

ISSN 1816-0301 (Print)  
ISSN 2617-6963 (Online)

## ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

### LOGICAL DESIGN

УДК 004.052.32+681.518.5

Поступила в редакцию 30.05.2018  
Received 30.05.2018

Д. В. Ефанов<sup>1</sup>, В. В. Сапожников<sup>2</sup>, Вл. В. Сапожников<sup>2</sup>, Д. В. Пивоваров<sup>2</sup>

<sup>1</sup>ООО «ЛокоТех-Сигнал», Российский университет транспорта (МИИТ),  
Москва, Россия

<sup>2</sup>Петербургский государственный университет путей сообщения Императора Александра I,  
Санкт-Петербург, Россия

### СИНТЕЗ САМОПРОВЕРЯЕМЫХ СХЕМ ВСТРОЕННОГО КОНТРОЛЯ НА ОСНОВЕ МЕТОДА ЛОГИЧЕСКОГО ДОПОЛНЕНИЯ ДО РАВНОВЕСНОГО КОДА «2 ИЗ 4»

**Аннотация.** Исследуются особенности синтеза самопроверяемых схем встроенного контроля по методу логического дополнения на основе равновесного кода «2 из 4». Описываются особенности реализации схем встроенного контроля по методу логического дополнения. Отмечается возможность синтеза структур дискретных устройств, имеющих меньшую структурную избыточность, чем при реализации схемы контроля по методу дублирования. Эффект в снижении структурной избыточности достигается за счет минимизации сложности технической реализации блока контрольной логики и использования более простых по своим структурам тестеров, чем компаратор в системе дублирования. Предлагается способ организации схемы встроенного контроля, основанный на доопределении значений контрольных функций с учетом обеспечения тестируемости элементов сложения по модулю два в блоке логического дополнения и тестера кода «2 из 4».

**Ключевые слова:** самопроверяемая схема встроенного контроля, метод логического дополнения, равновесный код «2 из 4», тестер, элемент сложения по модулю два, тестируемость компонентов

**Для цитирования.** Ефанов, Д. В. Применение равновесного кода «2 из 4» при организации самопроверяемых схем встроенного контроля на основе метода логического дополнения / Д. В. Ефанов [и др.] // Информатика. – 2018. – Т.15, № 4. – С. 71–85.

D. V. Efanov<sup>1</sup>, V. V. Sapozhnikov<sup>2</sup>, Vl. V. Sapozhnikov<sup>2</sup>, D. V. Pivovarov<sup>2</sup>

<sup>1</sup>"LocoTech-Signal" LCC, Russian University of Transport, Moscow, Russia

<sup>2</sup>Emperor Alexander I St. Petersburg State Transport University, Saint Petersburg, Russia

### THE SELF-CHECKING INTEGRATED CONTROL CIRCUITS SYNTHESIS BASED ON THE BOOLEAN COMPLEMENT METHOD TO "2-OUT-OF-4" CONSTANT-WEIGHT CODE

**Abstract.** The article explores the peculiarities of self-checking integrated control circuits synthesis by the Boolean complement method based on the "2-out-of-4" constant-weight code. The article describes the features of integrated control circuits implementation by the Boolean complement method. It is noted that it is possible to synthesize the structures of discrete devices, which have less structural redundancy than in situation of the control circuit implementation by the method of duplication. The effect in structural redundancy reducing is achieved by minimizing the complexity of the control logic block technical implementation and using checkers that are simpler in their structures than the comparator in the system of duplication. The article proposes a method of the integrated control circuit organization based on determining the values of control functions taking into account the maintenance of testability of elements of addition by modulo two in the Boolean complement block and the checker of the "2-out-of-4" code.

**Keywords:** self-checking control circuit, Boolean complement method, "2-out-of-4" constant-weight code, checker, element of addition by modulo two, testability of components

**For citation.** Efanov D. V., Sapozhnikov V. V., Sapozhnikov Vl. V., Pivovarov D. V. The self-checking integrated control circuits synthesis based on the Boolean complement method to "2-out-of-4" constant-weight code. *Informatics*, 2018, vol.15, no. 4, pp. 71–85 (in Russian).

**Введение.** При синтезе надежных компонентов систем автоматического управления на различных элементных базах широко используются самопроверяемые схемы встроенного контроля (ССВК) [1–4]. Их применение позволяет синтезировать устройства, наделенные свойством обнаружения и индикации наличия неисправностей, что, в свою очередь, дает возможность оперативного отключения от работы отказавших узлов в процессе эксплуатации с целью блокирования неверных результатов вычислений и запрета их передачи на последующие узлы или же непосредственно на объекты управления.

ССВК может быть построена различными способами. Одним из распространенных и повсеместно применяемых методов является метод дублирования [5–7]. Дублирование подразумевает применение в качестве схемы контроля копии объекта диагностирования и наличие самопроверяемого компаратора [8, 9]. Простота метода и возможность обнаружения любых неисправностей в объекте диагностирования определяют и повсеместное использование данной структурной схемы. Например, ее эффективно применяют при синтезе надежных компонентов систем управления движением поездов [10–12]. Альтернативой дублированию служит построение схемы контроля на основе заранее выбранного избыточного равномерного кода. В этом случае удастся уменьшить структурную избыточность ССВК, однако заметно снижаются возможности обнаружения различных видов неисправностей (например, полного множества сочетаний кратных неисправностей). На практике многие из них крайне маловероятны [13]. Учет этого фактора позволяет синтезировать более простые логические устройства с приемлемыми характеристиками обнаружения неисправностей. Например, если в качестве модели неисправностей выступает модель одиночной константной неисправности выходов внутренних логических элементов, то высокую эффективность при синтезе ССВК показывает применение равновесных кодов и разнообразных кодов с суммированием [14–17]. С учетом особенностей обнаружения данными кодами ошибок в кодовых словах (и информационных векторах) можно строить более простые структуры ССВК с обнаружением 100 % одиночных константных неисправностей. Примеры таких реализаций даны в работах [18, 19].

Одним из подходов к построению ССВК является метод логического дополнения, впервые, по всей видимости, описанный в [20]. Впоследствии данный метод исследовался во многих работах, например [4, 21–27]. Метод логического дополнения подразумевает преобразование в ССВК значений рабочих функций таким образом, чтобы их контроль можно было осуществить на основе использования какого-либо равновесного кода [28, 29] либо же с применением контроля по особым признакам получаемого сигнала, например контроля самодвойственного сигнала [4, 20].

Эффективным оказывается применение при синтезе ССВК равновесных кодов с малой длиной кодовых слов ( $n \leq 6$ ). Это объясняется тем фактом, что тестеры равновесных кодов при отмеченном ограничении имеют простые структуры, требуют малого количества тестовых комбинаций для полной проверки, а количество элементов сложения по модулю два (*XOR's*), необходимых для преобразования значений рабочих функций, невелико.

Особое место среди всех равновесных кодов занимает код «2 из 4» (2/4-код), тестер которого имеет наиболее простую структуру из всех известных равновесных кодов и требует для полной проверки формирования всего четырех кодовых комбинаций. Кроме того, при построении ССВК потребуется преобразование всего двух из четырех рабочих функций объекта диагностирования. Исследованиям применения 2/4-кода при организации ССВК посвящен ряд публикаций авторов данной статьи, в том числе [26, 27]. В настоящей работе описывается новый способ получения значений контрольных функций при синтезе ССВК по методу логического дополнения до 2/4-кода.

**Основные структурные схемы организации самопроверяемых логических устройств.** В этих схемах объект диагностирования – блок  $F(x)$ , реализующий некоторую систему булевых функций, – снабжается специализированной ССВК, позволяющей в процессе эксплуатации конечного устройства косвенно обнаруживать неисправности по их проявлениям в виде искажений рабочих сигналов.

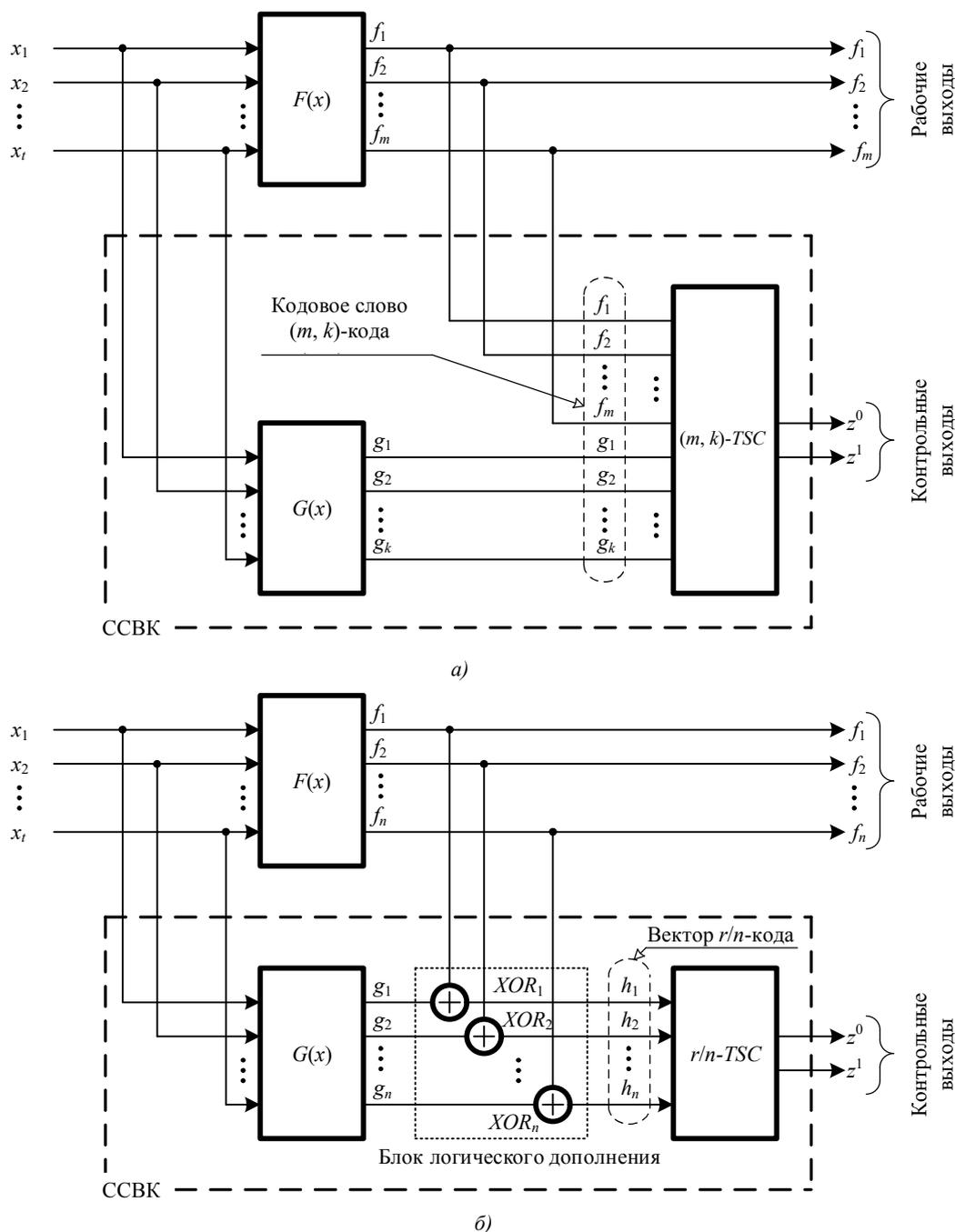


Рис. 1. Структурная схема организации контроля по методу вычисления контрольных разрядов (а) и по методу логического дополнения (б)

Широко применяемый метод вычисления контрольных разрядов (рис. 1, а) подразумевает синтез ССВК по правилам, определяемым правилами построения заранее выбранного разделимого  $(m, k)$ -кода ( $m$  и  $k$  – количество информационных и контрольных разрядов в кодовом слове кода). Информационный вектор  $(m, k)$ -кода отождествляется с вектором функций, формируемым на выходах блока  $F(x)$ , а контрольный вектор  $(m, k)$ -кода формируется в ССВК на выходах блока контрольной логики  $G(x)$ . В ССВК для контроля принадлежности формируемого на выходах обоих блоков  $F(x)$  и  $G(x)$  в любой момент времени кодового слова заранее выбранному  $(m, k)$ -коду используется тестер  $(m, k)$ -TSC (*totally self-checking checker*). При исправности всех

компонентов конечного устройства на выходах  $(m, k)$ -TSC формируется парафазный сигнал  $\langle 01 \rangle$  или  $\langle 10 \rangle$ . Неисправности в элементах структур блоков  $F(x)$  и  $G(x)$  искажают значения разрядов информационного или контрольного векторов. Это приводит к установлению на выходах тестера непарафазных сигналов  $\langle 00 \rangle$  или  $\langle 11 \rangle$ . Сам тестер реализуется в виде самопроверяемой схемы, что позволяет при любой его внутренней неисправности из заданного класса на выходах также формировать непарафазный сигнал  $\langle 00 \rangle$  или  $\langle 11 \rangle$ .

Часто при синтезе ССВК по методу вычисления контрольных разрядов применяют коды с повторением (схему дублирования), а также разнообразные коды с суммированием (коды Бергера и их модификации) [1, 2].

Альтернативным вариантом реализации ССВК является ее синтез по методу логического дополнения (рис. 1, б). Данный метод подразумевает преобразование (дополнение) любого информационного вектора, формируемого на выходах блока  $F(x)$ , в кодовое слово заранее выбранного неразделимого кода (например, равновесного) либо преобразование сигнала каждой рабочей функции в сигнал специального вида (например, самодвойственный). Для преобразования значений рабочих функций в ССВК применяется блок логического дополнения, реализующий функции дополнения по формуле

$$h_i = f_i \oplus g_i, \quad i = \overline{1, n}. \quad (1)$$

Из формулы (1) следует, что блок логического дополнения образуется каскадом параллельно установленных элементов сложения по модулю два (элементов  $XOR$ ). На каждый такой элемент поступают значения рабочей и контрольной функций. При этом значения контрольных функций подбираются таким образом, чтобы на выходе блока логического дополнения формировались функции заранее определенного вида, например функции вычисления разрядов равновесного кода « $r$  из  $n$ » ( $r/n$ -кода, где  $r$  – число единичных разрядов в кодовом слове длиной  $n$ ).

Если при реализации ССВК по методу вычисления контрольных разрядов возможно ее построение только одним вариантом для выбранного на этапе проектирования  $(m, k)$ -кода (при этом тестер и блок контрольной логики могут быть синтезированы различными способами), то при использовании метода логического дополнения появляется возможность синтеза весьма большого числа структур ССВК (это число определяется возможностями доопределения значений контрольных функций). Данная особенность метода логического дополнения позволяет разработчику ССВК варьировать характеристики конечного устройства и оптимизировать их структуры по различным критериям, например по минимуму сложности реализации ССВК (по структурной избыточности), по максимуму быстродействия ССВК и т. д.

При синтезе ССВК определяющее значение имеют два фактора – число входов объекта диагностирования  $F(x)$  и число его выходов. Число входов устройства  $F(x)$  определяет число возможных кодовых комбинаций, формируемых на его выходах, и в общем случае (при подаче на входы всех возможных воздействий) определяется величиной  $2^t$ . Таким образом, при синтезе ССВК по методу вычисления контрольных разрядов потребуется минимизация функций максимум от  $t$  переменных, а при использовании метода логического дополнения – сначала дополнение значений контрольных функций на  $2^t$  входных наборах, а затем минимизация этих функций. Подобное обстоятельство свидетельствует о явных ограничениях на возможности использования обоих методов при доопределении значений контрольных функций: методы применимы при малом числе входных переменных. Оно ограничено вычислительной мощностью компьютерной техники, используемой при проектировании (как правило, число  $t$  не превышает 30–35 входов). При большем числе переменных следует использовать методы декомпозиции логических устройств и организации отдельных подсхем контроля с объединением их выходов на входах самопроверяемого компаратора либо ограничиваться методом дублирования, не требующим анализа структуры блока  $F(x)$ . Однако на практике дублирование может быть не всегда оправданным и несколько избыточным с позиции обнаружения возникающих ошибок. К примеру, для того чтобы на всех выходах блока  $F(x)$  одновременно возникло искажение, необходимо, чтобы в его структуре был хотя бы один логический элемент, связанный путями со всеми выходами, и чтобы хотя бы на одном наборе не происходили компенсации

выходных сигналов [6]. Еще одним недостатком метода дублирования может оказаться сложность и даже невозможность обеспечения полной самопроверяемости компаратора [1]. Таким образом, использование структурных схем, приведенных на рис. 1, в ряде случаев оправдано.

Остановимся на рассмотрении метода логического дополнения и использования при контроле логических устройств равновесных кодов «2 из 4» (2/4-кодов).

**Контроль логических устройств на основе 2/4-кода.** Так как тестеры равновесных кодов с небольшой длиной кодовых слов имеют наиболее простые структурные схемы, а для их полной проверки требуется малое количество кодовых комбинаций, именно такие коды оказываются эффективными с позиции структурной избыточности синтезируемых логических устройств и обнаружения ошибок на их выходах.

Для использования приведенной на рис. 2 схемы на практике выходы объекта диагностирования разбиваются на подмножества, содержащие по четыре выхода. При этом подмножества выходов могут пересекаться. Далее для каждого подмножества синтезируется своя схема контроля на основании структурной схемы на рис. 2, а выходы отдельных схем контроля объединяются на входах самопроверяемого компаратора, реализуемого в виде самопроверяемой схемы сжатия парафазных сигналов [30].

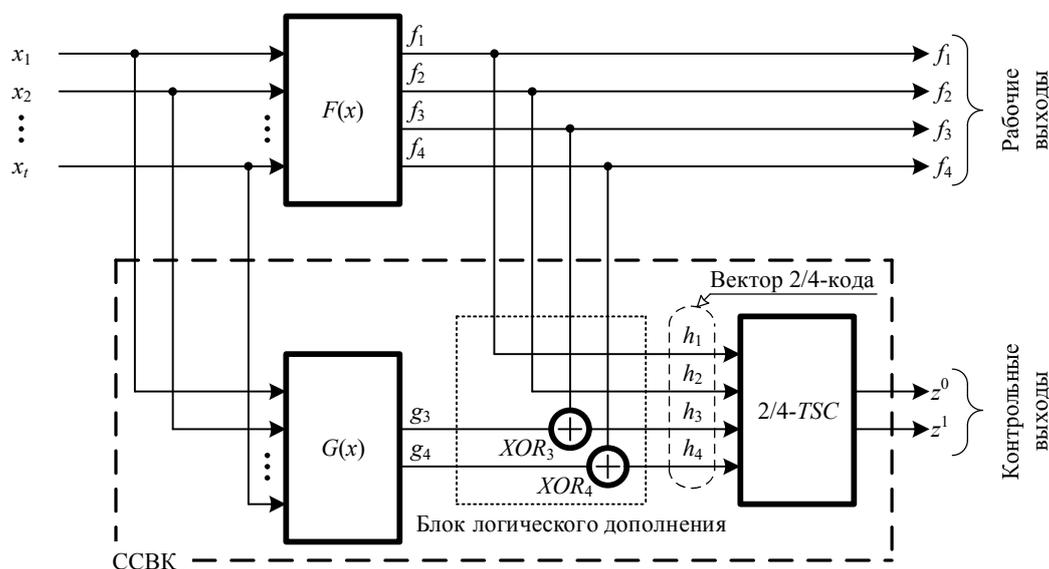


Рис. 2. Структурная схема организации контроля логического устройства на основе метода логического дополнения до равновесного кода «2 из 4»

Для преобразования любого информационного вектора  $\langle f_1 f_2 f_3 f_4 \rangle$  в кодовое слово 2/4-кода требуется изменение значений максимум двух рабочих функций. Данные преобразования реализуются в схеме контроля за счет использования блока логического дополнения, включающего в себя каскад элементов сложения по модулю два:

$$\begin{cases} h_1 = f_1, \\ h_2 = f_2, \\ h_3 = f_3 \oplus g_3, \\ h_4 = f_4 \oplus g_4. \end{cases} \quad (2)$$

На входы 2/4-TSC поступает кодовое слово  $\langle h_1 h_2 h_3 h_4 \rangle$ , принадлежащее 2/4-коду. Если в элементах ССВК и объекте диагностирования  $F(x)$  возникает неисправность, то хотя бы на одном входном наборе она должна проявиться в виде искажений сигналов рабочих или контрольных функций (или непосредственно разрядов кодового слова  $\langle h_1 h_2 h_3 h_4 \rangle$ ), что приводит

к появлению некодового слова на входах  $2/4-TSC$ . В этом случае на выходах последнего формируется непарафазный сигнал  $\langle 00 \rangle$  или  $\langle 11 \rangle$ , что свидетельствует о наличии неисправности.

В качестве преобразуемых функций могут выступать любые две рабочие функции. Это приводит к тому, что для каждого множества, состоящего из четырех выходов, возможна реализация ССВК  $C_4^2 = 6$  способами преобразования.

Для организации ССВК любого подмножества логического устройства, состоящего из четырех выходов, следует выполнить несколько условий:

1. Хотя бы по одному разу сформировать необходимые для полной проверки  $2/4-TSC$  комбинации  $2/4$ -кода. Для наиболее простого тестера (рис. 3) потребуется обеспечить формирование четырех кодовых комбинаций на его входах:  $\{0011; 0110; 1001; 1100\}$  [30].

2. Для элементов  $XOR$  необходимо обеспечить появление хотя бы по разу всех тестовых комбинаций:  $\{00; 01; 10; 11\}$  [31].

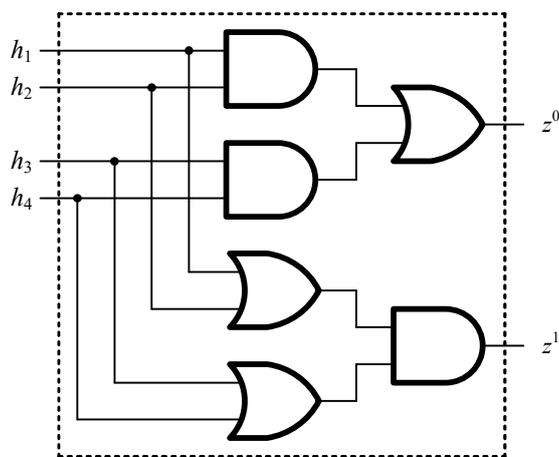


Рис. 3. Наиболее простая структурная схема  $2/4-TSC$

Важно отметить, что контрольные функции  $g_3$  и  $g_4$  должны быть наиболее простыми для сокращения сложности технической реализации блока контрольной логики. При этом необходимо обеспечить и сформулированные выше условия самопроверяемости структуры, приведенной на рис. 2.

Исследования показывают, что значения контрольных функций  $g_3$  и  $g_4$  на всех входных наборах могут быть доопределены произвольным образом, как это предложено в [26], или же могут быть выработаны специальные правила по «жесткому» доопределению значений, как это сделано в [27]. Возможности выбора двух доопределяемых функций из четырех в множестве выходов, выбора самих подмножеств выходов (при их числе, большем четырех), а также произвольного доопределения значений контрольных функций позволяют при организации ССВК варьировать показатели структурной избыточности конечного устройства. Таким образом, при реализации ССВК можно получать такие выражения для контрольных функций, которые будут давать наиболее простой блок  $G(x)$  в выбранном элементном базисе.

**Способ доопределения значений контрольных функций при синтезе ССВК.** Как отмечалось выше, для преобразования любого информационного вектора  $\langle f_1 f_2 f_3 f_4 \rangle$  в кодовое слово  $2/4$ -кода потребуется дополнение максимум двух функций. Поэтому значения двух из четырех рабочих функций могут быть напрямую поданы на входы  $2/4-TSC$   $h_1$  и  $h_2$ . На две преобразуемые функции при этом накладываются определенные ограничения: должны хотя бы по разу быть сформированы все возможные сочетания их значений  $\langle f_1 f_2 \rangle = \langle 00 \rangle$ ,  $\langle 01 \rangle$ ,  $\langle 10 \rangle$  и  $\langle 11 \rangle$ . Это следует из анализа множества тестовых комбинаций  $2/4-TSC$ .

В табл. 1 задано комбинационное логическое устройство, имеющее четыре входа и четыре выхода. Предположим, что  $h_1=f_1$  и  $h_2=f_2$ . Тогда при значениях подвекторов  $\langle h_1 h_2 \rangle = \langle 00 \rangle$  или  $\langle h_1 h_2 \rangle = \langle 11 \rangle$  значения подвекторов  $\langle h_3 h_4 \rangle$  кодового слова  $\langle h_1 h_2 h_3 h_4 \rangle$  определяются

«автоматически», исходя из того, что конечный вектор должен принадлежать 2/4-коду. Для подвекторов  $\langle h_1 h_2 \rangle = \langle 01 \rangle$  и  $\langle h_1 h_2 \rangle = \langle 10 \rangle$  заполнение столбцов  $h_3$  и  $h_4$  может быть произвольным: либо  $\langle h_3 h_4 \rangle = \langle 01 \rangle$ , либо  $\langle h_3 h_4 \rangle = \langle 10 \rangle$  (в таком случае кодовый вектор  $\langle h_1 h_2 h_3 h_4 \rangle$  будет принадлежать 2/4-коду). Это определяет возможности произвольного дополнения функций  $h_3$  и  $h_4$  и возможности влияния тем самым на вид контрольных функций  $g_3$  и  $g_4$ .

Таблица 1

Таблица истинности исходного комбинационного устройства и «автоматически» определяемые значения контрольных функций

$x_1$	$x_2$	$x_3$	$x_4$	$f_1$	$f_2$	$f_3$	$f_4$	$h_1$	$h_2$	$h_3$	$h_4$	$g_3$	$g_4$
0	0	0	0	0	1	1	0	0	1				
0	0	0	1	0	0	0	1	0	0	1	1	1	0
0	0	1	0	0	1	0	1	0	1				
0	0	1	1	1	0	1	1	1	0				
0	1	0	0	0	0	1	0	0	0	1	1	0	1
0	1	0	1	0	0	0	0	0	0	1	1	1	1
0	1	1	0	0	1	1	0	0	1				
0	1	1	1	0	0	1	0	0	0	1	1	0	1
1	0	0	0	1	0	1	1	1	0				
1	0	0	1	1	1	0	1	1	1	0	0	0	1
1	0	1	0	0	0	0	0	0	0	1	1	1	1
1	0	1	1	1	0	0	1	1	0				
1	1	0	0	0	1	1	1	0	1				
1	1	0	1	0	0	0	0	0	0	1	1	1	1
1	1	1	0	0	0	1	1	0	0	1	1	0	0
1	1	1	1	0	0	1	0	0	0	1	1	0	1

Вариант с произвольным дополнением функций описан, к примеру, в работах [4, 21, 24] для использования при синтезе ССВК 1/4-кода. Аналогично в [25] предлагается для оптимизации сложности ССВК подбирать значения функций дополнения при формировании кодовых слов 1/3-кода. Такой же подход транслирован для 2/4-кода в [26].

Альтернативный вариант вычисления функций логического дополнения – формирование их по заранее установленным правилам. Системы контрольных функций, обеспечивающие такое дополнение, приведены в [4, 21–23] для применения 1/3- и 1/4-кодов и в [27] – для применения 2/4-кода. Недостатком «жесткой» фиксации правил формирования значений контрольных функций является необходимость подачи на входы устройства определенного множества входных комбинаций для полной проверки ССВК, а также невозможность во всех случаях для любых структур блока  $F(x)$  сформировать контрольное множество комбинаций для полной проверки ССВК.

Способ доопределения логических функций, описанный в работе [26], не учитывает возможности подачи на входы 2/4-TSC (см. рис. 3) комбинаций  $\langle 0101 \rangle$  и  $\langle 1010 \rangle$ , а также связан с одновременным доопределением значений функций  $g_3$  и  $g_4$ . Представленный далее способ доопределения значений контрольных функций основан на последовательном заполнении значений контрольных функций и не исключает возможности использования кодовых комбинаций  $\langle 0101 \rangle$  и  $\langle 1010 \rangle$  в векторе  $\langle h_1 h_2 h_3 h_4 \rangle$ . Это позволяет для реального множества информационных векторов, которое может содержать только часть возможных информационных векторов, а не все (как в общем случае в [27]), обеспечить тестируемость элементов XOR в блоке контрольной логики, а также выбрать способ логического дополнения, дающий наиболее простой по структуре блок  $G(x)$ .

Заполнение характеристической таблицы (аналогичной табл. 1) может вестись в следующей последовательности:

1. Для множества из четырех выходов логического устройства определяются такие два выхода, для которых как минимум по одному разу формируются все возможные сочетания значений  $\langle f_1 f_2 \rangle = \{00; 01; 10; 11\}$ . Эти выходы не будут преобразовываться в блоке логического дополнения.

2. Заполняются столбцы значений функций  $h_1$  и  $h_2$ , а затем производится однозначное доопределение значений функций  $h_3$  и  $h_4$  на тех входных наборах, для которых  $\langle h_1 h_2 \rangle = \langle 01 \rangle$  и  $\langle h_1 h_2 \rangle = \langle 10 \rangle$ . Это позволяет также однозначно доопределить на данных наборах значения функций  $g_3$  и  $g_4$ . Кроме того, указанная операция позволяет сформировать на входах  $2/4-TSC$  две тестовые комбинации из четырех:  $\langle 0011 \rangle$  и  $\langle 1100 \rangle$ .

3. Осуществляется проверка формирования тестовых комбинаций для элементов  $XOR_3$  и  $XOR_4$  на входных наборах, на которых однозначно доопределены значения функций  $g_3$  и  $g_4$ , – фиксация комбинаций из множеств  $M_3 = \{00; 01; 10; 11\}$  и  $M_4 = \{00; 01; 10; 11\}$ . Для каждого из элементов  $XOR_3$  и  $XOR_4$  как минимум по одной комбинации из множеств  $M_3$  и  $M_4$  должны сформироваться на однозначно доопределенных значениях на предыдущем шаге.

4. Рассматривается задача доопределения значений контрольной функции  $g_3$  и значений функции  $h_3$  с учетом обеспечения формирования тестовых комбинаций из множества  $M_3$ . Из множества  $M_3$  исключаются уже сформированные тестовые комбинации и осуществляется поиск таких значений  $f_3$ , которые позволяют доопределить функцию  $g_3$  с учетом формирования необходимых тестовых комбинаций. Для этого потребуется проанализировать возможности доопределения функции  $g_3$  максимум на четырех входных наборах. На двух из данных наборов  $f_3=0$  и на двух  $f_3=1$ . Заполняются значения функций  $g_3$ , и однозначно определяются значения функции  $h_3$  (см. формулу (2)).

5. На каждом из наборов, которые позволяют доопределить значение функции  $g_3$  на предыдущем шаге, однозначно доопределяются значения функции  $h_4$  и, соответственно, функции  $g_4$ . Проверяется наличие на всех входных наборах, на которых доопределены все функции, тестовых комбинаций из множества  $M_4$ , а также двух оставшихся комбинаций для проверки  $2/4-TSC$  – комбинаций  $\langle 1001 \rangle$  и  $\langle 0110 \rangle$ . Следует отметить, что их может и не оказаться (если были сформированы векторы  $\langle h_1 h_2 h_3 h_4 \rangle = \langle 1010 \rangle$  или  $\langle 0101 \rangle$ ).

6. Рассматривается задача доопределения значений контрольной функции  $g_4$  и значений  $h_4$  с учетом обеспечения формирования тестовых комбинаций из множества  $M_4$ . Из множества  $M_4$  удаляются уже сформированные тестовые комбинации и осуществляется поиск таких значений  $f_4$ , которые позволяют доопределить функцию  $g_4$  с учетом формирования необходимых тестовых комбинаций. Для этого потребуется проанализировать возможности доопределения функции  $g_4$  максимум на четырех входных наборах. На двух из данных наборов  $f_4=0$  и на двух  $f_4=1$ . Заполняются значения функции  $g_4$ , и однозначно определяются значения функции  $h_4$  (см. формулу (2)).

7. Аналогично п. 5 однозначно заполняются строки таблицы, на которых доопределены значения функции  $g_4$ . Проверяется наличие тестовых комбинаций  $\langle 1001 \rangle$  и  $\langle 0110 \rangle$  для тестера.

8. Заполняются оставшиеся строки таблицы с учетом условий формирования всех тестовых комбинаций для  $2/4-TSC$ . На этом этапе (а также на этапах 4 и 6) целесообразно учитывать возможности доопределения значений функций  $g_3$  и  $g_4$  таким образом, чтобы минимизировать их логические выражения (операция перебора с поиском соседних конъюнкций).

9. Если при доопределении функций  $g_3$  или  $g_4$  не удастся получить все тестовые комбинации для элементов  $XOR_3$  и  $XOR_4$ , то функции могут быть поменяны местами при доопределении. Если же эта процедура не дает результата, могут быть доопределены другие две функции из четырех. Если при полном переборе всех вариантов не удастся достичь результата, изменится само множество из четырех функций, реализуемых логическим устройством, или же выбирается другой способ синтеза ССВК.

Вернемся к примеру логического устройства, заданного в табличной форме (см. табл. 1). В данной таблице приведены реализуемые логическим устройством функции и перечислены все информационные векторы  $\langle f_1 f_2 f_3 f_4 \rangle$ , формируемые на всех входных комбинациях. Далее, в столбцах  $h_1, h_2, h_3, h_4$  и  $g_3, g_4$  представлены однозначно определяемые значения, полученные «автоматически».

На следующем шаге после автоматического заполнения столбцов  $g_3$  и  $g_4$  проверим формирование контрольного множества комбинаций  $\{00; 01; 10; 11\}$  для элементов  $XOR_3$  и  $XOR_4$ .

Для табл. 1 проверка формирования тестовых комбинаций дает следующий результат:

$$\begin{aligned}
 p_{XOR_3}^{00} &= \overline{f_3 g_3} = \overline{x_1 x_2 x_3 x_4}, \\
 p_{XOR_3}^{01} &= \overline{f_3 g_3} = \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4}, \\
 p_{XOR_3}^{10} &= \overline{f_3 g_3} = \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4}, \\
 p_{XOR_3}^{11} &= f_3 g_3 = 0, \quad p_{XOR_4}^{00} = \overline{f_4 g_4} = 0, \\
 p_{XOR_4}^{01} &= \overline{f_4 g_4} = \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4}, \\
 p_{XOR_4}^{10} &= \overline{f_4 g_4} = \overline{x_1 x_2 x_3 x_4}, \quad p_{XOR_4}^{11} = f_4 g_4 = \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4}.
 \end{aligned}$$

Так как  $p_{XOR_3}^{11} = f_3 g_3 = 0$  и  $p_{XOR_4}^{00} = \overline{f_4 g_4} = 0$ , необходимо так доопределить значения оставшихся функций в табл. 1, чтобы хотя бы по разу были сформированы комбинации  $\langle f_3 g_3 \rangle = \langle 11 \rangle$  и  $\langle f_4 g_4 \rangle = \langle 00 \rangle$ . Для удовлетворения этого условия следует найти те строки табл. 1, для которых в столбце  $f_3$  будет записана 1, а в столбце  $f_4$  – 0. На данных входных наборах необходимо хотя бы по разу обеспечить формирование «недостающей» тестовой комбинации.

В табл. 2 приведены варианты доопределения разрядов вектора  $\langle h_1 h_2 h_3 h_4 \rangle$  с учетом необходимости формирования тестовых комбинаций элементов  $XOR_3$  и  $XOR_4$ . Знаком «+» отмечены те строки, доопределение которых единицами обеспечивает формирование тестовой комбинации  $\langle 11 \rangle$  на входах элемента  $XOR_3$ . Знаком «-» отмечены строки, доопределение которых нулями обеспечивает формирование тестовой комбинации  $\langle 00 \rangle$  на входах элемента  $XOR_4$ . Для формирования полного множества тестовых комбинаций для обоих элементов сложения по модулю два потребуется доопределить хотя бы одну клетку со знаком «+» до единицы и хотя бы одну клетку со знаком «-» до нуля. Ячейки в одной строке должны быть заполнены противоположными значениями, так как только в этом случае кодовый вектор  $\langle h_1 h_2 h_3 h_4 \rangle$  будет принадлежать 2/4-коду. Далее значения функций  $h_3$  и  $h_4$  заполняются однозначно.

Таблица 2

Расширенная таблица с учетом необходимости тестирования  $XOR_3$  и  $XOR_4$ 

$x_1$	$x_2$	$x_3$	$x_4$	$f_1$	$f_2$	$f_3$	$f_4$	$h_1$	$h_2$	$h_3$	$h_4$	$g_3$	$g_4$
0	0	0	0	0	1	1	0	0	1	+	-		
0	0	0	1	0	0	0	1	0	0	1	1	1	0
0	0	1	0	0	1	0	1	0	1				
0	0	1	1	1	0	1	1	1	0	+			
0	1	0	0	0	0	1	0	0	0	1	1	0	1
0	1	0	1	0	0	0	0	0	0	1	1	1	1
0	1	1	0	0	1	1	0	0	1	+	-		
0	1	1	1	0	0	1	0	0	0	1	1	0	1
1	0	0	0	1	0	1	1	1	0	+			
1	0	0	1	1	1	0	1	1	1	0	0	0	1
1	0	1	0	0	0	0	0	0	0	1	1	1	1
1	0	1	1	1	0	0	1	1	0				
1	1	0	0	0	1	1	1	0	1	+			
1	1	0	1	0	0	0	0	0	0	1	1	1	1
1	1	1	0	0	0	1	1	0	0	1	1	0	0
1	1	1	1	0	0	1	0	0	0	1	1	0	1

В табл. 3 представлен один из вариантов доопределения разрядов вектора  $\langle h_1 h_2 h_3 h_4 \rangle$ .

Таблица 3

Полностью определенные контрольные функции

$x_1$	$x_2$	$x_3$	$x_4$	$f_1$	$f_2$	$f_3$	$f_4$	$h_1$	$h_2$	$h_3$	$h_4$	$g_3$	$g_4$
0	0	0	0	0	1	1	0	0	1	1	0	0	1
0	0	0	1	0	0	0	1	0	0	1	1	1	0
0	0	1	0	0	1	0	1	0	1	0	1	0	0
0	0	1	1	1	0	1	1	1	0	0	1	1	0
0	1	0	0	0	0	1	0	0	0	1	1	0	1
0	1	0	1	0	0	0	0	0	0	1	1	1	1
0	1	1	0	0	1	1	0	0	1	1	0	0	0
0	1	1	1	0	0	1	0	0	0	1	1	0	1
1	0	0	0	1	0	1	1	1	0	1	0	0	1
1	0	0	1	1	1	0	1	1	1	0	0	0	1
1	0	1	0	0	0	0	0	0	0	1	1	1	1
1	0	1	1	1	0	0	1	1	0	1	0	1	1
1	1	0	0	0	1	1	1	0	1	0	1	1	0
1	1	0	1	0	0	0	0	0	0	1	1	1	1
1	1	1	0	0	0	1	1	0	0	1	1	0	0
1	1	1	1	0	0	1	0	0	0	1	1	0	1

Минимизируя полученные таким образом контрольные функции  $g_3$  и  $g_4$  (рис. 4), записываем выражения для полностью определенных контрольных функций, по которым синтезируется блок  $G(x)$ :

$$g_3 = x_1 x_2 x_3 \vee x_2 x_3 x_4 \vee x_1 x_2 x_4 \vee x_1 x_2 x_3,$$

$$g_4 = x_2 x_4 \vee x_1 x_2 \vee x_1 x_3 x_4.$$

Для упрощения структуры блока  $G(x)$  может быть использована скобочная форма записи функций  $g_3$  и  $g_4$ . Однако функции должны реализовываться отдельно.

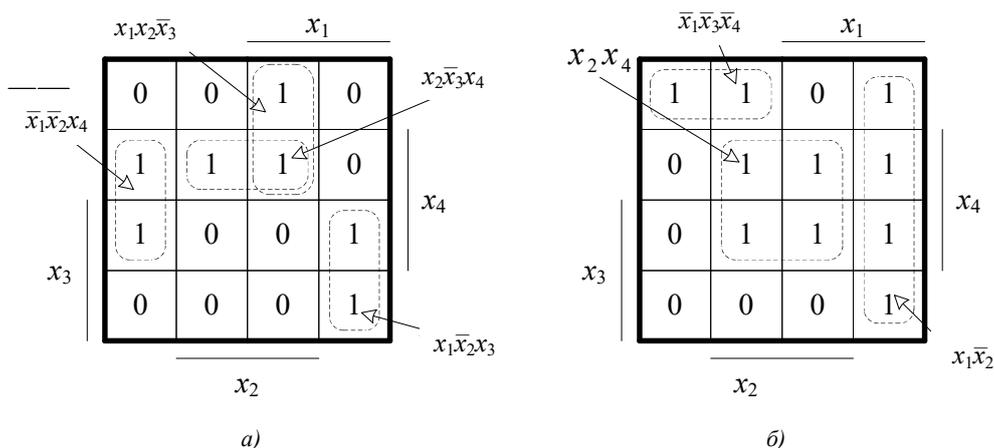


Рис. 4. Полностью определенные функции  $g_3$  (а) и  $g_4$  (б)

Алгоритм доопределения значений контрольных функций дает возможность построения ССВК для логических устройств. При этом чем больше входов у объекта диагностирования и чем чаще меняются значения в выходных векторах, тем проще обеспечить формирование

контрольного множества для блока логического дополнения и тестера в структуре ССВК. Если на каком-то этапе выполнения вычислительных процедур не удастся обеспечить формирование множества тестовых комбинаций для элемента сложения по модулю два, можно вернуться на шаг назад и выбрать другие неизменяемые функции, реализуемые блоком  $F(x)$ . Если блок  $F(x)$  многовыходной, можно для обеспечения тестируемости компонентов ССВК произвести замену функций в подмножествах контролируемых выходов и т. д. Конечный вариант реализации, если не удастся обеспечить самопроверяемость структуры схемы контроля, – это выбор другого способа преобразования или, вообще, другого метода синтеза ССВК.

Приведем пример, демонстрирующий возможности предложенного способа синтеза ССВК. В табл. 4 даны результаты экспериментов по оценке сложности технической реализации ССВК, выполненных по представленному в настоящей статье методу, для одной контрольной комбинационной схемы с четырьмя выходами – схемы «b1» из набора LGSynth'89 [32]. Сама схема в данном наборе представлена в формате \*.netblif, задающем ее структуру в виде списка логических элементов и конфигураций связей между входами и выходами схемы, а также входами и выходами внутренних логических элементов. Выбирая преобразуемые поочередно все возможные пары выходов схемы «b1», получаем шесть вариантов реализации ССВК (в каждом из них возможно, однако, различное доопределение контрольных функций, поэтому в реальности вариантов намного больше). При постановке экспериментов было принято решение выбрать произвольный вариант доопределения функций по представленному в данной работе способу. С использованием известного интерпретатора SIS [32] для каждой ССВК и ее компонентов был определен абсолютный показатель сложности технической реализации в условных единицах площади, занимаемой устройством на кристалле. В эксперименте использовалась стандартная библиотека функциональных элементов stdcell2\_2.genlib. Получаемые характеристики ССВК на основе 2/4-кода сравнивались с характеристиками сложности технической реализации схемы контроля по методу дублирования. В предпоследнем столбце табл. 4 рассчитан показатель  $\mu$ , характеризующий долю площади, занимаемой ССВК на основе 2/4-кода, по сравнению с площадью схемы контроля, реализованной по методу дублирования. Во всех случаях наблюдается существенное уменьшение площади ССВК, выполненной по предложенному методу, по сравнению с площадью системы дублирования. При этом в ряде случаев не удастся обеспечить формирование полного множества тестовых комбинаций для 2/4-TSC.

Таблица 4

Результаты эксперимента с контрольной комбинационной схемой «b1»

Преобразуемые функции	Площади компонентов ССВК, у. е.				Площади конечных устройств, у. е.		$\mu$ , %	Тестируемость компонентов ССВК на основе 2/4-кода
	$F(x)$	$G(x)$	Блок логического дополнения	2/4-TSC	Система дублирования	ССВК на основе 2/4-кода		
$f_3, f_4$	224	192	80	192	1088	688	63,235	Тестируются все компоненты
$f_2, f_4$		144				640	58,824	Тестируются все компоненты
$f_1, f_4$		184				680	62,5	Не формируется комбинация <1100> для 2/4-TSC
$f_2, f_3$		144				640	58,824	Не формируются комбинации <1100> и <0011> для 2/4-TSC
$f_1, f_3$		184				680	62,5	Тестируются все компоненты
$f_1, f_2$		184				680	62,5	Тестируются все компоненты

Проведенные исследования показывают возможности представленного способа синтеза ССВК логических устройств: существует большое количество вариантов реализации ССВК для одного и того же объекта диагностирования, что позволяет выбрать вариант с наименьшей сложностью технической реализации при обеспечении тестирования всех компонентов конечного устройства.

Следует отметить, что результаты экспериментов демонстрируют только частный случай, но тем не менее сама процедура доопределения значений контрольных функций вариативна и дает гибкость выбора конечного способа реализации ССВК с учетом всех предъявляемых к ней требований. Сам алгоритм имеет экспоненциальную сложность: с увеличением числа входных переменных существенно увеличивается число операций по доопределению значений контрольных функций. Это ограничивает применение данного способа синтеза ССВК небольшим количеством входных переменных объекта диагностирования (до 30–35).

**Заключение.** При синтезе ССВК для логических устройств автоматики и вычислительной техники эффективным может оказаться использование метода логического дополнения и применения в качестве «основы схемы контроля» равновесного кода «2 из 4». В отличие от представленных ранее результатов исследования применения данного кода в задачах синтеза ССВК описанный в настоящей статье способ позволяет за счет использования при дополнении не-тестовых комбинаций  $2/4-TSC$   $\langle 0101 \rangle$  и  $\langle 1010 \rangle$  влиять на сложность блока контрольной логики. Это также упрощает процедуру подбора значений при тестировании элементов сложения по модулю два в блоке контрольной логики по сравнению с иными способами логического дополнения.

Наиболее эффективным способ синтеза ССВК может оказаться для многовыходных логических схем, на выходах которых при подаче входных воздействий часто изменяются значения рабочих функций.

#### Список использованных источников

1. Согомоян, Е. С. Самопроверяемые устройства и отказоустойчивые системы / Е. С. Согомоян, Е. В. Слабаков. – М.: Радио и связь, 1989. – 208 с.
2. Nicolaidis, M. On-line testing for VLSI: state of the art and trends / M. Nicolaidis // *Integration, the VLSI Journal*. – 1998. – Vol. 26, iss. 1–2. – P. 197–209.
3. Lala, P. K. Self-Checking and Fault-Tolerant Digital Design / P. K. Lala. – San Francisco: Morgan Kaufmann Publishers, 2001. – 216 p.
4. New Methods of Concurrent Checking / M. Goessel [et al.]. – Ed. 1. – Dordrecht: Springer Science+Business Media B.V., 2008. – 184 p.
5. Goessel, M. Error Detection Circuits / M. Goessel, S. Graf. – London: McGraw-Hill, 1994. – 261 p.
6. Новые структуры систем функционального контроля логических схем / В. В. Сапожников [и др.] // *Автоматика и телемеханика*. – 2017. – № 2. – С. 127–143.
7. Borecký, J. Parity driven reconfigurable duplex system / J. Borecký, M. Kohlík, H. Kubátová // *Microprocessors and Microsystems*. – 2017. – Vol. 52. – P. 251–260.
8. Self-checking comparator with one periodic output / S. Kundu [et al.] // *IEEE Transactions on Computers*. – 1996. – Vol. 45, iss. 3. – P. 379–380.
9. Metra, C. Highly testable and compact single output comparator / C. Metra, M. Favalli, B. Ricco // *Proc. of 15th IEEE VLSI Test Symp.*, 27 Apr. – 1 May 1997, Monterey, CA, USA. – Monterey, 1997. – P. 210–215.
10. Методы построения безопасных микроэлектронных систем железнодорожной автоматики / В. В. Сапожников [и др.]; под ред. Вл. В. Сапожникова. – М.: Транспорт, 1995. – 272 с.
11. Kubalík, P. Fault tolerant system design method based on self-checking circuits / P. Kubalík, P. Fišer, H. Kubátová // *Proc. of 12th Intern. On-Line Testing Symp. 2006 (IOLTS'06)*. – Como, Italy, 2006. – P. 185–186.
12. Theeg, G. Railway Signalling & Interlocking – International Compendium / G. Theeg, S. Vlasenko. – Eurailpress, 2009. – 448 p.
13. Исследование вероятностных методов оценки логической уязвимости комбинационных схем / А. Л. Степковский [и др.] // *Проблемы разработки перспективных микро- и нанозлектронных систем (МЭС)*. – 2016. – № 4. – С. 121–126.
14. Piestrak, S. J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes / S. J. Piestrak. – Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. – 111 p.
15. Nicolaidis, M. On-line testing for VLSI – a compendium of approaches / M. Nicolaidis, Y. Zorian // *Journal of Electronic Testing: Theory and Applications*. – 1998. – Vol. 12, iss. 1–2. – P. 7–20.
16. Das, D. Synthesis of circuits with low-cost concurrent error detection based on bose-lin codes / D. Das, N. A. Touba // *Journal of Electronic Testing: Theory and Applications*. – 1999. – Vol. 15, iss. 1–2. – P. 145–155.

17. Low cost concurrent error detection based on modulo weight-based codes / D. Das [et al.] // Proc. of IEEE 6th Intern. On-Line Testing Workshop (IOLTW), Spain, Palma de Mallorca, 3–5 July 2000. – Palma de Mallorca, 2000. – P. 171–176.
18. Busaba, F. Y. Self-checking combinational circuit design for single and unidirectional multibit errors / F. Y. Busaba, P. K. Lala // Journal of Electronic Testing: Theory and Applications. – 1994. – Vol. 5, iss. 1. – P. 19–28.
19. A new design method for self-checking unidirectional combinational circuits / V. V. Sapozhnikov [et al.] // Journal of Electronic Testing: Theory and Applications. – 1998. – Vol. 12, iss. 1–2. – P. 41–53.
20. Self-dual parity checking – a new method for on line testing / V. V. Sapozhnikov [et al.] // Proc. of 14th IEEE VLSI Test Symp., 28 April – 1 May 1996, Princeton, NJ, USA. – Princeton, 1996. – P. 162–168.
21. Организация функционального контроля комбинационных схем методом логического дополнения / В. В. Сапожников [и др.] // Электронное моделирование. – 2002. – Т. 24, № 6. – С. 52–66.
22. Логическое дополнение – новый метод контроля комбинационных схем / М. Гессель [и др.] // Автоматика и телемеханика. – 2003. – № 1. – С. 167–176.
23. Контроль комбинационных схем методом логического дополнения / М. Гессель [и др.] // Автоматика и телемеханика. – 2005. – № 8. – С. 161–172.
24. Sen, S. K. A Self-checking circuit for concurrent checking by 1-out-of-4 code with design optimization using constraint don't cares / S. K. Sen // National Conf. on Emerging Trends and Advances in Electrical Engineering and Renewable Energy (NCEEERE 2010) / Sikkim Manipal Institute of Technology. – Sikkim, 2010.
25. Constraint don't cares for optimizing designs for concurrent checking by 1-out-of-3 codes / D. K. Das [et al.] // Proc. of the 10th Intern. Workshops on Boolean Problems, Freiberg, Germany, Sept. 2012. – Freiberg, 2012. – P. 33–40.
26. Sapozhnikov, V. Concurrent error detection of combinational circuits by the method of Boolean complement on the base of "2-out-of-4" code / V. Sapozhnikov, V. Sapozhnikov, D. Efanov // Proc. of 14th IEEE East-West Design & Test Symp. (EWDTS'2016), Yerevan, Armenia, 14–17 Oct. 2016. – Yerevan, 2016. – P. 126–133.
27. Сапожников, В. В. Построение самопроверяемых структур систем функционального контроля на основе равновесного кода «2 из 4» / В. В. Сапожников, В. В. Сапожников, Д. В. Ефанов // Проблемы управления. – 2017. – № 1. – С. 57–64.
28. Tarnick, S. Design of embedded m-out-of-n code checkers using complete parallel counters / S. Tarnick // 13<sup>th</sup> IEEE Intern. On-Line Testing Symp. (IOLTS 2007), 8–11 July 2007, Crete, Greece. – Crete, 2007. – P. 285–292.
29. Piestrak, S. Design of minimal-level PLA self-testing checkers for m-out-of-n codes / S. Piestrak // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. – 1996. – Vol. 4, iss. 2. – P. 264–272.
30. Сапожников, В. В. Самопроверяемые дискретные устройства / В. В. Сапожников, В. В. Сапожников. – СПб.: Энергоатомиздат, 1992. – 224 с.
31. Аксенова, Г. П. Необходимые и достаточные условия построения полностью проверяемых схем свертки по модулю 2 / Г. П. Аксенова // Автоматика и телемеханика. – 1979. – № 9. – С. 126–135.
32. Collection of Digital Design Benchmarks [Electronic resource]. – Mode of access: <http://ddd.fit.cvut.cz/prj/Benchmarks/>. – Date of access: 20.02.2018.
33. Sequential circuit design using synthesis and optimization / E. M. Sentovich [et al.] // Proc. IEEE Intern. Conf. on Computer Design: VLSI in Computers & Processors, 11–14 October 1992, Cambridge, MA, USA. – Cambridge, 1992. – P. 328–333.

## References

1. Sogomonyan E. S., Slabakov E. V. Samoproveryaemye ustrojstva i otkazoustojchivye sistemy. *Self-Checking Devices and Fault-Tolerance Systems*. Moscow, Radio i svyaz' Publ., 1989, 208 p. (in Russian).
2. Nicolaidis M. On-line testing for VLSI: state of the art and trends. *Integration, the VLSI Journal*, 1998, vol. 26, iss. 1–2, pp. 197–209. doi: 10.1016/S0167-9260(98)00028-5
3. Lala P. K. *Self-Checking and Fault-Tolerant Digital Design*. San Francisco, Morgan Kaufmann Publishers, 2001, 216 p.
4. Goessel M., Ocheretny V., Sogomonyan E., Marienfeld D. *New Methods of Concurrent Checking*. Dordrecht, Springer Science+Business Media B.V., 2008, 184 p.
5. Goessel M., Graf S. *Error Detection Circuits*. London, McGraw-Hill, 1994, 261 p.
6. Sapozhnikov V. V., Sapozhnikov V. V., Efanov D. V., Dmitriev V. V. Novye struktury sistem funkcional'nogo kontrolya logicheskikh skhem [New structures of the concurrent error detection systems for logic circuits]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 2017, no. 2, pp. 127–143 (in Russian).
7. Borecký J., Kohlík M., Kubátová H. Parity driven reconfigurable duplex system. *Microprocessors and Microsystems*, 2017, vol. 52, pp. 251–260. doi: 10.1016/j.micpro.2017.06.015
8. Kundu S., Sogomonyan E. S., Goessel M., Tarnick S. Self-checking comparator with one periodic output. *IEEE Transactions on Computers*, 1996, vol. 45, iss. 3, pp. 379–380. doi: 10.1109/12.485577
9. Metra C., Favalli M., Ricco B. Highly testable and compact single output comparator. *Proceedings of 15th IEEE VLSI Test Symposium, 27 April–1 May 1997, Monterey, CA, USA*, pp. 210–215. doi: 10.1109/VTEST.1997.600272

10. Sapozhnikov V. V., Sapozhnikov Vl. V., Hristov H. A., Gavzov D. V. Metody postroeniya bezopasnykh mikroelektronnykh sistem zheleznodorozhnoj avtomatiki. *Methods of Synthesis Safety Microelectronic Railway Automation Systems*. Moscow, Transport Publ., 1995, 272 p. (in Russian).
11. Kubalík P., Fišer P., Kubátová H. Fault tolerant system design method based on self-checking circuits. *Proceedings of 12th International On-Line Testing Symposium 2006 (IOLTS'06)*. Como, Italy, 2006, pp. 185–186.
12. Theeg G., Vlasenko S. *Railway Signalling & Interlocking – International Compendium*. Eurailpress, 2009, 448 p.
13. Stempkovsky A. L., Telpukhov D. V., Solovjov R. A., Telpukhova N. V. Issledovanie veroyatnostnykh metodov ocenki logicheskoy uyazvimosti kombinatsionnykh skhem [Probabilistic methods for reliability evaluation of combinational circuits]. Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES) [*Problems of Development of Perspective Micro- and Nanoelectronic Systems (MES)*], 2016, no. 4, pp. 121–126 (in Russian).
14. Piestrak S. J. *Design of Self-Testing Checkers for Unidirectional Error Detecting Codes*. Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 1995, 111 p.
15. Nicolaidis M., Zorian Y. On-line testing for VLSI – a compendium of approaches. *Journal of Electronic Testing: Theory and Applications*, 1998, vol. 12, iss. 1–2, pp. 7–20. doi: 10.1023/A:1008244815697
16. Das D., Touba N. A. Synthesis of circuits with low-cost concurrent error detection based on bose-lin codes. *Journal of Electronic Testing: Theory and Applications*, 1999, vol. 15, iss. 1–2, pp. 145–155. doi: 10.1023/A:1008344603814
17. 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, 3–5 July 2000*. Palma de Mallorca, 2000, pp. 171–176. doi: 10.1109/OLT.2000.856633
18. 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, vol. 5, iss. 1, pp. 19–28. doi: 10.1007/BF00971960
19. Sapozhnikov V. V., Morosov A. V., Sapozhnikov Vl. V., Gessel' M. A New design method for self-checking unidirectional combinational circuits. *Journal of Electronic Testing: Theory and Applications*, 1998, vol. 12, iss. 1–2, pp. 41–53. doi: 10.1023/A:1008257118423
20. Sapozhnikov Vl. V., Dmitriev A. V., Gessel' M., Sapozhnikov V. V. Self-dual parity checking – a new method for on line testing. *Proceedings of 14th IEEE VLSI Test Symposium, 28 Apr. – 1 May 1996, Princeton, NJ, USA*. Princeton, 1996, pp. 162–168. doi: 10.1109/VTEST.1996.510852
21. Sapozhnikov V. V., Sapozhnikov Vl. V., Dmitriev A. V., Morozov A. V., Gessel' M. Organizatsiya funktsional'nogo kontrolya kombinatsionnykh skhem metodom logicheskogo dopolneniya [The organization of functional control of combinational circuits by the method of complement]. *Elektronnoe modelirovanie [Electronic Modeling]*, 2002, vol. 24, no. 6, pp. 52–66 (in Russian).
22. Gessel' M., Morozov A. V., Sapozhnikov V. V., Sapozhnikov Vl. V. Logicheskoe dopolnenie – novyy metod kontrolya kombinatsionnykh skhem [Logic complement, a new method of checking the combinational circuits]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 2003, no. 1, pp. 167–176 (in Russian).
23. Gessel' M., Morozov A. V., Sapozhnikov V. V., Sapozhnikov Vl. V. Kontrol' kombinatsionnykh skhem metodom logicheskogo dopolneniya [Checking combinational circuits by the method of logic complement]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 2005, no. 8, pp. 161–172 (in Russian).
24. Sen S. K. A Self-checking circuit for concurrent checking by 1-out-of-4 code with design optimization using constraint don't cares. *National Conference on Emerging Trends and Advances in Electrical Engineering and Renewable Energy (NCEEERE 2010), Sikkim Manipal Institute of Technology*. Sikkim, 2010.
25. Das D. K., Roy S. S., Dmitriev A. V., Morozov A. V., Gessel' M. Constraint don't cares for optimizing designs for concurrent checking by 1-out-of-3 codes. *Proceedings of the 10th International Workshops on Boolean Problems, Freiberg, Germany, September 2012*. Freiberg, 2012, pp. 33–40.
26. Sapozhnikov V., Sapozhnikov Vl., Efanov D. Concurrent error detection of combinational circuits by the method of Boolean complement on the base of "2-out-of-4" code. *Proceedings of 14th IEEE East-West Design & Test Symposium (EWDTS'2016), Yerevan, Armenia, 14–17 October 2016*. Yerevan, 2016, pp. 126–133, doi: 10.1109/EWDTS.2016.7807677
27. Sapozhnikov V. V., Sapozhnikov Vl. V., Efanov D. V. Postroenie samoproveryaemykh struktur sistem funktsional'nogo kontrolya na osnove ravnovesnogo koda "2 iz 4" [Design of self-checking concurrent error detection systems based on "2-out-of-4" constant-weight code]. *Problemy upravleniya [Control Sciences]*, 2017, no. 1, pp. 57–64 (in Russian).
28. Tarnick S. Design of embedded m-out-of-n code checkers using complete parallel counters. *13th IEEE International On-Line Testing Symposium (IOLTS 2007), 8–11 July 2007, Crete, Greece*. Crete, 2007, pp. 285–292. doi: 10.1109/IOLTS.2007.27
29. Piestrak S. Design of minimal-level PLA self-testing checkers for m-out-of-n codes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 1996, vol. 4, iss. 2, pp. 264–272. doi: 10.1109/92.502198
30. Sapozhnikov V. V., Sapozhnikov Vl. V. Samoproveryaemye diskretnye ustrojstva. *Self-Checking Discrete Devi-ses*. Saint Petersburg, Energoatomizdat Publ., 1992, 224 p. (in Russian).
31. Aksonova G. P. Neobhodimye i dostatochnye usloviya postroeniya polnost'yu proveryaemykh skhem svertki po modulyu 2 [Necessary and sufficient conditions for design of completely checkable modulo convolution circuits]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 1979, no. 9, pp. 126–135 (in Russian).
32. *Collection of Digital Design Benchmarks*. Available at: <http://ddd.fit.cvut.cz/prj/Benchmarks/> (accessed 20.02.2018).
33. Sentovich E. M., Singh K. J., Moon C., Savoj H., Brayton R. K., Sangiovanni-Vincentelli A. Sequential circuit design using synthesis and optimization. *Proceedings IEEE International Conference on Computer Design: VLSI in Computers & Processors, 11–14 October 1992, Cambridge, MA, USA*. Cambridge, 1992, pp. 328–333. doi: 10.1109/ICCD.1992.276282

**Информация об авторах**

*Ефанов Дмитрий Викторович* – доктор технических наук, доцент, руководитель направления систем мониторинга и диагностики ООО «ЛокоТех-Сигнал» (ул. 3-я Рыбинская, 18, 107113, Москва, Российская Федерация); профессор кафедры «Автоматика, телемеханика и связь на железнодорожном транспорте» Российского университета транспорта (МИИТ) (ул. Образцова, 9, 127994, Москва, Российская Федерация). E-mail: TrES-4b@yandex.ru

*Сапожников Валерий Владимирович* – доктор технических наук, профессор, профессор кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения Императора Александра I (Московский пр., 9, 190031, Санкт-Петербург, Российская Федерация). E-mail: port.at.pgups@gmail.com

*Сапожников Владимир Владимирович* – доктор технических наук, профессор, профессор кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения Императора Александра I (Московский пр., 9, 190031, Санкт-Петербург, Российская Федерация). E-mail: at.pgups@gmail.com

*Пивоваров Дмитрий Вячеславович* – аспирант, ассистент кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения Императора Александра I (Московский пр., 9, 190031, Санкт-Петербург, Российская Федерация). E-mail: pivovarov.d.v.spb@gmail.com

**Information about the authors**

*Dmitry V. Efanov* – Dr. Sci. (Eng.), Associate Professor, Head of the Direction of Monitoring and Diagnostic Systems at "LocoTech-Signal" LCC (18, 3ed Rybinskaya Str., 107113, Moscow, Russian Federation); Professor of "Automation, Remote Control and Communication on Railway Transport" Department, Russian University of Transport (9, Obratsova Str., 127994, Moscow, Russian Federation). E-mail: TrES-4b@yandex.ru

*Valery V. Sapozhnikov* – Dr. Sci. (Eng.), Professor, Professor of "Automation and Remote Control on Railways" Department, Emperor Alexander I St. Petersburg State Transport University (9, Moscow Ave., 190031, Saint Petersburg, Russian Federation). E-mail: port.at.pgups@gmail.com

*Vladimir V. Sapozhnikov* – Dr. Sci. (Eng.), Professor, Professor of "Automation and Remote Control on Railways" Department, Emperor Alexander I St. Petersburg State Transport University (9, Moscow Ave., 190031, Saint Petersburg, Russian Federation). E-mail: at.pgups@gmail.com

*Dmitry V. Pivovarov* – Ph. D. Student, Assistant of "Automation and Remote Control on Railways" Department, Emperor Alexander I St. Petersburg State Transport University (9, Moscow Ave., 190031, Saint Petersburg, Russian Federation). E-mail: pivovarov.d.v.spb@gmail.com