УДК 004.052.32+681.518.5

Д. В. Ефанов
Петербургский государственный университет путей сообщения
Императора Александра I

СИНТЕЗ ГЕНЕРАТОРОВ ТЕСТЕРОВ МОДИФИЦИРОВАННЫХ КОДОВ БЕРГЕРА НА ОСНОВЕ СВОЙСТВ ЛИНЕЙНЫХ И ПРОСТЫХ СИММЕТРИЧНЫХ ФУНКЦИЙ

Приводится способ построения генераторов тестеров модифицированных кодов Бергера, основанный на использовании свойств простых симметричных и линейных функций алгебры логики. Данный способ позволяет в некоторых случаях строить более простые генераторы модифицированных кодов Бергера, чем генераторы классических кодов с суммированием, при сохранении характеристик быстродействия и контролепригодности. Более того, приведенный способ дает возможность построения генераторов тестеров модифицированных кодов Бергера с улучшенной характеристикой быстродействия по сравнению со способами, применявшимися ранее.

система функционального контроля, код с суммированием, код Бергера, модифицированный код Бергера, тестер, генератор, простая симметричная функция, линейная функция.

Введение

При построении надежных дискретных систем автоматики и вычислительной техники большое значение имеет возможность обеспечения свойства их контролепригодности [1]. Контролепригодность дискретной системы заключается в ее приспособленности к проведению контроля заданными средствами технического диагностирования. В свою очередь, характеристика контролепригодности тесно связана с возможностью обнаружения отказов в элементах структуры диагностируемого логического устройства. Невозможность обнаружения отказа может привести к неверному выполнению функций дискретной системой.

Существует два подхода к организации контроля дискретных систем – тестовое и функциональное диагностирование [2, 3]. В первом случае требуются ресурсы времени для подачи специальных проверочных воздействий на входы контролируемого объекта и для фиксации выходных значений [4, 5], во втором техническое состояние объекта определяется в его рабочем режиме [6, 7].

Данная статья посвящена вопросам развития теории синтеза самопроверяемых контрольных устройств (тестеров, детекторов и пр.) для систем функционального контроля, построенных с использованием свойств кодов с суммированием единичных информационных разрядов.

1 Системы функционального контроля логических устройств

При организации систем функционального контроля логических устройств автоматики и вычислительной техники (рис. 1) часто используются коды с суммированием [8–10]. Выходам блока основной логики $f(x)$ ставится в соответствие информационный вектор длины m, а выходам блока контрольной логики $g(x)$ – контрольный вектор длины k. На этапе проектирования системы функционального контроля устанавливается однозначное

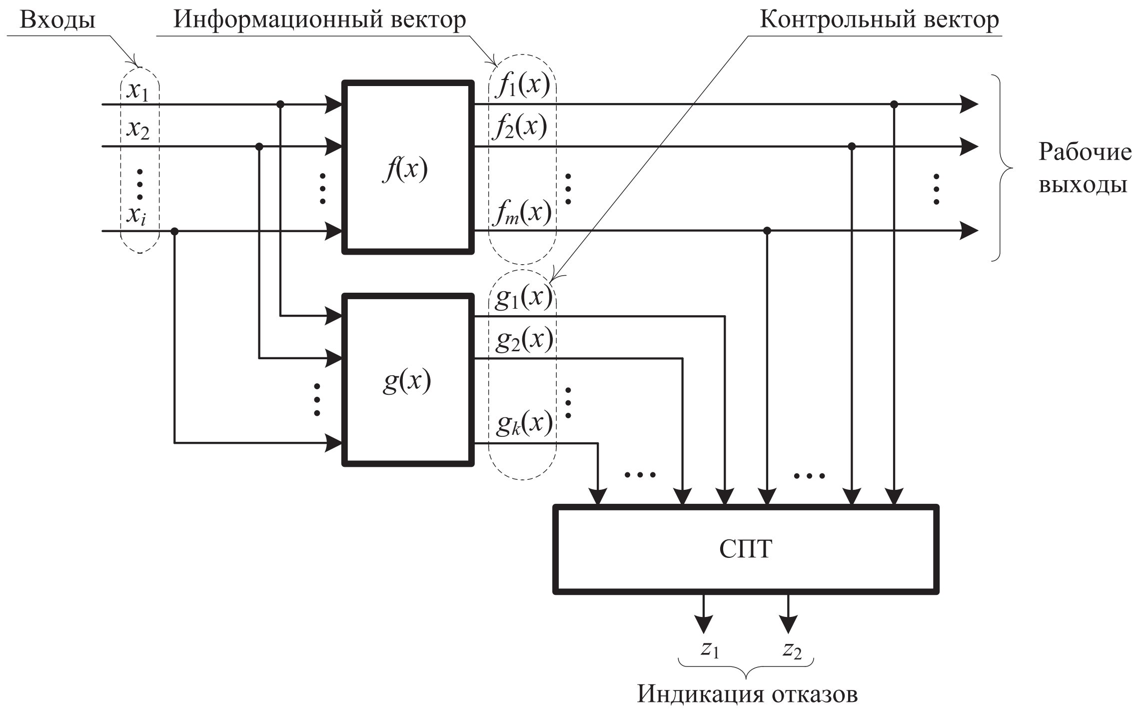

Рис. 1. Структура системы функционального контроля

соответствие между информационными и контрольными векторами с использованием правил построения кода с суммированием. При работе системы функционального контроля это соответствие в произвольный момент фиксируется самопроверяемым тестером (СПТ) [2, 7].

Основной целью организации системы функционального контроля является обеспечение тестирования блока $f(x)$ в его рабочем режиме, поэтому тестер должен фиксировать искажения в информационных разрядах при возникновении любых одиночных неисправностей в блоке $f(x)$. Кроме того, СПТ должен обеспечивать фиксацию ошибок как в блоке $g(x)$, так и в своей структуре. При исправной работе всех составляющих системы функционального контроля на выходах СПТ формируется парафазный сигнал <01> или <10>, возникновение неисправности нарушает парафазность выходного сигнала.

Наиболее простым вариантом построения системы функционального контроля является использование классических кодов с суммированием, или кодов Бергера [11]. В них значения разрядов контрольного вектора являются двоичным представлением числа единичных информационных разрядов (значения веса r информационного вектора). Коды Бергера обнаруживают любые монотонные (однонаправленные) ошибки, что эффективно используется при построении систем функционального контроля логических устройств [9, 12]. Далее коды Бергера будем обозначать как $S(m, k)$-коды.

Свойства $S(m, k)$-кодов по обнаружению ошибок в информационных векторах описываются в [13]. Например, установлено, что любой код Бергера не обнаруживает одинаковый процент ошибок четной кратности d в информационных векторах от общего числа ошибок той же кратности в информационных векторах:

$$\beta_d = 2^{-d} C_d^{\frac{d}{2}}. \qquad (1)$$

Из формулы (1) следует, что $S(m, k)$-кодами не обнаруживаются 50 % двукратных, 37,5 % четырехкратных, 31,25 % шестикратных ошибок и т. д.

Улучшение эффективности обнаружения ошибок $S(m, k)$-кодами в информационных векторах достигается путем модификации кода [14, 15].

2 Модифицированный код Бергера

В [15] предложен метод построения кода с суммированием, получающегося путем модификации известного кода Бергера по следующему алгоритму.

Алгоритм 1. Правила построения модифицированного кода Бергера:

1. Устанавливается модуль $M=2^{\lceil \log_2(m+1) \rceil -1}$;

2. Подсчитывается вес информационного вектора r (число единичных информационных разрядов);

3. Число r представляется по модулю M (другими словами, определяется вычет числа r по заданному модулю): $V=(r) \bmod M$;

4. Определяется поправочный коэффициент α, равный сумме по модулю «два» произвольного (но заранее установленного) числа любых разрядов информационного вектора;

5. Формируется число $W = V + \alpha M$;

6. Полученное число W представляется в двоичном виде и записывается в контрольный вектор.

Модифицированные коды Бергера обозначаются как $RS(m, k)$-коды. В общем случае $RS(m, k)$-коды почти вдвое эффективнее обнаруживают ошибки в информационных векторах, чем классические коды Бергера. Алгоритм 1 иллюстрируется на примере формирования контрольных векторов для нескольких информационных векторов $RS(6,3)$-кода с различным весом r, при этом поправочный коэффициент был вычислен по формуле $\alpha = x_1 \oplus x_2 \oplus x_3$ (табл. 1).

В отличие от классического $S(m, k)$-кода в $RS(m, k)$-коде информационные векторы с различным значением веса r могут иметь одинаковые контрольные векторы, что определяется формулой вычисления поправочного коэффициента α. Например, информационные векторы <000001> и <011111> в табл. 1 имеют одинаковые контрольные векторы. Это улучшает характеристики кода по обнаружению ошибок в целом, но привносит в их количество часть монотонных (однонаправленных) ошибок кратности $d = M$.

В зависимости от числа информационных разрядов, используемых для вычисления поправочного коэффициента α, может быть построено $\sum_{i=1}^{m-1} 2^i = 2^m - 2$ модифицированных кодов Бергера. Однако, как показано в [15], в общем случае характеристики $RS(m, k)$-кодов зависят только от количества суммируемых

ТАБЛИЦА 1. Векторы $RS(6,3)$-кода

Информационный вектор						r	V	α	W	Контрольный вектор		
x_1	x_2	x_3	x_4	x_5	x_6					y_1	y_2	y_3
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	1	0	0	1
0	0	0	0	1	1	2	2	0	2	0	1	0
0	0	0	1	1	1	3	3	0	3	0	1	1
0	0	1	1	1	1	4	0	1	4	1	0	0
0	1	1	1	1	1	5	1	0	1	0	0	1
1	1	1	1	1	1	6	2	1	6	1	1	0

в поправочном коэффициенте разрядов, но не от того, какие именно это разряды. При $m < 8$ все $RS(m, k)$-коды имеют одинаковое общее количество необнаруживаемых ошибок в информационных векторах, но различные распределения необнаруживаемых ошибок по кратностям. При $m \geq 8$ минимальное общее количество необнаруживаемых ошибок в $RS(m, k)$-коде достигается при суммировании в поправочном коэффициенте $\left\lfloor \frac{m}{2} \right\rfloor$ разрядов информационного вектора, где запись $\lfloor ... \rfloor$ обозначает целое снизу от вычисляемого значения. Данные свойства $RS(m, k)$-кодов позволяют на практике обосновано выбирать тот или иной вариант кодирования в зависимости от свойств контролируемого логического устройства.

Свойства $RS(m, k)$-кодов по обнаружению ошибок в системах функционального контроля достаточно полно описаны в [14–17]. Открытым остается вопрос синтеза генераторов тестеров $RS(m, k)$-кодов.

Генератор входит в структуру тестера (рис. 2) и предназначен для формирования контрольных функций по значениям информационных разрядов. Сформированные значения контрольных функций сравниваются в компараторе с одноименными значениями функций, подсчитанных блоком контрольной логики. Компараторы для кодов с суммированием строятся путем каскадного соединения модулей сравнения парафазных сигналов [11]. Более сложной задачей является синтез генераторов.

В [18] предложено строить генераторы на основе полных сумматоров, полусумматоров и сумматоров по модулю «два». В [19] этот метод применен для построения генераторов с наименьшей сложностью для модульных кодов с суммированием и для построения генераторов $RS(m, k)$-кодов.

Под сложностью понимается число входов логических элементов без учета инверсий на входах. Генераторы оцениваются также контролепригодностью и быстродействием. Контролепригодность генератора – это число тестовых наборов, на которых обеспечивается проверка всех одиночных неисправностей в его структуре. Быстродействие определяется числом уровней в схеме генератора.

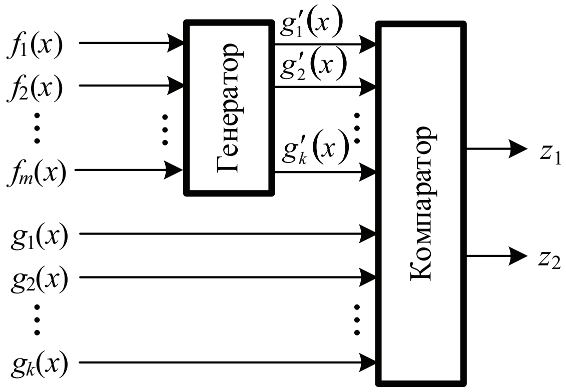

Рис. 2. Структура тестера $RS(m, k)$-кода

Рассмотрим подход, позволяющий синтезировать тестеры с улучшенной характеристикой быстродействия в сравнении с подходом, описанным в [19].

3 Новый подход к синтезу генераторов модифицированных кодов Бергера

Определение [20]. Простой симметричной функцией называется функция алгебры логики вида

$$f^{v}(x_1, x_2, ..., x_n) = \\ = \bigvee_{i_1, i_2, ..., i_v \in (1, 2, ..., n)} x_{i_1} x_{i_2} \cdot ... \cdot x_{i_v}, \quad (2)$$

где знаками дизъюнкции объединены все конъюнкции без инверсий одного ранга m (с одинаковым количеством переменных).

В [21] на основе анализа таблицы истинности $S(m, k)$-кода предлагается строить генератор по системе функций S_i, описывающих i-е разряды контрольного вектора $< S_0 \, S_1 \, \ldots S_{h-2} \, S_{h-1} >$:

$$S_i = f^{v_1} \overline{f^{v_1+2^i}} \vee f^{v_2} \overline{f^{v_2+2^i}} \vee ... \vee f^{a} \overline{f^{b}}, \quad (3)$$

где $v_1 = 2^i$, $v_{j+1} = v_j + 2^{i+1}$ $(j \in \{1, 2, 3, ...\})$, $a = p' \cdot 2^{i+1} + 2^i$, $p' = \left\lceil \frac{p}{2^{i+1}} \right\rceil$, $p = v - 2^i$, $b = a + 2^i$, если $a + 2^i \le v$, и $b = 0$ – в противном случае. Здесь и далее для простоты обозначений опустим переменные x_i в обозначении простой симметричной функции.

Из введенных определений, формул и обозначений следует алгоритм построения генераторов $S(m, k)$-кодов [22], который может быть непосредственно применен и для синтеза генераторов $RS(m, k)$-кодов.

Алгоритм 2. Модифицированный алгоритм синтеза тестеров $RS(m, k)$-кодов:

1. Генератор строится из трех блоков (A_1, A_2 и A_3) так, как это показано на рис. 3;

2. Блок A_1 реализует систему простых симметричных функций с четными верхними индексами и строится следующим образом:

2.1. Множество входных переменных $M = \{x_1, \dots, x_p, \dots, x_m\}$ разбивается на два подмножества $M_1 = \{x_1, \dots, x_p\}$ и $M_2 = \{x_{p+1}, \dots, x_m\}$, где p – ближайшее целое четное значение к величине $\frac{m}{2}$;

2.2. На каждом этапе разбиения множества входных переменных для реализации простых симметричных функций используется формула разложения:

$$\begin{gathered} f^i(x_1, ..., x_p, ..., x_m) = \\ = f^i(x_1, ..., x_p) \vee f^i(x_{p+1}, ..., x_m) \vee \\ f^{i-1}(x_1, ..., x_p) f^1(x_{p+1}, ..., x_m) \vee \\ \vee f^{i-2}(x_1, ..., x_p) f^2(x_{p+1}, ..., x_m) \vee \\ ... \vee f^1(x_1, ..., x_p) f^{m-1}(x_{p+1}, ..., x_m); \end{gathered} \tag{4}$$

2.3. Строится первый уровень блока A_1;

2.4. Получившиеся подмножества M_1 и M_2 также разбиваются на подмножества M_{11}, M_{12} и M_{21}, M_{22} аналогично тому, как это сделано в п. 2.1. К ним применяется п. 2.2;

2.5. Операции 2.2–2.4 повторяются до тех пор, пока не получаются множества из одной или двух переменных;

2.6. При непосредственном построении схемы генератора пользуются следующими элементарными функциями от одной и двух переменных:

$$f^1(x_i) = x_i,$$

$$f^1(x_i, x_j) = x_i \vee x_j,$$

$$f^2(x_i, x_j) = x_i x_j;$$

3. Блок A_2 реализует систему функций S_i (3), причем $i \in \{1, 2, ..., h-1\}$, где $h = k - 1$, $k = \log_2 M$ и $M = 2^{\lceil \log_2(m+1) \rceil - 1}$;

4. Блок A_3 реализует функцию младшего контрольного разряда S_0.

Для примера построим по предложенному алгоритму генератор тестера $RS(6,3)$-кода с формулой поправочного коэффициента $\alpha = x_1 \oplus x_2 \oplus x_3$.

Первый этап построения блока A_1 – разбиение множества входных переменных.

Разбиваем множество входных переменных $M = \{x_1 \div x_6\}$ на два подмножества: $M_1 = \{x_1 \div x_4\}$ и $M_2 = \{x_5, x_6\}$.

Пользуясь формулой (4), представляем функции $f_1 \div f_6$ в виде разложений:

$$f^1(x_1 \div x_6) = f^1(x_1 \div x_4) \vee f^1(x_5, x_6);$$

$$\begin{gathered} f^2(x_1 \div x_6) = f^2(x_1 \div x_4) \vee \\ f^2(x_5, x_6) \vee f^1(x_1 \div x_4) f^1(x_5, x_6); \end{gathered}$$

$$\begin{gathered} f^3(x_1 \div x_6) = f^3(x_1 \div x_4) \vee \\ f^2(x_1 \div x_4) f^1(x_5, x_6) \vee \\ f^1(x_1 \div x_4) f^2(x_5, x_6); \end{gathered}$$

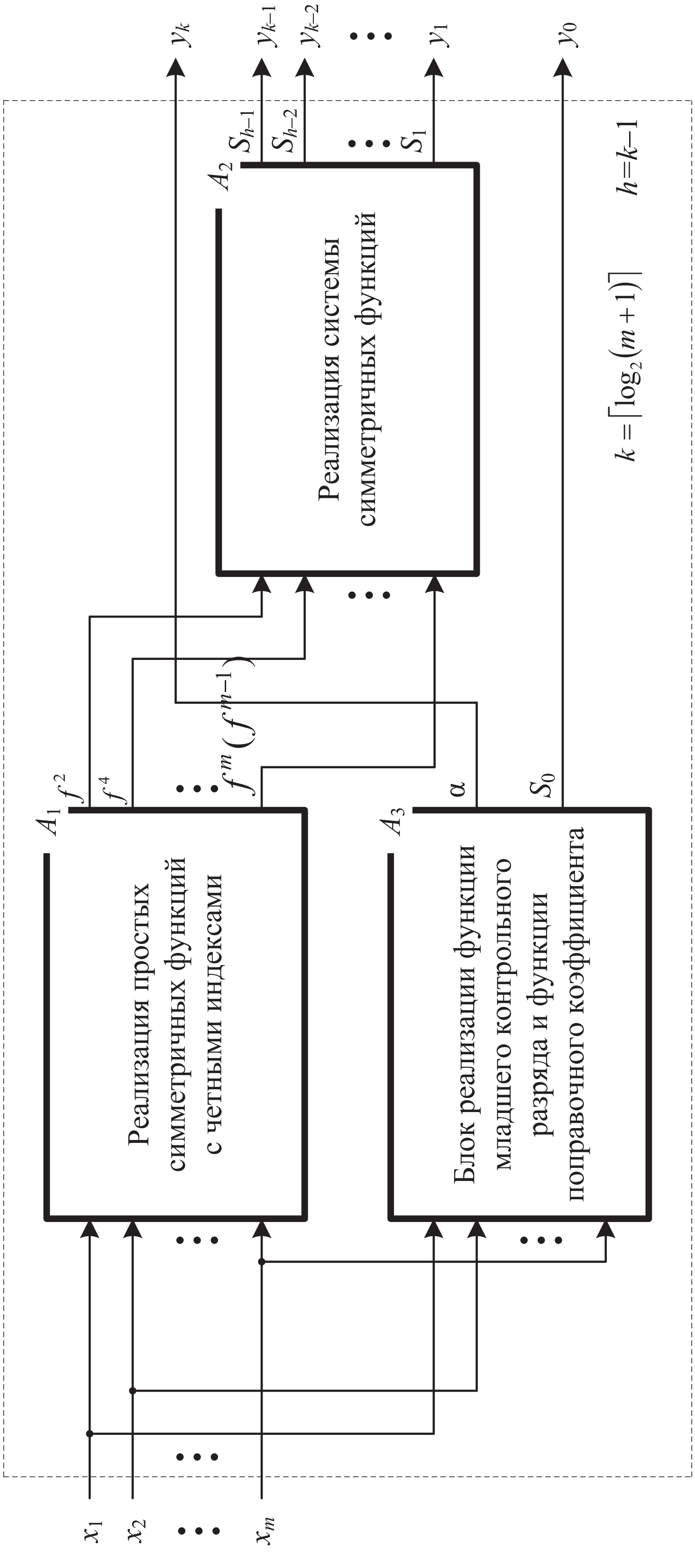

Рис. 3. Структура генератора $RS(m, k)$-кода

$$f^4(x_1 \div x_6) = f^4(x_1 \div x_4) \vee$$
$$f^3(x_1 \div x_4) f^1(x_5, x_6) \vee$$
$$f^2(x_1 \div x_4) f^2(x_5, x_6);$$

$$f^5(x_1 \div x_6) =$$
$$= f^4(x_1 \div x_4) f^1(x_5, x_6) \vee$$
$$f^3(x_1 \div x_4) f^2(x_5, x_6);$$

$$f^6(x_1 \div x_6) = f^4(x_1 \div x_4) f^2(x_5, x_6).$$

Функции $f_1 \div f_6$ входят в первый уровень схемы генератора.

Следующим этапом построения блока A_1 является разбиение подмножества M_1= $=\{x_1 \div x_4\}$. Подмножество $M_2=\{x_5, x_6\}$ разделять не требуется, так как оно описывает элементарные функции.

Для $M_1=\{x_1 \div x_4\}$:

$$f^1(x_1 \div x_4) = f^1(x_1, x_2) \vee f^1(x_3, x_4);$$

$$f^2(x_1 \div x_4) = f^2(x_1, x_2) \vee f^2(x_3, x_4) \vee$$
$$f^1(x_1, x_2) f^1(x_3, x_4);$$

$$f^3(x_1 \div x_4) = f^2(x_1, x_2) f^1(x_3, x_4) \vee$$
$$f^1(x_1, x_2) f^2(x_3, x_4);$$

$$f^4(x_1 \div x_4) = f^2(x_1, x_2) f^2(x_3, x_4).$$

Полученные разложения составляют второй уровень схемы генератора.

В третий уровень схемы генератора входят элементарные функции, входящие в правые части всех выражений для $f_1 \div f_6$.

Блок A_2 является технической реализацией следующей системы функций (см. рис. 2):

$$S_1 = f^2 \overline{f^4} \vee f^6.$$

Блок реализации функции младшего контрольного разряда (A_3) строится отдельно по формуле

$$S_0 = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6.$$

Функция поправочного коэффициента в модифицированном коде Бергера реализует старший контрольный разряд. При этом дополнительного блока для ее реализации не требуется, так как любое сочетание информационных разрядов, складываемых по модулю «два», может быть получено в блоке A_3.

Схема синтезированного генератора RS(6,3)-кода изображена на рис. 4, а на рис. 5 приводится моделирование ее работы в среде Multisim при поступлении на входы вектора <111101>. В этом случае, согласно правилам построения модифицированных кодов Бергера, формируется контрольный вектор <101>. Эксперимент с моделированием работы схемы генератора подтверждает корректность его работы.

В табл. 2 приводится сравнение характеристик генераторов S(6,3)- и RS(6,3)-кодов. Генератор RS(6,3)-кода имеет такие же характеристики быстродействия и контролепригодности, что и генератор S(6,3)-кода, построенный по данному методу; характеристика сложности (по числу входов логических элементов без учета инверсий) для генератора RS(6,3)-кода улучшена в сравнении со сложностью генератора S(6,3)-кода.

Генератор любого $RS(m, k)$-кода имеет такие же характеристики быстродействия и контролепригодности, как и генератор соответствующего $S(m, k)$-кода, и такую же или улучшенную характеристику сложности.

Утверждение следует из методики построения генераторов. Для любого $RS(m, k)$-кода генератор строится аналогичным образом, как и генератор $S(m, k)$-кода, однако ввиду того, что модуль $RS(m, k)$-кода $M = 2^{\lceil \log_2(m+1) \rceil - 1}$, а модуль $S(m, k)$-кода – $M = 2^{\lceil \log_2(m+1) \rceil}$, не требуется реализация функции старшего контрольного разряда, так как им является функция поправочного коэффициента. Исходя из этого, все характеристики, кроме сложности, остаются неизменными. А сложность остается неизменной, либо уменьшается.

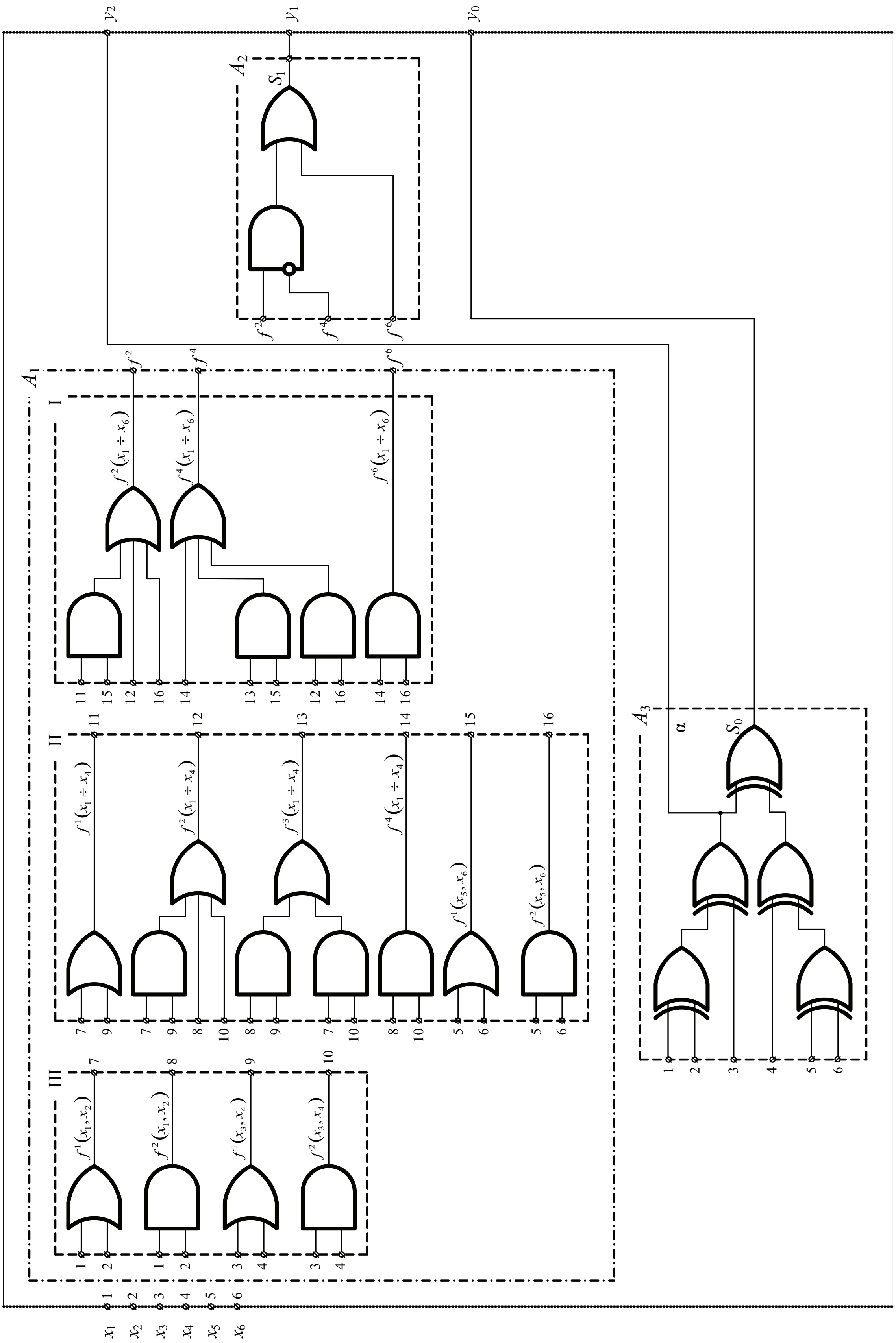

Рис. 4. Генератор тестера *RS*(6,3)-кода

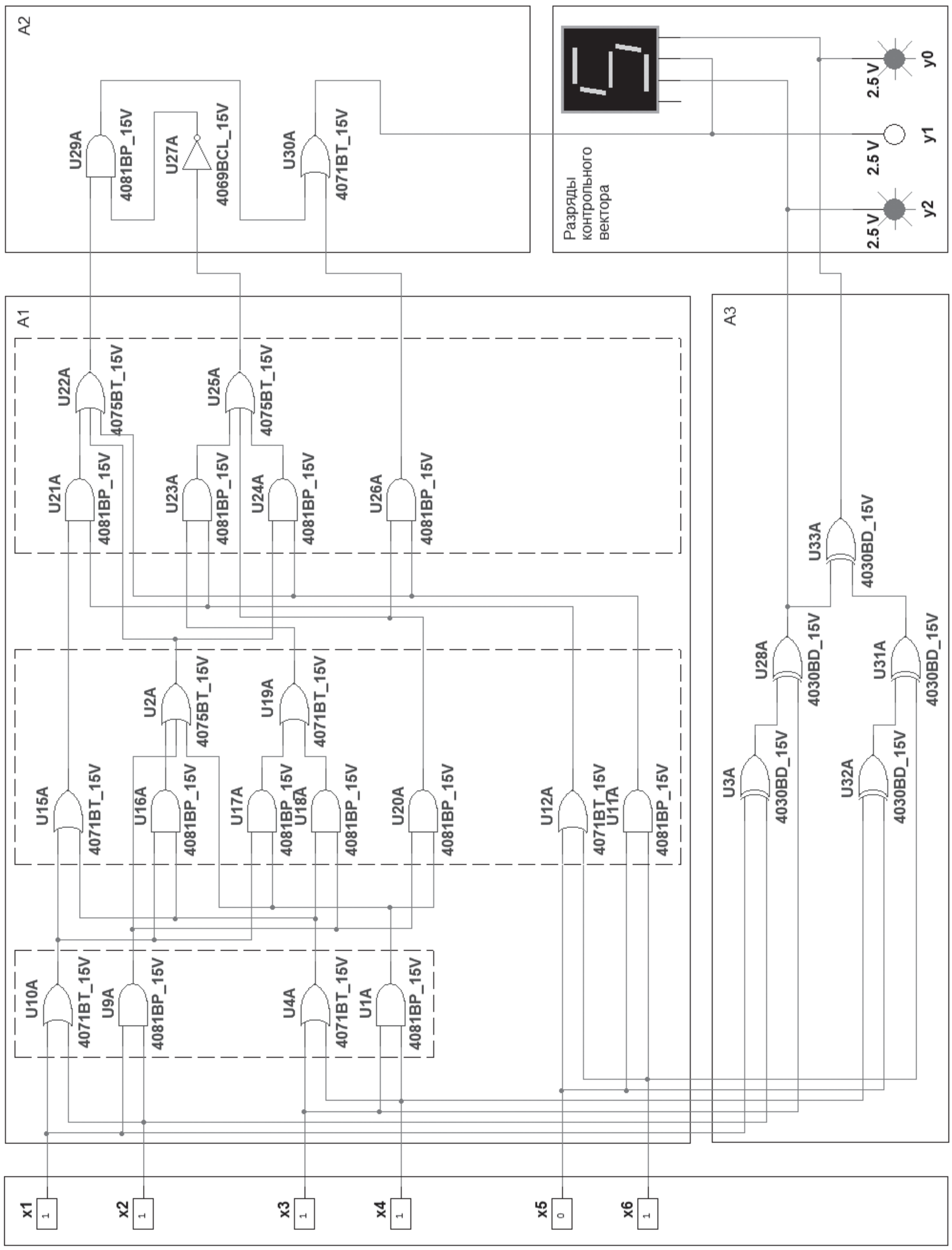

Рис. 5. Моделирование работы генератора тестера *RS*(6,3)-кода в Multisim

ТАБЛИЦА 2. Характеристики генераторов *S*(6,3)- и *RS*(6,3)-кодов

Код	Сложность, *L*	Быстродействие, *q*	Контролепригодность, *t*
S(6,3)	77	8	15
RS(6,3)	75	8	15

Сложность генератора *RS*(*m*, *k*)-кода не уменьшается в сравнении с генератором *S*(*m*, *k*)-кода при тех значениях *m*, при которых для реализации старшего контрольного разряда в блоке A_2 не требуется логических элементов. Из формулы (3) следует, что существуют такие значения длины информационных векторов, для которых функция старшего контрольного разряда будет равна простой симметричной функции, вычисленной блоком A_1. К примеру, для *S*(4,3)-кода $S_2 = f^4$, и для ее реализации логических элементов не требуется.

Заключение

В представленном исследовании описывается ранее не применявшийся способ построения генераторов тестеров модифицированных кодов Бергера, позволяющий в некоторых случаях получать генераторы с улучшенными характеристиками в сравнении с генераторами классических кодов с суммированием.

Результат данной работы может быть эффективно использован при построении надежных дискретных систем управления на базе программируемых интегральных схем FPGA [23, 24]. Например, в [25] рассматриваются вопросы синтеза самопроверяемых конечных автоматов, дается приложение результатов данных исследований для синтеза системы управления движением поездов на базе FPGA. Для обеспечения надежности и безопасности функционирующей системы предлагается использовать помехоустойчивое кодирование. Модифицированный код Бергера может быть использован для контроля комбинационных составляющих в конечных автоматах на базе FPGA. Соответственно, в этих структурах при синтезе генераторов (кодеров) *RS*(*m*, *k*)-кодов может быть использован подход, описанный в настоящей статье.

Приведенный в работе алгоритм построения генератора тестера *RS*(*m*, *k*)-кода может быть применен и для синтеза генераторов модульно модифицированных кодов с суммированием (*RSM*(*m*, *k*)-кодов, где *M* – значение модуля). При этом в зависимости от значения модуля $M \in \{2;4;8;...;2^{\lceil \log_2(m+1) \rceil - 2}\}$ при реализации блока A_2 необходимо строить только $k = \log_2 M$ функций, описывающих младшие разряды контрольного вектора, исключая самый младший из них – он реализуется отдельно блоком A_3. Для класса *RS*2(*m*, *k*)-кодов ограничиваются синтезом только блока A_3, так как остальные элементы генератора не требуются.

Библиографический список

1. **Самопроверяемые** дискретные устройства / В. В. Сапожников, Вл. В. Сапожников. – Санкт-Петербург : Энергоатомиздат, 1992. – 224 с.

2. **Основы** технической диагностики (оптимизация алгоритмов диагностирования, аппаратурные средства) / П. П. Пархоменко, Е. С. Согомонян. – Москва : Энергоатомиздат, 1981. – 320 с.

3. **McCluskey E. J.** Logic Design Principles: With Emphasis on Testable Semicustom Circuits. – NJ : Prentice Hall PTR, 1986. – 549 p.

4. **Ubar R.** Test Synthesis with Alternative Graphs // IEEE Design & Test of Computers. – 1996. – Vol. 13. – Is. 1. – P. 48–57.

5. **Обнаружение** несущественных путей логических схем на основе совместного анализа И-ИЛИ деревьев и SSBDD-графов / А. Ю. Матросова,

С. А. Останин, В. Сингх // Автоматика и телемеханика. – 2013. – № 7. – С. 126–142.

6. **Самотестируемая** структура для функционального обнаружения отказов в комбинационных схемах / М. Гессель, А. В. Дмитриев, В. В. Сапожников, Вл. В. Сапожников // Автоматика и телемеханика. – 1999. – № 11. – С. 162–174.

7. **Самопроверяемые** устройства и отказоустойчивые системы / Е. С. Согомонян, Е. В. Слабаков. – Москва : Радио и связь, 1989. – 208 с.

8. **Goessel M.**, Graf S. Error Detection Circuits. – L : McGraw-Hill, 1994. – 261 p.

9. **Fujiwara E.** Code Design for Dependable Systems : Theory and Practical Applications. – NY : John Wiley & Sons, 2006. – 720 p.

10. **Wang L.-T.**, Stroud C. E., Touba N. A. System-on-Chip Test Architectures : Nanometer Design for Testability. – Waltham : Morgan Kaufmann Publ., 2008. – 856 p.

11. **Berger J. M.** A Note on Error Detection Codes for Asymmetric Channels // Inform. and Control. – 1961. – Vol. 4. – Is. 1. – P. 68–73.

12. **Lo J.-C.** An SFS Berger Check Prediction ALU and Its Application to Self-Checking Processor Designs // Computer-Aided Design of Integrated Circuits and Systems. – 1992. – Vol. 11. – Is. 4. – P. 525–540.

13. **О свойствах** кода с суммированием в схемах функционального контроля / Д. В. Ефанов, В. В. Сапожников, Вл. В. Сапожников // Автоматика и телемеханика. – 2010. – № 6. – С. 155–162.

14. **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 Symp. (EWDTS`2012), Kharkov, Ukraine, Sept. 14–17, 2012. – Kharkov, 2012. – P. 114–117.

15. **Построение** модифицированного кода Бергера с минимальным числом необнаруживаемых ошибок информационных разрядов / А. А. Блюдов, Д. В. Ефанов, В. В. Сапожников, Вл. В. Сапожников // Электронное моделирование. – 2012. – Т. 34, № 6. – С. 17–29.

16. **Коды** с суммированием для организации контроля комбинационных схем / А. А. Блюдов, Д. В. Ефанов, В. В. Сапожников, Вл. В. Сапожников // Автоматика и телемеханика. – 2013. – № 6. – С. 153–164.

17. **О кодах** с суммированием единичных разрядов в системах функционального контроля / А. А. Блюдов, Д. В. Ефанов, В. В. Сапожников, Вл. В. Сапожников // Автоматика и телемеханика. – 2014. – № 8. – С. 131–145.

18. **Piestrak S. J.** Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. – Wrocław : Oficyna Wydawnicza Politechniki Wrocłavskiej, 1995. – 111 p.

19. **Исследование** модифицированных кодов с суммированием в системах технической диагностики и обработки информации в устройствах железнодорожной автоматики и телемеханики : дис. … канд. техн. наук (05.13.06) / А. А. Блюдов ; Петербург. гос. ун-т путей сообщения. – Санкт-Петербург, 2013. – 230 с.

20. **Синтез** самопроверяющихся тестеров для кодов с суммированием / А. Г. Мельников, В. В. Сапожников, Вл. В. Сапожников // Проблемы передачи информации. – 1986. – Т. XXII, № 2. – С. 85–97.

21. **Универсальный** алгоритм синтеза самопроверяющихся тестеров для кодов с постоянным весом / В. В. Сапожников, Вл. В. Сапожников // Проблемы передачи информации. – 1984. – Т. XX, № 2. – С. 65–76.

22. **О синтезе** тестеров кодов с суммированием на основе использования свойств простых и линейных функций / В. В. Сапожников, Вл. В. Сапожников, Д. В. Ефанов // Вестн. Урал. гос. ун-та путей сообщения. – 2011. – № 1. – С. 22–32.

23. **Проектирование** и верификация цифровых систем на кристаллах. Verilog & System Verilog / В. И. Хаханов, И. В. Хаханова, Е. И. Литвинова, О. А. Гузь. – Харьков : Новое слово, 2010. – 528 с.

24. **Navabi Z.** Digital System Test and Testable Design : Using HDL Models and Architectures. – NY : Springer Sci. + Bus. Media, LLC, 2011. – 435 p.

25. **Ubar R.**, Raik J., Vierhaus H.-T. Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source) // Hershey : NY : Inform. Sci. Reference, IGI Global, 2011. – 578 p.