (because $x_6 \oplus x_5 \oplus x_4 \oplus x_3 = 0$), *V=0* (because *W=0* and *α=0*). The check vector is $y_2y_1=00$.

Table 3. Words of the *RS2(8,6)*-code

Informational word						q	W	α	V	Check word	
x_6	x_5	x_4	x_3	x_2	x_1					y_2	y_1
0	0	0	1	1	1	3	1	1	3	1	1
0	1	1	0	1	1	4	0	0	0	0	0
0	0	1	0	0	0	1	1	1	3	1	1
1	1	1	1	1	1	6	0	0	0	0	0

4. Classification of codes with summation

The Berger code, modulo and modified codes make up a set of possible codes with summation. The classification of codes with summation is shown in Fig. 2 and all the codes with *m=16* are given in Tab. 4. The following characteristics of codes are given in the table: the number of check bits *k*, the number and percent of the undetectable errors of a small multiplicity (2 – 8), and of any multiplicity (2,4,…16). The analysis of such tables for different codes allows to deduce the following conclusions:

1. The *RS(n,m)*-codes have the best characteristics to detect errors. For example, the *RS(21,16)*-code (see Tab. 4) has about by two times less undetectable errors in comparing with the Berger code, and this concerns to the errors of any multiplicity.
2. Any modulo modified code with *M≥4* has the better detection characteristics than any non-modified code.
3. The modulo non-modified codes have an advantage in detecting errors among all the codes with the number of check bits *k=2*.

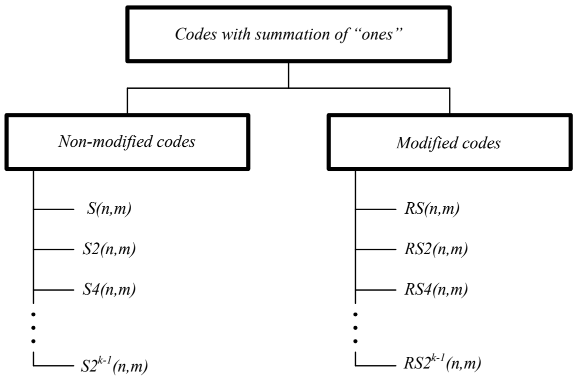

Fig. 2. Classification of codes

Table 4. Table of codes with *m=16*

Codes	*k*	Multiplicity of error				
		2	4	6	8	2, 4, …, 16
		The total number of errors of current multiplicity				
		7864320	119275520	524812288	843448320	4294901760
		Number/percent of undetectable errors				
S(21,16)	5	3932160 50%	44728320 37,50%	164003840 31,25%	230630400 27,34%	601014854 13,99%
S2(17,16)	1	7864320 100%	119275520 100%	524812288 100%	843448320 100%	2147418112 49,99%
S4(18,16)	2	3932160 50%	59637760 50%	262406144 50%	421724160 50%	1073709056 24,99%
S8(19,16)	3	3932160 50%	44728320 37,50%	164003840 31,25%	237219840 28,13%	622051456 14,48%
S16(20,16)	4	3932160 50%	44728320 37,50%	164003840 31,25%	230630400 27,34%	601014856 13,99%
RS2(18,16)	2	6029312 76,67%	71565312 60,00%	262406144 50,00%	393609216 46,67%	1073676288 25,00%
RS4(19,16)	3	2654208 33,75%	31096832 26,07%	126517248 24,10%	210862080 25,00%	536838144 12,50%
RS8(20,16)	4	2368512 30,12%	21790720 18,27%	81395712 15,51%	121700352 14,43%	312666048 7,28%
RS16(21,16)	5	2129920 27,08%	21749760 18,23%	82677760 15,75%	115315200 13,67%	300495008 6,99%

5. Conclusion

The table of codes, in which the main characteristics of all the available codes with summation with a current number of informational bits are given, allows to choose a code more reasonably at organizing the check of a combinational circuit. Herewith there is a possibility to consider the properties of a control circuit, for example, the possibility of appearing errors of the determined multiplicity at the outputs of the circuit.

6. References

[1] Saposhnikov V.V. and Saposhnikov Vl.V. Self-checking digital devices, St. Pb: Energoatomizdat, 1992, 224 p. – ISBN 5-283-04605-2.

[2] J.M. Berger "A note on error detection codes for asymmetric channels", *Information and Control*. – 1961. – 4, №3, pp. 68-73.

[3] D.V. Efanov, V.V. Saposhnikov and Vl.V. Saposhnikov "On the properties of code with summation in functional control circuits", *Automation and remote control*, 2010, №6, pp. 155-162.

[4] A. Morozov, V.V. Saposhnikov, Vl.V. Saposhnikov and M. Goessel "New self-checking circuits by use of Berger-codes", *6th IEEE International On-Line Testing Workshop*, Palma de Mallorca, Spain, 2000, pp. 141-146.

[5] Vl. Moshanin, V. Ocheretnij and A. Dmitriev "The Impact of Logic Optimization of Concurrent Error Detection", *Proc. 4th IEEE International On-Line Testing Workshop*, Capry, Italy, 1998, pp. 81-84.

[6] Blyudov A.A., Saposhnikov V.V., Saposhnikov Vl.V. "Modified code with summation for combinational circuit test organization", *Automation and remote control*, 2012, №1, pp. 169-177.