



УДК 681.518.5:004.052.32

© 2015 г. **В.В. Сапожников**, д-р техн. наук,

Вл.В. Сапожников, д-р техн. наук,

Д.В. Ефанов, канд. техн. наук,

Д.А. Никитин

(Петербургский государственный университет путей сообщения
Императора Александра I)

МОДУЛЬНО ВЗВЕШЕННЫЙ КОД С СУММИРОВАНИЕМ ДЛЯ СИСТЕМ ТЕХНИЧЕСКОГО ДИАГНОСТИРОВАНИЯ

Авторами описывается способ построения кода с суммированием с минимальным общим количеством обнаруживаемых ошибок для фиксированных параметров длин информационных и контрольных векторов. Данный код относится к классу модульно взвешенных кодов с суммированием. В работе проводится сравнение рассматриваемого кода с классическим кодом Бергера. Даются результаты экспериментов с системой контрольных примеров LGSynth`89, используемой для оценки эффективности новых способов построения надежных дискретных систем.

Ключевые слова: система функционального контроля, код Бергера, модульно взвешенный код с суммированием, оптимальный код, информационные разряды, обнаруживаемая ошибка, тестирование комбинационных схем.

Введение

Для организации функционального контроля комбинационного логического устройства $f(x)$ используется специальное контрольное оборудование: блок контрольной логики $g(x)$ и тестер, состоящий из генератора контрольных разрядов G и компаратора TRC [1]. В процессе работы блок $f(x)$ вычисляет значения рабочих функций $f_1(x), f_2(x), \dots, f_m(x)$, а блок $g(x)$ – значения контрольных функций $g_1(x), g_2(x), \dots, g_k(x)$. Таким образом, на выходах блока $f(x)$ формируется некоторый информационный вектор длины m , а на выходах блока $g(x)$ – соответствующий ему контрольный вектор длины k . Соответствие между информационным и контрольным векторами устанавливается на этапе проектирования системы функционального контроля и в процессе ее работы контролируется тестером. В нем генератор G по значениям рабочих функций вычисляет контрольные функции $g^*_1(x), g^*_2(x), \dots, g^*_k(x)$, а компаратор сравнивает их с одноименными функциями, вычисленными блоком $g(x)$. Значения функций, вычисляемые блоком $g(x)$, предварительно инвертируются, что позволяет свести контроль правильности функционирования

схемы $f(x)$ к контролю парафазного сигнала на выходе компаратора. Сам компаратор строится на основе каскадного соединения модулей сжатия парафазных сигналов *TRC* (*two-rail checker*) [2].

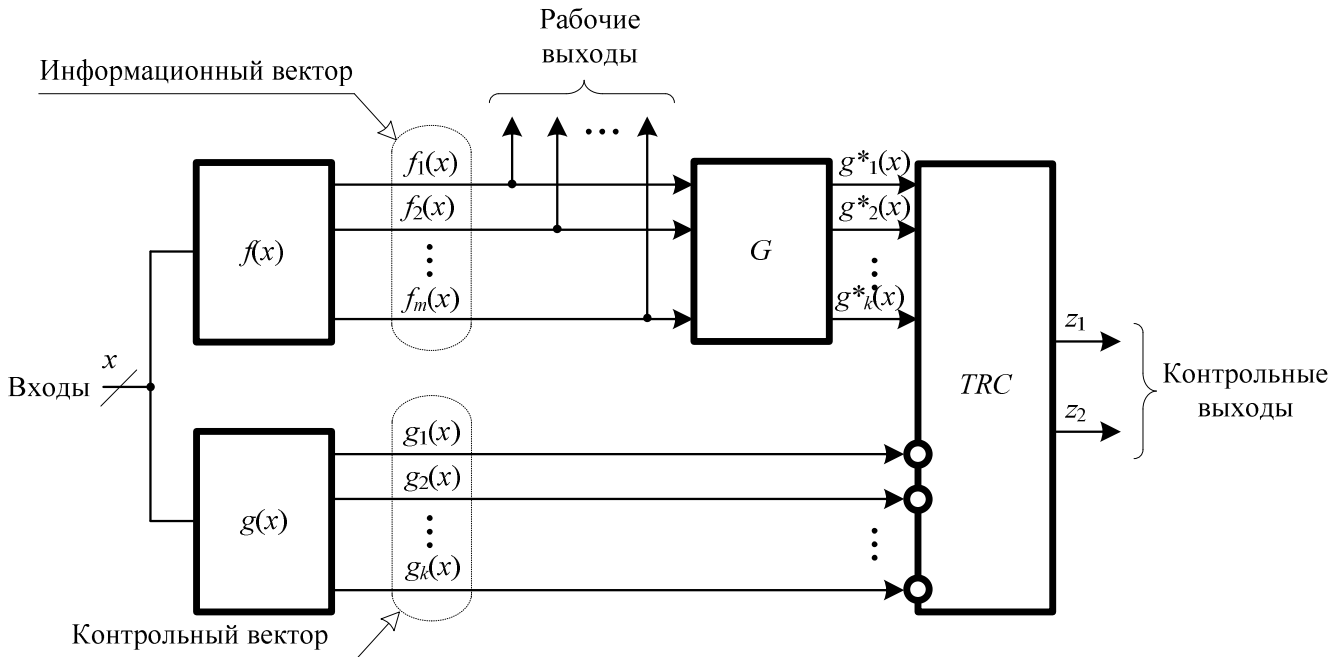


Рис. 1. Структура системы функционального контроля.

Такая организация системы функционального контроля (рис. 1) позволяет определять техническое состояние устройства $f(x)$ в его рабочем режиме. Возникающие в это время одиночные неисправности могут приводить к искажениям значений функций $f_1(x)$, $f_2(x)$, ..., $f_m(x)$. Учитывая тот факт, что в произвольный момент времени может произойти искажение только в одном из блоков системы функционального контроля, искажение рабочих функций можно обнаружить и отключить рабочие выходы системы от объектов управления.

Часто система функционального контроля строится на основе кода с суммированием, или (m,k) -кода [3 – 9]. Неисправности в блоке $f(x)$ искажают значения разрядов информационного вектора и при нарушении соответствия с разрядами контрольного вектора будут обнаружены тестером. Часть искажений, в зависимости от положенного в основу системы функционального контроля (m,k) -кода, может быть не обнаружена. Изучая свойства обнаружения ошибок в информационных векторах (m,k) -кодов, можно изучать и свойства самой структуры по обнаружению ошибок в контролируемом устройстве [10].

В данной работе описывается метод построения кода с суммированием, который обнаруживает больше ошибок в информационных векторах, чем классический код с суммированием (код Бергера [11]) в целом, а также в области двукратных ошибок. Приводятся результаты исследований характеристик систем функционального контроля, построенных на основе нового кода, для набора контрольных примеров LGSynth`89 [12], часто используемого при оценке эффективности вновь разрабатываемых подходов к техническому диагностированию логических устройств автоматики и вычислительной техники [13 – 17].

Классические и модульно взвешенные коды с суммированием

Классические коды с суммированием (коды Бергера или $S(m,k)$ -коды) часто используются при организации функционального контроля логических устройств, так как они обнаруживают 100% монотонных ошибок в информационных векторах [3, 5]. Контрольные векторы кодов Бергера образуются путем записи в контрольные разряды двоичного числа, равного сумме единичных информационных разрядов (равного весу r информационного вектора). Число контрольных разрядов в $S(m,k)$ -кодах определяется величиной $k = \lceil \log_2(m+1) \rceil$ (запись $\lceil \dots \rceil$ обозначает целое сверху от вычисляемого значения). Классические коды с суммированием не обнаруживают те ошибки в информационных векторах, при которых не нарушается вес вектора.

Количество таких ошибок определяется по формуле [10]:

$$N_m = \sum_{d=2}^{m(m-1)} N_{m,d} = \sum_{d=2}^{m(m-1)} 2^{m-d} C_m^d C_d^{\frac{d}{2}}, \quad (1)$$

где $N_{m,d}$ – число необнаруживаемых ошибок кратностью d ; N_m – общее число необнаруживаемых ошибок.

Кодом Бергера не фиксируется любая разнонаправленная ошибка с четной кратностью в информационном векторе, содержащая группу искажений $\{0 \rightarrow 1, 1 \rightarrow 0\}$. Для данного значения d вне зависимости от длины информационного вектора доля необнаруживаемых ошибок с данной кратностью от общего числа ошибок с этой же кратностью является постоянной величиной [10]:

$$\beta_d = 2^{-d} C_d^{\frac{d}{2}}. \quad (2)$$

Из (2) следует, что $S(m,k)$ -коды не обнаруживают большое количество ошибок с малой кратностью: 50% двукратных, 37,5% четырехкратных, 31,25% шестикратных ошибок и т.д. Это является существенным недостатком $S(m,k)$ -кодов. Более того, правила построения $S(m,k)$ -кодов таковы, что дают неравномерное распределение информационных векторов между контрольными векторами. В [18] показано, что $S(m,k)$ -коды не являются оптимальными с позиции использования контрольных разрядов. Возможно построение таких кодов, которые имеют равномерное распределение всех 2^m информационных векторов между 2^k контрольными векторами. В таких *оптимальных кодах* не обнаруживается минимальное количество ошибок в информационных векторах:

$$N_{m,k}^{\min} = 2^m (2^{m-k} - 1). \quad (3)$$

Например, $S(6,3)$ -код не обнаруживает 860 ошибок в информационных векторах, а оптимальный код – 448, что составляет 58,111% от числа необнаруживаемых ошибок кодом Бергера [19, 20].

Учитывая (3), можно оценивать эффективность каждого делимого кода следующим коэффициентом:

$$\xi_{m,k} = \frac{N_{m,k}^{\min}}{N_m}. \quad (4)$$

Чем ближе $\xi_{m,k}$ к 1, тем эффективнее делимый код. Далее коэффициент эффективности будем выражать в процентах.

В [21] показано, что коды с суммированием единичных информационных разрядов, включая и код Бергера, не являются оптимальными.

Рассмотрим следующий способ построения оптимального кода с суммированием.

1. Информационным разрядам приписываются весовые коэффициенты в виде последовательности натуральных чисел, начиная с младшего разряда.

2. Устанавливается модуль $M = 2^{\lceil \log_2(m+1) \rceil}$ (это модуль кода Бергера).

3. Подсчитывается суммарный вес единичных информационных разрядов V .

4. Определяется наименьший неотрицательный вычет числа V по установленному модулю:

$$W = (V) \bmod M. \quad (5)$$

5. Полученное число представляется в двоичном виде и записывается в контрольные разряды.

Модифицированный код с суммированием, полученный по приведенному алгоритму, обозначим как $WS(m,k)$ -код. Продемонстрируем работу алгоритма на примере получения контрольного вектора для информационного вектора $\langle 010011 \rangle$. Длина информационного вектора $m = 6$. Отсюда следует, что $M = 2^{\lceil \log_2(6+1) \rceil} = 2^3 = 8$. Число $V = 1 + 2 + 5 = 8$. Наименьший неотрицательный вычет числа V равен $W = (8) \bmod 8 = 0$. Таким образом, в контрольный вектор следует записать число $\langle 000 \rangle$.

$WS(m,k)$ -коды имеют меньшее количество необнаруживаемых ошибок, чем $S(m,k)$ -коды. Отметим, что число контрольных разрядов остается тем же. Уменьшение ошибок достигается за счет перераспределения информационных векторов между контрольными векторами. При этом в классе необнаруживаемых появляются ошибки нечетных кратностей и монотонные ошибки, которые обнаруживаются кодами Бергера.

Для определения характеристик обнаружения ошибок $WS(m,k)$ -кодами было разработано специальное программное обеспечение, с применением которого можно установить некоторые закономерности в свойствах $WS(m,k)$ -кодов. В табл. 1 приведены распределения необнаруживаемых ошибок по кратностям в информационных векторах $WS(m,k)$ -кодов при длинах информационных векторов $m = 3 \div 12$.

Таблица 1

m	k	Количество необнаруживаемых ошибок кратностью d												N_m	$\xi_{m,k}$
		1	2	3	4	5	6	7	8	9	10	11	12		
3	2	0	4	4										8	100
4	3	0	0	12	4									16	100
5	3	0	16	56	24	0								96	100
6	3	0	64	192	144	48	0							448	100
7	3	0	192	704	624	272	96	32						1920	100

8	4	0	0	1088	1408	816	360	136	32					3840	100
9	4	0	256	3200	4608	3680	2512	1296	320	0				15872	100
10	4	0	1024	8704	14848	15680	12704	7712	3200	640	0			64512	100
11	4	0	3072	23552	45824	57344	56960	43424	21984	6416	1264	256		260096	100
12	4	0	8192	61440	136192	199936	229120	203456	131008	56224	15328	3072	512	1044480	100

Из анализа табл. 1 следуют такие особенности $WS(m,k)$ -кодов по обнаружению ошибок в информационных векторах:

1. $WS(m,k)$ -коды являются помехоустойчивыми (обнаруживают все одиночные ошибки в информационных векторах).

2. Любой $WS(m,k)$ -код является оптимальным ($\xi_{m,k}=100\%$).

3. $WS(m,k)$ -коды с длинами информационных векторов $m = 2^t$ ($t = 1, 2, 3, \dots$) обнаруживают 100% двукратных ошибок в информационных векторах.

4. $WS(m,k)$ -коды в отличие от кодов Бергера имеют необнаруживаемые ошибки с нечетной кратностью и монотонные ошибки.

5. $WS(m,k)$ -коды с длинами информационных векторов $m = 2^t$ и $m = 2^t - 1$ ($t = 2, 3, 4, \dots$) имеют одинаковое количество необнаруживаемых ошибок с максимальной кратностью $d=m$.

6. $WS(m,k)$ -коды с длинами информационных векторов $m = 2$, $m = 1 + 4i$, $m = 2 + 4i$ ($i = 1, 2, 3, \dots$) обнаруживают 100% ошибок с максимальной кратностью $d = m$.

Объясним, за счет чего $WS(m,k)$ -коды обнаруживают все однократные ошибки в информационных векторах. Весовые коэффициенты информационных разрядов $\{1, 2, \dots, m\}$ подобраны таким образом, что каждый из них при определении вычета по модулю $M = 2^{\lceil \log_2(m+1) \rceil}$ никогда не будет равен 0.

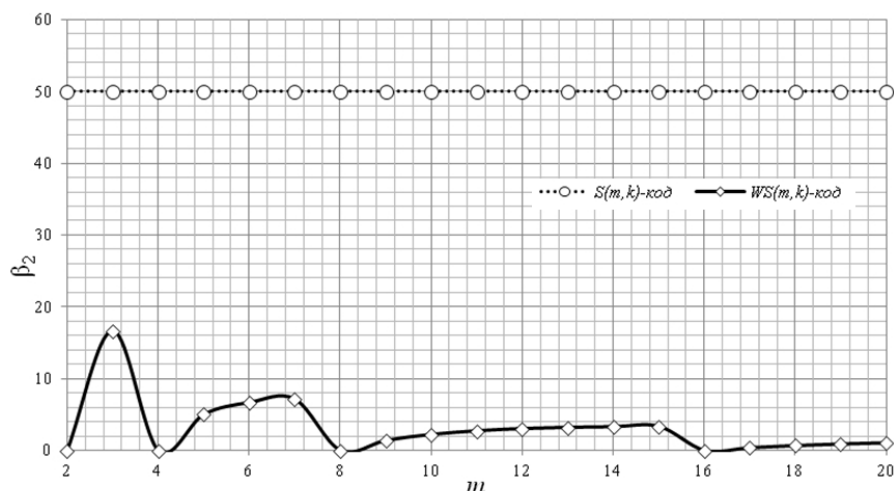
Действительно, значение веса самого старшего разряда всегда меньше значения модуля ($m < M$). Поэтому при подсчете разрядов контрольного вектора любой разряд информационного вектора будет учтен, и два информационных вектора с кодовым расстоянием $d = 1$ не будут иметь одного и того же контрольного вектора.

В табл. 2 приводятся характеристики рассматриваемых кодов с суммированием по обнаружению двукратных искажений в информационных векторах. Свойства кодов по обнаружению ошибок в информационных векторах можно оценить долей необнаруживаемых ошибок кратности d от общего количества ошибок той же кратностью – величиной β_d . В табл. 2 приведены значения величин β_2 для $S(m,k)$ и $WS(m,k)$ кодов. $S(m,k)$ -коды, вне зависимости от длины информационного вектора, не обнаруживают 50% двукратных ошибок в информационных векторах, что для некоторых значений m в десятки раз больше числа необнаруживаемых $WS(m,k)$ -кодами двукратных ошибок (рис. 2).

В столбце E_2 для каждого значения m указано, во сколько раз $WS(m,k)$ -код эффективнее $S(m,k)$ -кода с точки зрения обнаружения двукратных ошибок в информационных векторах.

Таблица 2

m	k	Количество необнаруживаемых ошибок кратности $d=2$		Общее количество двукратных ошибок, N_2	$\beta_2, \%$		E_2
		$S(m,k)$ -код	$WS(m,k)$ -код		$S(m,k)$ -код	$WS(m,k)$ -код	
2	2	2	0	4	50	0	–
3	2	12	4	24	50	16,667	3
4	3	48	0	96	50	0	–
5	3	160	16	320	50	5	10
6	3	480	64	960	50	6,667	7,5
7	3	1344	192	2688	50	7,143	7
8	4	3584	0	7168	50	0	–
9	4	9216	256	18432	50	1,389	36
10	4	23040	1024	46080	50	2,222	22,5
11	4	56320	3072	112640	50	2,727	18,333
12	4	135168	8192	270336	50	3,03	16,5
13	4	319488	20480	638976	50	3,205	15,6
14	4	745472	49152	1490944	50	3,297	15,167
15	4	1720320	114688	3440640	50	3,333	15
16	5	3932160	0	7864320	50	0	–
17	5	8912896	65536	17825792	50	0,368	136
18	5	20054016	262144	40108032	50	0,654	76,5
19	5	44826624	786432	89653248	50	0,877	57
20	5	99614720	2097152	199229440	50	1,053	47,5

Рис. 2. Зависимость величины β_2 от m для кодов с суммированием.

Общее количество необнаруживаемых ошибок в $WS(m,k)$ -кодах минимально для заданных m и k . По данному показателю $WS(m,k)$ -коды также превосходят классические коды Бергера. На рис. 3 даны графики зависимостей отношений числа необнаруживаемых ошибок к общему числу ошибок от значения длин информационных векторов для $S(m,k)$ и $WS(m,k)$ кодов (величина $\gamma_m, \%$). Отметим схожий вид графиков величин β_2 и γ_m для соответствующих кодов с суммированием.

Значение γ_m для данной длины информационного вектора равно:

$$\gamma_m = \frac{N_m}{N} = \frac{N_m}{2^m(2^m - 1)}. \quad (6)$$

Используя (3), (6), получим выражение для расчета γ_m в $WS(m,k)$ -кодах:

$$\gamma_m = \frac{N_m}{N} = \frac{2^m(2^{m-k} - 1)}{2^m(2^m - 1)} = \frac{2^{m-k} - 1}{2^m - 1}. \quad (7)$$

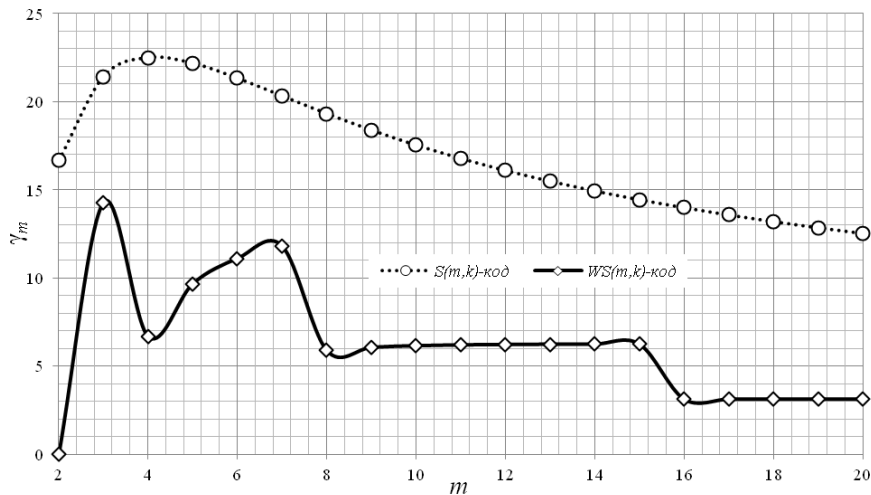


Рис. 3. Зависимость величины γ_m от m для кодов с суммированием.

С увеличением длины информационного вектора значение γ_m уменьшается и при $m \rightarrow \infty$ становится равным:

$$\lim_{m \rightarrow \infty} \gamma_m = \lim_{m \rightarrow \infty} \frac{2^{m-k} - 1}{2^m - 1} = \lim_{m \rightarrow \infty} \frac{\frac{2^{m-k}}{2^m} - \frac{1}{2^m}}{\frac{2^m}{2^m} - \frac{1}{2^m}} = \lim_{m \rightarrow \infty} \frac{2^{-k} - \frac{1}{2^m}}{1 - \frac{1}{2^m}} = 2^{-k}. \quad (8)$$

Из выражения (8) следует, что при больших значениях длин информационных векторов значение величины γ_m в $WS(m,k)$ -кодах зависит только от числа контрольных разрядов.

Практические результаты

Для оценки эффективности применения $WS(m,k)$ -кода при организации систем функционального контроля был поставлен ряд экспериментов со схемами, входящими в набор контрольных примеров LGSynth`89 [12]. Выбраны контрольные примеры в формате списка в базисе «ИЛИ-НЕ» (формат *.netblif). В данном формате логической схемы описываются все входящие в нее логические элементы, а также соединения их входов и выходов между собой и входами и выходами самой схемы.

Эксперимент заключался в моделировании работы схемы при внесении в нее последовательно всех одиночных неисправностей логических элементов на каждом наборе входных переменных. При этом фиксировались те ошибки на выходах схемы, которые не обнаруживаются при организации контроля на основе кодов $S(m,k)$ и $WS(m,k)$.

В табл. 3 приведены результаты экспериментов с 15 контрольными схемами с различным числом входов и выходов (I/O). В графе N указано количество ошибок на выходах схемы контрольного примера при моделировании всех одиночных неисправностей в ее структуре на всех входных наборах, а в графах N_m – количество ошибок, не обнаруженных соответствующим кодом. Столбец ΔN_m характеризует эффективность $WS(m,k)$ -кода в сравнении с кодом Бергера (отрицательное число говорит о неэффективности кода). В последнем столбце таблицы

приводится значение величины E_m , показывающей, во сколько раз $WS(m,k)$ -код эффективнее $S(m,k)$ -кода при организации системы функционального контроля.

Таблица 3

№	Контрольный пример	I/O	M	N	N_m		ΔN_m	$\gamma_m, \%$		E_m
					$S(m,k)$	$WS(m,k)$		$S(m,k)$	$WS(m,k)$	
1	alu2	10/6	8	61988	7672	805	6867	12,377	1,299	9,53
2	alu4	14/8	16	1966881	186142	54826	131316	9,464	2,787	3,395
3	cc	21/20	32	35167192	1870896	137864	1733032	5,32	0,392	13,571
4	cm42a	4/10	16	278	8	0	8	2,878	0	–
5	cm162a	14/5	8	317331	1920	6583	-4663	0,605	2,074	0,292
6	cm163a	16/5	8	1221312	10368	22400	-12032	0,849	1,834	0,463
7	cmb	16/4	8	288218	6	0	6	0,002	0	–
8	cu	14/11	16	137984	34048	22368	11680	24,675	16,211	1,522
9	f51m	8/8	16	13008	91	0	91	0,7	0	–
10	ldd	9/19	32	28198	0	96	-96	0	0,34	0
11	plce	19/9	16	17472087	188416	112481	75935	1,078	0,644	1,675
12	pm1	14/13	16	189440	6848	0	6848	3,615	0	–
13	sct	14/15	16	448510	4434	982	3452	0,989	0,219	4,515
14	x2	10/7	8	19708	104	518	-414	0,528	2,628	0,201
15	z4ml	7/4	8	4168	128	0	128	3,071	0	–

Для некоторых контрольных примеров применение при организации системы функционального контроля $WS(m,k)$ -кода, а не $S(m,k)$ -кода, имеющего такое же количество контрольных разрядов, более эффективно. В табл. 3 выделены данные по 5 контрольным примерам (cm42a, cmb, f51m, pm1 и z4ml), для которых применение $WS(m,k)$ -кода при организации системы функционального контроля обеспечивает обнаружение всех одиночных неисправностей на выходах внутренних логических элементов. Это свидетельствует о том, что применение $WS(m,k)$ -кода может оказаться даже эффективнее системы дублирования схем (при решении задачи обнаружения одиночных неисправностей)! Для контроля схем по кодам Бергера, как показано в [3, 22, 23], возможна реконфигурация структуры схемы в контролепригодную структуру, но это требует внесения избыточности, что усложняет систему функционального контроля в целом.

Для контрольных примеров, в которых $WS(m,k)$ -код обнаруживает 100% одиночных неисправностей, было проведено сравнение сложности технической реализации структуры, приведенной на рис. 1, с аналогичным показателем системы функционального контроля по методу дублирования. Для этого с использованием известного интерпретатора SIS [13] была определена площадь в условных единицах в библиотеке nor.genlib, содержащей элементы «ИЛИ-НЕ», всех составляющих структуры, приведенной на рис. 1, и структуры системы дублирования. При этом до определения параметров площади систем функционального контроля была проведена оптимизация схем для получения более простых структур блоков $g(x)$ и G [24, 25]. В табл. 4 дается сравнение обеих систем функционального контроля. В последнем столбце, обозначенном как ϵ , приводится «выигрыш» в сложности технической реализации системы функционального контроля, построенной по $WS(m,k)$ -коду, в сравнении с системой дублирования. Для всех кон-

трольных примеров, кроме pm1, показатель $\varepsilon > 1$, что говорит о более высокой эффективности $WS(m,k)$ -кода при построении системы со 100% обнаружением одиночных неисправностей во внутренней структуре контролируемых схем по сравнению с дублированием.

Таблица 4

Контрольный пример	m	Система функционального контроля по $WS(m,k)$ -коду	Система функционального контроля по методу дублирования	ε
cm42a	10	2231	2854	1,279
cmb	4	856	1132	1,322
f51m	8	2594	2698	1,04
pm1	13	7509	3878	0,516
z4ml	4	1070	1188	1,11

Заключение

В работе предлагается перспективный для решения задач технической диагностики $WS(m,k)$ -код, основанный на принципе взвешивания информационных разрядов. Новый код является оптимальным по критерию минимума общего количества необнаруживаемых ошибок в информационных векторах при длине контрольного вектора $k = \lceil \log_2(m+1) \rceil$, кроме того, рассматриваемый код эффективнее обнаруживает ошибки малых кратностей, чем классический код Бергера. При $m = 2^t$ ($t=1, 2, 3, \dots$) $WS(m,k)$ -код обнаруживает 100% двукратных ошибок в информационных векторах.

Применение для организации систем функционального контроля модульно взвешенных кодов с суммированием, полученных по приведенным в статье правилам, на практике позволяет существенно повысить эффективность обнаружения ошибок в контролируемых устройствах. В некоторых случаях удается достичь свойств 100% обнаружения одиночных неисправностей без модификации структуры контролируемого логического устройства.

ЛИТЕРАТУРА

1. *Touba N.A., McCluskey E.J.* Logic Synthesis of Multilevel Circuits with Concurrent Error Detection // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. –1997. –Vol. – P.783-789.
2. *Jha N.K.* Totally Self-Checking Checker Designs for Bose-Lin, Bose and Blaum Codes // IEEE Trans. Computer-Aided Design. – 1991. – Vol. CAD-10. – P.136-143.
3. *Согомонян Е.С., Слабаков Е.В.* Самопроверяемые устройства и отказоустойчивые системы. – М.: Радио и связь, 1989.
4. *Rao T.R., Fujiwara E.* Error Control Coding for Computer Systems. – NY.: Prentice Hall, 1989.
5. *Fujiwara E.* Code Design for Dependable Systems: Theory and Practical Applications. – John Wiley & Sons, 2006.
6. *Das D., Touba N.A.* Weight-Based Codes and Their Application to Concurrent Error Detection of Multilevel Circuits // Proc. 17th IEEE Test Symposium, USA, California, 1999. – P. 370-376.
7. *Das D., Touba 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. – 2000. – P. 171-176.
8. *Ghosh S., Lai K.W., Jone W.B., Chang S.C.* Scan Chain Fault Identification Using Weight-Based

- Codes for SOC Circuits // Proceedings of the 13th Asian Test Symposium. – 2004. – P.210-215.
9. *Matrosova A., Levin I., Ostanin S.* 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. – P.395-402.
 10. *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.
 11. *Berger J.M.* A Note on Error Detecting Codes for Asymmetric Channels // Information and Control. – 1961. – V.1. 4, №1. – P.68-73.
 12. *Benchmarks: LGSynth89* [Режим доступа: <http://www.cbl.ncsu.edu:16080/benchmarks/LGSynth89/mlexamples/>].
 13. *Sentovich E.M., Singh K.J., Lavagno L., Moon C., Murgai R., Saldanha A., Savoj H., Stephan P.R., Brayton R.K., Sangiovanni-Vincentelli A.* SIS: A System for Sequential Circuit Synthesis // Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1992.
 14. *Kunz W., Menon P.R.* Multi-Level Logic Optimization by Implication Analysis // Proceedings of IEEE/ACM International Conference on Computer Aided Design (ICCAD'94). – San Jose, CA, November. – 1994. – P.6-13.
 15. *Gopalakrishnan P., Rutenbar R.A.* Direct Transistor-Level Layout for Digital Blocks. – Boston: Kluwer Academic Publishers, 2004.
 16. *Moniot R.K.* SIS Tutorial: Combinational Logic Circuit Design, 2011.
 17. *Collection of Digital Design Benchmarks* [Режим доступа: <http://ddd.fit.cvut.cz/prj/Benchmarks/>].
 18. *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.
 19. *Blyudov A., Efanov D., Sapozhnikov V., Sapozhnikov Vl.* Properties of Code with Summation for Logical Circuit Test Organization // Proceedings of 10th IEEE East-West Design & Test Symposium (EWDTS'2012). – Kharkov, Ukraine. – 2012. – P.114-117.
 20. *Efanov D., Sapozhnikov V., Sapozhnikov Vl., Blyudov A.* On the Problem of Selection of Code with Summation for Combinational Circuit Test Organization // Proceedings of 11th IEEE East-West Design & Test Symposium (EWDTS'2013). – Rostov-on-Don, Russia. – 2013. – P.261-266.
 21. *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.
 22. *Morosow A., Sapozhnikov V.V., Sapozhnikov Vl.V., Goessel M.* Self-Checking Combinational Circuits with Unidirectionally Independent Outputs // VLSI Design. – 1998. – Vol. 5, Issue 4. – P. 333-345.
 23. *Matrosova A.Yu., Levin I., Ostanin S.A.* Self-Checking Synchronous FSM Network Design with Low Overhead // VLSI Design. – 2000. – Vol. 11, Issue 1. – P. 47-58.
 24. *Damiani M., Yang J.C.Y., De Micheli G.* Optimization of Combinational Logic Circuits Based on Compatible Gates // Proceedings of DAC'93. – 1993. – P.631-636.
 25. *Detjens E., Gannot G., Rudell R., Sangiovanni-Vincentelli A.L., Wang A.* Technology Mapping in MIS // Proceedings of ICCAD. – 1987. – P.116-119.

Статья представлена к публикации членом редколлегии С.В. Шалобановым.

E-mail:

Сапожников Валерий Владимирович – port.at.pgups1@gmail.com;

Сапожников Владимир Владимирович – port.at.pgups1@gmail.com;

Ефанов Дмитрий Викторович – TrES-4b@yandex.ru;

Никитин Дмитрий Александрович – nikitin239@gmail.com.