

УДК 681.518.5:004.052.32

DOI: 10.17223/19988605/37/2

Д.В. Ефанов

**К ВОПРОСУ СИНТЕЗА ГЕНЕРАТОРОВ МОДИФИЦИРОВАННЫХ КОДОВ
С СУММИРОВАНИЕМ ВЗВЕШЕННЫХ ИНФОРМАЦИОННЫХ РАЗРЯДОВ
С ПОСЛЕДОВАТЕЛЬНОСТЬЮ ВЕСОВЫХ КОЭФФИЦИЕНТОВ,
ОБРАЗУЮЩЕЙ НАТУРАЛЬНЫЙ РЯД ЧИСЕЛ**

Излагаются результаты, полученные автором в области синтеза генераторов контрольных векторов модифицированных кодов Бергера. Данные коды принадлежат к классу модифицированных модульно взвешенных кодов с суммированием. Приводятся общие структуры генераторов таких кодов, а также алгоритм построения генератора, позволяющий оптимизировать его структуру, сократив количество используемых функциональных элементов. Выводится формула подсчета общего количества двухходовых логических элементов, необходимых для технической реализации генератора. Предложенный алгоритм синтеза генераторов является универсальным и может быть использован для построения генераторов любых модифицированных модульно взвешенных кодов с суммированием.

Ключевые слова: техническая диагностика; система функционального контроля; код Бергера; модифицированный код Бергера; взвешенный код с суммированием; наименьший неотрицательный вычет; модуль; генератор; полусумматор; полный сумматор.

В задачах технической диагностики дискретных систем часто применяют разнообразные коды с суммированием [1–17]. Данные коды принадлежат к классу систематических – в кодовых словах таких кодов выделяют информационный и контрольный векторы с длинами m и k соответственно. Введем обозначение для кодов с суммированием – (m,k) -коды.

Применение (m,k) -кодов в задачах технической диагностики, например при синтезе систем с обнаружением отказов [18–20], связано с внесением избыточности в структуру «базового» объекта. При этом способ построения кода напрямую определяет характеристики избыточности получаемого технического объекта, а также особенности идентифицируемых в нем неисправностей.

Наиболее простым среди известных кодов с суммированием является классический код Бергера [21]. Данный код имеет $k = \lceil \log_2(m+1) \rceil$ разрядов в контрольных векторах (запись $\lceil \dots \rceil$ обозначает целое сверху от вычисляемого значения), принимающих значения разрядов двоичных чисел, равных сумме единичных информационных разрядов (равных весу информационного вектора). Такие особенности построения кода Бергера приводят к крайне неравномерному распределению информационных векторов между контрольными векторами, что в конечном итоге сказывается на свойствах обнаружения им ошибок. Кодами Бергера не обнаруживается любая симметричная ошибка в информационном векторе (такая ошибка не нарушает веса информационного вектора и происходит при одинаковом количестве искажаемых нулевых и единичных разрядов [22]). Вне зависимости от длины информационного вектора кодами Бергера не обнаруживается одинаковый процент ошибок четной кратностью d от общего количества ошибок данной кратностью [23–25]. Это, к примеру, 50% двукратных и 37,5% четырехкратных ошибок в информационных векторах. С неравномерностью распределения информационных векторов между контрольными связано и большое общее количество необнаруживаемых ошибок в информационных векторах кодов Бергера. Тем не менее данные коды нашли широкое применение за счет их свойства обнаружения любых монотонных ошибок в кодовых словах [26–31].

Любые модификации классических кодов связаны с перераспределением информационных векторов между контрольными векторами, что позволяет уменьшать количество необнаруживаемых в информационных векторах ошибок. В [32] введено понятие оптимального по критерию минимума общего количества необнаруживаемых ошибок (m,k) -кода с установленными значениями длин информационных и контрольных векторов и показано, что классический код Бергера далек от оптимального кода.

В [33] разработан перспективный для задач технической диагностики модифицированный код Бергера, основанный на идее взвешивания разрядов информационных векторов и последующих операциях модификации значения суммарного веса информационного вектора. Данная работа посвящена изложению способа синтеза генератора, или кодера, данного оптимального кода с суммированием.

1. Модифицированный код Бергера

Как отмечалось выше, способ построения оптимального (m,k) -кода базируется на первоначальном установлении неравноправия между разрядами в информационном векторе путем приписывания им различных весовых коэффициентов [10, 13, 21, 34–36], а значения разрядов контрольных векторов получаются следующим образом.

Алгоритм 1. Правила построения оптимального кода с суммированием:

1. Устанавливается последовательность весовых коэффициентов разрядов информационного вектора, образующая натуральный ряд чисел, начиная с младшего разряда.

2. Вычисляется значение модуля $M = 2^{\lceil \log_2(m+1) \rceil - 1}$.

3. Определяется значение суммарного веса единичных разрядов информационного вектора:

$$W = \sum_{i=1}^m w_i f_i, \quad (1)$$

где $f_i = 0$ или $f_i = 1$ – значения разрядов информационного вектора.

4. Определяется наименьший неотрицательный вычет числа W по модулю M :

$$[W]_M = W \pmod{M}. \quad (2)$$

5. Вычисляется специальный поправочный коэффициент α как функция паритета разрядов информационного вектора, занимающих позиции с четными номерами, начиная с младшего разряда:

$$\alpha = \begin{cases} f_2 \oplus f_4 \oplus \dots \oplus f_m, & \text{если } m - \text{четное;} \\ f_2 \oplus f_4 \oplus \dots \oplus f_{m-1}, & \text{если } m - \text{нечетное.} \end{cases} \quad (3)$$

6. Формируется значение результирующего веса информационного вектора:

$$V = [W]_M + \alpha M. \quad (4)$$

7. Число V представляется в двоичном виде и записывается в разряды контрольного вектора.

Получаемые по алгоритму 1 коды с суммированием обозначим как $RWS(m,k)$ -коды. Они принадлежат к классу модифицированных модульно взвешенных кодов с суммированием и имеют такое же количество разрядов в контрольных векторах, как и классические коды Бергера, однако обладают улучшенными характеристиками обнаружения ошибок в информационных векторах.

Пользуясь алгоритмом 1, приведем пример построения $RWS(4,3)$ -кода (табл. 1).

Таблица 1

Кодовые слова $RWS(4,3)$ -кода

№ п/п	Информационный вектор				W	W_M	A	V	Контрольный вектор		
	f_4	f_3	f_2	f_1					g_3	g_2	g_1
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	1	0	0	1
2	0	0	1	0	2	2	1	6	1	1	0
3	0	0	1	1	3	3	1	7	1	1	1
4	0	1	0	0	3	3	0	3	0	1	1
5	0	1	0	1	4	0	0	0	0	0	0
6	0	1	1	0	5	1	1	5	1	0	1
7	0	1	1	1	6	2	1	6	1	1	0
8	1	0	0	0	4	0	1	4	1	0	0
9	1	0	0	1	5	1	1	5	1	0	1
10	1	0	1	0	6	2	0	2	0	1	0
11	1	0	1	1	7	3	0	3	0	1	1
12	1	1	0	0	7	3	1	7	1	1	1
13	1	1	0	1	8	0	1	4	1	0	0
14	1	1	1	0	9	1	0	1	0	0	1
15	1	1	1	1	10	2	0	2	0	1	0

Данный код не обнаруживает 8 двухкратных и 8 четырехкратных ошибок в информационных векторах (для сравнения: кодом Бергера при данной длине информационного вектора не обнаруживается 48 двухкратных и 6 четырехкратных ошибок [23]). Вообще $RWS(m,k)$ -кодами идентифицируется гораздо больше ошибок малых кратностей, чем кодами Бергера. Кроме того, данный класс кодов, так же как и коды Бергера, обнаруживает любые ошибки нечетных кратностей в информационных векторах.

2. Синтез генераторов модифицированных кодов Бергера

Поскольку при построении $RWS(m,k)$ -кодов используются операции суммирования весовых коэффициентов и значений разрядов информационных векторов, в качестве элементной базы для их генераторов удобно использовать стандартные схемы сумматоров по модулю два (XOR), полусумматоров (HA) и полных сумматоров (FA) [7, 37–39]. На рис. 1 приводятся условные обозначения данных функциональных элементов. При этом в условных обозначениях полусумматора и полного сумматора на входах и выходах указаны веса разрядов двоичных чисел.

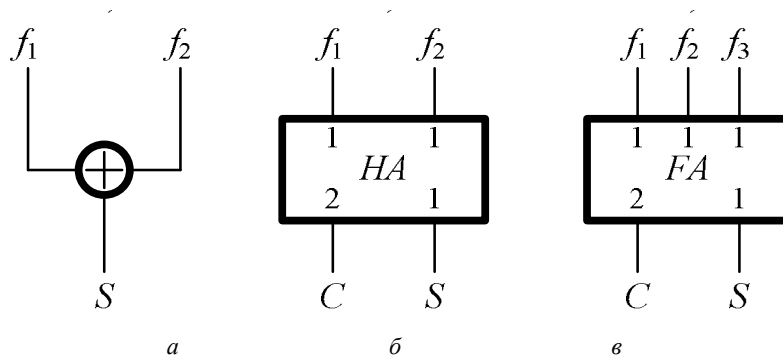


Рис. 1. Условные обозначения типовых функциональных элементов:
а – сумматора по модулю два; б – полусумматора; в – полного сумматора

Сумматор по модулю два реализует операцию сложения двух одноразрядных чисел без переноса и вычисляет функцию

$$S = f_1 \oplus f_2.$$

Схемы полусумматора и полного сумматора предназначены для сложения двух и трех двоичных чисел. Полусумматор имеет два входа и два выхода, на которых формируется двоичное число, равное суммарному количеству единиц на его входах:

$$\begin{cases} S = f_1 \oplus f_2; \\ C = f_1 f_2. \end{cases}$$

Аналогичную полусумматору функцию выполняет полный сумматор, имея при этом, однако, три входа и два выхода, описываемых следующими формулами:

$$\begin{cases} S = f_1 \oplus f_2 \oplus f_3; \\ C = f_1 f_2 \vee f_1 f_3 \vee f_2 f_3. \end{cases}$$

Согласно (1) генератор $RWS(m,k)$ -кода должен сначала подсчитать сумму весовых коэффициентов единичных разрядов информационного вектора, затем вычислить по формуле (2) наименьший неотрицательный вычет для полученного числа и прибавить к нему значение модуля при $\alpha = 1$ или сохранить неизменным при $\alpha = 0$. Таким образом, структурная схема генератора $RWS(m,k)$ -кода может иметь вид, представленный на рис. 2.

В генераторе выделяется четыре функциональных блока: B_1 предназначен для «взвешивания» разрядов; B_2 – устройство суммирования весовых коэффициентов; B_3 – схема определения наименьшего неотрицательного вычета суммарного веса единичных информационных разрядов; B_4 – формирователь старшего разряда контрольного вектора.

Схема генератора $RWS(m,k)$ -кода может быть построена и иным образом.

Преобразуем выражение (4), подставив в него выражения (1) и (2):

$$\begin{aligned}
 V &= [W]_M + \alpha M = W \pmod{M} + \alpha M = \left(\sum_{i=1}^m w_i f_i \right) \pmod{M} + \alpha M = \\
 &= (w_1 f_1 + w_2 f_2 + \dots + w_m f_m) \pmod{M} + \alpha M = \\
 &= w_1 f_1 \pmod{M} + w_2 f_2 \pmod{M} + \dots + w_m f_m \pmod{M} + \alpha M.
 \end{aligned}
 \tag{5}$$

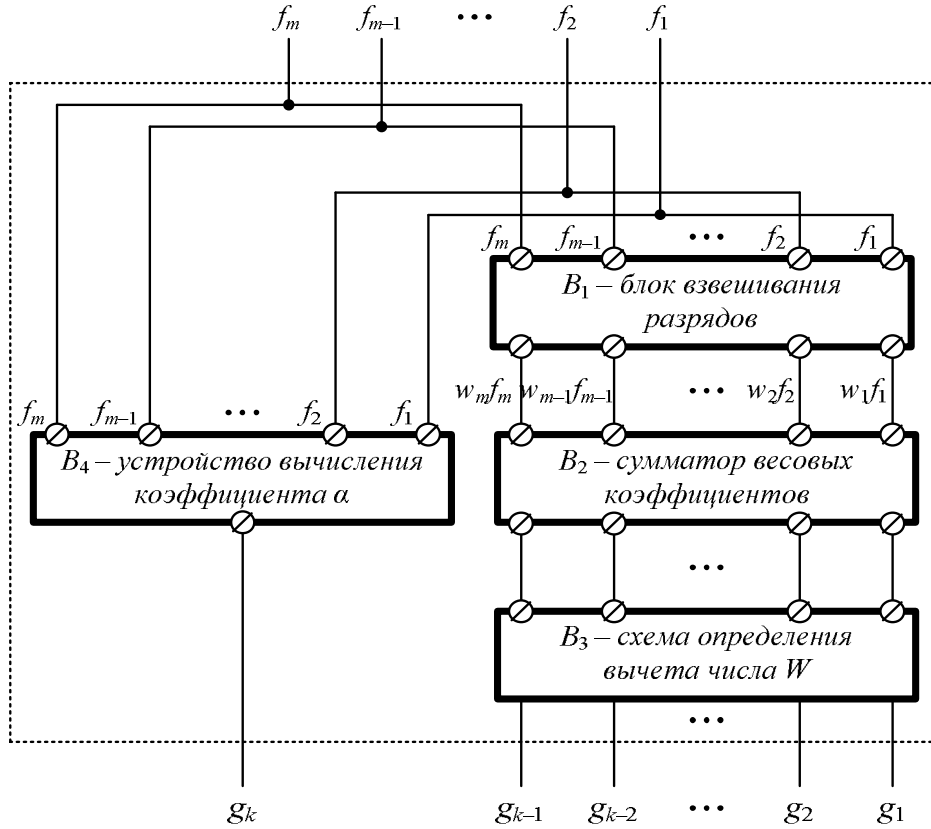


Рис. 2. Структурная схема первого типа генератора $RWS(m,k)$ -кода

Из формулы (5) следует, что перед тем, как определять суммарный вес единичных информационных разрядов, можно напрямую вычислять наименьшие неотрицательные вычеты для каждого весового коэффициента. Таким образом, структурная схема генератора $RWS(m,k)$ -кода примет вид, изображенный на рис. 3. В ней также выделяются четыре функциональных блока: B_1 предназначен для «взвешивания» разрядов; B_2 – устройство определения наименьших неотрицательных вычетов для каждого весового коэффициента; B_3 – схема суммирования наименьших неотрицательных вычетов весовых коэффициентов единичных разрядов по установленному модулю M ; B_4 – формирователь старшего разряда контрольного вектора.

Фактически блоки B_1 – B_3 могут быть построены путем каскадного соединения типовых схем сложения двоичных чисел по заданному модулю M . Такая типовая схема изображена на рис. 4. На ее входы подаются два двоичных числа $A = \langle a_1 a_2 \dots a_m \rangle$ и $B = \langle b_1 b_2 \dots b_m \rangle$, являющиеся двоичными представлениями наименьших неотрицательных вычетов по модулю M двух весовых коэффициентов. А на выходах формируется сумма чисел A и B за исключением переноса в старший разряд при переполнении – число $S = \langle s_1 s_2 \dots s_m \rangle$. В частности, на рис. 5 изображены типовые схемы сложения двоичных чисел по модулям $M = 4, 8$ и 16 , используемым при формировании $RWS(m,k)$ -кодов с длинами информационных векторов $m < 32$.

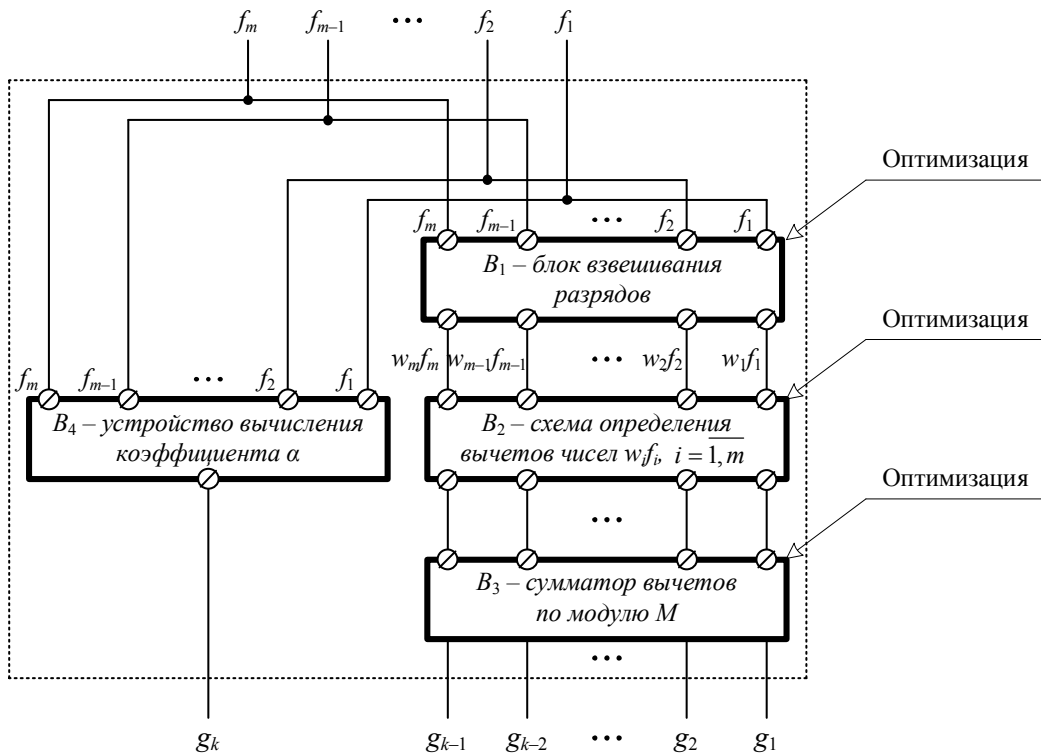


Рис. 3. Структурная схема второго типа генератора $RWS(m,k)$ -кода

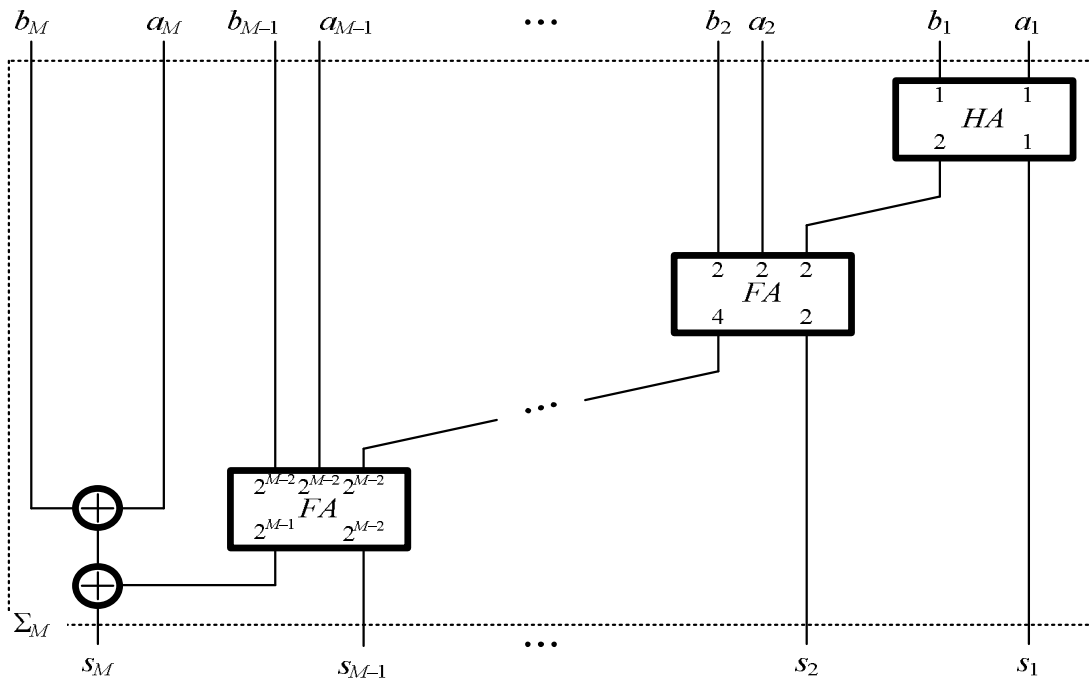


Рис. 4. Структурная схема функционального блока Σ_M

Целесообразно синтезировать генераторы по структурной схеме, приведенной на рис. 3, следующим способом.

Алгоритм 2. Правила синтеза генераторов $RWS(m,k)$ -кодов.

1. Устанавливается значение модуля M .
2. Определяется наименьший неотрицательный вычет по модулю M для каждого весового коэффициента w_i :

$$[w_i]_M = w_i \pmod{M}.$$

3. Каждое число $[w_i]_M$ представляется в двоичном виде.

4. Единичным разрядам двоичных чисел $[w_i]_M$ ставятся в соответствие значения разряда f_i .

5. Осуществляется разбиение двоичных чисел $[w_i]_M$ на пары с учетом оптимального заполнения рядов: числа $[w_i]_M$ разбиваются на $\left\lceil \frac{m}{2} \right\rceil$ групп, каждая из которых представляет собой пару двоичных чисел, полученных в п. 4 данного алгоритма, суммирование которых осуществляется без переносов.

6. Выполняется сложение чисел с учетом значений их разрядов и получение $\left\lceil \frac{m}{2} \right\rceil$ двоичных чисел.

7. С использованием функциональных блоков сложения двоичных чисел модулю M реализуется устройство суммирования $\left\lceil \frac{m}{2} \right\rceil$ двоичных чисел.

8. Реализуется блок вычисления старшего разряда контрольного вектора.

Из алгоритма 2, в частности, следует, что блоки B_1 и B_2 генератора не требуют функциональных элементов и содержат только провода, на которые подаются сигналы 0 или f_i . Более того, из п. 6 алгоритма следует, что фактически первым каскадом блока B_3 будут выступать только провода с поданными на них сигналами 0 или f_i . Таким образом, алгоритм 2 подразумевает некоторую оптимизацию структуры генератора $RWS(m,k)$ -кода.

Продемонстрируем работу алгоритма 2 на примере синтеза генератора $RWS(8,4)$ -кода.

Определим значения наименьших неотрицательных вычетов для всех весовых коэффициентов w_i и представим их в двоичном виде. После этого в каждом двоичном представлении вычета веса w_i все единичные разряды заменяем значениями соответствующего информационного разряда f_i , а все нулевые разряды оставляем заполненными значениями 0. Результат приведен в табл. 2.

Т а б л и ц а 2

Получение суммируемых двоичных чисел

Весовые коэффициенты							
w_8	w_7	w_6	w_5	w_4	w_3	w_2	w_1
8	7	6	5	4	3	2	1
$[w_i]_M$							
1000	0111	0110	0101	0100	0011	0010	0001
Двоичное число, соответствующее весу							
$f_8 000$	$0f_7f_7$	$0f_6f_6$	$0f_5f_5$	$0f_400$	$00f_3f_3$	$00f_20$	$000f_1$

Далее полученные двоичные числа разбиваются на $\left\lceil \frac{8}{2} \right\rceil = 4$ группы. При этом числа в каждой группе должны складываться без переносов (см. рис. 6). Данные двоичные числа требуется сложить по модулю $M = 8$ с использованием функциональных блоков, изображенных на рис. 5, б. Так как получено четыре двоичных числа, то потребуются три функциональных блока.

Отдельно реализуется функция паритета $\alpha = f_2 \oplus f_4 \oplus f_6 \oplus f_8$, вычисляющая старший разряд контрольного вектора $RWS(8,4)$ -кода.

Синтезированный генератор рассматриваемого кода представлен на рис. 7. На каждой его линии указаны логические сигналы 0 и 1, соответствующие вычислению значений разрядов контрольного вектора при поступлении на входы генератора информационного вектора $\langle 10100110 \rangle$.

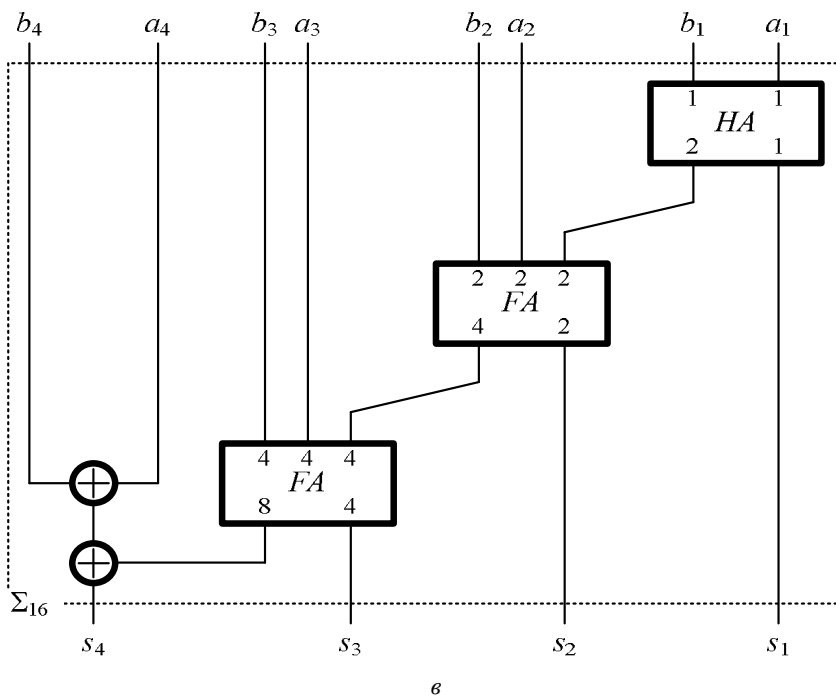
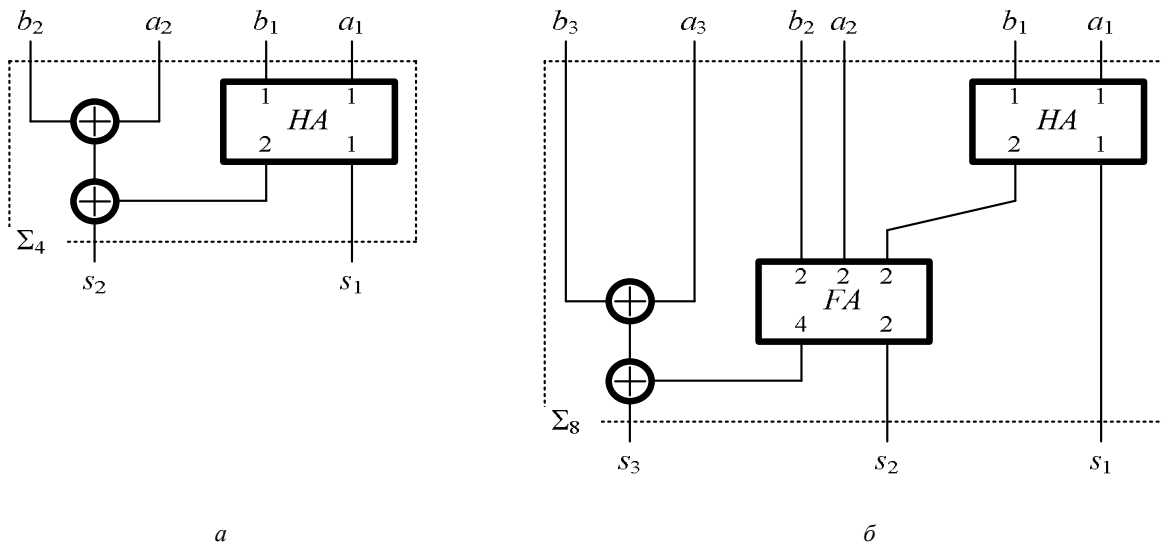


Рис. 5. Структурные схемы функциональных блоков Σ_M :
 а – сложения по модулю $M = 4$; б – сложения по модулю $M = 8$; в – сложения по модулю $M = 16$

$w_7 = \langle 0 f_7 f_7 f_7 \rangle$ $+$ $w_8 = \langle f_8 0 0 0 \rangle$ <hr style="width: 100%;"/> $w_7 + w_8 = \langle f_8 f_7 f_7 f_7 \rangle$ $(w_7 + w_8) \pmod{8} = \langle f_7 f_7 f_7 \rangle$	$w_1 = \langle 0 0 0 f_1 \rangle$ $+$ $w_6 = \langle 0 f_6 f_6 0 \rangle$ <hr style="width: 100%;"/> $w_1 + w_6 = \langle 0 f_6 f_6 f_1 \rangle$ $(w_1 + w_6) \pmod{8} = \langle f_6 f_6 f_1 \rangle$	$w_2 = \langle 0 0 f_2 0 \rangle$ $+$ $w_5 = \langle 0 f_5 0 f_5 \rangle$ <hr style="width: 100%;"/> $w_2 + w_5 = \langle 0 f_5 f_2 f_5 \rangle$ $(w_2 + w_5) \pmod{8} = \langle f_5 f_2 f_5 \rangle$	$w_3 = \langle 0 0 f_3 f_3 \rangle$ $+$ $w_4 = \langle 0 f_4 0 0 \rangle$ <hr style="width: 100%;"/> $w_3 + w_4 = \langle 0 f_4 f_3 f_3 \rangle$ $(w_3 + w_4) \pmod{8} = \langle f_4 f_3 f_3 \rangle$
---	---	---	---

Рис. 6. Получение двоичных чисел, суммируемых на входах функциональных блоков сложения по модулю $M = 8$

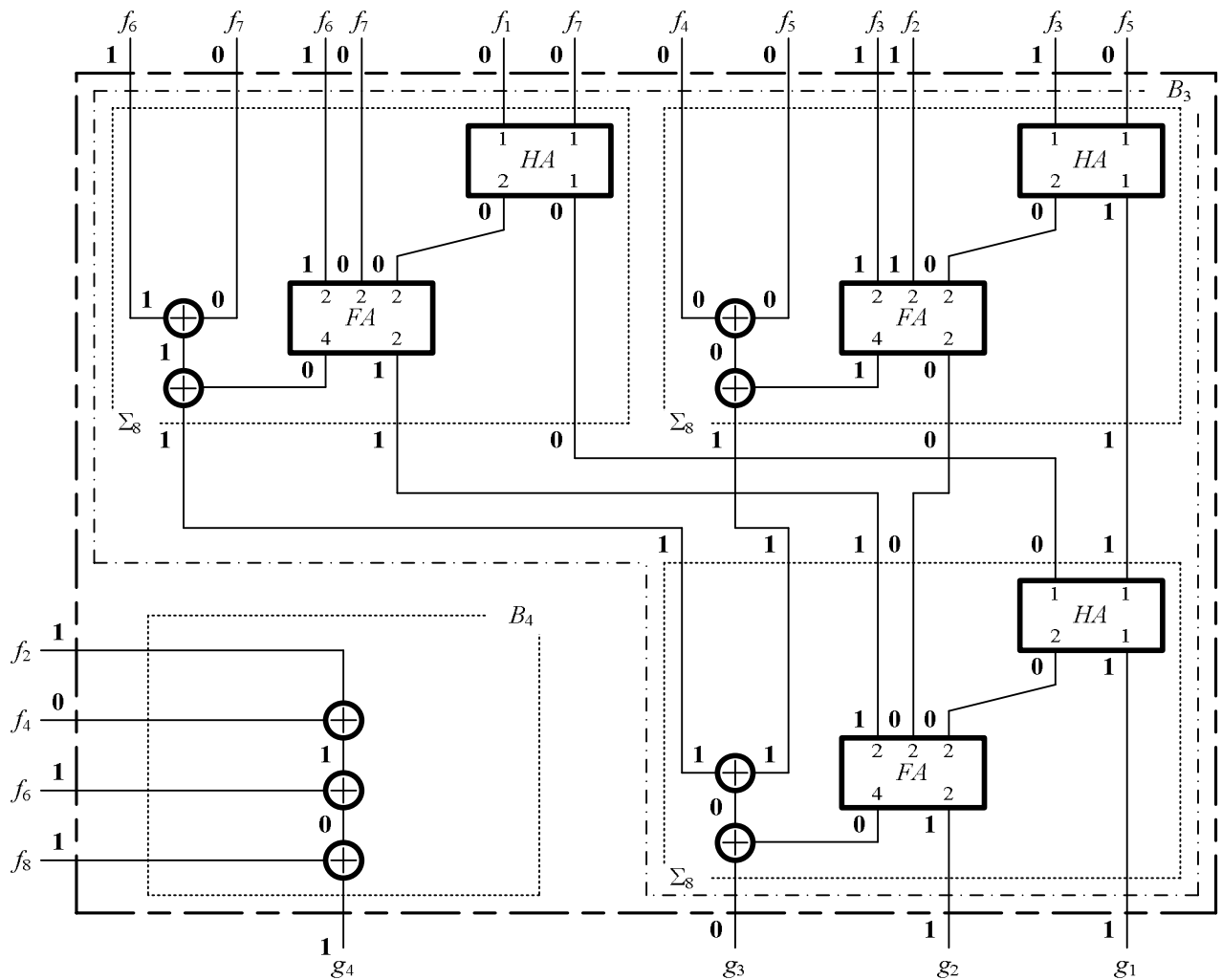


Рис. 7. Генератор разрядов контрольного вектора $RWS(8,4)$ -кода

Поскольку все весовые коэффициенты $RWS(m,k)$ -кода при использовании алгоритма 2 делятся на $\lfloor \frac{m}{2} \rfloor$ групп для их последующего суммирования, справедливо следующее положение.

Утверждение 1. Генератор $RWS(m,k)$ -кода содержит $\lfloor \frac{m}{2} \rfloor - 1$ функциональный блок Σ_M и блок формирования старшего контрольного разряда, включающий в себя $\lfloor \frac{m}{2} \rfloor - 1$ сумматоров по модулю два (запись $\lfloor \dots \rfloor$ обозначает целое снизу от вычисляемого значения).

Сформулированное выше положение о количестве функциональных элементов в структуре генератора модифицированного кода Бергера позволяет дать оценку сложности его технической реализации в зависимости от длины информационного вектора. Одним из таких показателей является количество двухвходовых элементарных логических элементов, требующихся для реализации генератора [6]. Данный показатель известен для сумматора по модулю два, полусумматора и полного сумматора [40]: $L_{XOR} = 3$, $L_{HA} = 3$, $L_{FA} = 7$. Зная эти данные, нетрудно оценить сложность технической реализации функциональных блоков Σ_M : $L_{\Sigma_4} = 9$, $L_{\Sigma_8} = 16$ и $L_{\Sigma_{16}} = 23$. В табл. 3 приводятся рассчитанные значения показателя сложности технической реализации генераторов $RWS(m,k)$ -кодов при длинах информационных векторов $m = 3 \div 20$, а на рис. 8 иллюстрируется зависимость сложности технической реализации генераторов от длины информационного вектора. Она носит линейный характер.

Формализуем процедуру оценки сложности технической реализации генераторов $RWS(m,k)$ -кодов.

Из утверждения 1 ясно, что сложность технической реализации генератора определяется суммой

$$L_{RWS} = \left(\left\lceil \frac{m}{2} \right\rceil - 1 \right) L_{\Sigma_M} + \left(\left\lceil \frac{m}{2} \right\rceil - 1 \right) L_{XOR}. \quad (6)$$

Таблица 3

Показатели сложности технической реализации генераторов $RWS(m,k)$ -кодов

m	Формула генератора	Количество двухвходовых элементов в структуре генератора
3	$2XOR$	6
4	$1XOR+1\Sigma_4$	12
5	$1XOR+2\Sigma_4$	21
6	$2XOR+2\Sigma_4$	24
7	$2XOR+3\Sigma_4$	33
8	$3XOR+3\Sigma_8$	57
9	$3XOR+4\Sigma_8$	73
10	$4XOR+4\Sigma_8$	76
11	$4XOR+5\Sigma_8$	92
12	$5XOR+5\Sigma_8$	95
13	$5XOR+6\Sigma_8$	111
14	$6XOR+6\Sigma_8$	114
15	$6XOR+7\Sigma_8$	130
16	$7XOR+7\Sigma_{16}$	182
17	$7XOR+8\Sigma_{16}$	205
18	$8XOR+8\Sigma_{16}$	208
19	$8XOR+9\Sigma_{16}$	231
20	$9XOR+9\Sigma_{16}$	234

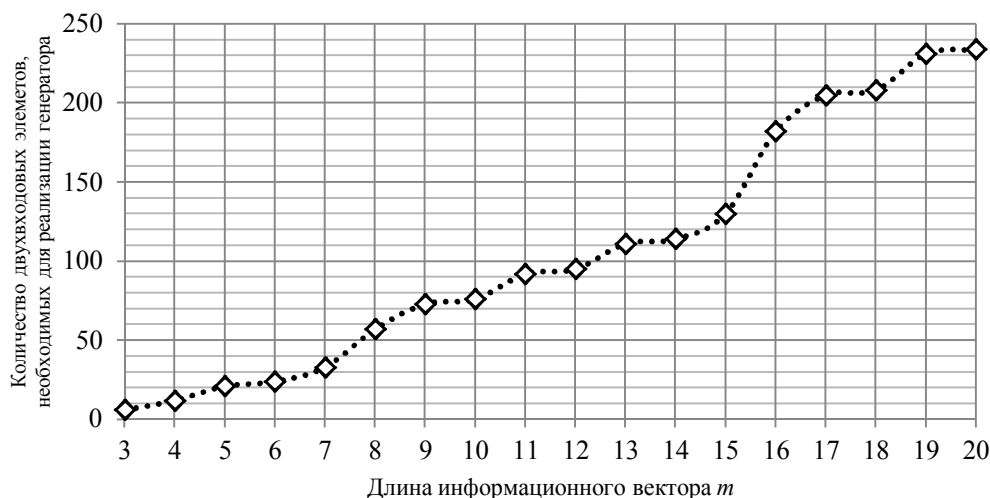


Рис. 8. Зависимость показателя сложности технической реализации генератора $RWS(m,k)$ -кода от длины информационного вектора

Величина L_{Σ_M} в формуле (6) зависит от количества разрядов в контрольных векторах. Каждый такой блок, за исключением случая $M = 2$, содержит один элемент HA , $\lceil \log_2(m+1) \rceil - 3$ элемента FA и два элемента XOR (см. рис. 4 и 5). Это объясняется тем, что на полусумматоре реализуется функция вычисления младшего разряда контрольного вектора $RWS(m,k)$ -кода, на $k-3$ полных сумматорах реализуются контрольные разряды g_2, g_3, \dots, g_{k-2} , а на двух сумматорах по модулю два – контрольный разряд g_{k-1} . Таким образом, сложность функционального блока Σ_M вычисляется по формуле

$$L_{\Sigma_M} = L_{HA} + (\lceil \log_2(m+1) \rceil - 3) L_{FA} + 2L_{XOR}. \quad (7)$$

Подставим (7) в (6), а также внесем туда значения $L_{XOR} = 3, L_{HA} = 3, L_{FA} = 7$.

Утверждение 2. Количество двухвходовых элементов в структуре генератора $RWS(m,k)$ -кода определяется выражением

$$\begin{aligned} L_{RWS(m,k)} &= \left(\left\lceil \frac{m}{2} \right\rceil - 1 \right) \left(L_{HA} + (\lceil \log_2(m+1) \rceil - 3) L_{FA} + 2L_{XOR} \right) + \left(\left\lfloor \frac{m}{2} \right\rfloor - 1 \right) L_{XOR} = \\ &= \left(\left\lceil \frac{m}{2} \right\rceil - 1 \right) \left(7 \cdot (\lceil \log_2(m+1) \rceil - 3) + 9 \right) + 3 \cdot \left(\left\lfloor \frac{m}{2} \right\rfloor - 1 \right) = \\ &= 7 \lceil \log_2(m+1) \rceil \left(\left\lceil \frac{m}{2} \right\rceil - 1 \right) - 12 \left\lfloor \frac{m}{2} \right\rfloor + 3 \left\lfloor \frac{m}{2} \right\rfloor + 9. \end{aligned} \quad (8)$$

Использование формулы (8) для рассмотренного выше $RWS(8,4)$ -кода дает следующий результат:

$$L_{RWS(8,4)} = 7 \lceil \log_2(8+1) \rceil \left(\left\lceil \frac{8}{2} \right\rceil - 1 \right) - 12 \left\lfloor \frac{8}{2} \right\rfloor + 3 \left\lfloor \frac{8}{2} \right\rfloor + 9 = 7 \lceil \log_2(9) \rceil \left(\left\lceil \frac{8}{2} \right\rceil - 1 \right) - 12 \left\lfloor \frac{8}{2} \right\rfloor + 3 \left\lfloor \frac{8}{2} \right\rfloor + 9 = 57.$$

Заключение

Предложенный в настоящей работе алгоритм синтеза генераторов модифицированных кодов Бергера позволяет строить данные устройства с простой структурой на основе каскадного соединения функциональных блоков сложения по модулю M . Сложность структур генераторов с увеличением длины информационного вектора возрастает линейно.

Описанный способ синтеза генераторов является универсальным и может быть применен для построения генераторов как кодов с суммированием взвешенных информационных разрядов [10, 34, 40–44], так и кодов с суммированием взвешенных переходов между разрядами, занимающими соседние позиции в информационных векторах [13, 35, 45–49]. Однако возможность оптимизации структур генераторов будет напрямую определяться значениями весовых коэффициентов кода.

ЛИТЕРАТУРА

1. Слабаков Е.В. Построение полностью самопроверяемых комбинационных устройств с использованием остаточных кодов // Автоматика и телемеханика. 1979. № 10. С. 133–141.
2. Слабаков Е.В., Согомонян Е.С. Самопроверяемые вычислительные устройства и системы (обзор) // Автоматика и телемеханика. 1981. № 11. С. 147–167.
3. McCluskey E.J. Logic Design Principles: With Emphasis on Testable Semicustom Circuits. N.J.: Prentice Hall PTR, 1986. 549 p.
4. Сагалович Ю.Л. Кодовая защита оперативной памяти ЭВМ от ошибок // Автоматика и телемеханика. 1991. № 5. С. 3–45.
5. Согомонян Е.С., Слабаков Е.В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
6. Сапожников В.В., Сапожников Вл.В. Самопроверяемые дискретные устройства. СПб.: Энергоатомиздат, 1992. 224 с.
7. Piestrak S.J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
8. Pradhan D.K. Fault-Tolerant Computer System Design. New-York: Prentice Hall, 1996. 560 p.
9. Das D., Toubia N.A. Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes // Journal of Electronic Testing: Theory and Applications. 1999. V. 15, Issue 1–2. P. 145–155.
10. Das D., Toubia N.A., Seuring M., Gossel M. Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes // Proceedings of IEEE 6th International On-Line Testing Workshop (IOLTW), Spain, Palma de Mallorca, July 3–5. 2000. P. 171–176.
11. Matrosova A.Yu., Levin I., Ostanin S.A. Self-Checking Synchronous FSM Network Design with Low Overhead // VLSI Design. 2000. V. 11, Issue 1. P. 47–58.
12. Fujiwara E. Code Design for Dependable Systems: Theory and Practical Applications. John Wiley & Sons, 2006. 720 p.
13. Mehov V., Saposhnikov V., Sapozhnikov V.I., Urganskov D. Concurrent Error Detection Based on New Code with Modulo Weighted Transitions between Information Bits // Proceedings of 7th IEEE East-West Design & Test Workshop (EWDWTW 2007), Erevan, Armenia, September 25–30. 2007. P. 21–26.
14. Блюдов А.А., Ефанов Д.В., Сапожников В.В., Сапожников Вл.В. Коды с суммированием для организации контроля комбинационных схем // Автоматика и телемеханика. 2013. № 6. С. 153–164.
15. Блюдов А.А., Ефанов Д.В., Сапожников В.В., Сапожников Вл.В. О кодах с суммированием единичных разрядов в системах функционального контроля // Автоматика и телемеханика. 2014. № 8. С. 131–145.
16. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Применение кодов с суммированием при синтезе систем железнодорожной автоматики и телемеханики на программируемых логических интегральных схемах // Автоматика на транспорте. 2015. Т. 1, № 1. С. 84–107.

17. Ефанов Д.В., Сапожников В.В., Сапожников Вл.В. Применение модульных кодов с суммированием для построения систем функционального контроля комбинационных логических схем // Автоматика и телемеханика. 2015. № 10. С. 152–169.
18. Nicolaidis M., Zorian Y. On-Line Testing for VLSI – A Compendium of Approaches // Journal of Electronic Testing: Theory and Applications. 1998. Issue 12. P. 7–20.
19. Mitra S., McClaskey E.J. Which Concurrent Error Detection Scheme to Choose? // Proceedings of International Test Conference, 2000, USA, Atlantic City, NJ, 03–05 October 2000. P. 985–994.
20. Ефанов Д.В., Грошев Г.М., Маликов О.Б. Способы организации систем функционального диагностирования логических схем без памяти // Электротехника. 2016. № 5. С. 53–56.
21. Berger J.M. A Note on Error Detecting Codes for Asymmetric Channels // Information and Control. 1961. V. 4, Issue 1. P. 68–73.
22. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Классификация ошибок в информационных векторах систематических кодов // Известия вузов. Приборостроение. 2015. Т. 58, № 5. С. 333–343.
23. Ефанов Д.В., Сапожников В.В., Сапожников Вл.В. О свойствах кода с суммированием в схемах функционального контроля // Автоматика и телемеханика. 2010. № 6. С. 155–162.
24. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Предельные свойства кода с суммированием // Известия Петербургского университета путей сообщения. 2010. № 3. С. 290–299.
25. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Черепанова М.Р. Модульные коды с суммированием в системах функционального контроля. I. Свойства обнаружения ошибок кодами в информационных векторах // Электронное моделирование. 2016. Т. 38, № 2. С. 27–48.
26. Busaba F.Y., Lala P.K. Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors // Journal of Electronic Testing: Theory and Applications. 1994. Issue 5. P. 19–28.
27. Гессель М., Морозов А.А., Сапожников В.В., Сапожников Вл.В. Построение комбинационных самопроверяемых устройств с монотонно независимыми выходами // Автоматика и телемеханика. 1994. № 7. С. 148–160.
28. Гессель М., Морозов А.А., Сапожников В.В., Сапожников Вл.В. Исследование комбинационных самопроверяемых устройств с независимыми и монотонно независимыми выходами // Автоматика и телемеханика. 1997. № 2. С. 180–193.
29. Morosow A., Sapozhnikov V.V., Sapozhnikov V.I., Goessel M. Self-Checking Combinational Circuits with Unidirectionally Independent Outputs // VLSI Design. 1998. V. 5, Issue 4. P. 333–345.
30. Sapozhnikov V.V., Morosov A., Sapozhnikov V.I., Göessel M. A New Design Method for Self-Checking Unidirectional Combinational Circuits // Journal of Electronic Testing: Theory and Applications. 1998. V. 12, Issue 1–2. P. 41–53.
31. Matrosova A., Levin I., Ostanin S. Survivable Self-Checking Sequential Circuits // Proceedings of 2001 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT 2001). October 24–26, San Francisco, CA, 2001. P. 395–402.
32. Блюдов А.А., Ефанов Д.В., Сапожников В.В., Сапожников Вл.В. Построение модифицированного кода Бергера с минимальным числом необнаруживаемых ошибок информационных разрядов // Электронное моделирование. 2012. Т. 34, № 6. С. 17–29.
33. Efanov D., Sapozhnikov V., Sapozhnikov V.I., Nikitin D. Sum Code Formation with Minimum Total Number of Undetectable Errors in Data Vectors // Proceedings of 13th IEEE East-West Design & Test Symposium (EWDTS'2015), Batumi, Georgia, September 26–29. 2015. P. 141–148.
34. Das D., Touba N.A. Weight-Based Codes and Their Application to Concurrent Error Detection of Multilevel Circuits // Proceedings of 17th IEEE Test Symposium, USA, California, 1999. P. 370–376.
35. Мехов В.Б., Сапожников В.В., Сапожников Вл.В. Контроль комбинационных схем на основе модифицированных кодов с суммированием // Автоматика и телемеханика. 2008. № 8. С. 153–165.
36. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Взвешенные коды с суммированием для организации контроля логических устройств // Электронное моделирование. 2014. Т. 36, № 1. С. 59–80.
37. Marouf M.A., Friedman A.D. Design of Self-Checking Checkers for Berger Codes // Proceedings of 8th Annual International Conference on Fault-Tolerant Computing, Toulouse, France, 1978. P. 179–183.
38. Lala P.K. Principles of Modern Digital Design. New Jersey : John Wiley & Sons, 2007. 436 p.
39. Blyudov A.A. On the Synthesis of Test Equipment for Modulo Codes with Summation // Proceedings of Petersburg Transport University. 2013. Issue 1. P. 53–58.
40. Дрозд А.В., Харченко В.С., Антошук С.Г., Дрозд Ю.В., Дрозд М.А., Сулима Ю.Ю. Рабочее диагностирование безопасных информационно-управляющих систем / под ред. А.В. Дрозда, В.С. Харченко. Харьков : Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», 2012. 614 с.
41. Favalli M., Metra C. Optimization of Error Detecting Codes for the Detection of Crosstalk Originated Errors // Design, Automation and Test in Europe (DATE), March 13–16, 2001. P. 290–296.
42. Ghosh S., Lai K.W., Jone W.B., Chang S.C. Scan Chain Fault Identification Using Weight-Based Codes for SoC Circuits // Proceedings of 13th Asian Test Symposium, 15–17 November. 2004. P. 210–215.
43. Sapozhnikov V., Sapozhnikov V.I., Efanov D., Nikitin D. Combinational Circuits Checking on the Base of Sum Codes with One Weighted Data Bit // Proceedings of 12th IEEE East-West Design & Test Symposium (EWDTS'2014), Kyev, Ukraine, September 26–29, 2014. P. 126–136. DOI 10.1109/EWDTS.2014.7027064.
44. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Никитин Д.А. Модульно взвешенный код с суммированием для систем технического диагностирования // Информатика и системы управления. 2015. № 3. С. 53–62.
45. Sapozhnikov V., Sapozhnikov V.I. New Code for Fault Detection in Logic Circuits // Proceedings of 4th International conference on Unconventional Electromechanical and Electrical Systems, St. Petersburg, Russia, June 21–24. 1999. P. 693–696.

46. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Дмитриев В.В. Свойства кодов с суммированием взвешенных переходов с прямой последовательностью весовых коэффициентов // Информатика и системы управления. 2014. № 4. С. 77–88.
47. Sapozhnikov V., Sapozhnikov Vl., Efanov D., Dmitriev V., Cherepanova M. Optimum Sum Codes, That Effectively Detect the Errors of Low Multiplicities // RadioElectronics & Informatics. 2015. No. 1. P. 17–22.
48. Дмитриев В.В. О двух способах взвешивания и их влиянии на свойства кодов с суммированием взвешенных переходов в системах функционального контроля логических схем // Известия Петербургского университета путей сообщения. 2015. № 3. С. 119–129.
49. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Дмитриев В.В., Черепанова М.Р. Способ построения кода с суммированием с улучшенными показателями обнаружения ошибок в информационных векторах // Автоматика на транспорте. 2016. Т. 2, № 1. С. 95–123.

Ефанов Дмитрий Викторович, канд. техн. наук, доцент. E-mail: TrES-4b@yandex.ru

Петербургский государственный университет путей сообщения Императора Александра I (г. Санкт-Петербург)

Поступила в редакцию 29 июня 2016 г.

Efanov Dmitry V. (Petersburg State Transport University, St. Petersburg, Russian Federation).

On the problem of modified codes with summation of weighted data bits with natural numbers sequence of weight indexes generators synthesis.

Keywords: technical diagnostics; concurrent error detection system (CED system); Berger code; modified Berger code; weight-based sum code; least nonnegative residue; modulus; generator; half-adder; full-adder.

DOI: 10.17223/19988605/37/2

Different codes with summation (most of which are Berger code modifications) are often used in reliable discrete devices design. Modified code with summation of weighted data bits refers to such modifications. It is systematic code (the same as Berger code) and has $k = \lceil \log_2(m+1) \rceil$ check bits (m – data vector length). This code formation involves attribution of natural numbers weight indexes, calculation of total active bits weight and its least nonnegative residue by modulo $M = 2^{\lceil \log_2(m+1) \rceil - 1}$. This value is placed in low bits of check vector. High bit is calculated as modulo two sum of data bits with even positions (that have even weight indexes). This code has better data bits error detection properties comparing with Berger code. Also it is the code with minimum total number of undetectable data bits errors for given length of data and check vectors.

The paper describes the method of modified Berger code generator synthesis. Component base list contains standard circuits of modulo two, half and full adders. They compose functional blocks of $M = 2^{\lceil \log_2(m+1) \rceil - 1}$ sum. Generator itself is formed by cascade connection of such blocks.

Two structural diagrams of modified Berger code generators are given; also the generator synthesis algorithm is described that optimizes its structure by separation of weight indexes by groups of summed numbers without transfers. These numbers are sent on inputs of counters of $M = 2^{\lceil \log_2(m+1) \rceil - 1}$ sum. High bit of check vector is formed on the output of the block that calculate modulo two sum of data bits with even positions. The algorithm is illustrated on the example synthesis of the generator for modified Berger code with $m = 8$.

The paper shows that of modified Berger code includes $\lceil \frac{m}{2} \rceil - 1$ functional blocks of sum by $M = 2^{\lceil \log_2(m+1) \rceil - 1}$ modulo and block of high bit formation that consists of $\lceil \frac{m}{2} \rceil - 1$ modulo two adders. Dependence between complexity and data vector length is stated. It is shown that number of two-input gates in generator structure is calculated with the following expression:

$$7 \lceil \log_2(m+1) \rceil \left(\left\lceil \frac{m}{2} \right\rceil - 1 \right) - 12 \left\lceil \frac{m}{2} \right\rceil + 3 \left\lfloor \frac{m}{2} \right\rfloor + 9.$$

The described way of generator synthesis is universal and could be used for the formation of generators for codes with summation of weighted data bits and for codes with summation of weighted transitions between adjacent bits as well.

REFERENCES

1. Slabakov, E.V. (1979) Design of Completely Self-Checking Combinational Devices with the use of Residual Codes. *Automation and Remote Control*. 10. pp. 133-141.
2. Slabakov, E.V. & Sogomonyan, E.S. (1981) Self-Checking Computing Devices and Systems (Review). *Automation and Remote Control*. 11. pp. 147-167.
3. McCluskey, E.J. (1986) *Logic Design Principles: With Emphasis on Testable Semicustom Circuits*. New Jersey: Prentice Hall PTR.
4. Sagalovich, Yu.L. (1991) Error-Correcting Codes for Computer Memories. *Automation and Remote Control*. 5. pp. 3-45.
5. Sogomonyan, E.S. & Slabakov, E.V. (1989) *Samoproveryaemye ustroystva i otkazoustoychivyye sistemy* [Self-Checking Devices and Fault-Tolerant Systems]. Moscow: Radio i svyaz'.
6. Sapozhnikov, V.V. & Sapozhnikov Vl.V. (1992) *Samoproveryaemye diskretnyye ustroystva* [Self-Checking Digital Devices]. St. Petersburg: Energoatomizdat. 224 p.
7. Piestrak, S.J. (1995) *Design of Self-Testing Checkers for Unidirectional Error Detecting Codes*. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej.

8. Pradhan, D.K. (1996) *Fault-Tolerant Computer System Design*. New-York: Prentice Hall.
9. Das, D. & Touba, N.A. (1999) Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes. *Journal of Electronic Testing: Theory and Applications*. 15(1-2). pp. 145-155. DOI: 10.1023/A:1008344603814
10. Das, D., Touba, N.A., Seuring, M. & Gossel, M. (2000) Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes. *Proceedings of IEEE 6th International On-Line Testing Workshop (IOLTW)*. Spain, Palma de Mallorca. July 3-5. pp. 171-176. DOI: 10.1109/OLT.2000.856633
11. Matrosova, A.Yu., Levin, I. & Ostanin, S.A. (2000) Self-Checking Synchronous FSM Network Design with Low Overhead. *VLSI Design*. 11(1). pp. 47-58.
12. Fujiwara, E. (2006) *Code Design for Dependable Systems: Theory and Practical Applications*. John Wiley & Sons.
13. Mehov, V., Sapozhnikov, V., Sapozhnikov, V.I. & Urganskov, D. (2007) Concurrent Error Detection Based on New Code with Modulo Weighted Transitions between Information Bits. *Proceedings of 7th IEEE East-West Design & Test Workshop (EWDTS'2007)*. Erevan, Armenia. September 25-30. pp. 21-26.
14. Blyudov, A.A., Efanov, D.V., Sapozhnikov, V.V. & Sapozhnikov, V.I. (2013) Summation Codes for Organization of Control of Combinational Circuits. *Automation and Remote Control*. 6. pp. 153-164. DOI: 10.1134/S0005117913060118
15. Blyudov, A.A., Efanov, D.V., Sapozhnikov, V.V. & Sapozhnikov, V.I. (2014) On Summation Code of Unit Bits in Concurrent Error Detection Systems. *Automation and Remote Control*. 8. pp. 131-145. DOI: 10.1134/S0005117914080098
16. Sapozhnikov, V.V., Sapozhnikov, V.I. & Efanov, D.V. (2015) Primenenie kodov s summirovaniem pri sinteze sistem zhelezodorozhnoy avtomatiki i telemekhaniki na programmiruemykh logicheskikh integral'nykh skhemakh [Application of Codes with Summation for the Synthesis of Railway Automation and Remote Control Systems Based on Field-Programmable Gate Arrays]. *Avtomatika na transporte – Automation on Transport*. 1(1). pp. 84-107.
17. Efanov, D.V., Sapozhnikov, V.V. & Sapozhnikov, V.I. (2015) Applications of Modular Summation Codes to Concurrent Error Detection Systems for Combinational Boolean Circuits. *Automation and Remote Control*. 10. pp. 152-169. DOI: 10.1134/S0005117915100112
18. Nicolaidis, M. & Zorian, Y. (1998) On-Line Testing for VLSI – A Compendium of Approaches. *Journal of Electronic Testing: Theory and Applications*. 12. pp. 7-20. DOI: 10.1023/A:1008244815697
19. Mitra, S. & McClaskey, E.J. (2000) Which Concurrent Error Detection Scheme to Choose? *Proceedings of International Test Conference*. USA, Atlantic City, NJ, October 3-5. pp. 985-994. DOI: 10.1109/TEST.2000.894311
20. Efanov, D.V., Groshev, G.M. & Malikov, O.B. (2016) Sposoby organizatsii sistem funktsional'nogo diagnostirovaniya logicheskikh skhem bez pamyati [Ways of Organization of Concurrent Error Detection Systems for Combinational Circuits]. *Elektrotehnika – Electrical Engineering*. 5. pp. 53-56.
21. Berger, J.M. (1961) A Note on Error Detecting Codes for Asymmetric Channels. *Information and Control*. 4(1). pp. 68-73. DOI: 10.1016/S0019-9958(61)80037-5
22. Sapozhnikov, V.V., Sapozhnikov, V.I. & Ephanov, D.V. (2015) Errors classification in information vectors of systematic codes. *Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroenie – Journal of Instrument Engineering*. 58(5). pp. 333-343. DOI: 10.17586/0021-3454-2015-58-5-333-343. (In Russian).
23. Efanov, D.V., Sapozhnikov, V.V. & Sapozhnikov, V.I. (2010) On Summation Code Properties in Functional Control Circuits. *Automation and Remote Control*. 6. pp. 155-162. DOI: 10.1134/S0005117910060123
24. Sapozhnikov, V.V., Sapozhnikov, V.I. & Efanov, D.V. (2010) Limit Property of Berger Code. *Izvestiya Peterburgskogo universiteta putey soobshcheniya – Proceedings of Petersburg Transport University*. 3. pp. 290-299. (In Russian).
25. Sapozhnikov, V.V., Sapozhnikov, V.I., Efanov, D.V. & Cherepanova, M.R. (2016) Modulo Codes with Summation in Concurrent Error Detection Systems. I. Ability to Detect Errors by Modulo Codes in Data Vectors. *Elektronnoe modelirovanie – Electronic Modeling*. 38(2). pp. 27-48. (In Russian).
26. Busaba, F.Y. & Lala, P.K. (1994) Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors. *Journal of Electronic Testing: Theory and Applications*. 5. pp. 19-28. DOI: 10.1007/BF00971960
27. Goessel, M., Morozov, A.A., Sapozhnikov, V.V. & Sapozhnikov, V.I. (1994) Postroenie kombinatsionnykh samoproveryaemykh ustroystv s monotonno nezavisimymi vykhodami [Construction of Combinational Self-Checking Devices with Monotonically Independent Outputs]. *Avtomatika i telemekhanika – Automation and Remote Control*. issue 7. pp. 148–160.
28. Goessel, M., Morozov, A.A., Sapozhnikov, V.V. & Sapozhnikov, V.I. (1997) Issledovanie kombinatsionnykh samoproveryaemykh ustroystv s nezavisimymi i monotonno nezavisimymi vykhodami [Investigation of Combination Self-Testing Devices Having Independent and Monotone Independent Outputs]. *Avtomatika i telemekhanika – Automation and Remote Control*. 2. pp. 180-193.
29. Morosov, A., Sapozhnikov, V.V., Sapozhnikov, V.I. & Goessel, M. (1998) Self-Checking Combinational Circuits with Unidirectionally Independent Outputs. *VLSI Design*. 5(4). pp. 333-345.
30. Sapozhnikov, V.V., Morosov, A., Sapozhnikov, V.I. & Goessel, M. (1998) A New Design Method for Self-Checking Unidirectional Combinational Circuits. *Journal of Electronic Testing: Theory and Applications*. 12(1–2). pp. 41-53. DOI: 10.1007/978-1-4757-6069-9_4
31. Matrosova, A., Levin, I. & Ostanin, S. (2001) Survivable Self-Checking Sequential Circuits. *Proceedings of 2001 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT 2001)*. October 24-26, San Francisco, CA. pp. 395-402.
32. Blyudov, A.A., Efanov, D.V., Sapozhnikov, V.V. & Sapozhnikov, V.I. (2012) Formation of the Berger Modified Code with Minimum Number of Undetectable Errors of Informational Bits. *Elektronnoe modelirovanie – Electronic Modeling*. 34(6). pp. 17-29. (In Russian).
33. Efanov, D., Sapozhnikov, V., Sapozhnikov, V.I. & Nikitin, D. (2015) Sum Code Formation with Minimum Total Number of Undetectable Errors in Data Vectors. *Proceedings of 13th IEEE East-West Design & Test Symposium (EWDTS'2015)*. Batumi, Georgia, September 26-29. pp. 141-148.

34. Das, D. & Touba, N.A. (1999) Weight-Based Codes and Their Application to Concurrent Error Detection of Multilevel Circuits. *Proceedings of 17th IEEE Test Symposium*. USA, California. pp. 370-376. DOI: 10.1109/43.644041
35. Mekhov, V.B., Sapozhnikov, V.V. & Sapozhnikov, VI.V. (2008) Checking of Combinational Circuits Basing on Modification Sum Codes. *Automation and Remote Control*. 8. pp. 153-165. DOI: 10.1134/S0005117908080134
36. Sapozhnikov, V.V., Sapozhnikov, VI.V. & Efanov, D.V. (2014) Weighted Codes with Summation for Organization of the Check of Logic Units. *Elektronnoe modelirovanie – Electronic Modeling*. 36(1). pp. 59-80. (In Russian).
37. Marouf, M.A. & Friedman, A.D. (1978) Design of Self-Checking Checkers for Berger Codes. *Proceedings of 8th Annual International Conference on Fault-Tolerant Computing*. Toulouse, France. pp. 179-183.
38. Lala, P.K. (2007) *Principles of Modern Digital Design*. New-Jersey: John Wiley & Sons.
39. Blyudov, A.A. (2013) On the Synthesis of Test Equipment for Modulo Codes with Summation. *Proceedings of Petersburg Transport University*. 1. pp. 53-58. (In Russian)
40. Drozd, A.V., Kharchenko, V.S., Antoshchuk, S.G., Drozd, Ju.V., Drozd, M.A. & Sulima, Yu.Yu. (2012) *Objects and Methods of On-Line Testing for Safe Instrumentation and Control Systems*. Kharkov: National Aerospace University. (In Russian)
41. Favalli, M. & Metra, C. (2001) Optimization of Error Detecting Codes for the Detection of Crosstalk Originated Errors. *Design, Automation and Test in Europe (DATE)*. March 13-16. pp. 290-296. DOI: 10.1109/DATE.2001.915039
42. Ghosh, S., Lai, K.W., Jone, W.B. & Chang, S.C. (2004) Scan Chain Fault Identification Using Weight-Based Codes for SoC Circuits. *Proceedings of 13th Asian Test Symposium*. November 15-17. pp. 210-215. DOI: 10.1109/ATS.2004.77
43. Sapozhnikov, V., Sapozhnikov, VI., Efanov, D. & Nikitin, D. (2014) Combinational Circuits Checking on the Base of Sum Codes with One Weighted Data Bit. *Proceedings of 12th IEEE East-West Design & Test Symposium (EWDTS'2014)*. Kyev, Ukraine. September 26-29. pp. 126-136. DOI: 10.1109/EWDTS.2014.7027064.
44. Sapozhnikov, V.V., Sapozhnikov, VI.V., Efanov, D.V. & Nikitin, D.A. (2015) Modulo Weighted Code with Summation for Technical Diagnostics Systems. *Informatika i sistemy upravleniya – Information Science and Control Systems*. 3. pp. 53-62.
45. Sapozhnikov, V. & Sapozhnikov, VI. (1999) New Code for Fault Detection in Logic Circuits. *Proceedings of 4th International conference on Unconventional Electromechanical and Electrical Systems*. St. Petersburg, Russia. June 21-24. pp. 693-696. (In Russian).
46. Sapozhnikov, V.V., Sapozhnikov, VI.V., Efanov, D.V. & Dmitriev, V.V. (2014) Properties of Codes with Summation of Transitions Between Data Bits With Direct Sequence of Weighting Coefficient. *Information Science and Control Systems*. issue 4. pp. 77–88.
47. Sapozhnikov, V., Sapozhnikov, VI., Efanov, D., Dmitriev, V. & Cherepanova, M. (2015) Optimum Sum Codes, That Effectively Detect the Errors of Low Multiplicities. *RadioElectronics & Informatics*. 1. pp. 17-22. (In Russian).
48. Dmitriev, V.V. (2015) On the Two Weighing Methods and Their Influence on Properties of Sum Codes of Weighted Transitions in Functional Logical Scheme Control Systems. *Izvestiya Peterburgskogo universiteta putey soobshcheniya – Proceedings of Petersburg Transport University*. 3. pp. 119-129. (In Russian).
49. Sapozhnikov, V.V., Sapozhnikov, VI.V., Efanov, D.V., Dmitriev, V.V. & Cherepanova, M.R. (2016) Method of Formation of Sum Code with Improved Efficiency of Error Detection in Data Vectors. *Avtomatika na transporte – Automation on Transport*. 2(1). pp. 95-123. (In Russian).