

ИССЛЕДОВАНИЕ АЛГОРИТМОВ СИНТЕЗА САМОПРОВЕРЯЕМЫХ ЦИФРОВЫХ УСТРОЙСТВ НА ОСНОВЕ ЛОГИЧЕСКОЙ КОРРЕКЦИИ СИГНАЛОВ С ПРИМЕНЕНИЕМ ВЗВЕШЕННЫХ КОДОВ БОУЗА — ЛИНА

ЕФАНОВ Дмитрий Викторович, д-р техн. наук, профессор, действительный член Международной Академии транспорта, член Института инженеров электротехники и электроники, профессор^{1,3,4}, заместитель генерального директора по научно-исследовательской работе²; e-mail: TrES-4b@yandex.ru
ЕЛИНА Есения Игоревна, аспирант¹; e-mail: eseniya-elina@mail.ru

¹ Санкт-Петербургский политехнический университет Петра Великого, Институт машиностроения, материалов и транспорта, Высшая школа транспорта, г. Санкт-Петербург

² Научно-исследовательский и проектный институт «Транспортной и строительной безопасности» (ООО «НИПИ «ТрансСтройбезопасность»), г. Санкт-Петербург

³ Российский университет транспорта (МИИТ), кафедра «Автоматика, телемеханика и связь на железнодорожном транспорте», г. Москва

⁴ Ташкентский государственный транспортный университет, кафедра «Автоматика и телемеханика», г. Ташкент

Предложено при синтезе самопроверяемых цифровых устройств на основе логической коррекции сигналов применять взвешенные коды Боуза — Лина, принципы построения которых подразумевают предварительное взвешивание информационных символов натуральными числами. Предложены две базовые структуры для синтеза схем встроенного контроля для групп из шести выходов объекта диагностирования. Структуры основаны на использовании взвешенных кодов Боуза — Лина с суммированием в кольце вычетов по модулю $M=4$. Таких помехозащищенных кодов с числом информационных символов $m=4$ существует 15, что позволяет выбирать в качестве базового кода в схеме встроенного контроля наилучший вариант по различным критериям, в том числе добиваться достижения свойства самопроверяемости даже в случаях, когда этого невозможно получить путем использования традиционных подходов, включая дублирование. Разработаны два алгоритма синтеза схем встроенного контроля на основе логической коррекции сигналов, позволяющие использовать коррекцию только двух из шести функций в базовой структуре. Для базовых структур существует 720 способов построения схемы встроенного контроля на основе логической коррекции сигналов с применением каждого взвешенного кода Боуза — Лина, что дает возможность выбора наилучшего способа реализации самопроверяемого устройства с учетом различных показателей (структурной избыточности, контролепригодности и пр.). Работа алгоритмов продемонстрирована на простых примерах. Приведены результаты экспериментов с тестовыми цифровыми схемами из набора *MCNC Benchmarks*, подтверждающие эффективность разработанных алгоритмов. Показано, что при большом числе выходов существует астрономическое количество способов организации схем встроенного контроля, что дает возможность построения самопроверяемых устройств с различными характеристиками. Логическая коррекция сигналов с применением взвешенных кодов Боуза — Лина может использоваться при разработке и проектировании самопроверяемых цифровых устройств на различной элементной базе.

Ключевые слова: тестирование цифровых схем, самопроверяемое устройство, схема встроенного контроля, логическая коррекция сигналов, взвешенный код с суммированием, взвешенный код Боуза — Лина.

DOI: 10.20295/2412-9186-2024-10-01-74-99

▼ Введение

Синтезу самопроверяемых цифровых устройств на основе двоичных избыточных кодов посвящено большое число публикаций. В основном рассматривались вопросы синтеза самопроверяемых устройств путем дополнения в схемах встроенного контроля (СВК)

рабочих функций, вычисляемых объектом диагностирования, некоторыми контрольными функциями для последующей проверки корректности вычислений по диагностическому признаку принадлежности формируемых кодовых слов заранее выбранному равномерно-му коду. Исследователями затрагивались как

особенности равномерных кодов, проявляющиеся при синтезе СВК [1–5], так и аспекты синтеза контрольного оборудования (детекторов, кодеров и тестеров кодов) [6–9]. Применение при синтезе СВК подхода, связанного с логической коррекцией сигналов (ЛКС) от объекта диагностирования, освещается в гораздо меньшем числе научных публикаций. Однако, по мнению авторов, ЛКС было уделено недостаточное внимание, поскольку она позволяет проектировщику самопроверяемого устройства получить ряд преимуществ перед традиционным подходом, связанным с дополнением в СВК рабочих функций некоторыми контрольными. Эти преимущества заключаются прежде всего в большей гибкости в обеспечении самопроверяемости структуры синтезируемого устройства и возможности влияния на показатели сложности технической реализации, поскольку в отличие от традиционного подхода ЛКС дает возможность для одного равномерного кода синтезировать большое количество СВК [10–13].

У ЛКС есть свои недостатки, среди которых следует выделить невозможность применения методов синтеза СВК с учетом поиска особых групп выходов по их видам (монотонно независимые или монотонно и асимметрично независимые) и преобразования структур объектов диагностирования в такие, которые будут обладать особыми свойствами (являться единой группой монотонно независимых или монотонно и асимметрично независимых) [14–16]. Такая особенность как раз и связана с применением при ЛКС немонотонного преобразования с использованием сумматоров по модулю $M=2$. Однако, как показывают исследования авторов, эффективным оказывается контроль по группам H^p -независимых выходов (p — максимальная кратность обнаруживаемых ошибок в группе выходов) и контроль устройств, выходы которых образуют единую H^p -независимую группу [17].

Можно выделить четыре основных направления исследований ЛКС при синтезе СВК. Первое направление связано с применением при синтезе СВК на основе ЛКС неразделимых кодов и учетом свойств отдельных видов кодов для обнаружения неисправностей, обеспечения тестируемости элементов в СВК и снижения

показателей сложности технической реализации [18, 19]. Второе направление связано с использованием при синтезе СВК на основе ЛКС диагностического признака принадлежности формируемых функций особым классам булевых функций, например самодвойственным [20, 21]. Третье направление связано с изучением особенностей применения разделимых кодов при синтезе СВК на основе ЛКС. Здесь можно отметить одну из первых работ [22]. Внимание именно этому направлению в литературе уделено действительно мало, по всей видимости, по причине недостаточно глубоких исследований способов формирования простых кодов с малой избыточностью и характеристик обнаружения ими ошибок как в информационных символах, так и в символах всего кодового слова. Четвертое направление связано с комбинацией свойств неразделимых и разделимых кодов, а также с контролем принадлежности формируемых в СВК функций к самодвойственным и «близким» к ним классам булевых функций [23, 24].

Настоящая статья посвящена развитию третьего обозначенного направления в исследованиях ЛКС и погружает читателя в особенности использования свойств взвешенных кодов Боуза — Лина для синтеза на их основе СВК.

1. Особенности применения двоичных избыточных кодов при синтезе схем встроенного контроля на основе логической коррекции сигналов

Структура организации СВК на основе ЛКС с применением двоичных избыточных кодов изображена на рис. 1. Объект диагностирования $F(x)$, имеющий t входов и n выходов и формирующий на каждом t -разрядном входном булевом векторе n -разрядный выходной булев вектор, снабжается СВК, состоящей из трех функциональных блоков. Блок коррекции сигналов (БКС) предназначен для преобразования функций, вычисляемых на рабочих выходах объекта диагностирования, в функции h_1, h_2, \dots, h_n , определяющие символы кодовых слов заранее выбранного двоичного избыточного кода. В качестве элементов преобразования используются двухвходовые сумматоры по модулю $M=2$ (элементы XOR). К первым

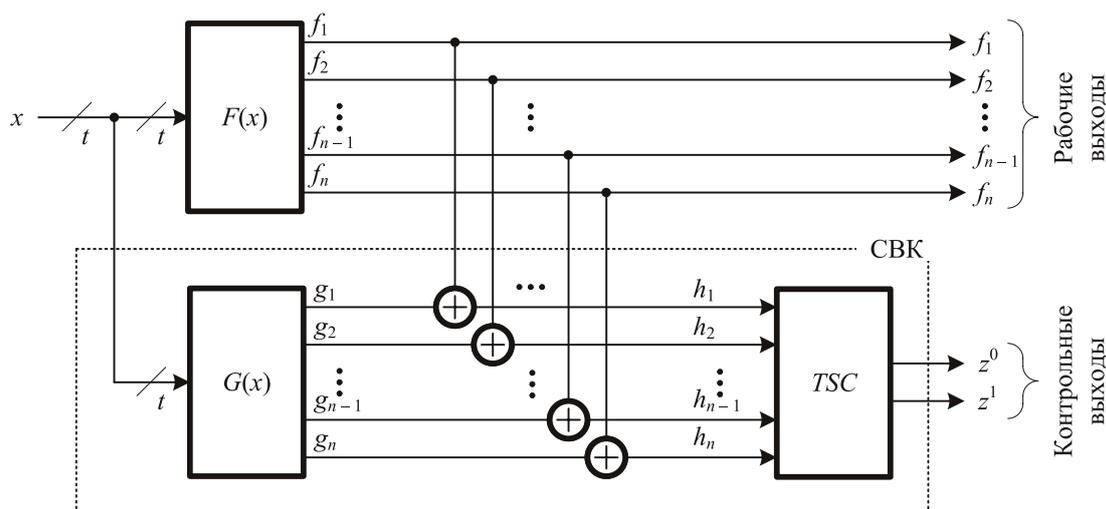


Рис. 1. Структура организации СВК на основе ЛКС

входам элементов XOR подключены рабочие выходы объекта диагностирования. Ко вторым входам подключены выходы блока $G(x)$, представляющего собой блок контрольной логики. Благодаря этому блоку и осуществляется вычисление функций логической коррекции: $g_i = f_i \oplus h_i, i = 1, n$. Таким образом, при поступлении входного вектора $\langle x_i, x_{i-1} \dots x_2, x_1 \rangle$ на выходах БКС формируется вектор $\langle h_n, h_{n-1} \dots h_2, h_1 \rangle$, принадлежащий выбранному на этапе проектирования СВК двоичному избыточному коду. Этот контроль осуществляется с помощью самопроверяемого тестера TSC (*totally self-checking checker*) [1, 6, 7]. TSC имеет два контрольных выхода z^0 и z^1 , на которых формируется парафазный сигнал контроля. При наличии неисправностей в объекте диагностирования или в одном из блоков СВК парафазность выходов z^0 и z^1 нарушается, что свидетельствует о наличии ошибки.

Обратимся именно к ветви теории синтеза СВК на основе ЛКС при использовании двоичных избыточных кодов. Наиболее глубоко исследованы вопросы применения равновесных кодов [25] при синтезе СВК на основе ЛКС. Свойства обнаружения ошибок, а также теория синтеза самопроверяемых тестеров данных кодов на различной элементной базе хорошо изучены, что позволяет эффективно использовать их для построения цифровых устройств.

Множество кодовых слов равновесного кода, или кода « r из n », образуют все кодовые слова с заданным числом n символов с одина-

ковым весом r (с одинаковым числом r единичных символов). Множество кодовых слов кода « r из n » образовано C_n^r кодовыми словами с одинаковым весом. Поэтому среди всего разнообразия видов ошибок с различными кратностями d такими кодами не обнаруживаются только симметричные ошибки в кодовых словах.

Ошибки в кодовых словах классифицируются на одиночные, монотонные, симметричные и асимметричные [5]. Одиночные ошибки связаны с однократными искажениями. Монотонные ошибки связаны с возникновением искажений только нулевых или только единичных символов в кодовых словах. Симметричные ошибки — это разнонаправленные ошибки четной кратностью d , в которых число искажений нулевых символов равно числу искажений единичных символов. Асимметричные ошибки — это разнонаправленные ошибки кратностью $d \geq 3$, в которых содержится неравное количество искажений нулевых и единичных символов. Различные равновесные коды обладают своими характеристиками обнаружения ошибок. Например, класс кодов «1 из n » не обнаруживает только двукратные симметричные ошибки, а все остальные их виды обнаруживает [26].

Поскольку свойства равновесных кодов хорошо изучены, в литературе известны методы синтеза самопроверяемых СВК на их основе путем выделения групп независимых и монотонно независимых выходов или преобразо-

вания структур объектов диагностирования в контролепригодные структуры с монотонно независимыми выходами [14]. Однако данные методы напрямую не применимы при синтезе СВК на основе ЛКС, поскольку сама коррекция осуществляется с использованием элементов, реализующих немонотонные функции.

При синтезе СВК на основе ЛКС можно контролировать сразу же все выходы объекта диагностирования с помощью одного равновесного кода, а можно использовать контроль вычислений по группам, применяя базовые структуры СВК для контроля устройств с малым числом выходов (как правило, до 10). Это обусловлено особенностями обеспечения самопроверяемости СВК на основе ЛКС и необходимостью формирования полного теста для элементов преобразования и тестера — равновесные коды с малыми значениями n имеют более простые тестеры, требующие меньшего количества комбинаций для полной их проверки [6].

В [22] предложено использовать для контроля вычислений на основе ЛКС коды Бергера [27]. При этом рассматривается построение базовой структуры для групп, состоящих из пяти выходов объектов диагностирования — три выхода не преобразуются и формируют информационные символы кода Бергера, а два оставшихся — преобразуются и формируют проверочные символы данного кода. Особенностью именно данного кода Бергера является то, что при числе информационных символов $m=3$ формируется полное множество контрольных векторов с $k=2$ проверочными символами, что позволяет обеспечивать полную проверку тестера. Более того, тестер данного кода Бергера имеет довольно простую структуру — он состоит из полного сумматора *FA* (*full-adder*) и модуля сжатия парафазных сигналов *TRC* (*two-rail checker*) [28]. Кодами Бергера, также как и равновесными кодами, не обнаруживаются только симметричные ошибки в информационных векторах. Однако, как показано в [29], ими могут не обнаруживаться ошибки любых видов и кратностей во всех символах кодовых слов. Поэтому при синтезе самопроверяемых СВК на основе данных кодов также нельзя использовать широко известные методы, подразумевающие контроль

вычислений по группам монотонно и группам монотонно и асимметрично независимых выходов.

В [30] учтена особенность кода Бергера, состоящая в том, что при условии $m \neq 2^j - 1$, $j \in \mathbb{N}$, $j \geq 2$ не формируется полное множество контрольных векторов с k проверочными символами, и предложено использовать модульные коды с суммированием для синтеза СВК для устройств с произвольным количеством выходов. При построении данных кодов устанавливается модуль $M \in \{2^1, 2^2, \dots, 2^{\lceil \log_2(m+1) \rceil - 1}\}$, а затем осуществляется суммирование единичных разрядов в кольце вычетов по заданному модулю. Число проверочных символов у модульного кода с суммированием определяется величиной $k = \log_2 M$. Поэтому среди множества выходов объекта диагностирования выделяется k выходов, участвующих в формировании проверочных символов, которые получаются при использовании логического преобразования в СВК, а оставшиеся $n-k$ выходов определяют число информационных символов кода. Свойства модульных кодов с суммированием схожи со свойствами кодов Бергера, однако ими не обнаруживается большее количество ошибок в информационных векторах. Во всех символах кодовых слов ими не обнаруживается такое же количество ошибок, что и у кодов Бергера с тем же значением m , а в классе обнаруживаемых оказываются монотонные, симметричные и асимметричные ошибки [31]. Отметим также, что могут рассматриваться коды с суммированием в кольце вычетов по произвольному значению $M \in \{2, 3, \dots, m+1\}$.

Существует большое разнообразие модификаций кодов с суммированием, которые могут эффективно применяться при синтезе СВК на основе ЛКС. Рассмотрим применение взвешенных кодов с суммированием с числом информационных символов $m=4$ и модулем $M=4$, которые могут применяться при синтезе СВК на основе ЛКС с контролем групп, состоящих из шести выходов объекта диагностирования. Поскольку модульные коды с суммированием единичных разрядов при $M=4$, 8 и 12 принято называть кодами Боуза — Лина [32], рассматриваемые взвешенные коды отнесем к классу так называемых взвешенных кодов Боуза — Лина.

2. Взвешенные коды Боуза — Лина

Взвешенные коды Боуза — Лина могут быть построены на основе принципа установления неравноправия между информационными символами путем их взвешивания [33]. Информационные символы из вектора $\langle f_4 f_3 f_2 f_1 \rangle$ взвешиваются при этом весовыми коэффициентами $[w_4, w_3, w_2, w_1]$ $w_i \in \mathbb{N}, i = 1, 4$. Затем определяется сумма весовых коэффициентов единичных информационных символов в кольце вычетов по модулю $M=4$:

$$W_4 = W \pmod{4} = \left(\sum_{i=1}^{m=4} f_i w_i \right) \pmod{4}.$$

Далее число W_4 представляется в двоичном виде и записывается в разряды контрольного вектора.

Обозначим взвешенные коды Боуза — Лина как $WS4(4, 2)$ коды (*weight-based sum codes*), где числа в скобках обозначают количество информационных и проверочных символов. Отдельно будем указывать массив весовых коэффициентов $[w_4, w_3, w_2, w_1]$.

Весовые коэффициенты из массива $[w_4, w_3, w_2, w_1]$ могут быть произвольными натуральными числами. Но поскольку вычисления производятся в кольце вычетов по модулю $M=4$, то только часть сочетаний значений весовых коэффициентов даст разнообразные по своим характеристикам коды. Весовые коэффициенты каждого информационного символа могут быть выбраны из множества $\{1, 2, 3\}$. Большие значения весовых коэффициентов не будут давать «новых» по характеристикам кодов. Взвешивание какого-либо информационного символа числом 0 не будет давать при построении помехозащищенного $WS4(4, 2)$ кода, что даст эффективное его использование только в некоторых частных случаях.

Всего существует $3^4=81$ вариант взвешивания информационных символов для построения взвешенного кода Боуза — Лина со значением $m=4$. Если рассматривать полное множество информационных векторов, то характеристики обнаружения ошибок будут определяться исключительно сочетанием значений весовых коэффициентов, а не их местоположением в массиве информационных векторов. Это дает возможность построения всего 15 помехозащищенных разнообразных $WS4(4, 2)$

кодов со следующими массивами весовых коэффициентов [34]: $[1, 1, 1, 1]$, $[1, 1, 1, 2]$, $[1, 1, 1, 3]$, $[1, 1, 2, 2]$, $[1, 1, 2, 3]$, $[1, 1, 3, 3]$, $[1, 2, 2, 2]$, $[1, 2, 2, 3]$, $[1, 2, 3, 3]$, $[1, 3, 3, 3]$, $[2, 2, 2, 2]$, $[2, 2, 2, 3]$, $[2, 2, 3, 3]$, $[2, 3, 3, 3]$, $[3, 3, 3, 3]$. Первый из представленных массивов даст классический код Боуза — Лина.

В табл. 1 приведены характеристики обнаружения ошибок кодами Боуза — Лина, которые необходимо учитывать при синтезе СВК на основе ЛКС. Каждому $WS4(4, 2)$ коду присвоены номера — римскими цифрами по мере возрастания весовых коэффициентов и увеличения их значений. Для каждого кода указано распределение необнаруживаемых ошибок в кодовых словах в виде: $Nv_{[w_4, w_3, w_2, w_1]} = Nv_d + N\sigma_d + N\alpha_d$, где N — число необнаруживаемых ошибок: v — всего в кодовых словах, v_d — число монотонных необнаруживаемых ошибок кратностью d , σ_d — число симметричных необнаруживаемых ошибок кратностью d , α_d — число асимметричных необнаруживаемых ошибок кратностью d .

Следует отметить такие важные факторы. Для каждого $WS4(4, 2)$ кода имеется свой код с отличными от других характеристиками сложности реализации, что определяется значениями весовых коэффициентов. У каждого $WS4(4, 2)$ кода имеется своя классификация информационных векторов по одинаковым контрольным векторам. К примеру, в табл. 2 приведена такая классификация для кода с номером II. Такая классификация является основополагающей при синтезе СВК на основе ЛКС, поскольку требуется выбирать способ доопределения на каждой входной комбинации $\langle x_i x_{i-1} \dots x_2 x_1 \rangle$ на выходах БКС вектора $\langle h_6 h_5 \dots h_2 h_1 \rangle$. Число способов доопределения разнообразно и зависит от проектировщика СВК. При этом могут использоваться специальные алгоритмы для доопределения.

Классификация информационных векторов по одинаковым контрольным векторам однозначно определяет характеристики обнаружения ошибок в информационных символах при безошибочности проверочных символов. Минимальное их общее количество достигается при равномерном распределении информационных векторов между всеми контрольными векторами. Исследования показывают,

Таблица 1. Характеристика обнаружения ошибок в кодовых словах WS4(4, 2) кодами

Номер кода	Массив весовых коэффициентов	Формула необнаруживаемых ошибок в кодовом слове
I	[1, 1, 1, 1]	$240v_{[1,1,1,1]} = 32u_2 + 48\sigma_2 + 36u_3 + 44\alpha_3 + 2u_4 + 6\sigma_4 + 32\alpha_4 + 8u_5 + 32\alpha_5$
II	[1, 1, 1, 2]	$240v_{[1,1,1,2]} = 32u_2 + 32\sigma_2 + 24u_3 + 72\alpha_3 + 6u_4 + 18\sigma_4 + 24\alpha_4 + 4u_5 + 28\alpha_5$
III	[1, 1, 1, 3]	$240v_{[1,1,1,3]} = 48u_2 + 32\sigma_2 + 20u_3 + 60\alpha_3 + 6u_4 + 18\sigma_4 + 16\alpha_4 + 4u_5 + 36\alpha_5$
IV	[1, 1, 2, 2]	$240v_{[1,1,2,2]} = 40u_2 + 32\sigma_2 + 12u_3 + 60\alpha_3 + 12u_4 + 24\sigma_4 + 36\alpha_4 + 2u_5 + 22\alpha_5$
V	[1, 1, 2, 3]	$240v_{[1,1,2,3]} = 44u_2 + 20\sigma_2 + 16u_3 + 80\alpha_3 + 20u_4 + 28\sigma_4 + 16\alpha_4 + 2u_5 + 4\sigma_5 + 10\alpha_5$
VI	[1, 1, 3, 3]	$240v_{[1,1,3,3]} = 48u_2 + 32\sigma_2 + 20u_3 + 60\alpha_3 + 6u_4 + 18\sigma_4 + 16\alpha_4 + 4u_5 + 36\alpha_5$
VII	[1, 2, 2, 2]	$240v_{[1,2,2,2]} = 64u_2 + 48\sigma_2 + 28u_4 + 36\sigma_4 + 48\alpha_4 + 2u_5 + 14\alpha_5$
VIII	[1, 2, 2, 3]	$240v_{[1,2,2,3]} = 40u_2 + 32\sigma_2 + 12u_3 + 60\alpha_3 + 12u_4 + 24\sigma_4 + 36\alpha_4 + 2u_5 + 22\alpha_5$
IX	[1, 2, 3, 3]	$240v_{[1,2,3,3]} = 32u_2 + 32\sigma_2 + 24u_3 + 72\alpha_3 + 6u_4 + 18\sigma_4 + 24\alpha_4 + 4u_5 + 28\alpha_5$
X	[1, 3, 3, 3]	$240v_{[1,3,3,3]} = 32u_2 + 48\sigma_2 + 36u_3 + 44\alpha_3 + 2u_4 + 6\sigma_4 + 32\alpha_4 + 8u_5 + 32\alpha_5$
XI	[2, 2, 2, 2]	$240v_{[2,2,2,2]} = 80u_2 + 80\sigma_2 + 10u_4 + 30\sigma_4 + 40\alpha_4$
XII	[2, 2, 2, 3]	$240v_{[2,2,2,3]} = 48u_2 + 48\sigma_2 + 32u_3 + 32\alpha_3 + 2u_4 + 6\sigma_4 + 8\alpha_4 + 8u_5 + 56\alpha_5$
XIII	[2, 2, 3, 3]	$240v_{[2,2,3,3]} = 24u_2 + 48\sigma_2 + 36u_3 + 36\alpha_3 + 36\sigma_4 + 36\alpha_4 + 6u_5 + 18\alpha_5$
XIV	[2, 3, 3, 3]	$240v_{[2,3,3,3]} = 4u_2 + 60\sigma_2 + 48u_3 + 48\alpha_3 + 4u_4 + 60\sigma_4 + 2u_5 + 12\sigma_5 + 2\alpha_5$
XV	[3, 3, 3, 3]	$240v_{[3,3,3,3]} = 80\sigma_2 + 20u_3 + 60\alpha_3 + 10u_4 + 30\sigma_4 + 40\alpha_5$

что если все весовые коэффициенты являются нечетными числами, то такое распределение оказывается неравномерным. Если хотя бы один весовой коэффициент взвешен четным числом, то распределение оказывается равномерным. Указанное обстоятельство влияет на алгоритмы доопределения в СВК функций логической коррекции сигналов.

3. Построение схемы встроенного контроля на основе логической коррекции сигналов при преобразовании только выходов, формирующих проверочные символы кода

На рис. 2 приведена базовая структура организации СВК на основе ЛКС с применением WS4(4, 2) кода при преобразовании только тех выходов объекта диагностирования, которые участвуют в формировании проверочных символов.

Значения информационных символов в структуре рис. 2 вычисляются объектом диагностирования без преобразования в БКС: $h_3=f_3, h_4=f_4, h_5=f_5, h_6=f_6$. Тогда каждому информационному вектору в СВК должен соответствовать единственный контрольный вектор, который можно однозначно доопределить на каждой входной комбинации.

Таблица 2. Классификация информационных векторов по одинаковым контрольным векторам для кода с номером II

W_4			
0	1	2	3
Контрольные векторы			
00	01	10	11
Информационные векторы			
0000	0010	0001	0011
0111	0100	0110	0101
1011	1000	1010	1001
1101	1111	1100	1110

При синтезе СВК требуется обеспечить выполнение следующих условий:

1. Должно формироваться множество проверочных комбинаций для тестера WS4(4, 2) кода, что связано с необходимостью подачи на его входы полного множества контрольных векторов хотя бы единожды (см. табл. 2).
2. Должно формироваться множество проверочных комбинаций для элементов преобразования, которое содержит все четыре комбинации {00, 01, 10, 11} при его канонической реализации [35].

С учетом этих условий и синтезируется СВК.

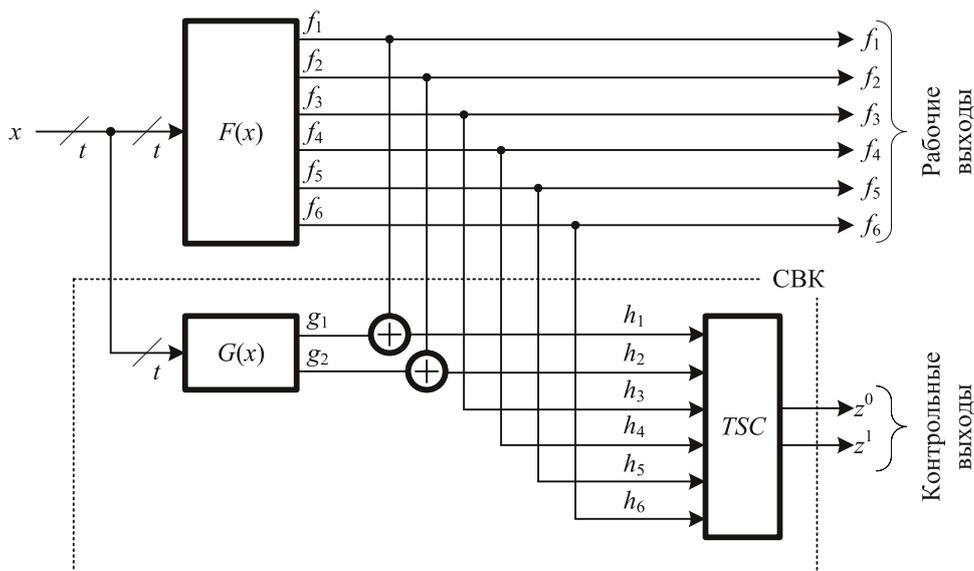


Рис. 2. Структура организации СВК с преобразованием части рабочих функций в функции, описывающие проверочные символы заданного кода

Существует $C_6^2 = 15$ способов выбора преобразуемых функций среди всех рабочих функций объекта диагностирования. Далее выходы в информационном векторе могут быть расставлены $P_4 = 24$ вариантами, а в контрольном — $P_2 = 2$ вариантами. Итого общее количество способов построения СВК на основе ЛКС с применением $WS4(4, 2)$ кода равно $C_6^2 P_4 P_2 = 15 \cdot 24 \cdot 2 = 720$.

Такая вариативность позволяет синтезировать большое количество СВК на основе ЛКС с применением базовой структуры рис. 2, среди которых можно выбрать те, которые будут давать структуры с наименьшими показателями сложности реализации при обеспечении самопроверяемости всех компонентов.

Алгоритм 1. Правила синтеза СВК на основе ЛКС с применением $WS4(4, 2)$ кода при преобразовании только тех рабочих функций, которые описывают проверочные символы кода:

1. Выделяем выходы объекта диагностирования, формирующие информационные и проверочные символы $WS4(4, 2)$ кода в СВК.
2. Выполняем проверку формирования для двух преобразуемых функций объекта диагностирования хотя бы на двух наборах 0 и на двух 1 (это необходимо для формирования проверочных комбинаций элементов XOR).

3. Вычисляем функции, описывающие информационные символы: $h_3 = f_3, h_4 = f_4, h_5 = f_5, h_6 = f_6$.
4. Однозначно по информационным векторам на каждой входной комбинации $\langle x_t, x_{t-1}, \dots, x_2, x_1 \rangle$ доопределяем разряды h_1 и h_2 .
5. Вычисляем функции g_1 и g_2 : $g_2 = f_2 \oplus h_2, g_1 = f_1 \oplus h_1$.
6. Проверяем формирование полного множества комбинаций для проверки тестера: {00, 01, 10, 11}.
7. Проверяем формирование полного теста для элементов преобразования: {00, 01, 10, 11}.
8. Оптимизируем функции g_2 и g_1 .
9. Синтезируем блок $G(x)$ в выбранном элементном базисе.

Покажем работу представленного алгоритма на примере синтеза СВК для устройства, описываемого таблицей истинности (табл. 3).

Будем синтезировать СВК непосредственно по структуре рис. 2 и в качестве преобразуемых выберем функции f_2 и f_1 . Процесс доопределения иллюстрируется табл. 4.

Табл. 4 заполнена с учетом табл. 2. Из анализа столбцов для g_2 и g_1 ясно, что полный тест для тестера формируется. Из анализа столбцов XOR_2 и XOR_1 ясно, что формируется и полный тест для элементов преобразования.

Таблица 3. Таблица истинности комбинационного устройства

№	x_4	x_3	x_2	x_1	f_6	f_5	f_4	f_3	f_2	f_1
0	0	0	0	0	0	0	1	1	1	0
1	0	0	0	1	0	1	0	1	0	1
2	0	0	1	0	1	0	1	1	0	1
3	0	0	1	1	0	0	0	0	1	0
4	0	1	0	0	0	1	1	0	1	0
5	0	1	0	1	0	1	1	0	0	1
6	0	1	1	0	1	0	0	1	0	0
7	0	1	1	1	1	0	1	0	0	0
8	1	0	0	0	0	1	0	0	1	0
9	1	0	0	1	1	0	0	1	0	0
10	1	0	1	0	0	1	0	0	1	1
11	1	0	1	1	1	1	1	1	1	1
12	1	1	0	0	1	0	1	1	0	0
13	1	1	0	1	1	1	1	0	1	0
14	1	1	1	0	0	1	0	1	0	0
15	1	1	1	1	0	0	1	0	0	0

Таблица 4. Сигналы на выходах $F(x)$ и линиях СВК при использовании алгоритма 1 и кода № II

№	x_4	x_3	x_2	x_1	f_6	f_5	f_4	f_3	f_2	f_1	h_6	h_5	h_4	h_3	h_2	h_1	g_2	g_1	XOR_2	XOR_1
0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	1	10	01
1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	0	01	10
2	0	0	1	0	1	0	1	1	0	1	1	0	1	1	0	0	0	1	00	11
3	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	11	00
4	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	10	00
5	0	1	0	1	0	1	1	0	0	1	0	1	1	0	1	0	1	1	01	11
6	0	1	1	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	01	01
7	0	1	1	1	1	0	1	0	0	0	1	0	1	0	1	0	1	0	01	00
8	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1	1	1	11	01
9	1	0	0	1	1	0	0	1	0	0	1	0	0	1	1	1	1	1	01	01
10	1	0	1	0	0	1	0	0	1	1	0	1	0	0	0	1	0	11	10	10
11	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	11	10
12	1	1	0	0	1	0	1	1	0	0	1	0	1	1	0	0	0	0	00	00
13	1	1	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	10	01
14	1	1	1	0	0	1	0	1	0	0	0	1	0	1	1	1	1	1	01	01
15	1	1	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0	1	00	01

Далее выписываются разрешенные наборы для функций g_2 и g_1 :

$$g_2 = \{1, 3, 5, 6, 7, 8, 9, 10, 11, 14\};$$

$$g_1 = \{0, 2, 5, 6, 8, 9, 13, 14, 15\}.$$

Затем синтезируется СВК в выбранном элементном базисе.

Для примера синтезируем самопроверяемое устройство по структуре рис. 2 и определим показатели сложности его технической реализации в условных единицах библиотеки функциональных элементов *stdcell2_2.genlib* с помощью *SIS* [36, 37].

Блок $F(x)$ синтезирован без оптимизации структуры. Сложность его реализации оценивается величиной $L_{F(x)}=1200$ (усл. ед.). Блок $G(x)$ синтезирован после оптимизации функций g_2 и g_1 с помощью процедуры *full_simplify*. Его сложность равна $L_{G(x)}=320$ (усл. ед.).

Показатель сложности реализации устройства с СВК определяется по формуле:

$$L_{CED} = L_{F(x)} + L_{G(x)} + 2L_{XOR} + L_{G(f)} + L_{TRC} + 2L_{NOT}, \quad (1)$$

где L_{CED} — показатель сложности реализации устройства с СВК¹, $L_{XOR}=40$ — показатель сложности одного элемента преобразования, $L_{NOT}=16$ — показатель сложности инвертора в TSC^2 , $L_{G(f)}=216$ — показатель сложности кодера в TSC , $L_{TRC}=192$ — показатель сложности TRC в TSC .

По формуле (1) имеем:

$$L_{CED} = 1200 + 320 + 2 \cdot 40 + 216 + 192 + 2 \cdot 16 = 2040.$$

Сравним с дублированием.

Для структуры дублирования формула вычисления показателя сложности имеет следующий вид:

$$L_D = 2L_{F(x)} + nL_{NOT} + (n-1)L_{TRC}, \quad (2)$$

где n — число выходов устройства $F(x)$.

По формуле (2) имеем:

$$L_D = 2 \cdot 1200 + 6 \cdot 16 + (6-1) \cdot 192 = 3456.$$

Показатель δ дает представление об эффективности синтезируемого устройства по предложенному методу в сравнении с использованием метода дублирования:

$$\delta = \frac{L_{CED}}{L_D} \cdot 100\%. \quad (3)$$

Вычислим показатель δ для рассматриваемого примера:

¹ Устройство с СВК в англоязычных источниках часто определяют как *CED System — concurrent error-detection system*. Отсюда нижний индекс у показателя сложности технической реализации устройства с СВК.

² Строго говоря, два инвертора устанавливаются либо на выходах $G(x)$, либо на выходах $G(f)$ для работы в парафазной логике элемента сравнения TRC .

$$\delta = \frac{2040}{3456} \cdot 100\% = 59,028\%.$$

Показатель структурной избыточности самопроверяемого устройства, синтезированного по алгоритму 1, составляет менее 60% от показателя структурной избыточности системы дублирования.

4. Построение схемы встроенного контроля на основе логической коррекции сигналов при преобразовании части выходов, которые формируют информационные символы кода

На рис. 3 приведена базовая структура организации СВК на основе ЛКС с применением $WS4(4, 2)$ кода при преобразовании части выходов объекта диагностирования, которые участвуют в формировании информационных символов.

Особенностью распределения информационных векторов между контрольными векторами для рассматриваемого $WS4(4, 2)$ кода № II является следующее свойство.

Утверждение 1. Для каждого контрольного вектора существует ровно по одному информационному вектору вида $\langle 00xx \rangle$, $\langle 01xx \rangle$, $\langle 10xx \rangle$, $\langle 11xx \rangle$, $x \in \{0, 1\}$.

Это свойство рассматриваемого $WS4(4, 2)$ кода можно использовать при синтезе СВК при преобразовании только части рабочих функций, участвующих в формировании информационного вектора.

Утверждение 2. Функции, описывающие старшие разряды в информационном векторе, могут напрямую вычисляться объектом диагностирования.

Доказательство. Обратимся к табл. 1. Для построения самопроверяемой СВК потребуется формирование хотя бы по разу контрольных векторов $\langle h_2 h_1 \rangle = \langle 00 \rangle$, $\langle 01 \rangle$, $\langle 10 \rangle$, $\langle 11 \rangle$. Положим, информационный вектор выглядит следующим образом: $\langle h_6 h_5 h_4 h_3 \rangle$. Покажем, что можно доопределить функции, описывающие старшие разряды информационного вектора, так: $h_6=f_6$ и $h_5=f_5$. В каждом столбце табл. 1 для функций h_6 и h_5 присутствует и 0, и 1, а значит, такое доопределение возможно всегда. ■

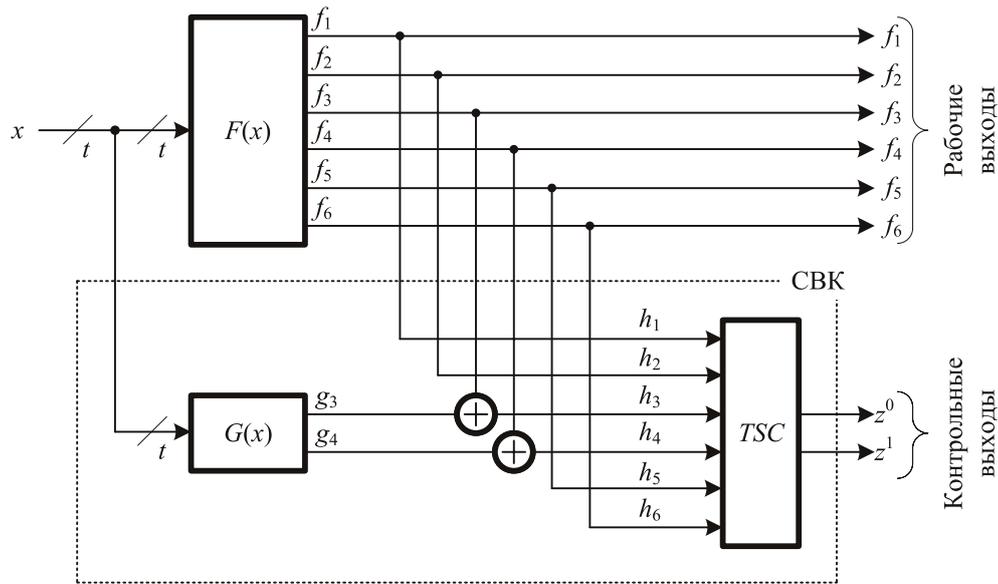


Рис. 3. Структура организации СВК с преобразованием части рабочих функций в функции, описывающие проверочные символы заданного кода

Алгоритм 2. Правила синтеза СВК на основе ЛКС с применением WS4(4, 2) кода при преобразовании двух рабочих функций, которые описывают информационные символы кода:

1. Выделяем выходы объекта диагностирования, формирующие информационные и проверочные символы WS4(4, 2) кода в СВК.
2. Вычисляем функции, описывающие проверочные символы: $h_2=f_2, h_1=f_1$.
3. Вычисляем функции, описывающие старшие разряды информационного вектора: $h_6=f_6, h_5=f_5$.
4. Однозначно по контрольным векторам и старшим разрядам информационного вектора на каждой входной комбинации $\langle x_i, x_{i-1}, \dots, x_2, x_1 \rangle$ доопределяем разряды h_4 и h_3 .
5. Вычисляем функции g_4 и g_3 : $g_4 = f_4 \oplus h_4, g_3 = f_3 \oplus h_3$.
6. Проверяем формирование полного множества комбинаций для проверки теста: {00, 01, 10, 11}.
7. Проверяем формирование полного теста для элементов преобразования: {00, 01, 10, 11}.
8. Оптимизируем функции g_2 и g_1 .
9. Синтезируем блок $G(x)$ в выбранном элементном базисе.

Для функций вычисления значений разрядов h_4 и h_3 нетрудно получить функцио-

нальную зависимость. Для этого рассмотрим таблицу кодовых слов WS4(4, 2) кода (табл. 5). В ней кодовые слова расположены в лексикографическом порядке, а каждому кодовому слову соответствует десятичный номер, соответствующий двоичному числу, записанному в информационном векторе.

Для получения функций, описывающих информационные символы h_4 и h_3 , поменяем местами столбцы h_4 и h_3 и h_2 и h_1 и поменяем строки

Таблица 5. Кодовые слова WS4(4, 2) кода

№	h_6	h_5	h_4	h_3	h_2	h_1
0	0	0	0	0	0	0
1	0	0	0	1	1	0
2	0	0	1	0	0	1
3	0	0	1	1	1	1
4	0	1	0	0	0	1
5	0	1	0	1	1	1
6	0	1	1	0	1	0
7	0	1	1	1	0	0
8	1	0	0	0	0	1
9	1	0	0	1	1	1
10	1	0	1	0	1	0
11	1	0	1	1	0	0
12	1	1	0	0	1	0
13	1	1	0	1	0	0
14	1	1	1	0	1	1
15	1	1	1	1	0	1

так, чтобы номера соответствовали двоичным числам, записанным в векторах $\langle h_6 h_5 h_2 h_1 \rangle$ (табл. 6).

Таблица 6. Таблица истинности функций h_4 и h_3

№	h_6	h_5	h_2	h_1	h_4	h_3
0	0	0	0	0	0	0
1	0	0	0	1	1	0
2	0	0	1	0	0	1
3	0	0	1	1	1	1
4	0	1	0	0	1	1
5	0	1	0	1	0	0
6	0	1	1	0	1	0
7	0	1	1	1	0	1
8	1	0	0	0	1	1
9	1	0	0	1	0	0
10	1	0	1	0	1	0
11	1	0	1	1	0	1
12	1	1	0	0	0	1
13	1	1	0	1	1	1
14	1	1	1	0	0	0
15	1	1	1	1	1	0

Используем метод Карно для получения логических выражений, описывающих функции h_4 и h_3 [38]. Получаем следующие функции, описывающие «недостающие» информационные символы:

$$\begin{aligned}
 h_4 &= \overline{h_6} \overline{h_5} h_1 \vee \overline{h_6} h_5 \overline{h_1} \vee h_6 h_5 h_1 \vee h_6 \overline{h_5} \overline{h_1} = \\
 &= \overline{h_6} (\overline{h_5} h_1 \vee h_5 \overline{h_1}) \vee h_6 (h_5 h_1 \vee \overline{h_5} \overline{h_1}) = \\
 &= h_6 \oplus h_5 \oplus h_1;
 \end{aligned} \tag{4}$$

$$\begin{aligned}
 h_3 &= \overline{h_6} \overline{h_5} h_2 \vee \overline{h_6} h_5 h_1 \vee \overline{h_6} h_2 h_1 \vee h_5 \overline{h_2} \overline{h_1} \vee h_6 h_5 \overline{h_2} \vee h_6 \overline{h_2} \overline{h_1} = \\
 &= \overline{h_6} \overline{h_5} h_2 \vee h_6 h_5 \overline{h_2} \vee h_2 h_1 (\overline{h_5} \vee \overline{h_6}) \vee \overline{h_2} \overline{h_1} (h_5 \vee h_6) = \\
 &= h_2 (\overline{h_6} \overline{h_5} \vee h_1 (\overline{h_5} \vee \overline{h_6})) \vee \overline{h_2} (h_6 h_5 \vee \overline{h_1} (h_5 \vee h_6)).
 \end{aligned} \tag{5}$$

Применяя (4) и (5), можно автоматически доопределить значения функций логической коррекции сигналов g_4 и g_3 .

Аналогично рассмотренному выше примеру с доопределением функций логической кор-

рекции сигналов для алгоритма 1, выполним те же процедуры для использования алгоритма 2 (табл. 7).

Проверочные комбинации для тестера и элементов преобразования формируются.

Выпишем разрешенные наборы для функций g_4 и g_3 :

$$\begin{aligned}
 g_4 &= \{0, 2, 5, 6, 8, 9, 13, 14, 15\}; \\
 g_3 &= \{1, 2, 3, 7, 10, 11\}.
 \end{aligned}$$

Далее СВК синтезируется в выбранном элементном базисе.

Используя *SIS* и библиотеку *stdcell2_2.genlib*, синтезируем СВК и оценим показатели сложности ее реализации.

Блок $G(x)$ синтезирован после оптимизации функций g_4 и g_3 с помощью процедуры *full_simplify*. Его сложность определяется величиной $L_{G(x)}=240$ (усл. ед.). Используем формулу (1) для определения показателя структурной избыточности самопроверяемого устройства:

$$L_{CED} = 1200 + 240 + 2 \cdot 40 + 216 + 192 + 2 \cdot 16 = 1960.$$

По формуле (3) определим показатель δ для рассматриваемого примера:

$$\delta = \frac{1960}{3456} \cdot 100 \% = 56,713 \%$$

В данном примере получено еще большее уменьшение показателя структурной избыточности самопроверяемого устройства по сравнению с дублированием, чем в предыдущем.

Могут рассматриваться и другие алгоритмы синтеза СВК: например, связанные с преобразованием всех рабочих функций от объекта диагностирования [39].

5. Эксперименты с тестовыми схемами

Для проверки эффективности предлагаемых способов синтеза СВК был проведен ряд экспериментов с тестовыми комбинационными схемами из набора *MCNC Benchmarks* [40, 41]. В ходе экспериментов оценивались показатели структурной избыточности и тестопригодности элементов СВК.

Тестовые схемы из набора *MCNC Benchmarks* имеют различные параметры. Обозначим число входов схемы, как и ранее, через t , а число выходов — через n .

Таблица 7. Сигналы на выходах $F(x)$ и линиях СВК

№	x_4	x_3	x_2	x_1	f_6	f_5	f_4	f_3	f_2	f_1	h_6	h_5	h_4	h_3	h_2	h_1	g_4	g_3	XOR_4	XOR_3
0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0	11	10
1	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	00	11
2	0	0	1	0	1	0	1	1	0	1	1	0	0	0	0	1	1	1	11	11
3	0	0	1	1	0	0	0	0	1	0	0	0	0	1	1	0	0	1	00	01
4	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	10	00
5	0	1	0	1	0	1	1	0	0	1	0	1	0	0	0	1	1	0	11	00
6	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	0	1	0	01	10
7	0	1	1	1	1	0	1	0	0	0	1	0	1	1	0	0	0	1	10	01
8	1	0	0	0	0	1	0	0	1	0	0	1	1	0	1	0	1	0	01	00
9	1	0	0	1	1	0	0	1	0	0	1	0	1	1	0	0	1	0	01	10
10	1	0	1	0	0	1	0	0	1	1	0	1	0	1	1	1	0	1	00	01
11	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	10	11
12	1	1	0	0	1	0	1	1	0	0	1	0	1	1	0	0	0	0	10	10
13	1	1	0	1	1	1	1	0	1	0	1	1	0	0	1	0	1	0	11	00
14	1	1	1	0	0	1	0	1	0	0	0	1	1	1	0	0	1	0	01	10
15	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	11	00

Приведенные выше алгоритмы синтеза СВК основаны на использовании $WS4(4, 2)$ кода № II и подразумевают контроль вычислений в группе из 6 выходов. Поэтому при синтезе СВК для схем с произвольным числом выходов требуется предварительно выделить контролируемые группы. Число способов их выделения велико. Среди n выходов можно в качестве первой группы выделить выходы C_n^6 вариантами. Среди оставшихся выходов — C_{n-6}^6 вариантами и т. д. Общее же число вариантов выбора групп определяется величиной³

$$N = \prod_{j=1}^{\lfloor \frac{n}{6} \rfloor - 1} C_{n-6(j-1)}^6 \quad (6)$$

К примеру, при $n=15$ формула (6) дает следующее число вариантов выбора групп:

$$N = \prod_{j=1}^{\lfloor \frac{15}{6} \rfloor - 1} C_{15-6(j-1)}^6 = \prod_{j=1}^2 C_{15-6(j-1)}^6 = C_{15}^6 C_{11}^6 = 2\,312\,310.$$

³ Следует отметить, что представленная формула дает возможность определения общего количества вариантов выбора групп по принципу «минимум выделенных групп для покрытия всех выходов». Другими словами, не учтено возможное повторное использование выходов в разных группах, что может быть сделано из соображений повышения обнаруживающих характеристик метода.

Кроме того, в каждой выделенной группе можно, как было показано выше, синтезировать СВК $C_6^2 P_4 P_2 = 15 \cdot 24 \cdot 2 = 720$ способами. Тогда с учетом (6) имеем следующее количество способов синтеза СВК с выделением групп по 6 выходов по 1 и 2 алгоритмам соответственно:

$$N_{CED}^I = N_{CED}^{II} = C_6^2 P_4 P_2 \prod_{j=1}^{\lfloor \frac{n}{6} \rfloor - 1} C_{n-6(j-1)}^6 = 720 \cdot \prod_{j=1}^{\lfloor \frac{n}{6} \rfloor - 1} C_{n-6(j-1)}^6 \quad (7)$$

Для рассмотренного выше примера с $n=15$ формула (7) дает соответственно $N_{CED}^I = 720 \cdot 2\,312\,310 = 1\,664\,863\,200$. Это число астрономическое, поэтому в ходе эксперимента был использован следующий алгоритм синтеза СВК для тестовых схем.

Алгоритм 3. Последовательность синтеза СВК на основе ЛКС с применением $WS4(4, 2)$ кода на основе базовых алгоритмов:

1. Для рассматриваемой тестовой схемы выделяем последовательно в соответствии с описанием схемы группы по 6 выходов, причем если $n \pmod 6 = 0$, то выделяем ровно $\lfloor \frac{n}{6} \rfloor$ групп по 6 выходов; если $n \pmod 6 = r, r \in \{1, 2, 3, 4, 5\}$, то выделяем $\lfloor \frac{n}{6} \rfloor$ групп по 6 выходов и одну

группу, в которую включаем s оставшихся выходов и $6-s$ выходов, расположенных рядом в описании. Таким образом, выделяется $\lceil \frac{n}{6} \rceil$ групп по 6 выходов.

2. Для каждой из $\lceil \frac{n}{6} \rceil$ групп синтезируем СВК с использованием базовой структуры рис. 2 по алгоритму 1 и отдельно с использованием базовой структуры рис. 3 по алгоритму 2.
3. Выходы полученных СВК объединяем на входах самопроверяемого компаратора, реализованного на основе $\lceil \frac{n}{6} \rceil - 1$ модуля сжатия паразитных сигналов *TRC*.

Пользуясь алгоритмом 3, реализуем СВК для ряда тестовых схем из набора *MCNC Benchmarks*. Используем исключительно *WS4(4, 2)* код № II и базовые алгоритмы 1 и 2. При синтезе устройств используем тот же подход, что и в описаниях примеров в предыдущих разделах, интерпретатор *SIS* и библиотеку функциональных элементов *stdcell2_2.genlib*. При этом блок $F(x)$ (тестовая схема) синтезировалась без оптимизации структуры. Блоки же контрольной логики каждой СВК объединены и синтезированы совместно при оптимизации процедурой *full_simplify*. Остальные элементы СВК являются типовыми.

В табл. 8 и 9, а также на рис. 4 и 5 приведены результаты экспериментов для 21 тестовой схемы. В последних двух графах табл. 8 и 9 приведены относительные показатели сложности реализации устройств с СВК — показатель δ , рассчитанный по формуле (3), и показатель γ , рассчитанный по формуле:

$$\gamma = \frac{L_{G(x)}}{L_{F(x)}} \cdot 100\%. \quad (8)$$

По показателю δ можно судить об эффективности реализации по сравнению со стандартным методом дублирования. При использовании алгоритма 1 в качестве базового для

19 из 21 схемы получают более эффективные по показателю δ структуры устройств. Причем для 10 схем получено улучшение более чем на 25%. В среднем для всех схем получен показатель $\delta_{cp} = 78,079\%$. При использовании алгоритма 2 в качестве базового для 18 из 21 схемы получают более эффективные по показателю δ структуры устройств. Причем, так же как и для алгоритма 1, для 10 схем получено улучшение более чем на 25%. В среднем для всех схем получен показатель $\delta_{cp} = 78,948\%$. В целом оба алгоритма дают соизмеримые по показателю δ структуры для каждой тестовой схемы. Разброс в значениях для всех схем, кроме *newcpla1*, не превышает $\pm 5\%$. Однако среди представленных вариантов синтеза устройств можно выбрать наилучший по показателю δ .

Показатель γ введен здесь для того, чтобы учесть вклад в общее значение абсолютного показателя сложности реализации устройства блока контрольной логики $G(x)$. Остальные составляющие каждой СВК типовые, естественно, влияют на сложность реализации, но при довольно сложных блоках $G(x)$ значения δ становятся близкими к 100% и превышающими данный порог. К примеру, при значениях $\gamma < 100\%$ при использовании обоих алгоритмов синтеза СВК в качестве базовых получены значения $\delta < 100\%$. Только в тех случаях, когда $\gamma > 100\%$, и показатель $\delta > 100\%$. Это говорит о весомом влиянии на показатели сложности реализации всего устройства показателя для $G(x)$.

Из сравнения рис. 4 и 5 следует, что характер величин δ для одинаковых схем при использовании разных алгоритмов синтеза СВК схож.

Опишем далее результаты экспериментов характеристик тестопригодности элементов СВК. В табл. 10 и 11 приведены данные о числе комбинаций, на которых формируются все возможные контрольные векторы для *WS4(4, 2)* кода № II для каждой контролируемой группы выходов.

Таблица 8. Результаты экспериментов по использованию в качестве базового алгоритма 1 при синтезе СВК

№	Схема	t	n	q	$L_{F(x)}$ усл. ед.	$L_{G(x)}$ усл. ед.	L_{CED} усл. ед.	$L_{D'}$ усл. ед.	$\gamma, \%$	$\delta, \%$
1	dc1	4	7	2	976	488	2696	3216	50	83,831
2	dekoder	4	7	2	736	464	2432	2736	63,043	88,889
3	wim	4	7	2	712	472	2416	2688	66,292	89,881
4	newbyte	5	8	2	592	376	2200	2656	63,514	82,831
5	p82	5	14	3	2368	952	5264	7456	40,203	70,601
6	m1	6	12	2	3064	912	5208	8432	29,765	61,765
7	newapla2	6	7	2	600	488	2320	2464	81,333	94,156
8	sqr6	6	12	2	2648	1536	5416	7600	58,006	71,263
9	inc	7	9	2	2376	1624	5232	6432	68,35	81,343
10	newcpla2	7	10	2	1896	1032	4160	5680	54,43	73,239
11	max128	7	24	4	20192	1352	24200	45184	6,696	53,559
12	m2	8	16	3	10096	2536	14576	23328	25,119	62,483
13	m3	8	16	3	13464	3272	18680	30064	24,302	62,134
14	m4	8	16	3	18704	4848	25496	40544	25,92	62,885
15	mlp4	8	8	2	7224	8432	16888	15920	116,722	106,08
16	tms	8	16	3	6784	1920	10648	16704	28,302	63,745
17	dk27	9	9	2	528	888	2648	2736	168,182	96,784
18	max512	9	6	1	9632	4256	14408	20320	44,186	70,906
19	newcpla1	9	16	3	2520	2080	6544	8176	82,54	80,039
20	newxcpla1	9	23	4	3760	1992	8408	12112	52,979	69,419
21	sex	9	14	3	1360	2888	6192	5440	212,353	113,824
Среднее									64,868	78,079

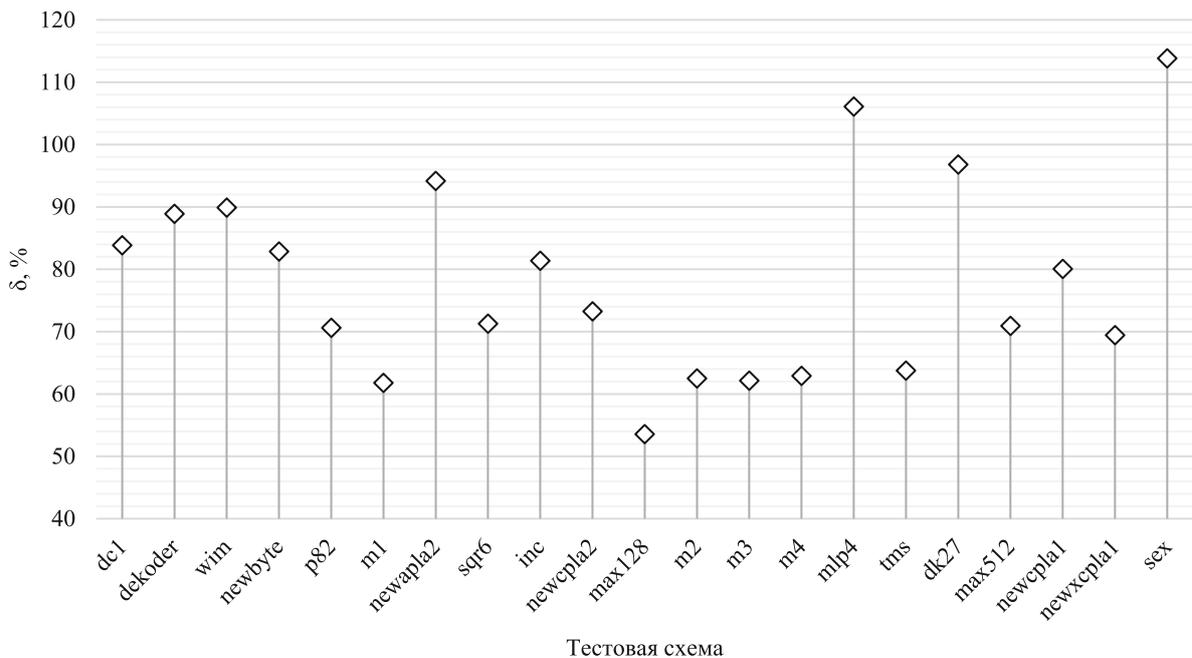


Рис. 4. Графическое представление показателя δ для тестовых схем при использовании в качестве базового алгоритма 1 при синтезе СВК

Таблица 9. Результаты экспериментов по использованию в качестве базового алгоритма 2 при синтезе СВК

№	Схема	t	n	q	$L_{F(x)}$ усл. ед.	$L_{G(x)}$ усл. ед.	L_{CED} усл. ед.	L_{D} усл. ед.	$\gamma, \%$	$\delta, \%$
1	dc1	4	7	2	976	344	2552	3216	35,246	79,353
2	dekoder	4	7	2	736	408	2376	2736	55,435	86,842
3	wim	4	7	2	712	464	2408	2688	65,169	89,583
4	newbyte	5	8	2	592	424	2248	2656	71,622	84,639
5	p82	5	14	3	2368	1072	5384	7456	45,27	72,21
6	m1	6	12	2	3064	808	5104	8432	26,371	60,531
7	newapla2	6	7	2	600	488	2320	2464	81,333	94,156
8	sqr6	6	12	2	2648	1672	5552	7600	63,142	73,053
9	inc	7	9	2	2376	1688	5296	6432	71,044	82,338
10	newcpla2	7	10	2	1896	1064	4192	5680	56,118	73,803
11	max128	7	24	4	20192	1408	24256	45184	6,973	53,683
12	m2	8	16	3	10096	2288	14328	23328	22,662	61,42
13	m3	8	16	3	13464	3088	18496	30064	22,935	61,522
14	m4	8	16	3	18704	4944	25592	40544	26,433	63,122
15	mlp4	8	8	2	7224	7728	16184	15920	106,977	101,658
16	tms	8	16	3	6784	2080	10808	16704	30,66	64,703
17	dk27	9	9	2	528	1056	2816	2736	200	102,924
18	max512	9	6	1	9632	4432	14584	20320	46,013	71,772
19	newcpla1	9	16	3	2520	2840	7304	8176	112,698	89,335
20	newxcpla1	9	23	4	3760	2304	8720	12112	61,277	71,995
21	sex	9	14	3	1360	3184	6488	5440	234,118	119,265
Среднее									68,643	78,948

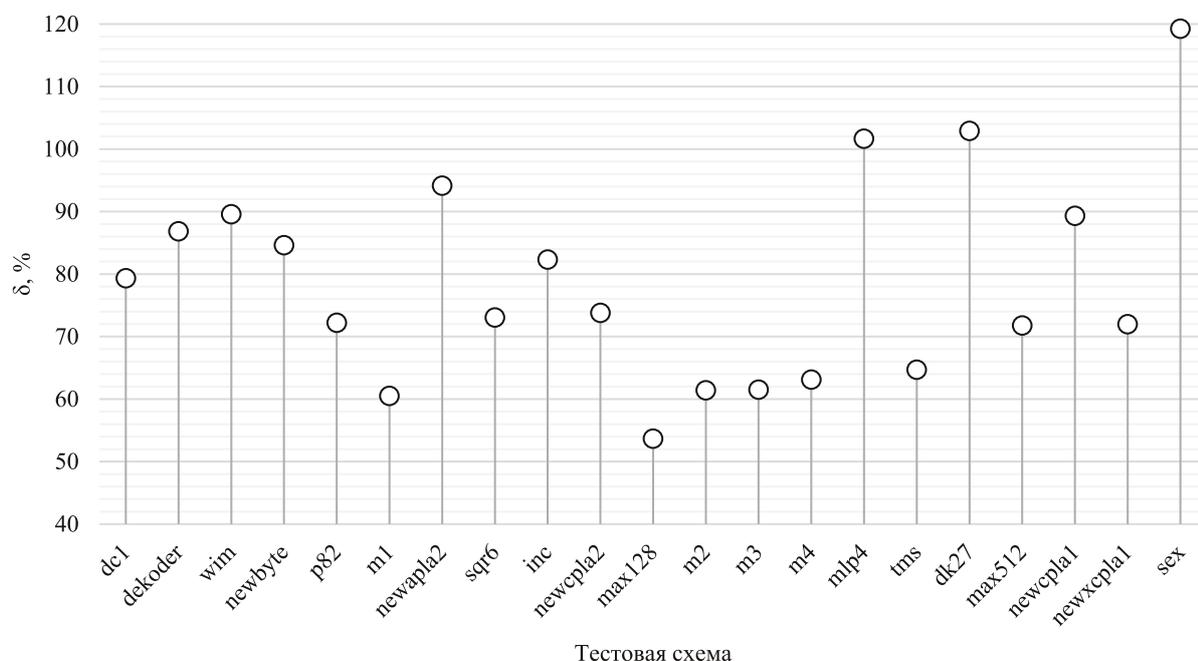


Рис. 5. Графическое представление показателя δ для тестовых схем в качестве базового алгоритма 2 при синтезе СВК

Таблица 10. Число контрольных векторов, поступающих на тестеры в СВК, синтезированных с применением алгоритма 1

№	Схема	Группа 1 — TSC ₁						Группа 2 — TSC ₂						Группа 3 — TSC ₃						Группа 4 — TSC ₄				
		Контрольные векторы						Контрольные векторы						Контрольные векторы						Контрольные векторы				
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10
1	dc1	11	4	1	0	11	2	0	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	decoder	4	2	5	5	6	7	2	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	wim	8	3	0	5	3	4	4	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	newbyte	28	3	1	0	28	3	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	p82	18	8	3	3	27	1	0	4	28	3	1	0	-	-	-	-	-	-	-	-	-	-	-
6	m1	51	4	7	2	32	25	0	7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	newapla2	60	3	1	0	60	3	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	sqrb	12	20	20	12	23	13	15	13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	inc	58	34	26	10	43	27	37	21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	newcpla2	106	17	4	1	73	55	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	max128	77	44	0	7	81	4	33	10	83	32	12	1	73	55	0	0	-	-	-	-	-	-	-
12	m2	210	15	20	11	181	38	15	22	160	95	0	1	-	-	-	-	-	-	-	-	-	-	-
13	m3	191	20	30	15	158	47	22	29	128	117	0	11	-	-	-	-	-	-	-	-	-	-	-
14	m4	138	40	49	29	57	111	32	56	12	202	0	42	-	-	-	-	-	-	-	-	-	-	-
15	mip4	79	55	65	57	93	56	59	48	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	tms	184	37	25	10	196	57	0	3	165	20	59	12	-	-	-	-	-	-	-	-	-	-	-
17	dk27	304	80	64	64	80	304	32	96	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	max512	117	160	121	114	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	newcpla1	258	226	14	14	278	52	110	72	262	20	0	230	-	-	-	-	-	-	-	-	-	-	-
20	newxcpla1	42	74	394	2	410	98	2	2	80	80	328	24	20	356	132	4	-	-	-	-	-	-	-
21	sex	176	244	88	4	47	110	255	100	207	126	159	20	-	-	-	-	-	-	-	-	-	-	-

Таблица 1 1. Число контрольных векторов, поступающих на тестеры в СВК, синтезированных с применением алгоритма 2

№	Схема	Группа 1 — TSC ₁						Группа 2 — TSC ₂						Группа 3 — TSC ₃						Группа 4 — TSC ₄			
		Контрольные векторы						Контрольные векторы						Контрольные векторы						Контрольные векторы			
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11		
1	dc1	6	2	1	7	6	4	2	4	6	4	2	4	-	-	-	-	-	-	-			
2	decoder	2	2	1	11	3	7	1	5	3	7	1	5	-	-	-	-	-	-	-			
3	wim	0	2	3	11	0	2	2	12	0	2	2	12	-	-	-	-	-	-	-			
4	newbyte	30	1	1	0	30	1	1	0	30	1	1	0	-	-	-	-	-	-	-			
5	p82	17	4	7	4	14	5	5	8	14	5	5	8	25	0	1	6	-	-	-			
6	m1	35	14	10	5	35	6	8	15	35	6	8	15	-	-	-	-	-	-	-			
7	newapla2	62	1	1	0	62	1	1	0	62	1	1	0	-	-	-	-	-	-	-			
8	sqg6	32	32	0	0	20	16	16	12	20	16	16	12	-	-	-	-	-	-	-			
9	inc	80	20	24	4	68	20	40	0	68	20	40	0	-	-	-	-	-	-	-			
10	newcpla2	76	12	33	7	114	0	10	4	114	0	10	4	-	-	-	-	-	-	-			
11	max128	76	12	33	7	120	0	7	1	120	0	7	1	59	0	36	33	114	0	10	4		
12	m2	172	39	26	19	254	1	0	1	254	1	0	1	169	12	18	57	-	-	-			
13	m3	146	51	32	27	251	4	0	1	251	4	0	1	139	15	23	79	-	-	-			
14	m4	43	115	54	44	239	13	2	2	239	13	2	2	33	31	45	147	-	-	-			
15	mip4	128	32	64	32	80	56	64	56	80	56	64	56	-	-	-	-	-	-	-			
16	tms	218	1	37	0	162	2	20	72	162	2	20	72	163	12	7	74	-	-	-			
17	dk27	120	128	136	128	368	128	16	0	368	128	16	0	-	-	-	-	-	-	-			
18	max512	109	120	138	145	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
19	newcpla1	391	103	9	9	417	72	23	0	417	72	23	0	432	64	16	0	-	-	-			
20	newxcpla1	492	4	16	0	24	8	40	440	24	8	40	440	32	0	468	12	504	4	4	0		
21	sex	320	64	128	0	200	0	184	128	200	0	184	128	116	244	140	12	-	-	-	-		

Таблица 12. Число тестовых комбинаций для элементов преобразования в СВК, синтезированных с применением алгоритма 1

№	Схема	Группа 1						Группа 2						Группа 3						Группа 4					
		XOR	Тестовые комбинации			XOR	Тестовые комбинации			XOR	Тестовые комбинации			XOR	Тестовые комбинации			XOR	Тестовые комбинации						
			00	01	10		11	00	01		10	11	00		01	10	11		00	01	10	11			
1	dc1	XOR ₄	8	0	1	7	XOR ₅	8	1	5	2	-	-	-	-	-	-	-	-	-					
		XOR ₃	6	1	3	6	XOR ₄	8	0	5	3	-	-	-	-	-	-	-	-	-					
2	dekode	XOR ₄	1	3	7	5	XOR ₅	7	3	0	6	-	-	-	-	-	-	-	-	-					
		XOR ₃	2	1	6	7	XOR ₄	2	2	6	6	-	-	-	-	-	-	-	-	-					
3	wim	XOR ₄	2	0	5	9	XOR ₅	2	0	9	5	-	-	-	-	-	-	-	-	-					
		XOR ₃	1	2	6	7	XOR ₄	1	1	8	6	-	-	-	-	-	-	-	-	-					
4	newbyte	XOR ₄	30	1	0	1	XOR ₆	30	1	0	1	-	-	-	-	-	-	-	-	-					
		XOR ₃	28	3	0	1	XOR ₅	28	3	0	1	-	-	-	-	-	-	-	-	-					
5	p82	XOR ₄	20	1	5	6	XOR ₁₀	18	1	3	10	XOR ₁₂	25	0	1	6	-	-	-	-					
		XOR ₃	16	8	3	5	XOR ₉	16	3	2	11	XOR ₁₁	24	2	1	5	-	-	-	-					
6	m1	XOR ₄	44	5	4	11	XOR ₁₀	39	2	5	18	-	-	-	-	-	-	-	-	-					
		XOR ₃	41	4	2	17	XOR ₉	32	11	21	0	-	-	-	-	-	-	-	-	-					
7	newapla2	XOR ₄	62	1	0	1	XOR ₅	62	1	0	1	-	-	-	-	-	-	-	-	-					
		XOR ₃	60	3	0	1	XOR ₄	60	3	0	1	-	-	-	-	-	-	-	-	-					
8	sqr6	XOR ₄	32	32	0	0	XOR ₁₀	24	14	14	14	-	-	-	-	-	-	-	-	-					
		XOR ₃	16	16	16	16	XOR ₉	22	14	12	16	-	-	-	-	-	-	-	-	-					
9	inc	XOR ₄	68	32	4	24	XOR ₇	52	36	22	18	-	-	-	-	-	-	-	-	-					
		XOR ₃	73	31	13	11	XOR ₆	60	48	0	20	-	-	-	-	-	-	-	-	-					
10	newcpla2	XOR ₄	88	0	5	35	XOR ₇	114	0	0	14	-	-	-	-	-	-	-	-	-					
		XOR ₃	98	11	7	12	XOR ₆	69	55	0	4	-	-	-	-	-	-	-	-	-					
11	max128	XOR ₄	88	0	7	33	XOR ₁₀	77	43	0	8	XOR ₁₆	47	12	1	68	XOR ₂₂	114	0	14					
		XOR ₃	77	32	19	0	XOR ₉	113	14	0	1	XOR ₁₅	62	33	0	33	XOR ₂₁	69	55	4					

Таблица 13. Число тестовых комбинаций для элементов преобразования в СВК, синтезированных с применением алгоритма 2

№	Схема	Группа 1				Группа 2				Группа 3				Группа 4					
		XOR	Тестовые комбинации			XOR	Тестовые комбинации			XOR	Тестовые комбинации			XOR	Тестовые комбинации				
			00	01	10		11	00	01		10	11	00		01	10	11	00	01
1	dc1	XOR ₄	7	2	2	5	XOR ₅	8	1	5	2	-	-	-	-	-	-	-	-
		XOR ₃	6	4	1	5	XOR ₄	9	0	2	5	-	-	-	-	-	-	-	-
2	decoder	XOR ₄	2	1	6	7	XOR ₅	1	1	7	7	-	-	-	-	-	-	-	-
		XOR ₃	5	5	3	3	XOR ₄	1	2	7	6	-	-	-	-	-	-	-	-
3	wim	XOR ₄	4	6	3	3	XOR ₅	0	4	9	3	-	-	-	-	-	-	-	-
		XOR ₃	2	0	6	8	XOR ₄	8	2	1	5	-	-	-	-	-	-	-	-
4	newbyte	XOR ₄	28	3	0	1	XOR ₆	28	3	0	1	-	-	-	-	-	-	-	-
		XOR ₃	28	3	0	1	XOR ₅	28	3	0	1	-	-	-	-	-	-	-	-
5	p82	XOR ₄	19	11	0	2	XOR ₁₀	16	9	2	5	XOR ₁₂	24	7	1	0	-	-	-
		XOR ₃	21	7	4	0	XOR ₉	17	9	4	2	XOR ₁₁	25	6	1	0	-	-	-
6	m1	XOR ₄	41	17	2	4	XOR ₁₀	32	0	21	11	-	-	-	-	-	-	-	-
		XOR ₃	46	9	4	5	XOR ₉	37	2	7	18	-	-	-	-	-	-	-	-
7	newapla2	XOR ₄	60	3	0	1	XOR ₅	60	3	0	1	-	-	-	-	-	-	-	-
		XOR ₃	60	3	0	1	XOR ₄	60	3	0	1	-	-	-	-	-	-	-	-
8	sqf6	XOR ₄	24	24	8	8	XOR ₁₀	24	16	10	14	-	-	-	-	-	-	-	-
		XOR ₃	32	16	8	8	XOR ₉	24	13	14	13	-	-	-	-	-	-	-	-
9	inc	XOR ₄	74	34	12	8	XOR ₇	46	42	14	26	-	-	-	-	-	-	-	-
		XOR ₃	56	58	11	3	XOR ₆	64	44	8	12	-	-	-	-	-	-	-	-
10	newcpa2	XOR ₄	98	22	7	1	XOR ₇	59	22	10	37	-	-	-	-	-	-	-	-
		XOR ₃	102	25	1	0	XOR ₆	59	22	37	10	-	-	-	-	-	-	-	-
11	max128	XOR ₄	88	32	8	0	XOR ₁₀	113	1	0	14	XOR ₁₆	55	33	7	33	XOR ₂₂	59	22
		XOR ₃	127	0	0	1	XOR ₉	73	51	4	0	XOR ₁₅	48	61	0	19	XOR ₂₁	59	22

Из табл. 10 следует, что только для 8 тестовых схем (около 40%) с использованием базового алгоритма 1 получены структуры, где формируются все тестовые комбинации для компараторов в СВК для каждой из контролируемых групп. Еще для 7 схем (около 35%) не формируется только одна из тестовых комбинаций для одного из тестеров.

Из табл. 11 следует, что для 6 тестовых схем (около 30%) с использованием базового алгоритма 2 получены структуры, где формируются все тестовые комбинации для компараторов в СВК для каждой из контролируемых групп. Еще для 7 схем (около 35%) не формируется только одна из тестовых комбинаций для одного из тестеров.

Также важным является и формирование теста для каждого элемента преобразования в каждой СВК. В табл. 12 и 13 сведены данные по числу тестовых комбинаций элементов преобразования в каждой СВК для каждой тестовой схемы.

Из анализа табл. 12 следует, что только для 3 тестовых схем (примерно 15%) использование алгоритма 3 совместно с алгоритмом 1 дает возможность тестирования всех элементов преобразования и еще для 6 тестовых схем (примерно 30%) не формируется только одна комбинация для одного из элементов преобразования.

Из анализа табл. 13 получаем, что только для 7 тестовых схем (примерно 35%) обеспечивается тестируемость всех элементов преобразования в каждой СВК и еще для 5 схем (около 25%) не формируется только одна комбинация для одного из элементов преобразования.

Напрямую применением алгоритма 3 во многих случаях не удается обеспечивать полную самопроверяемость структур для рассмотренных тестовых комбинационных схем. Следует, однако, отметить тот факт, что никаких перестановок выходов в экспериментах не осуществлялось и фиксировались результаты, получаемые для структур, реализованных по алгоритму 3. Учитывая (7), можно сделать вывод о возможном достижении тестопригодности элементов в структурах СВК путем выбора иного способа построения. Эту задачу авторы оставили за рамками настоящего исследова-

ния, показав принципиальную возможность построения самопроверяемых структур по предлагаемому способам.

Заключение

В подавляющем числе случаев при использовании предложенных в статье способов синтеза СВК удается построить устройства с функциями автоматического обнаружения неисправностей с показателями сложности технической реализации, меньшими, чем показатели для стандартного метода дублирования. В эксперименте показано, что в среднем для рассмотренных тестовых схем при использовании алгоритма 1 в качестве базового получена величина $\delta_{cp} = 78,079\%$, а при использовании алгоритма 2 — $\delta_{cp} = 78,948\%$. При этом в ряде случаев достигается уменьшение показателя сложности реализации по сравнению с дублированием более чем на 30%. Однако при этом не всегда удается обеспечивать самопроверяемость структур единичным применением алгоритма 3 без перестановки выходов — требуется поиск такого способа выделения выходов и их расположения в информационном и контрольном векторах, при котором условия обеспечения самопроверяемости выполняются. Тем не менее следует подчеркнуть, что достичь полной самопроверяемости с применением представленных в статье способов синтеза СВК удастся всегда, когда это возможно в принципе (речь идет о том, что для некоторых устройств может просто не быть достаточного количества значений 0 или 1 на всех входных комбинациях для какой-либо из функций). Учитывая тот факт, что можно эти функции сделать непреобразуемыми, эффект может быть достигнут. Таким образом, наиболее сложной задачей для представленных способов является обеспечение тестируемости элементов преобразования и тестеров в составах СВК. Тем не менее в статье показано, что базовые алгоритмы 1 и 2 могут эффективно применяться при синтезе СВК с использованием $WS4(4, 2)$ кодов.

Перспективой дальнейших исследований отметим направления изучения $WS4(4, 2)$ кодов с иными весовыми коэффициентами рядов информационного вектора, а также в целом модульных взвешенных кодов с суммированием с произвольными значениями

m и модулями, выбираемыми из множества $M \in \{2^2, 2^3, \dots, 2^{\lceil \log_2(m+1) \rceil - 1}\}$. Кроме того, может оказаться эффективным применение преобразования не только части выходов объекта диагностирования, но и всех выходов для достижения требуемых характеристик. К примеру, подобная задача исследовалась нами ранее в частном случае для одного из способов взвешивания при построении $WS4(4, 2)$ кода в [42]. Также интерес представляют исследования в области учета характеристик взвешенных кодов с суммированием при выборе групп контролируемых выходов или организации контроля вычислений единой группы по коду с соответствующими значениями m и k .

В заключение отметим, что, по мнению авторов, ЛКС — это до конца не исследованный и перспективный инструмент для синтеза самопроверяемых устройств, который во многих случаях позволяет реализовывать высоконадежные блоки для систем критического применения в промышленности и на транспорте. ▲

Библиографический список

1. Согомонян Е. С., Слабаков Е. В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
2. Mitra S., McCluskey E. J. Which Concurrent Error Detection Scheme to Choose? // Proceedings of International Test Conference, 2000, USA, Atlantic City, NJ, 3–5 October 2000. P. 985–994. DOI: 10.1109/TEST.2000.894311.
3. Стемповский А. Л., Тельпухов Д. В., Жукова Т. Д. и др. Синтез схемы функционального контроля на основе спектрального R-кода с разбиением выходов на группы // Микроэлектроника. 2019. Т. 48, № 4. С. 284–294.
4. Stempkovsky A. L., Zhukova T. D., Telpukhov D. V., Gurov S. I. CICADA: A New Tool to Design Circuits with Correction and Detection Abilities // International Siberian Conference on Control and Communications (SIBCON), 13–15 May 2021, Kazan, Russia. P. 1–5. DOI: 10.1109/SIBCON50419.2021.9438900.
5. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Теория синтеза самопроверяемых цифровых систем на основе кодов с суммированием. СПб.: Лань, 2021. 580 с.
6. Сапожников В. В., Сапожников Вл. В. Самопроверяемые дискретные устройства. СПб: Энергоатомиздат, 1992. 224 с.
7. Piestrak S. J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
8. Останин С. А., Матросова А. Ю., Буторина Н. Б. и др. Синтез монотонных детекторов для подмножества равновесных кодов // Известия вузов. Физика. 2016. Т. 59, № 8–2. С. 79–81.
9. Butorina N., Burkatovskaya Yu., Pakhomova E. On the Self-Testing (m, n)-code Checker Design // IOP Conference Series: Materials Science and Engineering, Volume 1019, 14th International Forum on Strategic Technology (IFOST 2019) 14–17 October 2019, Tomsk, Russian Federation. P. 336–340. DOI: 10.1088/1757-899X/1019/1/012098.
10. Сапожников В. В., Сапожников Вл. В., Дмитриев А. В. и др. Организация функционального контроля комбинационных схем методом логического дополнения // Электронное моделирование. 2002. Т. 24, № 6. С. 52–66.
11. Гессель М., Морозов А. В., Сапожников В. В. и др. Логическое дополнение — новый метод контроля комбинационных схем // Автоматика и телемеханика. 2003. № 1. С. 167–176.
12. Гессель М., Морозов А. В., Сапожников В. В. и др. Контроль комбинационных схем методом логического дополнения // Автоматика и телемеханика. 2005. № 8. С. 161–172.
13. Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science + Business Media B. V., 2008. 184 p.
14. Morosow A., Sapozhnikov V. V., Sapozhnikov V. V., et al. Self-Checking Combinational Circuits with Unidirectionally Independent Outputs // VLSI Design. 1998. Vol. 5, Issue 4. P. 333–345. DOI: 10.1155/1998/20389.
15. Ефанов Д. В., Сапожников В. В., Сапожников Вл. В. Условия обнаружения неисправности логического элемента в комбинационном устройстве при функциональном контроле на основе кода Бергера // Автоматика и телемеханика. 2017. № 5. С. 152–165.
16. Efanov D. V., Sapozhnikov V. V., Sapozhnikov V. V. 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. Vol. 54, Issue 4. P. 279–290. — DOI: 10.3103/S0146411620040045.
17. Ефанов Д. В., Сапожников В. В., Сапожников Вл. В. Модифицированные коды с суммированием взвешенных переходов в системах функционального контроля комбинационных схем // Труды Института системного программирования РАН. 2017. Т. 29, № 5. С. 39–60. DOI: 10.15514/ISPRAS-2017-29(5)-3.
18. Das D. K., Roy S. S., Dmitiriev A., et al. 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. P. 33–40.
19. Пивоваров Д. В. Построение систем функционального контроля многовыходных комбинационных схем методом логического дополнения по равновесным кодам // Автоматика на транспорте. 2018. Т. 4, № 1. С. 131–149.

20. Saposhnikov V.I., Dmitriev A., Goessel M., et al. Self-Dual Parity Checking — a New Method for on Line Testing // Proceedings of 14th IEEE VLSI Test Symposium, USA, Princeton, 1996. P. 162–168.
21. Гессель М., Дмитриев А. В., Сапожников В. В. и др. Самотестируемая структура для функционального обнаружения отказов в комбинационных схемах // Автоматика и телемеханика. 1999. № 11. С. 162–174.
22. Morozov M., Saposhnikov V.V., Saposhnikov V.I., et al. New Self-Checking Circuits by Use of Bergercodes // Proceedings of 6th IEEE International On-Line Testing Workshop, Palma De Mallorca, Spain, 3–5 July 2000. P. 171–176.
23. 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.
24. 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.
25. Freiman C.V. Optimal Error Detection Codes for Completely Asymmetric Binary Channels // Information and Control. 1962. Vol. 5, Issue 1. P. 64–71. DOI: 10.1016/S0019-9958(62)90223-1.
26. Efanov D.V., Sapozhnikov V.V., Sapozhnikov V.I., et al. The Synthesis Conditions of Completely Self-Testing Embedded-Control Circuits Based on the Boolean Complement Method to the «1-out-of-m» Constant-Weight Code // Automatic Control and Computer Sciences. 2020. Vol. 54. Issue 2. P. 89–99. DOI: 10.3103/S0146411620020042.
27. Berger J.M. A Note on Error Detection Codes for Asymmetric Channels // Information and Control. 1961. Vol. 4, Issue 1. P. 68–73. DOI: 10.1016/S0019-9958(61)80037-5.
28. 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, N. Y., 10 p.
29. Efanov D., Osadchy G., Zueva M. Special Aspects of Errors Definition via Sum Codes within Embedded Control Schemas Being Realized by Means of Boolean Complement Method // Proceedings of 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2021). Cracow, Poland. September 22–25 2021. Vol. 1. P. 424–431. DOI: 10.1109/IDAACS53288.2021.9660837.
30. Efanov D.V., Sapozhnikov V.V., Sapozhnikov V.I. The Self-Checking Concurrent Error-Detection Systems Synthesis Based on the Boolean Complement to the Bose — Lin Codes with the Modulo Value $M=4$ // Electronic Modeling. 2021. Vol. 43, Issue 1. P. 28–45. DOI: 10.15407/emodel.43.01.028.
31. Efanov D., Osadchy G., Zueva M. Specifics of Error Detection with Modular Sum Codes in Concurrent Error-Detection Circuits Based on Boolean Complement Method // Proceedings of 19th IEEE East-West Design & Test Symposium (EWDTS'2021), Batumi, Georgia, September 10–13, 2021. P. 59–69. DOI: 10.1109/EWDTS52692.2021.9581036.
32. Bose B., Lin D.J. Systematic Unidirectional Error-Detection Codes // IEEE Transaction on Computers. Nov. 1985. Vol. C-34, P. 1026–1032.
33. Das D., Toubia N.A., Seuring M., et al. Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes // Proceedings of the IEEE 6th International On-Line Testing Workshop (IOLTW). Spain, Palma de Mallorca. July 3–5 2000. P. 171–176. DOI: 10.1109/OLT.2000.856633.
34. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Об одном классе кодов, пригодных для синтеза систем функционального контроля логических устройств // Вестник Ростовского государственного университета путей сообщения. 2014. № 2. С. 48–58.
35. Аксёнова Г. П. Необходимые и достаточные условия построения полностью проверяемых схем свертки по модулю 2 // Автоматика и телемеханика. 1979. № 9. С. 126–135.
36. Sentovich E.M., Singh K.J., Moon C., et al. Sequential Circuit Design Using Synthesis and Optimization // Proceedings IEEE International Conference on Computer Design: VLSI in Computers & Processors. Cambridge, MA, USA. 11–14 October 1992, P. 328–333. DOI: 10.1109/ICCD.1992.276282.
37. SIS: A System for Sequential Circuit Synthesis / E. M. Sentovich, K.J. Singh, L. Lavagno, et al. // Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkeley. 4 May 1992. 45 p.
38. Закревский А. Д., Поттосин Ю. В., Черемисинова Л. Д. Логические основы проектирования дискретных устройств. М.: Физматлит, 2007. 592 с.
39. Ефанов Д. В. Синтез самопроверяемых комбинационных устройств на основе метода логической коррекции сигналов с применением кодов Боуза — Лина // Информационные технологии. 2023. Т. 29, № 10. С. 503–511. DOI: 10.17587/it.29.503-511.
40. McElvain K. IWLS'93 Benchmark Set: Version 4.0. — Distributed as a part of IWLS'93 benchmark set, May 1993.
41. Collection of Digital Design Benchmarks [Электронный ресурс]. URL: <https://ddd.fit.cvut.cz/www/prj/Benchmarks/> (дата обновления: 10.02.2024).
42. Efanov D.V., Yelina Y.I. Synthesis of Concurrent Error-Detection Circuits Based on Boolean Signals Correction Using Modular Weight-Based Sum Codes // Proceedings of the 2024 Conference of Young Researchers in Electrical and Electronic Engineering (EICon). St. Petersburg, Russia. 29–30 January 2024. P. 350–355.

TRANSPORT AUTOMATION RESEARCH. 2024. Vol. 10, No. 1. P. 74–99.
DOI: 10.20295/2412-9186-2024-10-01-74-99

Study of algorithms for synthesis of self-checking digital devices based on Boolean correction of signals using weighted Bose — Lin codes

Information about authors

Efanov D. V., D. Sc., Professor, Full-member of International Transport Academy, IEEE Member, Professor^{1,3,4}, General Director Deputy on Research²
E-mail: TrES-4b@yandex.ru

Yelina Y. I., Ph. D. student¹. E-mail: eseniya-elina@mail.ru

¹ Peter the Great St. Petersburg Polytechnic University, Institute of Mechanical Engineering, Materials and Transport, Higher School of Transport, St. Petersburg

² NIPI "TransStroybezopasnost" LLC, St. Petersburg

³ Russian University of Transport, Department of Automation, Remote Control and Communications in Railway Transport, Moscow

⁴ Tashkent State Transport University, Department of Automation and Remote Control, Tashkent

Abstract. When synthesizing self-checking digital devices based on Boolean correction of signals, it is proposed to use weight-based Bose – Lin codes, the construction principles of which imply preliminary weighting of data symbols by natural numbers. Two "basic" structures are proposed for the synthesis of built-in control circuits for groups of six outputs of the diagnostic object. The structures are based on weight-based Bose — Lin codes with summation in the residue ring modulo $M=4$. There are 15 such noise-protected codes with the number of data symbols $m=4$, which allows to select the best option as a base code in the built-in control circuit according to various criteria, including achieving self-checking properties even in cases where this cannot be achieved using traditional approaches, including duplication. Two algorithms for the synthesis of built-in control circuits based on Boolean signal correction have been developed, allowing the use of correction of only two of the six functions in the basic structure. For basic structures, there are 720 ways to construct an integrated control circuit based on Boolean correction of signals using each weight-based Bose — Lin code, which makes it possible to choose the best way to implement a self-checking device, considering various indicators (structural redundancy, testability, etc.). The operation of the algorithms is demonstrated on simple examples. The results of experiments with test digital circuits from the *MCNC Benchmarks* set confirming the efficiency of the developed algorithms are given. It is shown that with a large number of outputs, there is an astronomical number of ways to organize built-in control circuits, which makes it possible to build self-checking devices with various characteristics. The use of Boolean correction of signals using weight-based Bose – Lin codes can be used in the development and design of self-checking digital devices on various element bases.

Keywords: testing of digital circuits; self-checking device; built-in control circuit; Boolean signal correction; weight-based sum code; weight-based Bose – Lin code.

References

- Sogomonyan E. S., Slabakov E. V. Samoproveryaemye ustrojstva i otkazoustojchivye sistemy. M.: Radio i svyaz', 1989. 208 s. (In Russian)
- Mitra S., McCluskey E. J. Which Concurrent Error Detection Scheme to Choose? // Proceedings of International Test Conference, 2000, USA, Atlantic City, NJ, 3–5 October 2000. P. 985–994. DOI: 10.1109/TEST.2000.894311.
- Stempkovskij A. L., Telpuhov D. V., Zhukova T. D. i dr. Sintez skhemy funkcional'nogo kontrolya na osnove spektral'nogo R-koda s razbieniem vyhodov na gruppy // Mikroelektronika. 2019. T. 48, № 4. S. 284–294. (In Russian)

- Stempkovskij A. L., Zhukova T. D., Telpuhov D. V., et al. CICADA: A New Tool to Design Circuits with Correction and Detection Abilities // International Siberian Conference on Control and Communications (SIBCON), 13–15 May 2021, Kazan, Russia. P. 1–5. DOI: 10.1109/SIBCON50419.2021.9438900.
- Sapozhnikov V. V., Sapozhnikov V. I., Efanov D. V. Teoriya sinteza samoproveryaemykh cifrovyykh sistem na osnove kodov s summirovaniem. SPb.: Lan', 2021. 580 s. (In Russian)
- Sapozhnikov V. V., Sapozhnikov V. I. Samoproveryaemye diskretnye ustrojstva. SPb.: Energoatomizdat, 1992. 224 s. (In Russian)
- Piestrak S. J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
- Ostanin S. A., Matrosova A. YU., Butorina N. B. i dr. Sintez monotonykh detektorov dlya podmnozhestva ravnovesnykh kodov // Izvestiya vuzov. Fizika. 2016. T. 59, № 8–2. S. 79–81. (In Russian)
- Butorina N., Burkatovskaya Yu., Pakhomova E. On the Self-Testing (m, n)-code Checker Design // IOP Conference Series: Materials Science and Engineering, Volume 1019, 14th International Forum on Strategic Technology (IFOST 2019) 14th-17th October 2019, Tomsk, Russian Federation. P. 336–340. DOI: 10.1088/1757-899X/1019/1/012098.
- Sapozhnikov V. V., Sapozhnikov V. I., Dmitriev A. V. i dr. Organizatsiya funkcional'nogo kontrolya kombinatsionnykh skhem metodom logicheskogo dopolneniya // Elektronnoe modelirovanie. 2002. T. 24, № 6. S. 52–66. (In Russian)
- Gessel' M., Morozov A. V., Sapozhnikov V. V. i dr. Logicheskoe dopolnenie — novyy metod kontrolya kombinatsionnykh skhem // Avtomatika i telemekhanika. 2003. № 1. S. 167–176. (In Russian)
- Gessel' M., Morozov A. V., Sapozhnikov V. V. i dr. Kontrol' kombinatsionnykh skhem metodom logicheskogo dopolneniya // Avtomatika i telemekhanika. 2005. № 8. S. 161–172. (In Russian)
- Göessel M., Ocheretny V., Sogomonyan E., et al. New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science + Business Media B. V., 2008. 184 p.
- Morosov A., Sapozhnikov V. V., Sapozhnikov V. I., et al. Self-Checking Combinational Circuits with Unidirectionally Independent Outputs // VLSI Design. 1998. Vol. 5, Issue 4. P. 333–345. DOI: 10.1155/1998/20389.
- Efanov D. V., Sapozhnikov V. V., Sapozhnikov V. I. Usloviya obnaruzheniya neispravnosti logicheskogo elementa v kombinatsionnom ustrojstve pri funkcional'nom kontrole na osnove koda Bergera // Avtomatika i telemekhanika. 2017. № 5. S. 152–165. (In Russian)
- 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. Vol. 54, Issue 4. P. 279–290. DOI: 10.3103/S0146411620040045.
- Efanov D. V., Sapozhnikov V. V., Sapozhnikov V. I. Modificirovannyye kody s summirovaniem vzveshennykh perekhodov v sistemakh funkcional'nogo kontrolya kombinatsionnykh skhem // Trudy Instituta sistemnogo programmirovaniya RAN. 2017. T. 29, № 5. S. 39–60. DOI: 10.15514/ISPRAS-2017-29(5)-3. (In Russian)
- Das D. K., Roy S. S., Dmitriev A., et al. 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. P. 33–40.
- Pivovarov D. V. Postroyeniye sistem funkcional'nogo kontrolya mnogovyhodnykh kombinatsionnykh skhem metodom logicheskogo dopolneniya po ravnovesnykh kodam // Avtomatika na transporte. 2018. T. 4, № 1. S. 131–149. (In Russian)
- Sapozhnikov V. I., Dmitriev A., Goessel M., et al. Self-Dual Parity Checking — a New Method for on Line Testing // Proceedings of 14th IEEE VLSI Test Symposium, USA, Princeton, 1996. P. 162–168.
- Gessel' M., Dmitriev A. V., Sapozhnikov V. V., et al. Samotestiruemaya struktura dlya funkcional'nogo obnaruzheniya otkazov v kombinatsionnykh skhemah // Avtomatika i telemekhanika. 1999. № 11. S. 162–174. (In Russian)

22. Morozov M., Sapozhnikov V. V., Sapozhnikov VI. V., et al. New Self-Checking Circuits by Use of Berger-codes // Proceedings of 6th IEEE International On-Line Testing Workshop, Palma De Mallorca, Spain, 3–5 July 2000. P. 171–176.
23. Efanov D. V., Pogodina T. S. Issledovanie svoystv samodvoystvennykh kombinatsionnykh ustrojstv s kontrol'em vychislenij na osnove kodov Hemminga // Informatika i avtomatizatsiya. 2023. T. 22, № 2. S. 349–392. DOI: 10.15622/ia.22.2.5. (In Russian)
24. Efanov D. V. Osobennosti realizatsii samoproveryaemykh struktur na osnove metoda invertirovaniya dannykh i lineynykh kodov // Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika. 2023. № 65. S. 126–138. DOI: 10.17223/19988605/65/13. (In Russian)
25. Freiman C. V. Optimal Error Detection Codes for Completely Asymmetric Binary Channels // Information and Control. 1962. Vol. 5, Issue 1. P. 64–71. DOI: 10.1016/S0019-9958(62)90223-1.
26. Efanov D. V., Sapozhnikov V. V., Sapozhnikov VI. V., et al. The Synthesis Conditions of Completely Self-Testing Embedded-Control Circuits Based on the Boolean Complement Method to the «1-out-of-m» Constant-Weight Code // Automatic Control and Computer Sciences. 2020. Vol. 54, Issue 2. P. 89–99. DOI: 10.3103/S0146411620020042.
27. Berger J. M. A Note on Error Detection Codes for Asymmetric Channels // Information and Control. 1961. Vol. 4, Issue 1. P. 68–73. DOI: 10.1016/S0019-9958(61)80037-5.
28. 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, N. Y., 10 p.
29. Efanov D., Osadchy G., Zueva M. Special Aspects of Errors Definition via Sum Codes within Embedded Control Schemas Being Realized by Means of Boolean Complement Method // Proceedings of 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2021). Cracow, Poland. September 22–25 2021. Vol. 1. P. 424–431. DOI: 10.1109/IDAACS53288.2021.9660837.
30. Efanov D. V., Sapozhnikov V. V., Sapozhnikov VI. V. The Self-Checking Concurrent Error-Detection Systems Synthesis Based on the Boolean Complement to the Bose — Lin Codes with the Modulo Value $M=4$ // Electronic Modeling. 2021. Vol. 43, Issue 1. P. 28–45. DOI: 10.15407/emodel.43.01.028.
31. Efanov D., Osadchy G., Zueva M. Specifics of Error Detection with Modular Sum Codes in Concurrent Error-Detection Circuits Based on Boolean Complement Method // Proceedings of 19th IEEE East-West Design & Test Symposium (EWDTS'2021), Batumi, Georgia, September 10–13 2021. P. 59–69. DOI: 10.1109/EWDTS52692.2021.9581036.
32. Bose B., Lin D. J. Systematic Unidirectional Error-Detection Codes // IEEE Transaction on Computers. Nov. 1985. Vol. C-34. P. 1026–1032.
33. Das D., Toubia N. A., Seuring M., et al. Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes // Proceedings of the IEEE 6th International On-Line Testing Workshop (IOLTW). Spain, Palma de Mallorca. July 3–5, 2000. P. 171–176. DOI: 10.1109/OLT.2000.856633.
34. Sapozhnikov V. V., Sapozhnikov VI. V., Efanov D. V. Ob odnom klasse kodov, prigodnykh dlya sinteza sistem funktsional'nogo kontrolya logicheskikh ustrojstv // Vestnik Rostovskogo gosudarstvennogo universiteta putej soobshcheniya. 2014. № 2. S. 48–58. (In Russian)
35. Aksonova G. P. Neobhodimye i dostatochnye usloviya postroeniya polnost'yu proveryaemykh skhem svertki po modulyu 2 // Avtomatika i telemekhanika. 1979. № 9. S. 126–135. (In Russian)
36. Sentovich E. M., Singh K. J., Moon C., et al. Sequential Circuit Design Using Synthesis and Optimization // Proceedings IEEE International Conference on Computer Design: VLSI in Computers & Processors. Cambridge, MA, USA. 11–14 October 1992. P. 328–333. DOI: 10.1109/ICCD.1992.276282.
37. SIS: A System for Sequential Circuit Synthesis / E. M. Sentovich, K. J. Singh, L. Lavagno, et al. // Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkeley. 4 May 1992. 45 p.
38. Zakrevskij A. D., Pottosin YU. V., Cherenisina L. D. Logicheskie osnovy proektirovaniya diskretnykh ustrojstv. M.: Fizmatlit, 2007. 592 s. (In Russian)
39. Efanov D. V. Sintez samoproveryaemykh kombinatsionnykh ustrojstv na osnove metoda logicheskoy korrektsii signalov s primeneniem kodov Bouza – Lina // Informatsionnye tekhnologii. 2023. T. 29, № 10. S. 503–511. DOI: 10.17587/it.29.503-511. (In Russian)
40. McElvain K. IWLS'93 Benchmark Set: Version 4.0. — Distributed as a part of IWLS'93 benchmark set, May 1993.
41. Collection of Digital Design Benchmarks. [Elektronnyj resurs]. URL: <https://ddd.fit.cvut.cz/www/prj/Benchmarks/> (data obnovleniya: 10.02.2024).
42. Efanov D. V., Yelina Y. I. Synthesis of Concurrent Error-Detection Circuits Based on Boolean Signals Correction Using Modular Weight-Based Sum Codes // Proceedings of the 2024 Conference of Young Researchers in Electrical and Electronic Engineering (ECon). St. Petersburg, Russia. 29–30 January 2024. P. 350–355.