Properties of code with summation for logical circuit test organization

Anton Blyudov, Dmitry Efanov, Valery Sapožnikov, Vladimir Sapožnikov
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 \( f_1(x), f_2(x), \ldots, f_m(x) \). The block of an additional logic \( g(x) \) realizes such functions \( g_1(x), g_2(x), \ldots, g_k(x) \), so the operative output vectors \( (f_1, f_2, \ldots, f_m, g_1, g_2, \ldots, g_k) \) are the code words of some previously chosen code. The check bits of code words are calculated at the outputs \( g_1, g_2, \ldots, g_k \), the generator \( G \) also calculates them by the values of the functions \( f_1(x), f_2(x), \ldots, f_m(x) \). The module \( MC \) compares the values of the same name signals at the outputs of the blocks \( g(x) \) and \( G \).

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

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 \( k = \log_2 (m+1) \). 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

<table>
<thead>
<tr>
<th>Informational words are representatives of isolated groups</th>
<th>check words</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S(12,8) )</td>
<td></td>
</tr>
<tr>
<td>( S(10,8) )</td>
<td></td>
</tr>
<tr>
<td>( x_1 )</td>
<td>( x_2 )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 1. The structure of a functional control system
properties of a code with summation which determines 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 $1 \rightarrow 0$ equal to the distortions $0 \rightarrow 1$, 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 $\beta_t$.

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

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

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 $t_{\text{max}}$ of the error at the outputs $f_1(x), f_2(x), \ldots, f_{\text{m}}(x)$ 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 $M$ $(M = 2^i, i=1,2,\ldots)$, 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 $\beta_t$ 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 $\beta_t$ 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 $M=4$ doesn’t detect 50% of errors of informational bits of multiplicity 2.

7. The $SM(n,m)$-code with $M=8$ 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 $\beta_t$ equal or less than the same characteristic for the $SM'(n,m)$-code $(M' > M)$ for every $t$.

<table>
<thead>
<tr>
<th>Table 2. Values of the characteristic $\beta_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
</tr>
<tr>
<td>$S(n,m)$</td>
</tr>
<tr>
<td>$S2(n,m)$</td>
</tr>
<tr>
<td>$S4(n,m)$</td>
</tr>
<tr>
<td>$S8(n,m)$</td>
</tr>
<tr>
<td>$S16(n,m)$</td>
</tr>
<tr>
<td>$S32(n,m)$</td>
</tr>
</tbody>
</table>

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 $M=2,4,\ldots,64$. Let’s also suppose that the experiment with the block $f(x)$ determined the maximum multiplicity of the error as $t_{\text{max}}=8$. 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 $\beta_t$ for errors of the multiplicity $t \leq 8$. Therefore in this case it
is advisable to use the $S_8(103,100)$-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 $M=2,4,8,...,2^b$
($b = \lceil \log_2(m+1) \rceil - 2$) 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 $\alpha$
is determined. If $x_m \oplus x_{m-1} \oplus ... \oplus x_0 = 0$
($p = \lceil \log_2 M \rceil + 2$), then $\alpha=0$, otherwise $\alpha=1$. 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 $q=6$, $W=6(mod2)=0$,
$p = \lceil \log_2 2 \rceil + 2 = 3$, $\alpha=0$
(because $x_6 \oplus x_5 \oplus x_4 \oplus x_3 = 0$), $V=0$ (because $W=0$ and $\alpha=0$).
The check vector is $y_{2y_1}=00$.

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

<table>
<thead>
<tr>
<th>Informational word</th>
<th>q</th>
<th>W</th>
<th>α</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_6$ $s_5$ $s_4$ $s_3$ $s_2$ $s_1$</td>
<td>0 0 0 1 1 1</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 1 0 1 1</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1 0 0 1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1 1</td>
</tr>
<tr>
<td>1 1 1 0 1 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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 $M \geq 4$ 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>
<thead>
<tr>
<th>Non-modified codes</th>
<th>Modified codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S(n,m)$</td>
<td>$RS(n,m)$</td>
</tr>
<tr>
<td>$S2(n,m)$</td>
<td>$RS2(n,m)$</td>
</tr>
<tr>
<td>$S4(n,m)$</td>
<td>$RS4(n,m)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$S2^k(n,m)$</td>
<td>$RS2^k(n,m)$</td>
</tr>
</tbody>
</table>
Table 4. Table of codes with $m=16$

<table>
<thead>
<tr>
<th>Codes</th>
<th>$k$</th>
<th>Multiplicity of error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>7864320</td>
<td>119275520</td>
</tr>
</tbody>
</table>

The total number of errors of current multiplicity:

- $S(21,16)$: 7864320, 119275520, 524812288, 84348320, 4294901760
- $S(17,16)$: 3932160, 59637760, 262406144, 421724160, 1073709056
- $S(18,16)$: 2654208, 31096832, 126517248, 210862080, 534267024
- $S(19,16)$: 2368512, 21790720, 82677760, 115315200, 348095908
- $S(20,16)$: 2129920, 21749760, 82677760, 115315200, 348095908
- $S(21,16)$: 2129920, 21749760, 82677760, 115315200, 348095908

Number/percent of undetectable errors:

- $S(21,16)$: 50%, 37.50%, 31.25%, 27.34%, 13.99%
- $S(17,16)$: 100%, 100%, 100%, 100%, 49.99%
- $S(18,16)$: 50%, 50%, 50%, 50%, 24.99%
- $S(19,16)$: 50%, 50%, 50%, 50%, 24.99%
- $S(20,16)$: 50%, 50%, 50%, 50%, 24.99%
- $S(21,16)$: 50%, 50%, 50%, 50%, 24.99%

6. References


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.