

**ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ****DESIGNING AND DIAGNOSTICS OF COMPUTER SYSTEMS**

Научная статья

УДК 681.518.5 + 004.052.32

doi: 10.17223/19988605/72/12

**Использование равновесного кода «1 из 4» при синтезе самопроверяемых схем
встроенного контроля на основе логической коррекции сигналов****Дмитрий Викторович Ефанов***Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия**Российский университет транспорта, Москва, Россия**Ташкентский государственный транспортный университет, Ташкент, Узбекистан**TrES-4b@yandex.ru*

Аннотация. Предложен способ получения группы методов синтеза самопроверяемых дискретных устройств со схемами встроенного контроля, синтезируемыми на основе логической коррекции сигналов и равновесного кода «1 из 4». Схемы встроенного контроля строятся для групп, состоящих из четырех выходов объекта диагностирования. В отличие от известных методов подразумевается коррекция сигналов в схеме встроенного контроля от всех четырех выходов из выделенной группы выходов объекта диагностирования, что позволяет получить большое количество методов синтеза схемы встроенного контроля. Показано, как устанавливается зависимость между значениями, формируемыми на выходах блока вычисления функций коррекции, и значениями, формируемыми на выходах объекта диагностирования, учитывающая подачу проверяющих тестов на элементы преобразования и тестер в схеме встроенного контроля в процессе эксплуатации самопроверяемого устройства. Приводится пример установления такой зависимости, а также определены множества проверяющих тестов для схемы встроенного контроля. Показано, что для полной проверки схемы встроенного контроля требуется не более восьми рабочих комбинаций, генерируемых на выходах объекта диагностирования. Полученные в статье результаты могут использоваться при разработке самопроверяемых дискретных устройств на различной элементной базе.

Ключевые слова: самопроверяемые дискретные устройства; схема встроенного контроля; логическая коррекция сигналов; равновесный код «1 из 4»; контроль вычислений на выходах дискретных устройств.

Для цитирования: Ефанов Д.В. Использование равновесного кода «1 из 4» при синтезе самопроверяемых схем встроенного контроля на основе логической коррекции сигналов // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2025. № 72. С. 114–133. doi: 10.17223/19988605/72/12

Original article

doi: 10.17223/19988605/72/12

**Using the “1-out-of-4” constant-weight code in the synthesis of self-checking
concurrent error-detection circuit based on Boolean signal correction****Dmitry V. Efanov***Peter the Great St. Petersburg Polytechnic University, Saint Petersburg, Russian Federation**Russian University of Transport, Moscow, Russian Federation*

Abstract. In this paper proposed a method for obtaining a group of methods for synthesizing self-checking discrete devices with concurrent error-detection (CED circuit) synthesized based on Boolean signal correction and a «1-out-of-4» constant-weight code. The CED circuits are made for groups that consist of four outputs from the diagnostic object. As opposed known methods, it implies correction of signals in the CED circuit from all four outputs from a selected group of diagnostic object outputs, which allows obtaining a mass of methods for synthesizing a CED circuit. Also, author showing how the relationship is established between the values formed at the correction computing check unit outputs and the values formed at the outputs of the diagnostic object, considering the supply of detection test set to the gates and the checker in the CED circuit during operation of the self-checking device. An example of establishing such a dependence is given, and detection test set for the CED circuit are defined. It is shown that no more than eight working combinations generated at the outputs of the diagnostic object are required for a full check the CED circuit. These results demonstrate the effectiveness of the proposed solutions. The results obtained in the paper can be used in developing self-checking discrete devices on various element bases.

Keywords: self-checking discrete devices; concurrent error-detection circuit; Boolean signal correction; «1-out-of-4» constant-weight code; computing check at the discrete device's outputs.

For citation: Efanov, D.V. (2025) Using the «1-out-of-4» constant-weight code in the synthesis of self-checking concurrent error-detection circuit based on Boolean signal correction. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naja tehnika i informatika – Tomsk State University Journal of Control and Computer Science*. 72. pp. 114–133. doi: 10.17223/19988605/72/12

Введение

Одной из ключевых задач при синтезе дискретных устройств, входящих в структуры систем критического применения, является своевременное обнаружение неисправностей и ошибок в вычислениях [1–5]. Недостаточно синтезировать устройство, действующее в соответствии с требуемым алгоритмом. Важно наделить его свойством фиксации корректности собственной работы, чтобы избежать распространения неверных данных в системе и тем самым нарушения обеспечиваемого технологического процесса. Все остальные задачи, например формирование отказоустойчивых и самовосстанавливаемых структур для устройств, решаются, как правило, с учетом возможности обнаружения ошибок, вызываемых неисправностями.

Для обеспечения обнаружения неисправностей в дискретных устройствах используются различные подходы, в том числе распространено использование самопроверяемых схем встроенного контроля (СВК) [6–8]. СВК решает задачу обнаружения неисправностей косвенно по оценке корректности вычислений на рабочих выходах исходного устройства (объекта диагностирования) или в специально выведенных для этого контрольных точках [9]. В качестве диагностических признаков для СВК могут использоваться принадлежность вычисляемых булевых функций к особым или «близким» к ним классам булевых функций [10, 11] или же принадлежность формируемых кодовых слов заранее выбранным двоичным блоковым равномерным кодам [12, 13]; оба диагностических признака можно комбинировать [14, 15].

Известны две основные структуры организации СВК. Первая (классическая и широко используемая на практике) подразумевает отождествление выходов объекта диагностирования с информационными символами, которые в СВК дополняются контрольными символами так, чтобы формируемое в ней кодовое слово принадлежало заранее выбранному избыточному коду [9]. Контроль принадлежности формируемых кодовых слов выбранному коду осуществляется с использованием тестера [16, 17]. Вторая структура (альтернативная) подразумевает использование логической коррекции сигналов (ЛКС) с выходов объекта диагностирования в СВК [18]. Ее применение позволяет преобразовать любой вектор, формируемый на выходах исходного дискретного устройства, в кодовое слово заданного кода [19, 20], либо же функции, реализуемые объектом диагностирования, в функции специального вида [10]. Альтернативная структура организации СВК на основе ЛКС позволяет гораздо проще обеспечивать решение наиболее сложной задачи, возникающей в ходе проектирования самопроверяемого устройства, – наделения структуры СВК свойством самопроверяемости. Примеры и эксперименты

с тестовыми схемами показывают, что ее использование позволяет решать задачу обеспечения самопроверяемости даже тогда, когда этого невозможно добиться с использованием классической структуры и широко известных методов, например дублирования или контроля вычислений по классическим кодам с суммированием [21].

Настоящая статья раскрывает особенности получения целой группы методов организации СВК на основе ЛКС с применением равновесного кода «1 из 4», дающих возможность простого синтеза самопроверяемых устройств и дополняющих известные способы использования данного кода в аналогичных задачах.

1. Структура организации схемы встроенного контроля на основе логической коррекции сигналов

В [22] предложена, а затем описана в большом количестве работ, включая монографию [18], структура организации СВК для комбинационных составляющих (комбинационных схем) дискретных устройств, основанная на логической коррекции сигналов (ЛКС). Она представлена на рис. 1.

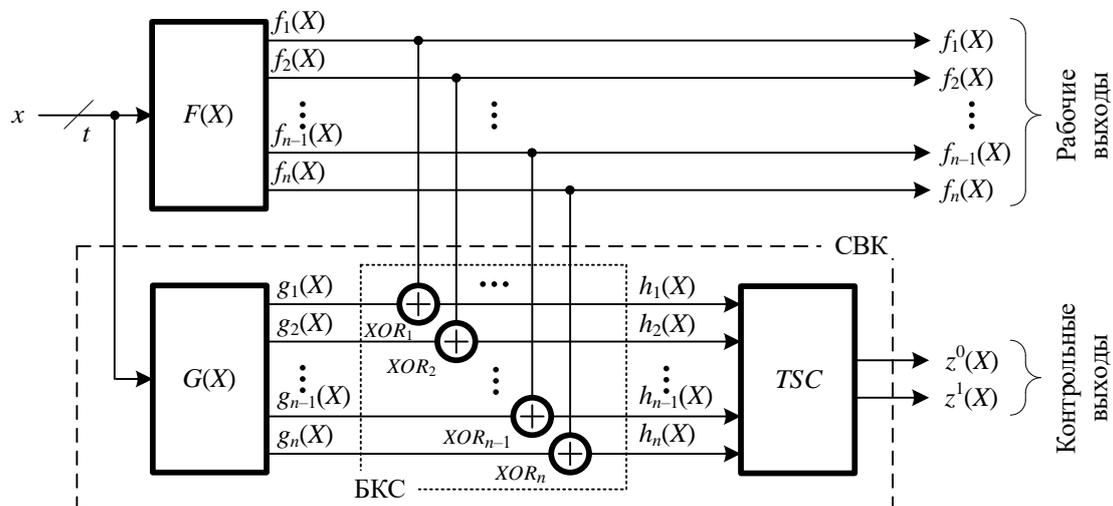


Рис. 1. Структура организации СВК на основе ЛКС

Fig. 1. The structure of the organization of the CED system based on the Boolean signal correction

В описываемой структуре объектом диагностирования является блок $F(X)$. На входы данного блока поступают комбинации $\langle x_t x_{t-1} \dots x_2 x_1 \rangle = \langle X \rangle$, а на выходах формируются значения булевых функций $f_i(X)$, $i = \overline{1, n}$. Объект диагностирования снабжен СВК, состоящей из трех функциональных блоков. Блок $G(X)$ является блоком вычисления значений функций коррекции сигналов и формирует значения булевых функций $g_i(X)$, $i = \overline{1, n}$, предназначенных для коррекции сигналов от объекта диагностирования. Одноименные выходы блоков $F(X)$ и $G(X)$ подключены к входам двухвходовых элементов коррекции сигналов, которые реализуют функцию сложения по модулю 2 (XOR). Двухвходовые элементы преобразования объединены в блок коррекции сигналов (БКС). На выходах БКС, таким образом, при подаче на входы какой-либо комбинации $\langle X \rangle$ формируются следующие сигналы:

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

По сути, преобразование (1) позволяет вектор $\langle f_n(X) f_{n-1}(X) \dots f_2(X) f_1(X) \rangle = \langle F \rangle$, формируемый на выходах блока $F(X)$, трансформировать в вектор $\langle h_n(X) h_{n-1}(X) \dots h_2(X) h_1(X) \rangle = \langle H \rangle$, наделяемый особыми свойствами, например принадлежности заданному двоичному равномерному коду. Принадлежность этого вектора выбранному коду проверяется с помощью тестера TSC (totally self-checking checker), выходы $z^0(X)$ и $z^1(X)$ которого являются и контрольными выходами СВК. На выходах тестера фиксируется парафазный сигнал $\langle 01 \rangle$ или $\langle 10 \rangle$. Если в объекте диагностирования или элементах СВК будет присутствовать неисправность или появится ошибка в вычислениях, вызванная внешними

дестабилизирующими факторами, то на выходах $z^0(X)$ и $z^1(X)$ будет установлен непарафазный сигнал $\langle 00 \rangle$ или $\langle 11 \rangle$.

Для обеспечения самопроверяемости всей системы, представленной на рис. 1, требуется соблюдение некоторых условий:

1. Во-первых, структуры блоков $F(X)$ и $G(X)$ должны быть проверяемыми [23].

2. Во-вторых, на входах каждого из элементов преобразования в БКС должен формироваться проверяющий тест, содержащий все четыре комбинации $\{00, 01, 10, 11\}$ при их канонической реализации [24].

Здесь отметим, что при неканонической реализации с учетом свойств функции «сложение по модулю 2» (XOR) число тестовых комбинаций уменьшается: тестовыми будут являться комбинации либо из подмножества $\{00, 01, 11\}$, либо из $\{00, 01, 10\}$, либо из $\{00, 10, 11\}$, либо из $\{01, 10, 11\}$. Однако далее рассматривается именно каноническая реализация функции «сложение по модулю 2», поскольку все остальные случаи с позиции тестирования окажутся частными, а результаты, полученные в настоящей статье, с легкостью смогут быть распространены и на случай применения других реализаций элементов «сложение по модулю 2».

3. В-третьих, на входах тестера выбранного кода при эксплуатации устройства также должен формироваться проверяющий тест, определяемый выбранным кодом и способом реализации тестера [16].

Задача обеспечения проверяемости блоков $F(X)$ и $G(X)$ связана с их контролепригодной реализацией таким образом, чтобы любая неисправность из установленной модели неисправностей для каждого из этих устройств проявлялась на их выходах в виде искажений сигналов хотя бы на одной входной комбинации $\langle x_t x_{t-1} \dots x_2 x_1 \rangle$. Задачи же формирования проверяющих тестов для элементов БКС и тестера должны решаться в процессе проектирования СВК по разрабатываемому методу.

Рассмотрим далее использование равновесного кода «1 из 4» (1/4-кода) для решения задачи организации полностью самопроверяемой СВК на основе ЛКС.

2. Использование равновесного кода «1 из 4» при организации схемы встроенного контроля на основе логической коррекции сигналов

1/4-код является одним из простейших неразделимых кодов, которые обладают рядом особенностей, эффективно применяющихся при синтезе самопроверяемых структур. Множество кодовых слов 1/4-кода включает в себя следующие булевы векторы: $\{0001, 0010, 0100, 1000\}$. Отсюда становится ясно, что данным кодом не обнаруживается всего 12 ошибок в кодовых словах из возможных 240, и все эти ошибки являются двукратными разнонаправленными (симметричными) [25]. Наиболее простой полностью самопроверяемый тестер для 1/4-кода представлен на рис. 2 [16].

На его входы поступают кодовые векторы $\langle h_4(X) h_3(X) h_2(X) h_1(X) \rangle$, а на выходах $\langle z^0(X) z^1(X) \rangle$ фиксируется парафазный сигнал, если вектор представляет собой кодовое слово 1/4-кода, и фиксируется непарафазный сигнал в том случае, если на входы поступил вектор, не принадлежащий 1/4-коду, или же возникли неисправности самого тестера. Для полной проверки тестера 1/4-кода достаточно на его входы подать единожды каждое из его кодовых слов.

На рис. 3 изображена структура организации СВК на основе ЛКС с применением 1/4-кода. Она строится для группы из четырех выходов объекта диагностирования.

Поскольку при организации СВК используется 1/4-код, то можно учесть тот факт, что для получения его кодового слова достаточно преобразовать три из четырех символов в кодовом слове $\langle f_4(X) f_3(X) f_2(X) f_1(X) \rangle$. Это обстоятельство позволяет несколько упростить структуру организации СВК (рис. 4). В упрощенной структуре используется три элемента преобразования, что влияет и на структурную избыточность СВК, и на выполняемые на этапе ее проектирования процедуры для обеспечения самопроверяемости.

Отметим важный факт. Во всех исследованиях применения 1/4-кода до настоящей работы рассматривался только такой вариант организации СВК на основе ЛКС, который подразумевает минимизацию числа элементов преобразования. В любых случаях их достаточно три, а в частных – два и даже один (см., напр: [19]).

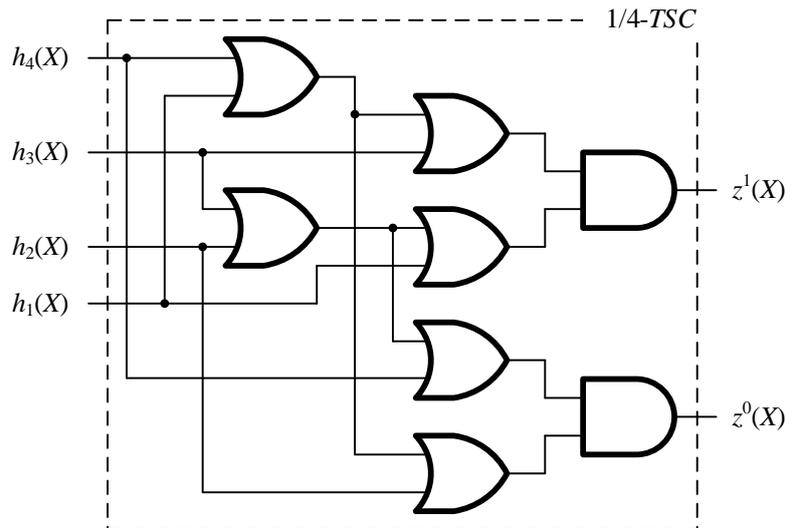


Рис. 2. Тестер 1/4-кода
Fig. 2. 1/4-code checker

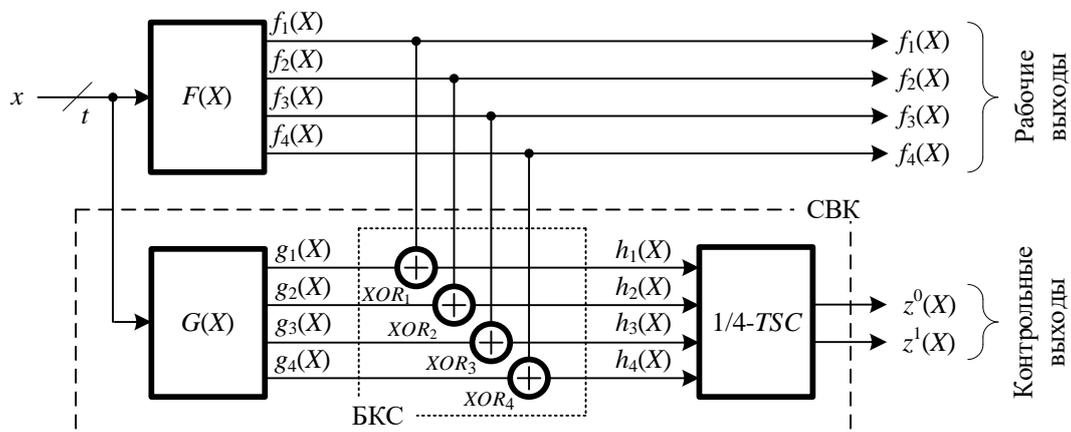


Рис. 3. Структура организации СВК на основе ЛКС с применением 1/4-кода
Fig. 3. The structure of the organization of the CED system based on the Boolean signal correction with 1/4-code

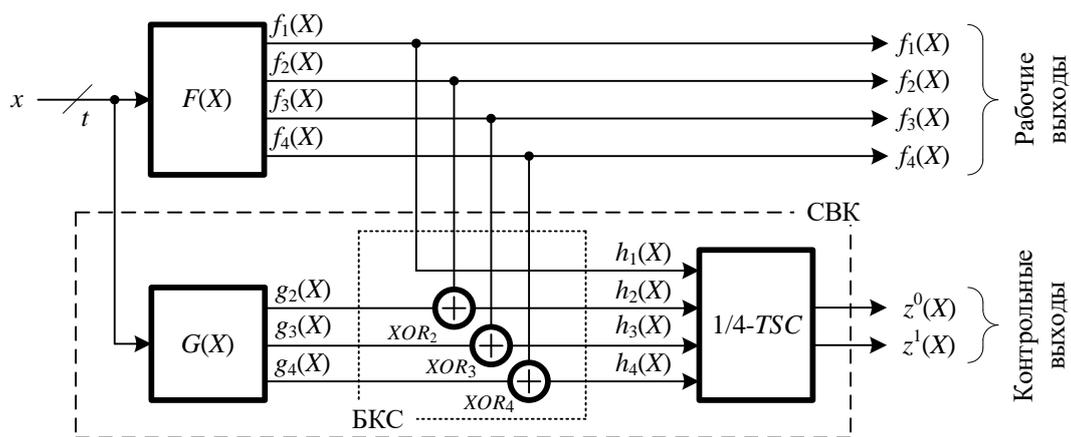


Рис. 4. Упрощенная структура организации СВК на основе ЛКС с применением 1/4-кода
Fig. 4. Simplified structure of the organization of the CED system based on the Boolean signal correction with 1/4-code

Первой работой, в которой описано использование ЛКС совместно с неразделимыми кодами, по всей видимости, является статья [22]. В ней приведены особенности использования любых равновесных кодов для организации СВК на основе ЛКС. На примере контроля шестивыходного устройства

показано, что использование классической структуры с дополнением сигналов от объекта диагностирования контрольными сигналами не всегда дает возможность синтеза полностью самопроверяемых структур, и показано эффективное использование в этих целях ЛКС и равновесного кода «1 из 6». В этой же статье представлен способ синтеза СВК на основе ЛКС с применением 1/4-кода, в основе которого лежит следующая зависимость между значениями функций коррекции сигналов и функций, описывающих выходы объекта диагностирования:

$$\begin{cases} g_4(X) = 0; \\ g_3(X) = f_4(X) f_3(X); \\ g_2(X) = (f_4(X) \vee f_3(X)) f_2(X); \\ g_1(X) = (f_4(X) \vee f_3(X) \vee f_2(X)) f_1(X) \vee \overline{f_4(X) \vee f_3(X) \vee f_2(X)} \vee f_1(X). \end{cases} \quad (2)$$

Следует обратить внимание на то, что нумерация выходов в настоящей работе отличается от нумерации выходов, принятой в [19, 22], и сделана слева направо, как это и принято при нумерации разрядов в кодовых векторах. Поэтому в системе функций (2) нумерация функций изменена по сравнению с источником. Для полного соответствия изложенного работе [19, 22] нужно выполнить замену индексов функций: $4 \rightarrow 1, 3 \rightarrow 2, 2 \rightarrow 3, 1 \rightarrow 4$.

Пользуясь (2), получим сигналы, формируемые на линиях СВК, синтезированной на основе ЛКС с применением 1/4-кода, с условием формирования полного множества четырехбитных кодовых векторов на выходах объекта диагностирования (табл. 1).

Из табл. 1 следует, что на выходах БКС формируется полное множество кодовых слов 1/4-кода. Однако читатель может обратить внимание на последние три столбца таблицы, где приведены тестовые комбинации элементов преобразования: для элементов XOR_3 и XOR_2 невозможно сформировать комбинации $\langle 01 \rangle$ и $\langle 10 \rangle$. Другими словами, использование (2) не позволяет синтезировать полностью самопроверяемые СВК на основе ЛКС. Обратите внимание на табл. 2 из [19]. Из сравнения столбцов f_i и g_i становится ясным, что для рассмотренного авторами примера (как, впрочем, и для любого примера) не обеспечивается самопроверяемость всех элементов преобразования.

Таблица 1

Сигналы на линиях СВК, полученные при использовании системы функций (2)

Выходы $F(X)$				Выходы $G(X)$				Выходы БКС				Тестовые комбинации элементов преобразования		
$f_4(X)$	$f_3(X)$	$f_2(X)$	$f_1(X)$	$g_4(X)$	$g_3(X)$	$g_2(X)$	$g_1(X)$	$h_4(X)$	$h_3(X)$	$h_2(X)$	$h_1(X)$	$\langle f_3(X) g_3(X) \rangle$	$\langle f_2(X) g_2(X) \rangle$	$\langle f_1(X) g_1(X) \rangle$
0	0	0	0	0	0	0	1	0	0	0	1	00	00	01
0	0	0	1	0	0	0	0	0	0	0	1	00	00	10
0	0	1	0	0	0	0	0	0	0	1	0	00	10	00
0	0	1	1	0	0	0	1	0	0	1	0	00	10	11
0	1	0	0	0	0	0	0	0	1	0	0	10	00	00
0	1	0	1	0	0	0	1	0	1	0	0	10	00	11
0	1	1	0	0	0	1	0	0	1	0	0	10	11	00
0	1	1	1	0	0	1	1	0	1	0	0	10	11	11
1	0	0	0	0	0	0	0	1	0	0	0	00	00	00
1	0	0	1	0	0	0	1	1	0	0	0	00	00	11
1	0	1	0	0	0	1	0	1	0	0	0	00	11	00
1	0	1	1	0	0	1	1	1	0	0	0	00	11	11
1	1	0	0	0	1	0	0	1	0	0	0	11	00	00
1	1	0	1	0	1	0	1	1	0	0	0	11	00	11
1	1	1	0	0	1	1	0	1	0	0	0	11	11	00
1	1	1	1	0	1	1	1	1	0	0	0	11	11	11

В [21, 26] считается, что сигналы на выходах БКС изначально не определены, а задача синтеза СВК решается подбором доопределяемых значений функций на выходах БКС так, чтобы были обеспечены условия тестируемости элементов преобразования. Кроме того, в [21] сформулирована и доказана теорема, определяющая условия тестируемости элементов преобразования и тестера при организации СВК на основе равновесных кодов «1 из n », где $n \geq 3$ – число выходов в контролируемой группе.

Явным недостатком метода [19, 22], проявляющимся при попытках автоматизации расчетов, является необходимость подбора значений на выходах БКС по некоторому правилу для обеспечения тестируемости элементов преобразования и тестера 1/4-кода. Поэтому в [27] описана группа методов, базирующихся на установлении зависимости между значениями функций, описывающих выходы блока вычисления функций коррекции в СВК, и функций, реализуемых на выходах объекта диагностирования. Эта зависимость, в отличие от (2), учитывает необходимость формирования проверяющего теста для каждого элемента преобразования. Авторами предложен и научно обоснован подход, позволяющий сформировать 72 варианта синтеза СВК на основе ЛКС с применением 1/4-кода с использованием зависимости между значениями функций, описывающих выходы блока вычисления функций коррекции сигналов, и функций, реализуемых на выходах объекта диагностирования. Один из таких вариантов рассмотрен подробнее:

$$\begin{cases} g_4(X) = 0; \\ g_3(X) = \overline{f_4(X)f_3(X)}f_2(X) \vee f_3(X)(f_4(X) \vee f_1(X)); \\ g_2(X) = \overline{f_4f_3f_2f_1} \vee f_2(X)(f_4(X) \vee f_3(X) \vee f_1(X)); \\ g_1(X) = \overline{f_4f_3f_2f_1} \vee f_1(X)(f_4(X) \vee f_3(X) \vee f_2(X)). \end{cases} \quad (3)$$

Также доказано, что для обеспечения полной самопроверяемости СВК необходимо формирование на выходах объекта диагностирования не менее семи конкретных векторов. Это требуется для обеспечения полной проверки элементов БКС и тестера в структуре СВК.

Еще в одной статье [28] приводится метод синтеза СВК на основе ЛКС с применением 1/4-кода, который подразумевает изначально неопределенность в значениях функций на выходах БКС на входных комбинациях. Он связан с пошаговым доопределением значений функций на каждой входной комбинации с учетом введения неопределенностей для некоторых значений доопределяемых функций, последующей минимизации функций коррекции, проверкой формирования полного множества проверяющих комбинаций и, при необходимости, выполнением иного доопределения значений функций на выходах БКС для обеспечения формирования полного теста соответствующего элемента преобразования. Здесь процедуры доопределения оказываются частично автоматизированными за счет установления зависимости между значениями функций, формируемых на выходах БКС. Тем не менее присутствуют частично определенные булевы функции, что требует дополнительного анализа для однозначного доопределения их значений на полном множестве входных комбинаций.

Исследования автора настоящей статьи показывают, что существует еще один подход к формированию целой группы методов синтеза СВК, основанных на установлении зависимости между функциями, описывающими выходы блока вычисления функций коррекции в СВК, и функциями, реализуемыми на выходах объекта диагностирования. И здесь используется уже СВК с преобразованием сигналов с четырех выходов объекта диагностирования (см. рис. 3). Использование описываемых далее результатов существенно расширяет число способов синтеза самопроверяемых СВК на основе ЛКС с применением 1/4-кода, а полученные автором результаты восполняют пробел в теории синтеза СВК на основе ЛКС, связанный с недостатками известных методов для решения аналогичной задачи.

3. Использование 1/4-кода при организации схемы встроенного контроля на основе логической коррекции сигналов с преобразованием всех сигналов от объекта диагностирования

Рассмотрим все возможные двоичные векторы, которые могут быть сформированы на выходах объекта диагностирования. Они приведены в табл. 2. Также в табл. 2 представлены тестовые комбинации,

формирующиеся на входах элементов преобразования в виде двухбитного вектора, где первый символ соответствует значению функции $f_i(X)$, а второй – значению функции $g_i(X)$, $i \in \{1, 2, 3, 4\}$. Знаком «~» показаны значения, которые на данном этапе не определены, поскольку связаны с тем, в какой именно вектор будет осуществлено преобразование исходного вектора $\langle F \rangle$.

Таблица 2

Векторы, формируемые на выходах объекта диагностирования, и тестовые комбинации элементов преобразования

$f_4(X)$	$f_3(X)$	$f_2(X)$	$f_1(X)$	XOR_4	XOR_3	XOR_2	XOR_1
0	0	0	0	0~	0~	0~	0~
0	0	0	1	0~	0~	0~	1~
0	0	1	0	0~	0~	1~	0~
0	0	1	1	0~	0~	1~	1~
0	1	0	0	0~	1~	0~	0~
0	1	0	1	0~	1~	0~	1~
0	1	1	0	0~	1~	1~	0~
0	1	1	1	0~	1~	1~	1~
1	0	0	0	1~	0~	0~	0~
1	0	0	1	1~	0~	0~	1~
1	0	1	0	1~	0~	1~	0~
1	0	1	1	1~	0~	1~	1~
1	1	0	0	1~	1~	0~	0~
1	1	0	1	1~	1~	0~	1~
1	1	1	0	1~	1~	1~	0~
1	1	1	1	1~	1~	1~	1~

Каждый вектор $\langle F \rangle$ может быть преобразован в кодовые слова $\langle H \rangle$, принадлежащие 1/4-коду четырьмя способами: в векторы $\langle 0001 \rangle - \alpha$, $\langle 0010 \rangle - \beta$, $\langle 0100 \rangle - \gamma$ и $\langle 1000 \rangle - \delta$. Преобразования осуществляются при подаче на входы объекта диагностирования векторов $\langle X \rangle$. При этом на каждом входном векторе $\langle X \rangle$ может сформироваться только один из векторов $\langle H \rangle$. Припишем каждому вектору $\langle F \rangle$ индекс i , соответствующий десятичному эквиваленту вектора двоичного числа, записываемому в нем. Условимся обозначать буквами a_i^j , $i \in \{0, 1, \dots, 15\}$, $j \in \{\alpha, \beta, \gamma, \delta\}$, каждый из вариантов преобразования i -го вектора, формируемого на выходах объекта диагностирования, в j -е кодовое слово 1/4-кода.

Можно преобразовать каждый i -й вектор, формируемый на выходах объекта диагностирования, в j -е кодовое слово 1/4-кода четырьмя способами. Всего преобразуемых векторов 16. Тогда существует $4^{16} = 4\,294\,967\,296$ вариантов преобразований. Однако не все они будут устраивать условиям обеспечения тестируемости элементов СВК. В табл. 3 приведены для каждого элемента преобразования все тестовые комбинации, а также для каждого вектора, формируемого на выходах объекта диагностирования, указано, какие варианты преобразования дают ту или иную тестовую комбинацию. Тогда из табл. 3 непосредственно можно выделить условия, выполнение которых обеспечит тестируемость каждого из элементов преобразования.

Для каждого элемента преобразования должны быть выполнены условия формирования хотя бы единожды каждой тестовой комбинации.

Для XOR_4 , XOR_3 , XOR_2 , XOR_1 это означает выполнение хотя бы единожды следующих условий соответственно:

$$\begin{cases} \forall k, l \in \{0, 1, 2, 3, 4, 5, 6, 7\}, k \neq l: (a_k^\alpha \vee a_k^\beta \vee a_k^\gamma) a_l^\delta = 1, \\ \forall k, l \in \{8, 9, 10, 11, 12, 13, 14, 15\}, k \neq l: (a_k^\alpha \vee a_k^\beta \vee a_k^\gamma) a_l^\delta = 1. \end{cases} \quad (4)$$

$$\begin{cases} \forall k, l \in \{0, 1, 2, 3, 8, 9, 10, 11\}, k \neq l: (a_k^\alpha \vee a_k^\beta \vee a_k^\delta) a_l^\gamma = 1, \\ \forall k, l \in \{4, 5, 6, 7, 12, 13, 14, 15\}, k \neq l: (a_k^\alpha \vee a_k^\beta \vee a_k^\delta) a_l^\gamma = 1. \end{cases} \quad (5)$$

$$\begin{cases} \forall k, l \in \{0, 1, 4, 5, 8, 9, 12, 13\}, k \neq l: (a_k^\alpha \vee a_k^\gamma \vee a_k^\delta) a_l^\beta = 1, \\ \forall k, l \in \{2, 3, 6, 7, 10, 11, 14, 15\}, k \neq l: (a_k^\alpha \vee a_k^\gamma \vee a_k^\delta) a_l^\beta = 1. \end{cases} \quad (6)$$

$$\begin{cases} \forall k, l \in \{0, 2, 4, 6, 8, 10, 12, 14\}, k \neq l: (a_k^\beta \vee a_k^\gamma \vee a_k^\delta) a_l^\alpha = 1, \\ \forall k, l \in \{1, 3, 5, 7, 9, 11, 13, 15\}, k \neq l: (a_k^\beta \vee a_k^\gamma \vee a_k^\delta) a_l^\alpha = 1. \end{cases} \quad (7)$$

Таблица 3

Условия формирования тестовых комбинаций для элементов преобразования

<F>	XOR ₄				XOR ₃				XOR ₂				XOR ₁			
	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	α, β, γ	δ	–	–	α, β, δ	γ	–	–	α, γ, δ	β	–	–	β, γ, δ	α	–	–
0001	α, β, γ	δ	–	–	α, β, δ	γ	–	–	α, γ, δ	β	–	–	–	–	α	β, γ, δ
0010	α, β, γ	δ	–	–	α, β, δ	γ	–	–	–	–	β	α, γ, δ	β, γ, δ	α	–	–
0011	α, β, γ	δ	–	–	α, β, δ	γ	–	–	–	–	β	α, γ, δ	–	–	α	β, γ, δ
0100	α, β, γ	δ	–	–	–	–	γ	α, β, δ	α, γ, δ	β	–	–	β, γ, δ	α	–	–
0101	α, β, γ	δ	–	–	–	–	γ	α, β, δ	α, γ, δ	β	–	–	–	–	α	β, γ, δ
0110	α, β, γ	δ	–	–	–	–	γ	α, β, δ	–	–	β	α, γ, δ	β, γ, δ	α	–	–
0111	α, β, γ	δ	–	–	–	–	γ	α, β, δ	–	–	β	α, γ, δ	–	–	α	β, γ, δ
1000	–	–	δ	α, β, γ	α, β, δ	γ	–	–	α, γ, δ	β	–	–	β, γ, δ	α	–	–
1001	–	–	δ	α, β, γ	α, β, δ	γ	–	–	α, γ, δ	β	–	–	–	–	α	β, γ, δ
1010	–	–	δ	α, β, γ	α, β, δ	γ	–	–	–	–	β	α, γ, δ	β, γ, δ	α	–	–
1011	–	–	δ	α, β, γ	α, β, δ	γ	–	–	–	–	β	α, γ, δ	–	–	α	β, γ, δ
1100	–	–	δ	α, β, γ	–	–	γ	α, β, δ	α, γ, δ	β	–	–	β, γ, δ	α	–	–
1101	–	–	δ	α, β, γ	–	–	γ	α, β, δ	α, γ, δ	β	–	–	–	–	α	β, γ, δ
1110	–	–	δ	α, β, γ	–	–	γ	α, β, δ	–	–	β	α, γ, δ	β, γ, δ	α	–	–
1111	–	–	δ	α, β, γ	–	–	γ	α, β, δ	–	–	β	α, γ, δ	–	–	α	β, γ, δ

Следуя за условиями (4)–(7) и анализируя табл. 3, сделаем такие выводы:

а) можно преобразование <F> → <H> выполнить с учетом обеспечения формирования проверяющего теста для элемента XOR₄ таким образом, чтобы формировалось подмножество кодовых слов 1/4-кода {0001, 1000}, либо {0010, 1000}, либо {0100, 1000};

б) можно преобразование <F> → <H> выполнить с учетом обеспечения формирования проверяющего теста для элемента XOR₃ таким образом, чтобы формировалось подмножество кодовых слов 1/4-кода {0001, 0100}, либо {0010, 0100}, либо {1000, 0100};

в) можно преобразование <F> → <H> выполнить с учетом обеспечения формирования проверяющего теста для элемента XOR₂ таким образом, чтобы формировалось подмножество кодовых слов 1/4-кода {0001, 0010}, либо {0100, 0010}, либо {1000, 0010};

г) можно преобразование <F> → <H> выполнить с учетом обеспечения формирования проверяющего теста для элемента XOR₁ таким образом, чтобы формировалось подмножество кодовых слов 1/4-кода {0010, 0001}, либо {0100, 0001}, либо {1000, 0001}.

Отсюда явно следует

Утверждение 1. При обеспечении формирования проверяющих тестов для всех элементов преобразования гарантированно формируется и проверяющий тест для тестера 1/4-кода.

Получим далее один из способов преобразования.

Предварительно зафиксируем условие единственности выполнения преобразований:

$$\forall i: a_i^\alpha a_i^\beta = a_i^\alpha a_i^\gamma = a_i^\alpha a_i^\delta = a_i^\beta a_i^\gamma = a_i^\beta a_i^\delta = a_i^\gamma a_i^\delta = 0. \quad (8)$$

Из данных табл. 3 и условий (4)–(7) становится понятно, что существует большое число способов установления зависимостей между значениями функций коррекции сигналов и функций, реализуемых на выходах объекта диагностирования. Получим для примера один из них. Для этого потребуется однозначно заполнить табл. 3 с учетом обеспечения формирования проверяющих тестов для элементов преобразования.

Сформируем таблицу покрытий (табл. 4). При заполнении будем учитывать, что в каждом столбце при выборе способа преобразования должен оказаться хотя бы один знак покрытия «×». Исходя из этого, получим предварительное заполнение таблицы покрытий с учетом данного условия. Будем следовать по шагам:

1. Рассмотрим получение тестовых комбинаций <01> и <10> для элемента XOR_4 , поскольку они получаются при единственном способе преобразования: $\langle F \rangle \rightarrow \langle 1000 \rangle$. На комбинациях с десятичными эквивалентами {0, 1, 2, 3, 4, 5, 6, 7} и {8, 9, 10, 11, 12, 13, 14, 15} требуется единожды выполнить такое преобразование. Заполним табл. 4 знаками покрытия для строк с десятичными эквивалентами 7 и 8, а в скобках укажем способ преобразования.

2. Так как способ преобразования в строке согласно условию (8) единственный, то можно определить, какие комбинации дает такое заполнение для других элементов преобразования. Выполним это и укажем аналогично знаками покрытия и в скобках способом преобразования в соответствующих столбцах для каждого элемента преобразования. Для каждого из элементов преобразования XOR_3 , XOR_2 , XOR_1 формируется по две комбинации <00> и <11>.

3. Для элемента XOR_3 две тестовые комбинации уже сформированы. Требуется сформировать комбинации <01> и <10>. Это реализуется в единственном случае преобразования: $\langle F \rangle \rightarrow \langle 0100 \rangle$. На комбинациях с десятичными эквивалентами {0, 1, 2, 3, 8, 9, 10, 11} и {4, 5, 6, 7, 12, 13, 14, 15} требуется единожды выполнить такое преобразование. Заполним табл. 4 знаками покрытия для строк с десятичными эквивалентами 3 и 12, а в скобках укажем способ преобразования. Такой способ заполнения неслучаен: именно на противоположных половинах таблицы оказываются подмножества {0, 1, 2, 3, 4, 5, 6, 7} и {8, 9, 10, 11, 12, 13, 14, 15}, для которых формируются комбинации <00> и <11> элемента XOR_4 .

4. Аналогично п. 2 укажем для элементов преобразования, получаемые при заполнении строк с номерами 3 и 4 вариантом γ тестовые комбинации. После указанного заполнения для элементов XOR_4 и XOR_3 сформированы все тестовые комбинации, а для элементов XOR_2 и XOR_1 остаются комбинации <01> и <10>.

Таблица 4

Таблица покрытий после первого этапа заполнения

$\langle F \rangle$	XOR_4				XOR_3				XOR_2				XOR_1			
	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	× (α)				× (α)				× (α)					× (α)		
0001	× (β)				× (β)					× (β)						× (β)
0010																
0011	× (γ)					× (γ)						× (γ)				× (γ)
0100																
0101																
0110																
0111		× (δ)					× (δ)				× (δ)					× (δ)
1000			× (δ)		× (δ)				× (δ)				× (δ)			
1001																
1010																
1011																
1100				× (γ)			× (γ)		× (γ)				× (γ)			
1101																
1110				× (β)				× (β)			× (β)		× (β)			
1111				× (α)				× (α)				× (α)			× (α)	

5. Для элемента XOR_2 две тестовые комбинации уже сформированы. Требуется сформировать комбинации <01> и <10>. Это реализуется в единственном случае преобразования: $\langle F \rangle \rightarrow \langle 0010 \rangle$. На комбинациях с десятичными эквивалентами {0, 1, 4, 5, 8, 9, 12, 13} и {2, 3, 6, 7, 10, 11, 14, 15} требуется единожды выполнить такое преобразование. Заполним табл. 4 знаками покрытия для строк с десятичными эквивалентами 2 и 14, а в скобках укажем способ преобразования.

6. Заполним для всех элементов преобразования столбцы, соответствующие формируемым тестовым комбинациям, способом преобразования β .

7. Для элемента XOR_1 две тестовые комбинации уже сформированы. Требуется сформировать комбинации $\langle 01 \rangle$ и $\langle 10 \rangle$. Это реализуется в единственном случае преобразования: $\langle F \rangle \rightarrow \langle 0001 \rangle$. Сформируем их на строках с номерами 1 и 15.

Преобразования $\langle F \rangle \rightarrow \langle H \rangle$ однозначно установлены для восьми векторов, формируемых на выходах объекта диагностирования, составляющих множество $\{0000, 0001, 0011, 0111, 1000, 1100, 1110, 1111\}$. Такое доопределение обеспечивает условия тестируемости элементов преобразования и тестера в СВК. Однако требуется заполнить таблицу покрытий и на оставшихся строках, поскольку необходимо обеспечить формирование кодовых слов 1/4-кода.

Утверждение 2. Для обеспечения тестируемости элементов преобразования и тестера в СВК достаточно на выходах объекта диагностирования обеспечить формирование восьми рабочих кодовых векторов из множества $\{0000, 0001, 0011, 0111, 1000, 1100, 1110, 1111\}$.

Вторым условием заполнения таблицы покрытий является то, что в каждой строке должен оказаться только один знак покрытия. При этом оставшиеся преобразования могут быть произвольными – $\alpha, \beta, \gamma, \delta$. Это уже при имеющемся способе преобразования восьми векторов дает $4^8 = 65\,536$ способов установления функциональной зависимости.

Продолжим заполнение таблицы покрытий с учетом формирования не единожды тестовых комбинаций $\langle 01 \rangle$ и $\langle 10 \rangle$ для каждого элемента преобразования:

1. Строки с десятичными эквивалентами 2 и 13 заполним способом δ . Для элемента XOR_4 будут сформированы тестовые комбинации $\langle 01 \rangle$ и $\langle 10 \rangle$ по два раза.

2. Строки с десятичными эквивалентами 4 и 11 заполним способом γ . Для элемента XOR_3 будут сформированы тестовые комбинации $\langle 01 \rangle$ и $\langle 10 \rangle$ по два раза.

3. Строки с десятичными эквивалентами 5 и 10 заполним способом β . Для элемента XOR_2 будут сформированы тестовые комбинации $\langle 01 \rangle$ и $\langle 10 \rangle$ по два раза.

4. Строки с десятичными эквивалентами 6 и 9 заполним способом α . Для элемента XOR_1 будут сформированы тестовые комбинации $\langle 01 \rangle$ и $\langle 10 \rangle$ по два раза.

Итоговое заполнение таблицы покрытий представлено в таблице 5.

Таблица 5

Таблица покрытий после второго этапа заполнения

$\langle F \rangle$	XOR_4				XOR_3				XOR_2				XOR_1			
	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	$\times (\alpha)$				$\times (\alpha)$				$\times (\alpha)$					$\times (\alpha)$		
0001	$\times (\beta)$				$\times (\beta)$					$\times (\beta)$						$\times (\beta)$
0010		$\times (\delta)$			$\times (\delta)$							$\times (\delta)$	$\times (\delta)$			
0011	$\times (\gamma)$					$\times (\gamma)$						$\times (\gamma)$				$\times (\gamma)$
0100	$\times (\gamma)$						$\times (\gamma)$		$\times (\gamma)$				$\times (\gamma)$			
0101	$\times (\beta)$							$\times (\beta)$		$\times (\beta)$						$\times (\beta)$
0110	$\times (\alpha)$							$\times (\alpha)$				$\times (\alpha)$		$\times (\alpha)$		
0111		$\times (\delta)$						$\times (\delta)$				$\times (\delta)$				$\times (\delta)$
1000			$\times (\delta)$		$\times (\delta)$				$\times (\delta)$				$\times (\delta)$			
1001				$\times (\alpha)$	$\times (\alpha)$				$\times (\alpha)$						$\times (\alpha)$	
1010				$\times (\beta)$	$\times (\beta)$						$\times (\beta)$		$\times (\beta)$			
1011				$\times (\gamma)$		$\times (\gamma)$						$\times (\gamma)$				$\times (\gamma)$
1100				$\times (\gamma)$			$\times (\gamma)$		$\times (\gamma)$				$\times (\gamma)$			
1101			$\times (\delta)$					$\times (\delta)$	$\times (\delta)$							$\times (\delta)$
1110				$\times (\beta)$				$\times (\beta)$			$\times (\beta)$		$\times (\beta)$			
1111				$\times (\alpha)$				$\times (\alpha)$				$\times (\alpha)$			$\times (\alpha)$	

Из табл. 5 следует, что для каждого элемента преобразования тестовые комбинации $\langle 00 \rangle$ и $\langle 11 \rangle$ формируются на шести векторах, а комбинации $\langle 01 \rangle$ и $\langle 10 \rangle$ – на двух. При эксплуатации самопро-

веряемого устройства потребуется обеспечивать появление всего восьми векторов на выходах объекта диагностирования. Более того, каждое из кодовых слов 1/4-кода при таком заполнении таблицы покрытий формируется при использовании четырех векторов. Но, исходя из утверждения 1, достаточно выполнить условия по формированию тестовых комбинаций элементов преобразования, что обеспечит и формирование проверяющего теста для тестера в СВК. Это наилучшее заполнение с позиции равномерности формирования тестовых комбинаций тестера 1/4-кода и элементов преобразования.

Проверяющие тесты для элементов преобразования и тестера определяются из следующих выражений:

$$T_{XOR_4} = (0000 \vee 0001 \vee 0011 \vee 0100 \vee 0101 \vee 0110)(0010 \vee 0111) \& \quad (9)$$

$$\& (1000 \vee 1101)(1001 \vee 1010 \vee 1011 \vee 1100 \vee 1110 \vee 1111);$$

$$T_{XOR_3} = (0000 \vee 0001 \vee 0010 \vee 1000 \vee 1001 \vee 1010)(0011 \vee 1011) \& \quad (10)$$

$$\& (0100 \vee 1100)(0101 \vee 0110 \vee 0111 \vee 1101 \vee 1110 \vee 1111);$$

$$T_{XOR_2} = (0000 \vee 0100 \vee 1000 \vee 1001 \vee 1100 \vee 1101)(0001 \vee 0101) \& \quad (11)$$

$$\& (1010 \vee 1110)(0010 \vee 0011 \vee 0110 \vee 0111 \vee 1011 \vee 1111);$$

$$T_{XOR_1} = (0010 \vee 0100 \vee 1000 \vee 1010 \vee 1100 \vee 1110)(0000 \vee 0110) \& \quad (12)$$

$$\& (1001 \vee 1111)(0001 \vee 0011 \vee 0101 \vee 0111 \vee 1011 \vee 1101);$$

$$T_{1/4-TSC} = (0000 \vee 0110 \vee 1001 \vee 1111)(0001 \vee 0101 \vee 1010 \vee 1110) \& \quad (13)$$

$$\& (0011 \vee 0100 \vee 1011 \vee 1100)(0010 \vee 0111 \vee 1000 \vee 1101).$$

Проверяющий тест для СВК будет определяться по формуле

$$T_{СВК} = T_{1/4-TSC} T_{XOR_4} T_{XOR_3} T_{XOR_2} T_{XOR_1}. \quad (14)$$

Минимальная длина проверяющего теста для СВК составляет 8. Это генерация, например, комбинаций из множества, зафиксированного в утверждении 2.

Получим далее саму зависимость между значениями функций, описывающими выходы объекта диагностирования, и функций коррекции сигналов. Для этого сформируем табл. 6. В таблице получены и значения функций коррекции сигналов, исходя из следующего соображения:

$$h_i(X) = f_i(X) \oplus g_i(X) \Rightarrow g_i(X) = f_i(X) \oplus h_i(X). \quad (15)$$

Путем минимизации методом Карно получим функции, описывающие выходы БКС:

$$\begin{cases} g_4(X) = f_4(X) f_3(X) \overline{f_1(X)} \vee f_4(X) \overline{f_3(X)} f_1(X) \vee f_3(X) f_2(X) f_1(X) \vee \overline{f_3(X)} f_2(X) \overline{f_1(X)}; \\ g_3(X) = f_3(X) f_1(X) \vee f_3(X) f_2(X) \vee f_2(X) f_1(X); \\ g_2(X) = \overline{f_4(X)} f_1(X) \vee \overline{f_4(X)} f_2(X) \vee f_2(X) f_1(X); \\ g_1(X) = \overline{f_4(X)} \overline{f_3(X)} f_2(X) \vee \overline{f_4(X)} f_3(X) f_2(X) \vee f_3(X) \overline{f_2(X)} f_1(X) \vee \overline{f_3(X)} f_2(X) f_1(X). \end{cases} \quad (16)$$

Элементарные преобразования позволяют несколько упростить выражения в (16), хотя это не столь существенно, поскольку итоговая сложность реализации функций коррекции сигналов будет зависеть в том числе и от сложности самих функций, формируемых на выходах объекта диагностирования:

$$\begin{cases} g_4(X) = f_4(X) (f_3(X) \oplus f_1(X)) \vee f_2(X) \overline{f_3(X) \oplus f_1(X)}; \\ g_3(X) = f_3(X) \# f_2(X) \# f_1(X); \\ g_2(X) = \overline{f_4(X)} \# f_2(X) \# f_1(X); \\ g_1(X) = f_4(X) f_3(X) \oplus f_2(X) \vee f_1(X) (f_3(X) \oplus f_2(X)). \end{cases} \quad (17)$$

Значения сигналов на выходах самопроверяемого устройства

Выходы $F(X)$				Выходы БКС				Выходы $G(X)$			
$f_4(X)$	$f_3(X)$	$f_2(X)$	$f_1(X)$	$h_4(X)$	$h_3(X)$	$h_2(X)$	$h_1(X)$	$g_4(X)$	$g_3(X)$	$g_2(X)$	$g_1(X)$
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	1	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	0	0	0	0	0
0	1	0	1	0	0	1	0	0	1	1	1
0	1	1	0	0	0	0	1	0	1	1	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0	0
1	0	1	0	0	0	1	0	1	0	0	0
1	0	1	1	0	1	0	0	1	1	1	1
1	1	0	0	0	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0	0	1	0	1
1	1	1	0	0	0	1	0	1	1	0	0
1	1	1	1	0	0	0	1	1	1	1	0

Использование системы (16) или (17) позволяет синтезировать блок вычисления функций коррекции сигналов $G(X)$. При этом функции коррекции сигналов будут наделены свойством, которое обеспечивает тестируемость всех компонентов СВК.

4. Пример синтеза схемы встроенного контроля на основе предложенного метода и некоторые результаты экспериментов по оценке структурной избыточности самопроверяемых дискретных устройств

Приведем пример использования системы (16) или (17) (они устанавливают одну и ту же зависимость) для синтеза СВК для произвольного комбинационного устройства, заданного табл. 7. Процедура синтеза, таким образом, крайне проста и не требует никакого построчного анализа работы устройства $F(X)$. В таблице указаны сигналы на выходах самопроверяемого устройства. Как следует из ее анализа, на все элементы преобразования и тестер поступают проверяющие тесты.

Объект диагностирования описывается следующими функциями (они получены при отдельной минимизации по методу Карно):

$$\begin{cases} f_4(X) = \overline{x_3} \overline{x_2} \overline{x_1} \vee \overline{x_4} \overline{x_3} \overline{x_2} \vee \overline{x_4} \overline{x_3} \overline{x_1} \vee \overline{x_4} \overline{x_3} \overline{x_2} \overline{x_1}; \\ f_3(X) = \overline{x_4} \overline{x_3} \overline{x_2} \overline{x_1} \vee \overline{x_4} \overline{x_2} \overline{x_1} \vee \overline{x_3} \overline{x_2} \overline{x_1} \vee \overline{x_3} \overline{x_2} \overline{x_1}; \\ f_2(X) = \overline{x_4} \overline{x_2} \overline{x_1} \vee \overline{x_4} \overline{x_3} \overline{x_1} \vee \overline{x_4} \overline{x_3} \overline{x_1} \vee \overline{x_3} \overline{x_2} \overline{x_1}; \\ f_1(X) = \overline{x_4} \overline{x_2} \overline{x_1} \vee \overline{x_4} \overline{x_2} \overline{x_1} \vee \overline{x_3} \overline{x_2} \overline{x_1} \vee \overline{x_3} \overline{x_2} \overline{x_1}. \end{cases}$$

Минимизация функций коррекции сигналов дает следующий результат:

$$\begin{cases} g_4(X) = \overline{x_4} \overline{x_3} \overline{x_2} \vee \overline{x_4} \overline{x_3} \overline{x_1} \vee \overline{x_4} \overline{x_2} \overline{x_1} \vee \overline{x_3} \overline{x_2} \overline{x_1} \vee \overline{x_4} \overline{x_3} \overline{x_2} \overline{x_1}; \\ g_3(X) = \overline{x_3} \overline{x_2} \overline{x_1} \vee \overline{x_4} \overline{x_3} \overline{x_1} \vee \overline{x_3} \overline{x_2} \overline{x_1}; \\ g_2(X) = \overline{x_4} \overline{x_2} \overline{x_1} \vee \overline{x_4} \overline{x_3} \overline{x_1} \vee \overline{x_2} \overline{x_1}; \\ g_1(X) = \overline{x_4} \overline{x_2} \overline{x_1} \vee \overline{x_4} \overline{x_2} \overline{x_1} \vee \overline{x_4} \overline{x_2} \overline{x_1} \vee \overline{x_4} \overline{x_3} \overline{x_2}. \end{cases}$$

Оценим в метрике числа входов внутренних логических элементов сложность технической реализации самопроверяемого устройства (может быть использована и другая метрика, например в числе транзисторов, занимаемых устройством на кристалле, или в числе литер в дизъюнктивной нормальной

форме каждой функции из полученных выражений, однако эти показатели имеют корреляцию с числом входов элементов) [16]. Учтем, что показатель сложности реализации элемента преобразования $L_{XOR} = 6$ (вх.) [16], а тестера (см. рис. 2) – $L_{1/4-TSC} = 16$ (вх.). Сложность объекта диагностирования оценивается величиной $L_{F(X)} = 66$ (вх.). Сложность блока вычисления функций коррекции $L_{G(X)} = 60$ (вх.). Итого сложность самопроверяемого устройства оценивается величиной $L_{CED} = 66 + 60 + 4 \cdot 6 + 16 = 166$ (вх.).

Таблица 7

Сигналы на выходах самопроверяемого устройства и тестовые комбинации элементов преобразования

Входы <X>				Выходы F(X)				Выходы G(X)				Выходы БКС				Тестовые комбинации элементов преобразования			
x ₄	x ₃	x ₂	x ₁	f ₄ (X)	f ₃ (X)	f ₂ (X)	f ₁ (X)	g ₄ (X)	g ₃ (X)	g ₂ (X)	g ₁ (X)	h ₄ (X)	h ₃ (X)	h ₂ (X)	h ₁ (X)	XOR ₄	XOR ₃	XOR ₂	XOR ₁
0	0	0	0	1	1	0	0	1	0	0	0	0	1	0	0	11	10	00	00
0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	01	00	11	00
0	0	1	0	0	0	0	1	0	0	1	1	0	0	1	0	00	00	01	11
0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	1	00	00	00	01
0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	10	00	00	00
0	1	0	1	1	1	1	1	1	1	1	0	0	0	0	1	11	11	11	10
0	1	1	0	1	1	1	1	1	1	1	0	0	0	0	1	11	11	11	10
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	00	00	00	01
1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	10	00	00	00
1	0	0	1	0	0	1	1	0	1	1	1	0	1	0	0	00	01	11	11
1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	00	00	00	01
1	0	1	1	1	1	1	0	1	1	0	0	0	0	1	0	11	11	10	00
1	1	0	0	0	0	1	0	1	0	1	0	1	0	0	0	01	00	11	00
1	1	0	1	0	1	0	1	0	1	1	1	0	0	1	0	00	11	01	11
1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	0	01	11	11	11
1	1	1	1	0	1	0	0	0	0	0	0	0	1	0	0	00	10	00	00

Если сравнивать со стандартным методом дублирования, то получаем следующую величину: $L_D = 2 \cdot 66 + 4 \cdot 1 + 3 \cdot 12 = 176$ (вх.). Здесь учтено, что требуются три инвертора с показателями сложности реализации $L_{NOT} = 1$ (вх.) и 3 модуля сжатия парафазных сигналов с показателями сложности реализации $L_{TRC} = 12$ (вх.) [16].

По показателю сложности реализации самопроверяемое устройство, построенное по предложенному методу, имеет выигрыш перед устройством, синтезированным с СВК по методу дублирования: показатель сложности реализации для нового метода составил 96,5% от показателя сложности реализации устройства с СВК на основе дублирования. Для устройств F(X) с различной конфигурацией элементов и большим числом выходов выигрыш может оказаться гораздо существеннее. Например, в табл. 8 приведены некоторые результаты экспериментов по оценке показателя структурной избыточности самопроверяемых устройств, реализованных по предложенному в статье методу (см. систему функций (17)), для тестовых комбинационных схем из набора MCNC Benchmarks [29].

Для каждой из 25 представленных в табл. 8 тестовых схем F(X) была синтезирована СВК на основе ЛКС с применением системы функций (17). Выходы F(X) разбивались последовательно в их описании в тестовой схеме, представленной в формате *.pla (аналог таблицы истинности), на группы по четыре выхода в каждой; оставшиеся при разбиении выходы выделялись в неполную группу «свободных» выходов. Для групп из четырех выходов строилась СВК на основе ЛКС с применением 1/4-кода, а «свободные» выходы контролировались на основе метода дублирования [9, 16, 18]. Контрольные выходы отдельных подсхем контроля сжимались с использованием двухвходовых модулей сжатия парафазных сигналов TRC (two-rail checkers) [30] для получения единственного контрольного выхода.

Тестовые схемы и файлы-описания элементов СВК обрабатывались с помощью известного интерпретатора SIS и библиотеки функциональных элементов stdcell2_2.genlib [31]. Это позволило определить показатели сложности структурной реализации устройств (L – площади, занимаемой устройством на

кристалле) в условных единицах библиотеки функциональных элементов stdcell2_2.genlib. Типовые элементы СВК (см. рис. 3) имеют показатели сложности реализации $L_{1/4-TSC} = 256$ (усл. ед.) и $L_{XOR} = 192$ (усл. ед.).

Таблица 8

Результаты по оценке показателей структурной избыточности самопроверяемых устройств на основе предлагаемого метода

Тестовая схема	Число входов	Число выходов	Число полных групп	Число «свободных» выходов	Формула тестера	Значения показателя площади, L (усл. ед.)						μ , %
						$F(X)$	$G(X)$	Тестеры	БКС	На основе ЛКС	На основе дублирования	
sqrt8	8	4	1	0	1/4-TSC	1 160	1 232	256	160	2 808	2 960	94,865
sao2	10	4	1	0	1/4-TSC	3 056	2 568	256	160	6 040	6 752	89,455
dist	8	5	1	1	1/4-TSC+TRC	9 976	6 448	448	160	17 032	20 800	81,885
newcwp	4	5	1	1	1/4-TSC+TRC	440	584	448	160	1 632	1 728	94,444
root	8	5	1	1	1/4-TSC+TRC	6 152	2 752	448	160	9 512	13 152	72,324
max512	9	6	1	2	1/4-TSC+2×TRC	15 016	7 224	640	160	23 040	31 088	74,112
dc1	4	7	1	3	1/4-TSC+3×TRC	808	632	832	160	2 432	2 880	84,444
dekoder	4	7	1	3	1/4-TSC+3×TRC	1 128	488	832	160	2 608	3 520	74,091
wim	4	7	1	3	1/4-TSC+3×TRC	1 104	496	832	160	2 592	3 472	74,654
newapla2	6	7	1	3	1/4-TSC+3×TRC	600	480	832	120	2 032	2 464	82,468
dc2	8	7	1	3	1/4-TSC+3×TRC	2 616	2 440	832	160	6 048	6 496	93,103
newbyte	5	8	2	0	2×1/4-TSC+TRC	592	384	704	560	2 240	2 656	84,337
mlp4	8	8	2	0	2×1/4-TSC+TRC	7 504	7 520	704	640	16 368	16 480	99,32
f51m	8	8	2	0	2×1/4-TSC+TRC	5 176	5 528	704	640	12 048	11 824	101,894
inc	7	9	2	1	2×1/4-TSC+2×TRC	3 160	1 712	896	640	6 408	8 000	80,1
dk27	9	9	2	1	2×1/4-TSC+2×TRC	1 592	768	896	640	3 896	4 864	80,099
newcpla2	7	10	2	2	2×1/4-TSC+3×TRC	1 896	1 392	1 088	640	5 016	5 680	88,31
sqr6	6	12	3	0	3×1/4-TSC+2×TRC	2 896	2 672	1 152	1 400	8 120	8 096	100,296
m1	6	12	3	0	3×1/4-TSC+2×TRC	2 208	1 096	1 152	1 440	5 896	6 720	87,738
p82	5	14	3	2	3×1/4-TSC+4×TRC	2 432	1 712	1 536	1 400	7 080	7 584	93,354
m2	8	16	4	0	4×1/4-TSC+3×TRC	7 408	4 288	1 600	2 560	15 856	17 952	88,324
m3	8	16	4	0	4×1/4-TSC+3×TRC	8 768	5 536	1 600	2 560	18 464	20 672	89,319
m4	8	16	4	0	4×1/4-TSC+3×TRC	17 216	8 344	1 600	2 560	29 720	37 568	79,11
tms	8	16	4	0	4×1/4-TSC+3×TRC	6 144	3 928	1 600	2 520	14 192	15 424	92,012
max128	7	24	6	0	6×1/4-TSC+5×TRC	13 384	4 528	2 496	4 320	24 728	31 568	78,332
Среднее значение												86,336

Для каждой тестовой схемы был вычислен относительный показатель структурной избыточности самопроверяемого устройства – доля площади, занимаемой устройством с СВК на основе ЛКС и 1/4-кода на кристалле, от площади, занимаемой устройством с СВК на основе дублирования (показатель μ , %). В среднем для 25 представленных схем он составил 86,336%. Для 17 схем показатель $\mu < 90\%$, среди которых 6 схем имеют показатель $\mu < 80\%$. Все это говорит о «хороших» показателях структурной избыточности самопроверяемых устройств, синтезируемых по разработанному методу.

Заключение

До сих пор было известно 72 способа установления зависимости между значениями функций, формируемых на выходах блоков $G(X)$ и $F(X)$, в СВК на основе ЛКС для группы из четырех выходов при контроле вычислений по 1/4-коду [27]. При этом в БКС использовалось наименьшее достаточное количество элементов преобразования – три из четырех. Результаты исследований, приведенные в настоящей статье, позволяют расширить число способов синтеза СВК на основе ЛКС с применением 1/4-кода и открыть более 4,2 млрд способов установления зависимостей между значениями функций, формируемых на выходах блоков $G(X)$ и $F(X)$, при использовании в БКС четырех элементов преобразования. Естественно, вручную получить их все не только трудоемко, но и нецелесообразно. В статье

показан принцип, позволяющий установить зависимости между значениями функций, формируемых на выходах блоков $G(X)$ и $F(X)$, с учетом обеспечения формирования проверяющих тестов всех компонентов СВК. Данный принцип открывает группу методов синтеза СВК на основе ЛКС с применением равновесного 1/4-кода.

В представленном в статье способе установления зависимости минимальная длина проверяющего теста равна восьми комбинациям из шестнадцати возможных. Таким образом, в процессе эксплуатации самопроверяемого устройства потребуется формировать на выходах объекта диагностирования половину возможных четырехбитных векторов для каждой контролируемой группы для обеспечения полной проверки СВК. Это, в свою очередь, накладывает ограничения на число входов объекта диагностирования – оно должно быть $t \geq 3$. Причем даже сам случай $t = 3$ даст возможность генерации проверяющего теста для СВК только в крайне редких случаях. На практике число входов должно быть $t \geq 4$. И чем больше входов будет у устройства, тем более просто будет сформировать проверяющий тест для СВК. Конечно, здесь требуют рассмотрения и отдельные частные случаи для устройств: например, таких устройств, на входы которых поступает неполное множество наборов входных аргументов.

Отметим также, что для различных структур объектов диагностирования использование преобразования максимального числа сигналов в БКС будет давать различные по показателям сложности СВК. Здесь нельзя однозначно сказать, будет ли эффективным применение преобразований всех сигналов или только трех из четырех [27]. В каждом конкретном случае потребуется моделирование работы устройства до разработки его самопроверяемой реализации. Преобразование четырех сигналов в БКС может дать и более простые реализации, чем преобразование только трех из них, в том числе может позволить решить задачу обеспечения самопроверяемости в том случае, если этого невозможно добиться использованием преобразования только трех сигналов из четырех.

Дальнейшие исследования применения самого принципа установления зависимостей между значениями функций, формируемых на выходах блоков $G(X)$ и $F(X)$, при организации СВК на основе ЛКС и 1/4-кода могут быть сосредоточены на поиске таких способов, в которых некоторые функции коррекции сигналов окажутся одинаковыми, что уменьшит показатели сложности реализации блока $G(X)$. Кроме того, интересными могут оказаться такие способы, которые дают равномерное (или близкое к таковому) распределение числа формируемых комбинаций на входах элементов преобразования и тестера для упрощения процедуры их тестирования в процессе эксплуатации самопроверяемого устройства. Интересными являются и исследования, связанные с обобщениями предложенного в статье принципа и использования его для других равновесных кодов, например кода «1 из 5», и сравнения с известными результатами использования данного кода для синтеза СВК на основе ЛКС [32]. В том числе могут быть учтены результаты работы [33], в которой путем коррекции всех сигналов в группе из шести выходов объекта диагностирования на основе ЛКС осуществляется контроль вычислений с применением взвешенных кодов с суммированием в кольце вычетов по модулю $M = 4$. Здесь уже рассмотрено применение разделимых блоковых кодов при синтезе СВК на основе ЛКС. К слову, число способов синтеза СВК при использовании разделимых кодов также велико, что позволяет синтезировать различные по своим характеристикам самопроверяемые устройства [34].

В заключение отметим, что для построения полностью самопроверяемого устройства потребуется также покрыть ошибки, вызываемые неисправностями из рассматриваемой модели, на выходах объекта диагностирования с помощью 1/4-кода. Это требует исключения всех двукратных симметричных ошибок, возникающих в векторе $\langle f_4(X) f_3(X) f_2(X) f_1(X) \rangle$. Принципам выделения групп выходов, исключаяющих такого рода ошибки, а также преобразованию исходных структур комбинационных устройств в контролепригодные структуры с учетом данного свойства посвящены статьи [35, 36], развивающие идеи работ [37–40].

Использование представленного в статье принципа установления зависимостей между значениями функций, формируемых на выходах блоков $G(X)$ и $F(X)$, при организации СВК на основе ЛКС и 1/4-кода перспективно при построении самопроверяемых дискретных устройств на различной, в том числе программируемой, элементной базе.

Список источников

1. Kharchenko V., Kondratenko Yu., Kacprzyk J. Green IT Engineering: Concepts, Models, Complex Systems Architectures. Springer, 2017. 305 p. doi: 10.1007/978-3-319-44162-7 (Springer Book series "Studies in Systems, Decision and Control"; v. 74).
2. Mikoni S. Top Level Diagnostic Models of Complex Objects // Lecture Notes in Networks and Systems. 2022. V. 442. P. 238–249. doi: 10.1007/978-3-030-98832-6_21
3. Gharibi W., Hahanov V., Chumachenko S., Litvinova E., Hahanov I., Hahanova I. Vector-Logic Computing for Faults-As-Address Deductive Simulation // IAES International Journal of Robotics and Automation (IJRA). 2023. V. 12 (3). P. 274–288. doi: 10.11591/ijra.v12i3.pp274-288
4. Hahanov V., Litvinova E., Davitadze Z., Chumachenko S., Devadze Z., Abdullaev V.H. Truth Table Based Intelligent Computing // 2024 31st International Conference on Mixed Design of Integrated Circuits and System (MIXDES), 27–28 June 2024. Gdansk, 2024. P. 199–204. doi: 10.23919/MIXDES62605.2024.10614035
5. Ubar R., Raik J., Jenihhin M., Jutman A. Structural Decision Diagrams in Digital Test: Theory and Applications. Birkhäuser, 2024, XIII + 595 p. doi: 10.1007/978-3-031-44734-1 (Computer Science Foundations and Applied Logic).
6. Nicolaidis M., Zorian Y. On-Line Testing for VLSI – A Compendium of Approaches // Journal of Electronic Testing: Theory and Applications. 1998. V. 12. P. 7–20. doi: 10.1023/A:1008244815697
7. Mitra S., McCluskey E.J. Which Concurrent Error Detection Scheme to Choose? // Proc. of International Test Conference. 2000. USA. Atlantic City, NJ. 03–05 October 2000. P. 985–994. doi: 10.1109/TEST.2000.894311
8. Sahana A.R., Chiraag V., Suresh G., Thejaswini P., Nandi S. Application of Error Detection and Correction Techniques to Self-Checking VLSI Systems: An Overview // Proc. of 2023 IEEE Guwahati Subsection Conference (GCON). Guwahati. 2023. doi: 10.1109/GCON58516.2023.10183449
9. Согомоян Е.С., Слабаков Е.В. Самопроверяемые устройства и отказоустойчивые системы. М. : Радио и связь, 1989. 208 с.
10. Dmitriev A., Saposhnikov V., Saposhnikov V., Goessel M. New Self-Dual Circuits for Error Detection and Testing // VLSI Design. 2000. V. 11, is. 1. P. 1–21. doi: 10.1155/2000/84720
11. Ефанов Д.В. Особенности реализации самопроверяемых структур на основе метода инвертирования данных и линейных кодов // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2023. № 65. С. 126–138. doi: 10.17223/19988605/65/13
12. Das D., Toubia N.A. Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes // Journal of Electronic Testing: Theory and Applications. 1999. V. 15, is. 1-2. P. 145–155. doi: 10.1023/A:1008344603814
13. Tshagharyan G., Harutyunyan G., Shoukourian S., Zorian Y. Experimental Study on Hamming and Hsiao Codes in the Context of Embedded Applications // Proc. of 15th IEEE East-West Design & Test Symposium (EWDTS'2017). Novi Sad, Serbia. Sept. 29–Oct. 2. 2017. P. 25–28. doi: 10.1109/EWDTS.2017.8110065
14. Ефанов Д.В., Погодина Т.С. Исследование свойств самодвойственных комбинационных устройств с контролем вычислений на основе кодов Хэмминга // Информатика и автоматизация. 2023. Т. 22, № 2. С. 349–392. doi: 10.15622/ia.22.2.5
15. Ефанов Д.В. Особенности использования кодов Хэмминга при синтезе самопроверяемых цифровых устройств на основе метода инвертирования данных // Известия высших учебных заведений. Электроника. 2024. Т. 29, № 3. С. 379–392. doi: 10.24151/1561-5405-2024-29-3-379-392
16. Сапожников В.В., Сапожников Вл.В. Самопроверяемые дискретные устройства. СПб. : Энергоатомиздат, 1992. 224 с.
17. Piestrak S.J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław : Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 s.
18. Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking. Ed. 1. Dordrecht : Springer Science + Business Media B.V., 2008. 184 p.
19. Сапожников В.В., Сапожников Вл.В., Дмитриев А.В., Морозов А.В., Гессель М. Организация функционального контроля комбинационных схем методом логического дополнения // Электронное моделирование. 2002. Т. 24, № 6. С. 52–66.
20. Das D.K., Roy S.S., Dmitriev A., Morozov A., Gössel M. Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes // Proc. of the 10th International Workshops on Boolean Problems. Freiburg. Germany. September, 2012. P. 33–40.
21. Гессель М., Морозов А.В., Сапожников В.В., Сапожников Вл.В. Контроль комбинационных схем методом логического дополнения // Автоматика и телемеханика. 2005. № 8. С. 161–172.
22. Goessel M., Saposhnikov V., Saposhnikov V., Dmitriev A. A New Method for Concurrent Checking by Use of a 1-out-of-4 Code // Proc. of the 6th IEEE International On-line Testing Workshop, 3–5 July 2000. Palma de Mallorca, Spain. P. 147–152.
23. Пархоменко П.П., Согомоян Е.С. Основы технической диагностики (оптимизация алгоритмов диагностирования, аппаратные средства). М. : Энергоатомиздат, 1981. 320 с.
24. Аксёнова Г.П. Необходимые и достаточные условия построения полностью проверяемых схем свертки по модулю 2 // Автоматика и телемеханика. 1979. № 9. С. 126–135.
25. Ефанов Д.В. Некоторые особенности обнаружения ошибок равномерными неразделимыми кодами // Известия высших учебных заведений. Приборостроение. 2019. Т. 62, № 7. С. 621–631. doi: 10.17586/0021-3454-2019-62-7-621-631
26. Saposhnikov V.V., Saposhnikov V.V., Morozov A., Goessel M., Osadchy G. Design of totally self-checking combinational circuits by use of complementary circuits // Proc. of 2th IEEE East-West Design & Test Symposium (EWDTS'2004). Crimea, Ukraine. Sept. 15–17, 2004. P. 83–87.

27. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Пивоваров Д.В. Метод логического дополнения на основе равновесного кода «1 из 4» для построения полностью самопроверяемых структур систем функционального контроля // Электронное моделирование. 2017. Т. 39, № 2. С. 15–34.
28. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Пивоваров Д.В. Организация контроля комбинационных схем на основе метода логического дополнения до равновесного кода «1 из 4» // Известия высших учебных заведений. Приборостроение. 2018. Т. 61, № 12. С. 1025–1035. doi: 10.17586/0021-3454-2018-61-12-1025-1035
29. Collection of Digital Design Benchmarks. URL: <https://ddd.fit.cvut.cz/www/prj/Benchmarks/> (accessed: 07.11.2024).
30. Carter W.C., Duke K.A., Schneider P.R. Self-Checking Error Checker for Two-Rail Coded Data // United States Patent Office. Filed July 25. 1968; ser. no. 747533; patented Jan. 26, 1971. New York. 10 p.
31. Sentovich E.M., Singh K.J., Moon C., Savoj H., Brayton R.K., Sangiovanni-Vincentelli A. Sequential Circuit Design Using Synthesis and Optimization // Proc. IEEE International Conference on Computer Design: VLSI in Computers & Processors. 11–14 Oct. 1992. Cambridge, MA, USA. P. 328–333. doi: 10.1109/ICCD.1992.276282
32. Пивоваров Д.В. Организация систем функционального контроля комбинационных логических схем на основе метода логического дополнения по равновесному коду «1 из 5» // Автоматика на транспорте. 2017. Т. 3, № 4. С. 605–624.
33. Ефанов Д.В., Елина Е.И. Синтез самопроверяемых цифровых устройств на основе логической коррекции сигналов с применением взвешенных кодов Боуза-Лина // Проблемы управления. 2024. № 4. С. 26–43. doi: 10.25728/ru.2024.4.3
34. Efanov D.V., Yelina Y.I. Investigation of Ways of Synthesizing Concurrent Error-Detection Circuits Based on Boolean Signal Correction Using Uniform Separable Codes // Russian Microelectronics. 2024. V. 53 (5). P. 471–482. doi: 10.1134/S1063739724600456
35. Efanov D.V., Sapozhnikov V.V., Sapozhnikov V.I. Organization of a Fully Self-Checking Structure of a Combinational Device Based on Searching for Groups of Symmetrically Independent Outputs // Automatic Control and Computer Sciences. 2020. V. 54, is. 4. P. 279–290. doi: 10.3103/S0146411620040045
36. Ефанов Д.В. Синтез самопроверяемых вычислительных устройств на основе полной системы особых групп выходов объекта диагностирования // Известия высших учебных заведений. Приборостроение. 2023. Т. 66, № 5. С. 355–372. doi: 10.17586/0021-3454-2023-66-5-355-372
37. Sogomonyan E.S., Gössel M. Design of Self-Testing and On-Line Fault Detection Combinational Circuits with Weakly Independent Outputs, Journal of Electronic Testing: Theory and Applications. 1993. V. 4, is. 4. P. 267–281. doi: 10.1007/BF00971975
38. Matrosova A.Yu., Ostanin S.A. Self-Checking Synchronous Sequential Circuit Design for Unidirectional Error // Proc. of the IEEE European Test Workshop (ETW'98), 27–29 May 1998, Sitges, Barcelona, Spain.
39. Saposhnikov V.V., Morosov A., Saposhnikov V.I., Gössel M. A New Design Method for Self-Checking Unidirectional Combinational Circuits // Journal of Electronic Testing: Theory and Applications. 1998. V. 12, is. 1-2. P. 41–53. doi: 10.1023/A:1008257118423
40. Morosow A, Saposhnikov V.V., Saposhnikov V.I., Goessel M. Self-Checking Combinational Circuits with Unidirectionally Independent Outputs // VLSI Design. 1998. V. 5, is. 4. P. 333–345. doi: 10.1155/1998/20389

References

1. Kharchenko, V., Kondratenko, Yu. & Kacprzyk, J. (2017) *Green IT Engineering: Concepts, Models, Complex Systems Architectures*. Springer. DOI: 10.1007/978-3-319-44162-7
2. Mikoni, S. (2022) Top Level Diagnostic Models of Complex Objects. *Lecture Notes in Networks and Systems*. 442. pp. 238–249. DOI: 10.1007/978-3-030-98832-6_21
3. Gharibi, W., Hahanov, V., Chumachenko, S., Litvinova, E., Hahanov, I. & Hahanova, I. (2023) Vector-Logic Computing for Faults-As-Address Deductive Simulation. *IAES International Journal of Robotics and Automation (IJRA)*. 12(3). pp. 274–288. DOI: 10.11591/ijra.v12i3.pp274-288
4. Hahanov, V., Litvinova, E., Davitadze, Z., Chumachenko, S., Devadze, Z. & Abdullaev, V.H. (2024) Truth Table Based Intelligent Computing. *31st International Conference on Mixed Design of Integrated Circuits and System (MIXDES)*. June 27–28, 2024. Gdansk, Poland. DOI: 10.23919/MIXDES62605.2024.10614035
5. Ubar, R., Raik, J., Jenihhin, M. & Jutman, A. (2024) *Structural Decision Diagrams in Digital Test: Theory and Applications*, Birkhäuser: Springer. DOI: 10.1007/978-3-031-44734-1
6. Nicolaidis, M. & Zorian, Y. (1998) On-Line Testing for VLSI – A Compendium of Approaches. *Journal of Electronic Testing: Theory and Applications*. 12. pp. 7–20. DOI: 10.1023/A:1008244815697
7. Mitra, S. & McCluskey, E.J. (2000) Which Concurrent Error Detection Scheme to Choose? *Proc. of International Test Conference*. USA, Atlantic City, NJ, October 3–5, 2000. pp. 985–994. DOI: 10.1109/TEST.2000.894311
8. Sahana, A.R., Chiraag, V., Suresh, G., Thejaswini, P. & Nandi, S. (2023) Application of Error Detection and Correction Techniques to Self-Checking VLSI Systems: An Overview. *Proceedings of 2023 IEEE Guwahati Subsection Conference (GCON)*. Guwahati. DOI: 10.1109/GCON58516.2023.10183449
9. Sogomonyan, E.S. & Slabakov, E.V. (1989) *Samoproveryaemye ustroystva i otkazoustoychivyye sistemy* [The Self-Checking Devices and Fault-Tolerant Systems]. Moscow: Radio i svyaz'.
10. Dmitriev, A., Saposhnikov, V., Saposhnikov, V. & Goessel, M. (2000) New Self-Dual Circuits for Error Detection and Testing. *VLSI Design*. 11(1). pp. 1–21. DOI: 10.1155/2000/84720

11. Efanov, D.V. (2023) The Self-Checking Structures Implementation Features Based on the Inverting Data and Linear Block Code Method. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 65. pp. 126–138. DOI: 10.17223/19988605/65/13
12. Das, D. & Touba, N.A. (1999) Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes. *Journal of Electronic Testing: Theory and Applications*. 15(1-2). pp. 145–155. DOI: 10.1023/A:1008344603814
13. Tshagharyan, G., Harutyunyan, G., Shoukourian, S. & Zorian, Y. (2017) Experimental Study on Hamming and Hsiao Codes in the Context of Embedded Applications. *Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017)*. Novi Sad, Serbia. September 29 – October 2, 2017. pp. 25–28. DOI: 10.1109/EWDTS.2017.8110065
14. Efanov, D.V. & Pogodina, T.S. (2023) Issledovanie svoystv samodvoystvennykh kombinatsionnykh ustroystv s kontrol'em vychisleniy na osnove kodov Khemminga [Properties Investigation of Self-Dual Combinational Devices with Calculation Control Based on Hamming Codes]. *Informatika i avtomatizatsiya*. 22(2). pp. 349–392. DOI: 10.15622/ia.22.2.5
15. Efanov, D.V. (2024) Osobennosti ispol'zovaniya kodov Khemminga pri sinteze samoproveryaemykh tsifrovyykh ustroystv na osnove metoda invertirovaniya dannykh [Special Aspects of Hamming Codes Use in the Self-Checking Digital Devices Synthesis Based on the Data Inversion Method]. *Izvestiya vysshikh uchebnykh zavedeniy. Elektronika*. 29(3). pp. 379–392. DOI: 10.24151/1561-5405-2024-29-3-379-392
16. Sapozhnikov, V.V. & Sapozhnikov, V.I. (1992) *Samoproveryaemye diskretnye ustroystva* [Self-Checking Digital Devices]. St. Petersburg: Energoatomizdat.
17. Piestrak, S.J. (1995) *Design of Self-Testing Checkers for Unidirectional Error Detecting Codes*. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej.
18. Göessel, M., Ocheretny, V., Sogomonyan, E. & Marienfeld, D. (2008) *New Methods of Concurrent Checking*. Dordrecht: Springer.
19. Sapozhnikov, V.V., Sapozhnikov, V.I., Dmitriev, A.V., Morozov, A.V. & Goessel, M. (2002) Organizatsiya funktsional'nogo kontrolya kombinatsionnykh skhem metodom logicheskogo dopolneniya [Organization of Functional Control of Combinational Circuits by the Method of Boolean Complement]. *Elektronnoe modelirovanie*. 24(6). pp. 52–66.
20. Das, D.K., Roy S.S., Dmitriev, A., Morozov, A. & Gössel, M. (2012) 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*. Freiburg, Germany. September. pp. 33–40.
21. Goessel, M., Morozov, A.V., Sapozhnikov, V.V. & Sapozhnikov, V.I. (2005) Checking Combinational Circuits by the Method of Logic Complement. *Automation and Remote Control*. 8. pp. 161–172.
22. Goessel, M., Saposhnikov, V.I., Saposhnikov, V. & Dmitriev, A. (2000) A New Method for Concurrent Checking by Use of a 1-out-of-4 Code. *Proceedings of the 6th IEEE International On-line Testing Workshop*. July 3–5, 2000. Palma de Mallorca, Spain. pp. 147–152.
23. Parkhomenko, P.P. & Sogomonyan, E.S. (1981) *Osnovy tekhnicheskoy diagnostiki (optimizatsiya algoritmov diagnostirovaniya, apparaturnye sredstva)* [Fundamentals of technical diagnostics (optimization of diagnostic algorithms, hardware)]. Moscow: Energoatomizdat.
24. Aksenova, G.P. (1979) Necessary and Sufficient Conditions for Design of Completely Checkable Modulo 2 Convolution Circuits. *Automation and Remote Control*. 9. pp. 126–135.
25. Efanov, D.V. (2019) Some Features of Error Detection by Uniform Indivisible Codes. *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie – Journal of Instrument Engineering*. 62(7). pp. 621–631. DOI: 10.17586/0021-3454-2019-62-7-621-631
26. Saposhnikov, V.V., Saposhnikov, V.I., Morozov, A., Goessel, M. & Osadchy, G. (2004) Design of totally self-checking combinational circuits by use of complementary circuits. *Proceedings of 2th IEEE East-West Design & Test Symposium (EWDTS'2004)*. Crimea, Ukraine. September 15–17. pp. 83–87.
27. Sapozhnikov, V.V., Sapozhnikov, V.I., Efanov, D.V. & Pivovarov, D.V. (2017) Boolean Complement Method Based on Constant-Weight Code "1-out-of-4" for Formation of Totally Self-Checking Concurrent Error Detection Systems. *Elektronnoe modelirovanie*. 39(2). pp. 15–34.
28. Sapozhnikov, V.V., Sapozhnikov, V.I., Efanov, D.V. & Pivovarov, D.V. (2018) Organization of Control of Combinational Circuits Based on the Method of Logical Complement to Equilibrium 1-out-of-4 Code. *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie – Journal of Instrument Engineering*. vol. 61. is. 12. pp. 1025–1035. DOI: 10.17586/0021-3454-2018-61-12-1025-1035
29. *Collection of Digital Design Benchmarks*. [Online] Available from: <https://ddd.fit.cvut.cz/www/prj/Benchmarks/> (Accessed: 7th November 2024).
30. Carter, W.C., Duke, K.A., & Schneider, P.R. (1971) *Self-Checking Error Checker for Two-Rail Coded Data*. United States Patent Office, filed July 25, 1968, ser. No. 747533. Patented Jan. 26. New York.
31. Sentovich, E.M., Singh, K.J., Moon, C., Savoj, H., Brayton, R.K. & Sangiovanni-Vincentelli, A. (1992) Sequential Circuit Design Using Synthesis and Optimization. *Proceedings IEEE International Conference on Computer Design: VLSI in Computers & Processors*. October 11–14, 1992. Cambridge, MA, USA. pp. 328–333. DOI: 10.1109/ICCD.1992.276282
32. Pivovarov, D.V. (2017) Organization of Concurrent Error-Detection Systems of Combinational Logical Circuits on the Basis of Boolean Complement Method with "1-out-of-5" Constant-Weight Code. *Avtomatika na transporte – Transport Automation Research*. 3(4). pp. 605–624.
33. Efanov, D.V. & Yelina, Y.I. (2024) Design of Self-Checking Digital Devices with Boolean Signals Correction Using Weight-Based Bose-Lin Codes. *Problemy upravleniya*. 4. pp. 26–43. DOI: 10.25728/pu.2024.4.3

34. Efanov, D.V. & Yelina, Y.I. (2024) Investigation of Ways of Synthesizing Concurrent Error-Detection Circuits Based on Boolean Signal Correction Using Uniform Separable Codes. *Russian Microelectronics*. 53(5). pp. 471–482. DOI: 10.1134/S1063739724600456
35. Efanov, D.V., Sapozhnikov, V.V. & Sapozhnikov, V.I. (2020) Organization of a Fully Self-Checking Structure of a Combinational Device Based on Searching for Groups of Symmetrically Independent Outputs. *Automatic Control and Computer Sciences*. 54(4). pp. 279–290. DOI: 10.3103/S0146411620040045
36. Efanov, D.V. (2023) Synthesis of Self-Checking Computing Devices Based on a Complete System of Special Groups of the Diagnostic Object Outputs. *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie – Journal of Instrument Engineering*. 66(5). pp. 355–372. DOI: 10.17586/0021-3454-2023-66-5-355-372
37. Sogomonyan, E.S. & Gössel, M. (1993) Design of Self-Testing and On-Line Fault Detection Combinational Circuits with Weakly Independent Outputs. *Journal of Electronic Testing: Theory and Applications*. 4(4). pp. 267–281. DOI: 10.1007/BF00971975
38. Matrosova, A.Yu. & Ostanin, S.A. (1998) Self-Checking Synchronous Sequential Circuit Design for Unidirectional Error. *Proceedings of the IEEE European Test Workshop (ETW'98)*. May 27–29, 1998. Sitges, Barcelona, Spain.
39. Saposhnikov, V.V., Morosov, A., Saposhnikov, V.I. & Gössel, M. (1998) A New Design Method for Self-Checking Unidirectional Combinational Circuits. *Journal of Electronic Testing: Theory and Applications*. 12(1-2). pp. 41–53. DOI: 10.1023/A:1008257118423
40. Morosow, A., Saposhnikov, V.V., Saposhnikov, V.I. & Goessel, M. (1998) Self-Checking Combinational Circuits with Unidirectionally Independent Outputs. *VLSI Design*. 5(4). pp. 333–345. DOI: 10.1155/1998/20389

Информация об авторе:

Ефанов Дмитрий Викторович – доктор технических наук, профессор, действительный член Международной Академии транспорта, член Института инженеров электротехники и электроники (IEEE); профессор Высшей школы транспорта Института машиностроения, материалов и транспорта, профессор Высшей школы управления кибер-физическими системами Санкт-Петербургского политехнического университета Петра Великого (Санкт-Петербург, Россия); профессор кафедры автоматизации, телемеханики и связи на железнодорожном транспорте Российского университета транспорта (Москва, Россия); профессор кафедры автоматизации и телемеханики Ташкентского государственного транспортного университета (Ташкент, Узбекистан). E-mail: TrES-4b@yandex.ru

Автор заявляет об отсутствии конфликта интересов.

Information about the author:

Efanov Dmitry V. (Doctor of Technical Sciences, Professor, Full-member of the International Transport Academy, IEEE member, Professor of Transport Higher School of Mechanical Engineering, Material and Transport Institute and Professor at Peter the Great Saint Petersburg Polytechnic University, St. Petersburg, Russian Federation; Professor of Automation, Remote Control and Communication on Railway Transport Department, Russian University of Transport Moscow, Russian Federation; Professor of Automation and Remote Control Department of Tashkent State Transport University, Tashkent, Uzbekistan). E-mail: TrES-4b@yandex.ru

The author declares no conflicts of interests.

Поступила в редакцию 15.03.2025; принята к публикации 02.09.2025

Received 15.03.2025; accepted for publication 02.09.2025