

Д.В. ЕФАНОВ

**СИНТЕЗ САМОПРОВЕРЯЕМЫХ КОМБИНАЦИОННЫХ
УСТРОЙСТВ НА ОСНОВЕ КОДОВ С ЭФФЕКТИВНЫМ
ОБНАРУЖЕНИЕМ СИММЕТРИЧНЫХ ОШИБОК**

Ефанов Д.В. Синтез самопроверяемых комбинационных устройств на основе кодов с эффективным обнаружением симметричных ошибок.

Аннотация. При создании надежных и безопасных компонентов систем автоматического управления часто используются методы помехоустойчивого кодирования — как при передаче данных между узлами системы, так и на уровне архитектуры аппаратных и программных средств. Широко применяется избыточное кодирование при организации контроля комбинационных логических устройств. При этом используются коды, ориентированные именно на обнаружение, а не исправление ошибок. Такие особенности кодов позволяют реализовывать контролепригодные системы автоматики с приемлемой избыточностью, не превышающей избыточности при использовании дублирования. В статье освещается метод синтеза самопроверяемых комбинационных устройств, позволяющий учитывать при решении задачи построения технических средств диагностирования особенности архитектуры исходных устройств, а также свойства обнаружения ошибок избыточными кодами. Даются базовые сведения из теории синтеза контролепригодных дискретных систем на основе избыточных кодов с суммированием. Определены ключевые этапы анализа топологий объектов диагностирования с выделением специальных групп выходов — групп структурно и функционально симметрично независимых выходов устройств. Приводятся формулы, позволяющие установить наличие или отсутствие симметричной зависимости выходов объекта диагностирования. Дается пример, иллюстрирующий процесс вычислений. Сформулированы основные этапы анализа применения избыточных кодов при выявлении ошибок на функционально симметрично зависимых выходах. Дан алгоритм синтеза самопроверяемых логических устройств с учетом особенностей структуры объекта диагностирования и свойств избыточных кодов.

Ключевые слова: логические устройства автоматики, контролепригодная структура, техническая диагностика, диагностирование, контроль технического состояния, равномерный блочный код, код Бергера, коды с суммированием, обнаружение ошибок.

1. Введение. Важная задача разработки и конструирования современных аппаратно-программных систем автоматического управления и контроля — обеспечение высокого уровня надежности и безопасности функционирования их компонентов. Эта задача решается за счет разнообразных методов как на аппаратном, так и на программном уровнях системы: широко используются методы структурного, временного и информационного резервирования и технического диагностирования [1-7]. Для контроля корректности выполняемых вычислительных процедур по реализации ответственных технологических алгоритмов применяются методы сигнатурного анализа и сканирования, используются структуры устройств, обладающие свойством самопроверяемости, а также проводится тестирование блоков и компонентов в свободное от выполнения ими своих функций время [8, 9].

Еще на этапе разработки и синтеза компонентов систем управления в структуры будущих устройств закладываются такие технические решения, которые позволяют обеспечить наиболее рациональную реализацию процедур технического диагностирования [10, 11]. Это обеспечивает реализацию контролепригодных систем автоматического управления. При этом часто учитываются и другие требования по реализации компонентов, ориентированные на снижение избыточности аппаратных средств, повышение их быстродействия, снижение энергопотребления и в конечном итоге снижение капиталовложений на их разработку и затрат на последующую эксплуатацию [12-15].

Вопросам синтеза контролепригодных дискретных систем посвящено множество публикаций отечественных и зарубежных ученых [16-21]. Среди методов синтеза подобных систем особое место занимает применение равномерных блочных кодов, в которых глубоко проработаны вопросы использования равновесных кодов и кодов с суммированием [22-27]. Зачастую снижая требования к классам идентифицируемых неисправностей (и, например, ограничиваясь некоторой общепринятой моделью неисправностей), можно синтезировать устройства с простыми и контролепригодными структурами, а избыточность конечных блоков становится меньшей, чем при использовании традиционного подхода дублирования [16, 28].

Данная работа посвящена развитию методов применения избыточного кодирования при построении контролепригодных дискретных систем и освещает новый подход, который учитывает как особенности топологии исходного комбинационного логического устройства, так и свойства выбираемого для организации технического диагностирования избыточного кода.

2. Постановка задачи. Рассматриваемый класс логических устройств — это класс устройств, не обладающих памятью (комбинационные логические схемы). Такие устройства могут быть формально описаны различными способами. Если речь идет о разрабатываемом устройстве с известной элементной базой, например простейшие логические элементы, реализующие элементарные функции алгебры логики, то описание содержит: набор входов и выходов устройства; состав логических элементов; конфигурацию логических связей между входами и выходами устройства, а также между входами и выходами логических элементов. Такой подход в описании используется во всех современных средствах логического проектирования — как с изображением структурной схемы, так и с описанием на уровне языка логического проектирования [29].

Для организации контроля технического состояния комбинационных логических устройств используются схемы контроля, которые реализуются исходя из требований покрытия максимального количества физических дефектов (согласно выбранной или заданной модели неисправностей), а также с установленными ограничениями на структурную избыточность получаемой системы, ее энергопотребление, быстродействие и так далее.

В данной работе ставится следующая задача: разработать известные алгоритмы синтеза комбинационных логических устройств на основе избыточных кодов с обнаружением монотонных ошибок с целью более полного учета свойств кодов по обнаружению других классов ошибок для снижения структурной избыточности синтезируемой контролепригодной системы.

3. Особенности обнаружения ошибок кодами с суммированием. Использование избыточных равномерных кодов, ориентированных на обнаружение ошибок, а не на их исправление, при построении систем автоматики с обнаружением неисправностей связано с диагностическими способностями первых. Те свойства, которые присущи избыточным кодам и используются при организации сетей передачи данных, применяются и для решения задач технической диагностики [30-32]. Известно [33, 34], что равновесные коды, а также классические коды с суммированием (коды Бергера) обладают возможностью обнаружения любых однонаправленных искажений (монотонных ошибок) в разрядах кодовых слов. Эта особенность данных кодов применяется при условии использования асимметричных каналов передачи данных (в таких каналах возможны только монотонные проявления помех). С другой стороны, при построении надежных систем автоматического управления использование свойства обнаружения любых монотонных ошибок равновесными кодами и кодами Бергера позволяет осуществлять контроль технического состояния их компонентов именно по свойству монотонного проявления любых ошибок в структуре на их выходах. Это свойство широко используется как при решении задач тестового, так и рабочего диагностирования [32, 35-39]. Кроме того, при синтезе устройств автоматики еще на этапе кодирования состояний абстрактного автомата часто применяется кодирование этих состояний кодом с обнаружением монотонных ошибок, что исключает наличие состояний при работе устройства, а также облегчает процедуры технического диагностирования [40]. Свойство монотонного проявления неисправностей учитывается и в современных системах автоматизированного проектирования логических устройств автоматики и вычислительной техники.

С точки зрения использования равномерных блочных кодов при решении задач синтеза контролепригодных устройств автоматики и организации их диагностического обеспечения, ошибки в кодовых словах (или же в информационных векторах кодовых слов) классифицируются на монотонные и немонотонные ошибки (рисунок 1). В классе монотонных ошибок выделяются одиночные ошибки, которые обнаруживаются любыми помехоустойчивыми кодами. Немонотонные ошибки принято разделять на класс симметричных и класс асимметричных ошибок [41]. При возникновении немонотонной ошибки четной кратностью и равном количестве искажений нулевых и единичных разрядов ошибку относят к симметричному типу. Остальные немонотонные ошибки, связанные с искажениями неравного количества нулевых и единичных разрядов, являются асимметричными.

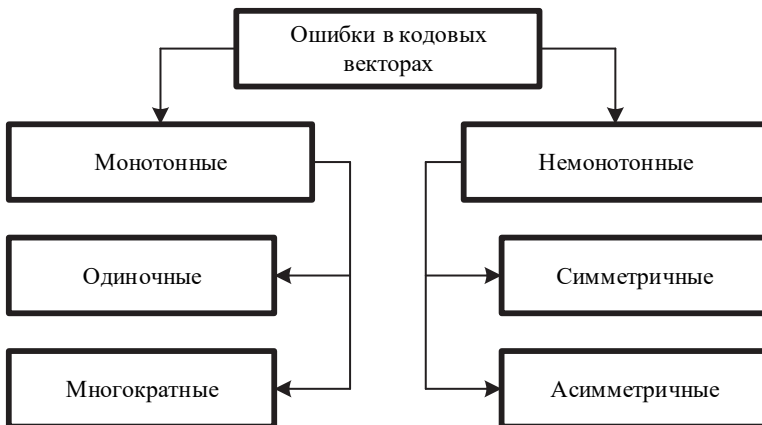


Рис. 1. Классификация ошибок в кодовых словах (или информационных векторах кодовых слов)

В [41] показано, что с увеличением количества разрядов в кодовых словах или информационных векторах кодовых слов доля монотонных ошибок от общего количества возможных ошибок стремительно уменьшается (в том числе и одиночных), тогда как доля асимметричных ошибок от общего их количества, наоборот, существенно возрастает. Симметричные же ошибки могут иметь только четную кратность, и их доля от общего количества ошибок является существенной. К примеру, при количестве разрядов, равным 10, распределение ошибок между одиночными, монотонными, симметричными и асимметричными ошибками следующее: 0,978%, 10,101%, 17,539%, 71,383%. Аналогичная закономерность наблюдается и при увеличении кратности монотонных, симметричных и асимметричных ошибок.

Остановимся на рассмотрении широкого класса кодов с суммированием [42]. Введем их обозначение — (m, k) -коды, где m и k — количество информационных и контрольных разрядов (длины информационных и контрольных векторов). Классические коды Бергера ($S(m, k)$ -коды), которые строятся путем подсчета числа нулевых или единичных разрядов в информационных векторах и записи полученного числа в двоичном виде в разряды контрольного вектора, обладают свойством идентификации любых монотонных и асимметричных ошибок. Класс таких кодов обозначим как $UAED(m, k)$ -коды (unidirectional and asymmetrical error-detection codes). Однако ценой этого свойства является невозможность обнаружения любых симметричных ошибок, число которых сравнительно велико (к примеру, это 50% двукратных и 37,5% четырехкратных ошибок в информационных векторах [43]). Избыточность $S(m, k)$ -кодов определяется величиной $k = \lceil \log_2(m+1) \rceil$, где запись $\lceil \dots \rceil$ обозначает целое сверху от вычисляемого значения. По своим характеристикам m и k и свойству обнаружения любых монотонных и асимметричных ошибок коды Бергера являются оптимальными кодами, обнаруживающими максимум обозначенных видов ошибок.

Существует большое количество кодов с суммированием, ориентированных на сохранение свойства обнаружения любых монотонных ошибок или любых монотонных ошибок до определенной кратности d_0 — так называемых, $UED(m, k)$ или d_0 - $UED(m, k)$ кодов [25]. Такие коды строятся путем различных модификаций классических кодов. Некоторые из них реализуемы только для частных случаев значений длин информационных векторов. Например, известны модификации кодов, связанные с использованием операции конкатенации контрольных векторов иных различных («базовых») кодов при образовании новых кодов с заданными свойствами. Известны также и коды, которые возможно строить для любых значений длин информационных векторов. Наверное, самыми известными из таких кодов являются коды Боуза — Лина (коды с суммированием единичных информационных разрядов в кольце вычетов по модулю $M=4$ или $M=8$ ($SM(m, k)$ -коды)). Эти коды относятся к 4 - $UED(m, k)$ и 8 - $UED(m, k)$ кодам соответственно.

При использовании свойства обнаружения любых монотонных ошибок в процессе решения задач технической диагностики и синтеза контролепригодных систем автоматики никак не учитываются возможности кодов по обнаружению симметричных и асимметричных ошибок. $S(m, k)$ и $SM(m, k)$ коды не обнаруживают 100% симметричных ошибок. Однако известны модификации данных кодов, которые обладают уменьшенной долей необнаруживаемых симметричных ошибок в информационных векторах. При этом имеется некоторая доля моно-

тонных и асимметричных ошибок. Такие модифицированные коды с суммированием строятся за счет различных правил, включающих в себя взвешивание разрядов или переходов между разрядами, занимающими соседние позиции в информационных векторах, вычисление поправочных коэффициентов в виде сверток по модулю два, выделение подмножеств разрядов информационных векторов и отдельный их контроль и прочее [42, 44, 45]. Используя свойства кодов с суммированием, а также особенности топологии устройств автоматики, можно синтезировать системы автоматики с уменьшенной по сравнению с дублированием избыточностью [16, 30, 37]. Возможности обнаружения (m,k) -кодом некоторой доли симметричных ошибок при обнаружении любых монотонных ошибок (либо любых монотонных ошибок до определенной кратности) могут быть использованы для сокращения структурной избыточности при преобразованиях схем объектов диагностирования (рисунок 2).

Рассмотрим особенности использования свойств кодов по обнаружению монотонных и симметричных ошибок при синтезе устройств с контролепригодными структурами.

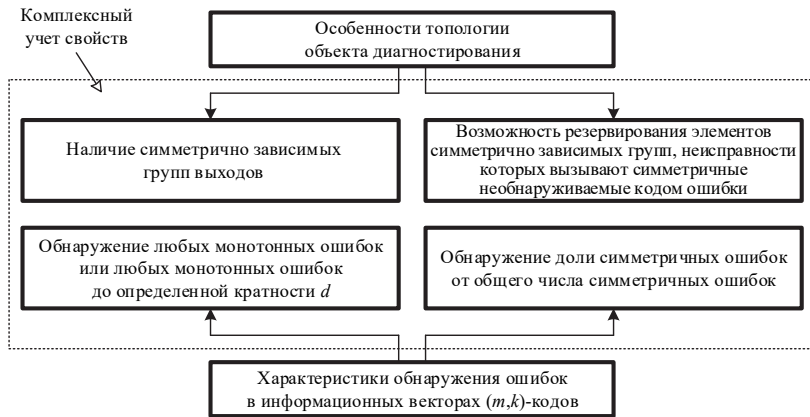


Рис. 2. Характеристики объекта диагностирования и (m,k) -кода

4. Структурно симметричная зависимость выходов. Обозначим как p_0 и p_1 — число путей, ведущих от выхода логического элемента G_q к выходам логического устройства через четное и через нечетное число инверсий соответственно (такие пути обозначим n_0 -пути и n_1 -пути). Тогда максимальная возможная кратность ошибки определяется суммой:

$$d_{\max} = p_0 + p_1. \tag{1}$$

Вид вызываемой внутренней неисправностью логического элемента ошибки на выходах логического устройства определяется соотношениями между числами p_0 и p_1 . Ошибка на выходах будет монотонной в том случае, если все n_0 -пути или же все n_1 -пути от конкретного логического элемента не будут существенными из-за компенсаций сигналов ошибок в схеме устройства (под компенсацией понимается событие поглощения ошибки корректным сигналом с исправного входа логического элемента). В противном случае, если на каком-либо входном наборе будут активизированы хотя бы один существенный n_0 -путь и хотя бы один существенный n_1 -путь, ведущие к разным выходам, это приведет к возникновению немонотонной ошибки (симметричной или асимметричной).

Анализируя топологию логического устройства, можно установить «предварительные» условия, которые будут указывать на невозможность возникновения монотонных ошибок определенной кратности.

Пусть p_0^{\max} и p_1^{\max} — это максимальное число n_0 -путей и n_1 -путей, ведущих от всех логических элементов структуры логического устройства к его выходам. Тогда справедливы следующие положения.

Утверждение 1. Монотонная ошибка на выходах логического устройства будет иметь кратность:

$$d_v \leq \max(p_0^{\max}; p_1^{\max}). \quad (2)$$

Утверждение 2. Симметричная ошибка на выходах логического устройства будет иметь кратность:

$$d_\sigma \leq 2n_{0/1}^{\max}, \quad (3)$$

где $n_{0/1}^{\max}$ — максимальное число случаев $p_0 = p_1$ для одного логического элемента.

Утверждение 3. Асимметричная ошибка на выходах логического устройства будет иметь кратность:

$$d_\alpha \leq \begin{cases} \max(p_0 + p_1), & \text{при } p_0, p_1 \neq 0; \\ \max(p_0 + p_1) - 1, & \text{при } p_0 = p_1. \end{cases} \quad (4)$$

Для примера поиска чисел d_v , d_σ и d_α рассмотрим комбинационное логическое устройство, приведенное на рисунке 3. Данное устройство имеет двухуровневую схему, снабжено четырьмя входами и шестью выходами. Описание структуры логического устройства дано в

таблице 1, где перечислены все логические элементы второго ранга и указаны характеристики путей, ведущих от каждого элемента к элементам первого ранга, выходы которых являются непосредственно и выходами самого устройства.

Таблица 1. Характеристики заданного комбинационного логического устройства

Логический элемент	p_0	p_1	n_0 -пути	n_1 -пути
G_1	2	2	G_8, G_9	G_6, G_7
G_2	3	0	G_6, G_{10}, G_{11}	–
G_3	1	1	G_7	G_{10}
G_4	2	1	G_{10}, G_{11}	G_8
G_5	1	0	G_9	–

Неисправности элемента G_5 могут вызывать только одиночные ошибки, так как от данного пути идет всего один n_0 -путь. Неисправности элемента G_3 могут вызывать либо одиночные ошибки, либо двукратные симметричные ошибки, так как для данного элемента имеется по одному n_0 -пути и n_1 -пути, ведущих к элементам G_7 и G_{10} . Ошибка на выходе элемента G_4 может трансформироваться либо в одиночную ошибку, либо в двукратную симметричную или монотонную ошибку, либо в трехкратную асимметричную ошибку. Это следует из соотношений между числом n_0 -путей и n_1 -путей. Ошибка на выходе G_1 может вызывать на выходах устройства те же варианты ошибок, плюс еще один — четырехкратную симметричную ошибку. И, наконец, неисправность элемента G_2 может повлечь за собой только одиночную или же двух- или трехкратную монотонную ошибку. Анализ показывает, что для приведенного на рисунке 2 логического устройства, согласно формулам (2)-(4):

$$d_v = 3 \leq \max(p_0^{\max}; p_1^{\max}) = \max(3; 2);$$

$$d_\sigma = 4 \leq 2n_{0/1}^{\max} = 2 \cdot 2;$$

$$d_\alpha = 3 \leq \max(p_0 + p_1) - 1 = \max(2 + 2) - 1, \text{ так как } p_0 = p_1 = 2.$$

Наличие информации о числах d_v , d_σ и d_α для конкретного логического устройства дает возможность подбора кода с суммированием для организации контроля. Поскольку при малом количестве разрядов в контрольных векторах всегда будут присутствовать симметричные необнаруживаемые ошибки [46], то и число d_σ для большого числа кодов будет равным $d_\sigma=2$ либо $d_\sigma=4$.

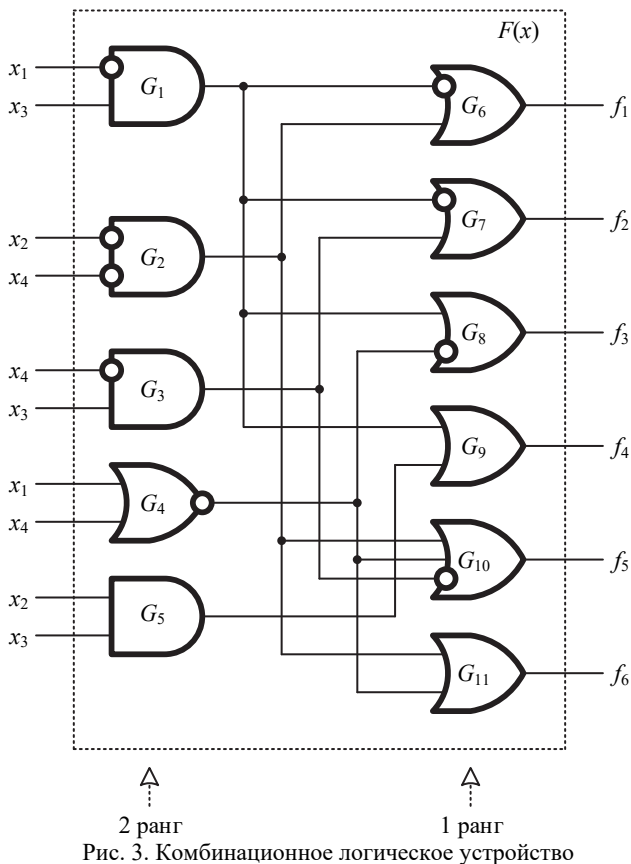


Рис. 3. Комбинационное логическое устройство

Выберем для контроля приведенного логического устройства двухмодульный код $TM(6,4)$, описанный в [47]. Данным кодом идентифицируются любые монотонные и асимметричные ошибки, однако в классе необнаруживаемых имеется 336 симметричных ошибок (192 двукратных и 144 четырехкратных). Как показано в [48], генератор $TM(6,4)$ -кода практически вдвое проще генератора классического кода Бергера, $S(6,3)$ -кода, обладающего свойством идентификации любых монотонных и асимметричных ошибок. При этом $S(6,3)$ -кодом не обнаруживается 860 симметричных ошибок (480 двукратных, 360 четырехкратных и 20 шестикратных).

Исходя из представленных характеристик $TM(6,4)$ -кода и установленных особенностей топологии комбинационного логического устройства можно сделать вывод о необходимости защиты от симмет-

ричных двукратных и четырехкратных ошибок, что реализуется путем резервирования элементов по методике, описанной в [36, 37].

Определение 1. Назовем группу выходов структурно симметрично зависимой группой выходов (ССЗ-группой), если анализ топологии показал наличие случаев $p_0 = p_1$ хотя бы для одного логического элемента в структуре логического устройства.

Из таблицы 1 следует, что три логических элемента G_1 , G_3 и G_4 связаны путями с выходами таким образом, что образуют следующие ССЗ-группы: $\{f_1; f_2; f_3; f_4\}$, $\{f_2; f_5\}$, $\{f_3; f_5; f_6\}$.

5. Функционально симметричная зависимость выходов.

Структурная симметричная зависимость выходов еще не означает реальной функциональной зависимости, поскольку визуальный анализ схемы говорит только о потенциальной возможности возникновения симметричной ошибки. Такой анализ никак не затрагивает функциональных особенностей логических элементов структуры логического устройства.

Определение 2. Назовем группу выходов функционально симметрично зависимой группой выходов (ФСЗ-группой), если хотя бы на одном входном наборе при неисправности какого-либо элемента, связанного путями с данными выходами, формируется симметричная ошибка.

Поиск ФСЗ-групп позволяет учесть и особенности структуры логического устройства и исключить случаи избыточного резервирования элементов при преобразовании схемы устройства в схему с контролепригодной топологией.

Теорема 1. Неисправность логического элемента G_q внутренней структуры логического устройства будет вызывать на выходах f_{i_1} и f_{i_2} симметричную ошибку в том случае, если выполнено условие:

$$\psi^{2,\sigma} = \frac{\partial f_{i_1}}{\partial y_q} \frac{\partial f_{i_2}}{\partial y_q} (f_{i_1} \oplus f_{i_2}) \neq 0, \quad (5)$$

где y_q – функция, реализуемая на выходе логического элемента G_q .

Доказательство. В формуле (5) множитель $\frac{\partial f_{i_1}}{\partial y_q} \frac{\partial f_{i_2}}{\partial y_q}$ определяет условия одновременного искажения выходов f_{i_1} и f_{i_2} (те входные наборы, которые вызывают двукратное искажение), а множитель $(f_{i_1} \oplus f_{i_2})$ позволяет установить вид ошибки. Если значения функций на каких-либо входных наборах были равны, то

двукратная ошибка на этих же входных наборах будет монотонной, в противном случае — симметричной. Таким образом, выражение (5) определяет те входные наборы, на которых возникает симметричная ошибка на выходах f_{i_1} и f_{i_2} . Это и зафиксировано в условии теоремы 1.

Формула (5) используется для поиска монотонно независимых групп выходов [16, 36, 37].

Теорема 2. Неисправность логического элемента G_q внутренней структуры логического устройства будет вызывать симметричную ошибку на четном количестве выходов $f_{i_1}, f_{i_2}, \dots, f_{i_d}$ в том случае, если выполнено условие:

$$\psi^{d,\sigma} = F^d F^\sigma \neq 0, \quad (6)$$

где $F^d = \frac{\partial f_{i_1}}{\partial y_q} \frac{\partial f_{i_2}}{\partial y_q} \dots \frac{\partial f_{i_d}}{\partial y_q}$, $F^\sigma = \bigvee_{f_{i_1} f_{i_2} \dots f_{i_d} \in R} f_{i_1} f_{i_2} \dots f_{i_d}$, где F^d —

функция, определяющая искажение кратностью d ; F^σ — функция, позволяющая выявить, является ли ошибка кратностью d симметричной или нет; $f_{i_1} f_{i_2} \dots f_{i_d}$ — конъюнкция значений выходных функ-

ций длиной d ; R — множество кодовых векторов с весом $\frac{d}{2}$ и длиной

d (их общее число определяется величиной $C_{d/2}^{d/2}$).

Доказательство. Функция F^d определяет входные наборы, на которых искажаются все рассматриваемые выходы, а функция F^σ — те входные наборы, на которых половина значений выходов в группе равна нулю, а половина — единице (при четном значении d ошибка будет симметричной, если хотя бы одна из конъюнкций равновесного кода « $\frac{d}{2}$ из d » не будет равна нулю). Таким образом, формула (6) позволяет определить те входные наборы, при которых происходит четное симметричное искажение рассматриваемой группы выходов.

Условие (6) необходимо проверять для всех значений четной кратности тех выходов, которые структурно допускают возможность возникновения симметричных ошибок.

Определим для рассматриваемой схемы, на каких входных наборах возникают симметричные ошибки.

Неисправности элемента G_3 могут давать только двукратные симметричные ошибки:

$$\begin{aligned}
 f_2 &= \overline{\overline{x_1 x_3} \vee \overline{x_4 x_3}} = \overline{\overline{x_1 x_3} \vee y_3}, \\
 f_5 &= \overline{x_2 x_4 \vee \overline{x_1 x_4} \vee \overline{x_4 x_3}} = \overline{x_2 x_4 \vee \overline{x_1 x_4} \vee y_3}, \\
 \frac{\partial f_2}{\partial y_3} &= \left(\overline{\overline{x_1 x_3} \vee (y_3 = 0)} \right) \oplus \left(\overline{\overline{x_1 x_3} \vee (y_3 = 1)} \right) = \overline{\overline{x_1 x_3}} \oplus 1 = \overline{x_1 x_3}, \\
 \frac{\partial f_5}{\partial y_3} &= \left(\overline{x_2 x_4 \vee \overline{x_1 x_4} \vee (y_3 = 0)} \right) \oplus \left(\overline{x_2 x_4 \vee \overline{x_1 x_4} \vee (y_3 = 1)} \right) = \\
 &= 1 \oplus \left(\overline{x_2 x_4 \vee \overline{x_1 x_4}} \right) = \overline{\overline{x_2 x_4 \vee \overline{x_1 x_4}}} = \overline{\overline{x_2 x_4} \cdot \overline{x_1 x_4}} = \\
 &= (\overline{x_2 \vee x_4}) x_1 x_4 = x_1 x_4, \\
 (\overline{x_1 x_3})(x_1 x_4) &\left(\left(\overline{\overline{x_1 x_3} \vee \overline{x_4 x_3}} \right) \oplus \overline{x_2 x_4 \vee \overline{x_1 x_4} \vee \overline{x_4 x_3}} \right) = 0.
 \end{aligned}$$

Таким образом, неисправности выхода G_3 не вызывают симметричных ошибок.

Неисправности элемента G_4 также могут давать только двукратные симметричные ошибки, причем, как следует из таблицы 1, либо в паре выходов $\{f_3; f_5\}$, либо $\{f_3; f_6\}$:

$$\begin{aligned}
 f_3 &= \overline{x_1 x_3 \vee x_1 \vee x_4} = \overline{x_1 x_3 \vee x_1 \vee y_4}, \\
 f_5 &= \overline{x_2 x_4 \vee \overline{x_1 x_4} \vee \overline{x_4 x_3}} = \overline{x_2 x_4 \vee y_4 \vee \overline{x_4 x_3}}, \\
 f_6 &= \overline{x_2 x_4 \vee \overline{x_1 x_4}} = \overline{x_2 x_4 \vee y_4}, \\
 \frac{\partial f_3}{\partial y_4} &= \left(\overline{x_1 x_3 \vee x_1 \vee (y_4 = 0)} \right) \oplus \left(\overline{x_1 x_3 \vee x_1 \vee (y_4 = 1)} \right) = \\
 &= 1 \oplus \left(\overline{x_1 x_3 \vee x_1} \right) = \overline{\overline{x_1 x_3 \vee x_1}} = \overline{x_1} (x_1 \vee \overline{x_3}) = \overline{x_1 x_3}, \\
 \frac{\partial f_5}{\partial y_4} &= \left(\overline{x_2 x_4 \vee (y_4 = 0) \vee \overline{x_4 x_3}} \right) \oplus \left(\overline{x_2 x_4 \vee (y_4 = 1) \vee \overline{x_4 x_3}} \right) = \\
 &= \left(\overline{x_2 x_4 \vee \overline{x_4 x_3}} \right) \oplus 1 = \overline{\overline{x_2 x_4 \vee \overline{x_4 x_3}}} = \\
 &= (x_4 \vee \overline{x_3})(x_2 \vee x_4) = x_4 \vee \overline{x_2 x_3}, \\
 \frac{\partial f_6}{\partial y_4} &= \left(\overline{x_2 x_4 \vee (y_4 = 0)} \right) \oplus \left(\overline{x_2 x_4 \vee (y_4 = 1)} \right) = \overline{\overline{x_2 x_4}} = x_2 \vee x_4,
 \end{aligned}$$

$$\begin{aligned}
 & \frac{\partial f_3}{\partial y_4} \frac{\partial f_5}{\partial y_4} (f_3 \oplus f_5) = (\overline{x_1 x_3}) (x_4 \vee x_2 \overline{x_3}) \times \\
 & \times \left((\overline{x_1 x_3} \vee x_1 \vee x_4) \oplus (\overline{x_2 x_4} \vee \overline{x_1 x_4} \vee \overline{x_4 x_3}) \right) = \\
 & = (\overline{x_1 x_3 x_4} \vee \overline{x_1 x_2 x_3}) \left(\overline{x_1 x_3} \vee x_1 \vee x_4 (\overline{x_2 x_4} \vee \overline{x_1 x_4} \vee \overline{x_4 x_3}) \vee \right. \\
 & \quad \left. \vee (\overline{x_1 x_3} \vee x_1 \vee x_4) \overline{x_2 x_4} \vee \overline{x_1 x_4} \vee \overline{x_4 x_3} \right) = \\
 & = (\overline{x_1 x_3 x_4} \vee \overline{x_1 x_2 x_3}) \left((x_1 \vee \overline{x_3}) \overline{x_1 x_4} (\overline{x_2 x_4} \vee \overline{x_1 x_4} \vee \overline{x_4 x_3}) \vee \right. \\
 & \quad \left. \vee (\overline{x_1 x_3} \vee x_1 \vee x_4) (x_2 \vee x_4) x_1 x_4 (\overline{x_3} \vee x_4) \right) = \\
 & = (\overline{x_1 x_3 x_4} \vee \overline{x_1 x_2 x_3}) (\overline{x_1 x_2 x_3 x_4} \vee x_1 x_4) = 0. \\
 & \frac{\partial f_3}{\partial y_4} \frac{\partial f_6}{\partial y_4} (f_3 \oplus f_6) = (\overline{x_1 x_3}) (x_2 \vee x_4) \times \\
 & \times \left((\overline{x_1 x_3} \vee x_1 \vee x_4) \oplus (\overline{x_2 x_4} \vee \overline{x_1 x_4}) \right) = \\
 & = (\overline{x_1 x_2 x_3} \vee \overline{x_1 x_3 x_4}) \left(\overline{x_1 x_3} \vee x_1 \vee x_4 (\overline{x_2 x_4} \vee \overline{x_1 x_4}) \vee \right. \\
 & \quad \left. \vee (\overline{x_1 x_3} \vee x_1 \vee x_4) \overline{x_2 x_4} \vee \overline{x_1 x_4} \right) = \\
 & = (\overline{x_1 x_2 x_3} \vee \overline{x_1 x_3 x_4}) \left((x_1 \vee \overline{x_3}) \overline{x_1 x_4} (\overline{x_1} \vee \overline{x_4}) \vee \right. \\
 & \quad \left. \vee (\overline{x_1 x_3} \vee x_1 \vee x_4) (x_2 \vee x_4) x_1 x_4 \right) = \\
 & = (\overline{x_1 x_2 x_3} \vee \overline{x_1 x_3 x_4}) (\overline{x_1 x_3 x_4} \vee x_1 x_4) = \overline{x_1 x_2 x_3 x_4} \neq 0.
 \end{aligned}$$

Неисправности выхода элемента G_4 (ошибка типа константа 0) вызывают симметричную ошибку в паре выходов $\{f_3; f_6\}$ при подаче на входы устройства двоичного набора $\langle 0100 \rangle$.

Анализ топологии логического устройства показывает, что неисправности элемента G_1 могут давать двукратные и четырехкратные симметричные ошибки. Двукратные симметричные ошибки могут возникать на выходах пар $\{f_1; f_3\}$, $\{f_1; f_4\}$, $\{f_2; f_3\}$ и $\{f_2; f_4\}$. Четырехкратные ошибки возникают во всей группе выходов $\{f_1; f_2; f_3; f_4\}$.

$$\begin{aligned}
 f_1 &= \overline{x_1 x_3} \vee \overline{x_2 x_4} = \overline{y_1} \vee \overline{x_2 x_4}, \\
 f_2 &= \overline{x_1 x_3} \vee \overline{x_4 x_3} = \overline{y_1} \vee \overline{x_4 x_3},
 \end{aligned}$$

$$f_3 = \overline{x_1 x_3} \vee x_1 \vee x_4 = y_1 \vee x_1 \vee x_4,$$

$$f_4 = \overline{x_1 x_3} \vee x_2 x_3 = y_1 \vee x_2 x_3,$$

$$\frac{\partial f_1}{\partial y_1} = \left((\overline{y_1 = 0}) \vee \overline{x_2 x_4} \right) \oplus \left((\overline{y_1 = 1}) \vee \overline{x_2 x_4} \right) = \overline{\overline{\overline{x_2 x_4}}} = x_2 \vee x_4,$$

$$\frac{\partial f_2}{\partial y_1} = \left((\overline{y_1 = 0}) \vee \overline{x_4 x_3} \right) \oplus \left((\overline{y_1 = 1}) \vee \overline{x_4 x_3} \right) = \overline{x_3} \vee x_4,$$

$$\frac{\partial f_3}{\partial y_1} = \left((\overline{y_1 = 0}) \vee x_1 \vee x_4 \right) \oplus \left((\overline{y_1 = 1}) \vee x_1 \vee x_4 \right) = \overline{x_1 x_4},$$

$$\frac{\partial f_4}{\partial y_1} = \left((\overline{y_1 = 0}) \vee x_2 x_3 \right) \oplus \left((\overline{y_1 = 1}) \vee x_2 x_3 \right) = \overline{x_2} \vee \overline{x_3},$$

$$\begin{aligned} \frac{\partial f_1}{\partial y_1} \frac{\partial f_3}{\partial y_1} (f_1 \oplus f_3) &= (x_2 \vee x_4) (\overline{x_1 x_4}) \left(\left(\overline{\overline{\overline{x_1 x_3} \vee x_2 x_4}} \right) \oplus \left(\overline{x_1 x_3} \vee x_1 \vee x_4 \right) \right) = \\ &= \overline{x_1 x_2 x_4} \left(\overline{\overline{\overline{x_1 x_3} \vee x_2 x_4}} (\overline{x_1 x_3} \vee x_1 \vee x_4) \vee \left(\overline{\overline{\overline{x_1 x_3} \vee x_2 x_4}} \right) \overline{\overline{\overline{x_1 x_3} \vee x_1 \vee x_4}} \right) = \\ &= \overline{x_1 x_2 x_4} (\overline{x_1 x_3} (x_2 \vee x_4) (\overline{x_1 x_3} \vee x_1 \vee x_4) \vee \left(\overline{\overline{\overline{x_1 x_3} \vee x_2 x_4}} \right) (x_1 \vee \overline{x_3}) \overline{x_1 x_4}) = \\ &= \overline{x_1 x_2 x_4} (\overline{x_1 x_2 x_3} \vee \overline{x_1 x_3 x_4} \vee \overline{x_1 x_3 x_4}) = \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \neq 0, \end{aligned}$$

$$\begin{aligned} \frac{\partial f_1}{\partial y_1} \frac{\partial f_4}{\partial y_1} (f_1 \oplus f_4) &= (x_2 \vee x_4) (\overline{x_2} \vee \overline{x_3}) \left(\left(\overline{\overline{\overline{x_1 x_3} \vee x_2 x_4}} \right) \oplus \left(\overline{x_1 x_3} \vee x_2 x_3 \right) \right) = \\ &= (\overline{x_2 x_4} \vee \overline{x_2 x_3} \vee \overline{x_3 x_4}) \left(\overline{\overline{\overline{x_1 x_3} \vee x_2 x_4}} (\overline{x_1 x_3} \vee x_2 x_3) \vee \right. \\ &\quad \left. \vee \left(\overline{\overline{\overline{x_1 x_3} \vee x_2 x_4}} \right) \overline{\overline{\overline{x_1 x_3} \vee x_2 x_3}} \right) = (\overline{x_2 x_4} \vee \overline{x_2 x_3} \vee \overline{x_3 x_4}) \times \\ &\quad \times \left(\overline{x_1 x_3} (x_2 \vee x_4) (\overline{x_1 x_3} \vee x_2 x_3) \vee \left(\overline{\overline{\overline{x_1 x_3} \vee x_2 x_4}} \right) (x_1 \vee \overline{x_3}) (\overline{x_2} \vee \overline{x_3}) \right) = \\ &= (\overline{x_2 x_4} \vee \overline{x_2 x_3} \vee \overline{x_3 x_4}) (\overline{x_1 x_2 x_3} \vee \overline{x_1 x_2} \vee \overline{x_3}) = \overline{x_1 x_2 x_4} \vee \overline{x_2 x_3} \vee \overline{x_3 x_4} \neq 0, \end{aligned}$$

$$\begin{aligned} \frac{\partial f_2}{\partial y_1} \frac{\partial f_3}{\partial y_1} (f_2 \oplus f_3) &= (\overline{x_3} \vee x_4) (\overline{x_1 x_4}) \left(\left(\overline{\overline{\overline{x_1 x_3} \vee x_4 x_3}} \right) \oplus \left(\overline{x_1 x_3} \vee x_1 \vee x_4 \right) \right) = \\ &= \overline{x_1 x_3 x_4} \left(\overline{\overline{\overline{x_1 x_3} \vee x_4 x_3}} (\overline{x_1 x_3} \vee x_1 \vee x_4) \vee \left(\overline{\overline{\overline{x_1 x_3} \vee x_4 x_3}} \right) \overline{\overline{\overline{x_1 x_3} \vee x_1 \vee x_4}} \right) = \\ &= \overline{x_1 x_3 x_4} (\overline{x_1 x_3} (\overline{x_3} \vee x_4) (\overline{x_1 x_3} \vee x_1 \vee x_4) \vee (x_1 \vee \overline{x_3} \vee \overline{x_4 x_3}) (x_1 \vee \overline{x_3})) = \overline{x_1 x_3 x_4} \neq 0, \end{aligned}$$

$$\begin{aligned} \frac{\partial f_2}{\partial y_1} \frac{\partial f_4}{\partial y_1} (f_2 \oplus f_4) &= (\overline{x_3 \vee x_4}) (\overline{x_2 \vee x_3}) \left(\overline{\overline{\overline{x_1 x_3 \vee x_4 x_3}}} \oplus (\overline{x_1 x_3 \vee x_2 x_3}) \right) = \\ &= (\overline{x_3 \vee x_2 x_4}) \left(\overline{\overline{\overline{x_1 x_3 \vee x_4 x_3}} (\overline{x_1 x_3 \vee x_2 x_3}) \vee \left(\overline{\overline{\overline{x_1 x_3 \vee x_4 x_3}}} \overline{\overline{\overline{x_1 x_3 \vee x_2 x_3}}} \right)} \right) = \\ &= (\overline{x_3 \vee x_2 x_4}) (\overline{x_1 x_3 x_4 \vee x_1 x_2 \vee x_3}) = \overline{x_1 x_2 x_3 x_4 \vee x_1 x_2 x_4 \vee x_3} \neq 0. \end{aligned}$$

Из приведенных выкладок следует, что все четыре выхода образуют попарно группы ФСЗ-выходов.

Обратимся к анализу группы из четырех выходов.

$$\begin{aligned} \frac{\partial f_1}{\partial y_1} \frac{\partial f_2}{\partial y_1} \frac{\partial f_3}{\partial y_1} \frac{\partial f_4}{\partial y_1} &= (x_2 \vee x_4) (\overline{x_3 \vee x_4}) (\overline{x_1 x_4}) (\overline{x_2 \vee x_3}) = \\ &= (x_2 \overline{x_3} \vee x_4) (\overline{\overline{\overline{x_1 x_2 x_4 \vee x_1 x_3 x_4}}}) = \overline{x_1 x_2 x_3 x_4}. \end{aligned}$$

Так как существует только один вариант возникновения четырехкратной симметричной ошибки в группе рассматриваемых выходов (см. рисунок 3), необходимо проверить только условие $F^\sigma = \overline{f_1 f_2 f_3 f_4} \vee f_1 f_2 \overline{f_3 f_4} \neq 0$:

$$\begin{aligned} \overline{f_1 f_2 f_3 f_4} &= \left(\overline{\overline{\overline{x_1 x_3 \vee x_2 x_4}}} \right) \left(\overline{\overline{\overline{x_1 x_3 \vee x_4 x_3}}} \right) (\overline{x_1 x_3 \vee x_1 \vee x_4}) (\overline{x_1 x_3 \vee x_2 x_3}) = \\ &= (\overline{x_1 x_3 (x_2 \vee x_4)}) (\overline{x_1 x_3 (x_3 \vee x_4)}) (\overline{x_1 x_3 \vee x_1 x_2 x_3 \vee x_2 x_3 x_4}) = \\ &= (\overline{x_1 x_2 x_3 \vee x_1 x_3 x_4}) (\overline{x_1 x_3 x_4}) (\overline{x_1 x_3 \vee x_1 x_2 x_3 \vee x_2 x_3 x_4}) = \overline{x_1 x_3 x_4}. \\ f_1 f_2 \overline{f_3 f_4} &= \left(\overline{\overline{\overline{x_1 x_3 \vee x_2 x_4}}} \right) \left(\overline{\overline{\overline{x_1 x_3 \vee x_4 x_3}}} \right) (\overline{\overline{\overline{x_1 x_3 \vee x_1 \vee x_4}}}) (\overline{\overline{\overline{x_1 x_3 \vee x_2 x_3}}}) = \\ &= (x_1 \vee \overline{x_3 \vee x_2 x_3 x_4}) (x_1 \vee \overline{x_3}) \overline{x_1 x_4} (x_1 \vee \overline{x_3}) (\overline{x_2 \vee x_3}) = \\ &= \overline{x_1 x_4} (x_1 \overline{x_2 \vee x_3}) = \overline{x_1 x_3 x_4}. \end{aligned}$$

Проверим условие (6):

$$\begin{aligned} \psi^{d=4, \sigma=4} &= F^{d=4} F^{\sigma=4} = \\ &= (\overline{\overline{\overline{x_1 x_2 x_3 x_4}}}) (\overline{\overline{\overline{x_1 x_3 x_4 \vee x_1 x_3 x_4}}}) = \overline{x_1 x_2 x_3 x_4} \neq 0, \end{aligned}$$

откуда следует, что симметричная ошибка на рассматриваемых выходах возникает при условии поступления на входы набора <0100> и формировании на выходе элемента G_1 сигнала типа константа 1.

6. Учет свойств кодов с суммированием при преобразовании структур. Если для контроля логического устройства выбирается классический код Бергера ($S(m,k)$ -код), то анализ возникающих ошибок не потребуется, поскольку любая симметричная ошибка ими обнаружена не будет [43]. В этом случае используют следующий алгоритм преобразования структур логических устройств в контролепригодные структуры [36, 37].

Алгоритм 1. Реконфигурация элементов и связей в схеме объекта диагностирования:

1. Определяется множество W таких элементов, которые допускают возникновение симметричных искажений на выходах (так называемых *немонотонных элементов*).

2. Для каждого элемента из множества W проводится анализ путей в схеме логического устройства, ведущих к выходам схемы: если на пути к выходу от немонотонного логического элемента встречается какой-либо логический элемент, то он также включается во множество W .

3. Каждый элемент $G_i \in W$ заменяется двумя своими копиями G_i^0 и G_i^1 .

4. В полученной структуре реконфигурируются соединения входов и выходов элементов по следующим правилам:

- если в исходной схеме логического устройства выход элемента G_i соединен со входом элемента G_j и соединение содержит четное число инверсий (при этом учитываются инверсии выхода элемента G_i и входа элемента G_j), то в преобразованной схеме соединяются элементы G_i и G_j с одинаковыми верхними индексами, в противном случае — с разными верхними индексами;

- если элемент G_i не был дублирован (не входил во множество W), то его выход соединяется с обеими копиями элемента G_j ;

- если выходной элемент G_i на своем выходе имеет инверсию, то соответствующий выход схемы соединяется с элементом G_i^1 , в противном случае — с G_i^0 ;

- все несвязанные с выходами схемы элементы удаляются.

Приведенный алгоритм для изображенной на рисунке 1 схемы дает результат преобразования, представленного на рисунке 4. Два логических элемента внутренней структуры логического устройства потребовалось резервировать.

При использовании двухмодульного кода для контроля заданного логического устройства целесообразно проанализировать те входные наборы, на которых вызываются симметричные ошибки и классифицировать их на обнаруживаемые и не обнаруживаемые кодом. Это

позволит установить необходимость резервирования каких-либо элементов. Другими словами, множество немонотонных элементов можно уменьшить за счет установления свойства схемы, связанного с тем, что на определенных входных наборах симметричная ошибка может оказаться обнаруженной двухмодульным кодом.

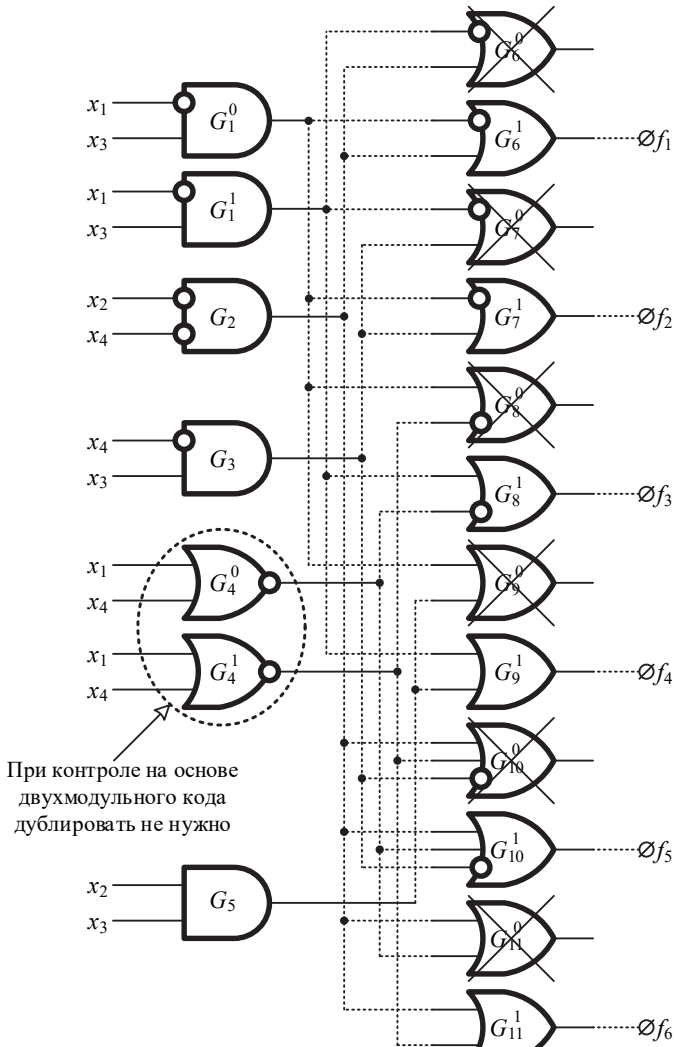


Рис. 4. Комбинационное логическое устройство с контролепригодной по коду Бергера структурой

Определение 3. Назовем группу выходов функционально симметрично независимой группой выходов при контроле кодом с суммированием (СН-группой), если ошибка, вызываемая неисправностью каждого логического элемента, связанного путями с этими выходами, идентифицируется заданным кодом с суммированием.

Для того чтобы установить, является ли группа выходов СН-группой, необходимо проанализировать «поведение» логического устройства при возникновении неисправностей при подаче на входы тех комбинаций, которые создают условия формирования симметричных ошибок на выходах устройства. Решим эту задачу, внося данные в таблицу 2.

Из таблицы 2 следует, что группы выходов $\{f_3; f_6\}$, $\{f_1; f_4\}$, $\{f_2; f_4\}$ являются СН-группами, а группы выходов $\{f_1; f_3\}$, $\{f_2; f_3\}$ и $\{f_1; f_2; f_3; f_4\}$ таковыми не являются. Отметим, что группа из четырех выходов СН-группой не является в силу того, что в нее полностью входят группы $\{f_1; f_3\}$ и $\{f_2; f_3\}$. Таким образом, при контроле схемы на основе двухмодульного кода элемент G_4 дублировать не требуется.

Отметим также, что для контроля схемы можно использовать и любой другой код, имеющий уменьшенное число симметричных обнаруживаемых ошибок. Например, к такому коду относится описанный в [49] модифицированный $RS(6,3)$ -код, для которого поправочный коэффициент вычислен по формуле $\alpha = f_3 \oplus f_4$. В классе обнаруживаемых у такого кода присутствует 224 двукратных симметричных ошибки, 56 четырехкратных монотонных и 168 симметричных ошибок и 20 шестикратных симметричных и 12 асимметричных ошибок. За счет дополнительной проверки $\alpha = f_3 \oplus f_4$ будет обеспечено обнаружение любых симметричных ошибок в парах $\{f_3; f_6\}$, $\{f_1; f_4\}$, $\{f_2; f_4\}$, $\{f_1; f_3\}$, $\{f_2; f_3\}$. Однако четырехкратная симметричная ошибка обнаружена не будет. Это потребует дублирования элемента G_1 , неисправности которого вызывают данную ошибку на входном наборе $\langle 0100 \rangle$.

Кроме приведенных кодов для контроля рассматриваемого логического устройства подходит взвешенный $WS(6,4)$ -код с последовательностью весовых коэффициентов $[w_6 \div w_1] = [112211]$ [50]. Данный код обнаруживает любые монотонные ошибки и не обнаруживает 224 двукратные симметричные, 120 четырехкратные симметричных, 12 шестикратных симметричных и 2 асимметричные ошибки. Взвешивание разрядов f_3 и f_4 обеспечит обнаружение симметричных ошибок во всех группах ФСЗ-выходов за счет разделения взвешиваемых выходов при контроле. В таблице 3 приводятся параметры об-

наружения монотонных и симметричных ошибок основными $UED(m,k)$ и d_v - $UED(m,k)$ кодами.

Таблица 2. Поиск СН-групп выходов

Элементы	Вид неисправности	Группа выходов	Комбинации	Формируемое кодовое слово без неисправности		Формируемое кодовое слово с неисправностью		Тип ошибки
				Информационный вектор	Контрольный вектор	Информационный вектор	Контрольный вектор	
G_4	Кон. 0	$\{f_3; f_6\}$	0100	110011	1010	111000	1100	Обн.
G_1	Кон. 1	$\{f_1; f_3\}$	0100	110011	1010	011011	1010	Необн.
	Кон. 0		0110	001100	0101	100100	0101	Необн.
G_1	Кон. 1	$\{f_1; f_4\}$	0001	111010	1101	011110	1010	Обн.
	Кон. 1		0100	110011	1010	010111	0111	Обн.
	Кон. 1		0101	111010	1101	011110	1010	Обн.
	Кон. 1		1001	111010	1101	011110	1010	Обн.
	Кон. 1		1011	111010	1101	011110	1010	Обн.
	Кон. 1		1100	111010	1101	011110	1010	Обн.
G_1	Кон. 1	$\{f_2; f_3\}$	0000	110011	1010	101011	1010	Необн.
	Кон. 1		0100	110011	1010	101011	1010	Необн.
G_1	Кон. 1	$\{f_2; f_4\}$	0000	110011	1010	100111	0111	Обн.
	Кон. 1		0001	111010	1101	101110	1010	Обн.
	Кон. 0		0011	001110	0110	011010	1001	Обн.
	Кон. 1		0100	110011	1010	100111	0111	Обн.
	Кон. 1		0101	111010	1101	101110	1010	Обн.
	Кон. 1		1000	111010	1101	101110	1010	Обн.
	Кон. 1		1001	111010	1101	101110	1010	Обн.
	Кон. 1		1011	111010	1101	101110	1010	Обн.
	Кон. 1		1100	111010	1101	101110	1010	Обн.
Кон. 1	1101	111010	1101	101110	1010	Обн.		
G_1	Кон. 1	$\{f_1; f_2; f_3; f_4\}$	0100	110011	1010	001111	0111	Обн.

Таблица 3. Параметры основных $UED(m,k)$ и d_0 - $UED(m,k)$ кодов при $m=8 \div 15$

Код	M	k	Параметры обнаружения монотонных ошибок	Доля необнаруживаемых симметричных ошибок от общего числа симметричных ошибок кратностью d			
				2	4	6	8
$S(m,k)$	2^{k*}	k^*	$UED(m,k)$ -код $\forall m$	100%	100%	100%	100%
$S8(m,k)$	8	3	8- $UED(m,k)$ -код $\forall m$	100%	100%	100%	100%
$SA(m,k)$	4	2	4- $UED(m,k)$ -код $\forall m$	100%	100%	100%	100%
$RS(m,k)$	2^{k*-1}	k^*	M - $UED(m,k)$ -код $\forall m$	42,875%	20,769%	42,857%	50,272%
				–	–	–	–
$RS4(m,k)$	4	3	4- $UED(m,k)$ -код $\forall m$	46,667%	54,286%	49,65%	100%
				–	–	–	–
$TM(m,k)$	4	4	$UED(m,k)$ -код при $m=6,$ 4- $UED(m,k)$ -код $\forall m$	42,875%	36,508%	25,714%	27,646%
				–	–	–	–
$WS(m,k,w)$	2^{k*}	k^*	$UED(m,k)$ -код $\forall m$	46,667%	54,286%	49,65%	100%
				–	–	–	–

Примечание. $k^* = \lceil \log_2(m+1) \rceil$.

7. Алгоритм синтеза самопроверяемого комбинационного устройства. Аккумулируя приведенные выше сведения, приведем алгоритм синтеза системы встроенного контроля для логических устройств с учетом особенностей их топологии и свойств кодов с суммированием.

Алгоритм 2. Синтез самопроверяемого комбинационного устройства:

1. На основании анализа топологии логического устройства определяются ССЗ-группы выходов.
2. Путем функционального анализа ССЗ-групп выходов выделяются ФСЗ-группы выходов.
3. Выбирается код с суммированием, свойства которого позволяют решать задачу обнаружения максимального числа симметричных

ошибок с минимальным резервированием элементов в структуре логического устройства при минимальной избыточности кода.

4. Выполняется процедура преобразования структуры исходного логического устройства в устройство с контролепригодной по выбранному коду структурой с учетом уменьшения множества W немонотонных элементов (см. алгоритм 1).

Оценим сложность алгоритма 2.

Сам алгоритм включает в себя четыре подчиненных алгоритма. Первый подалгоритм подразумевает анализ связей внутренних логических элементов исходного логического устройства с его выходами. Сложность данного подалгоритма напрямую связана с числом внутренних логических элементов без элементов первого каскада (N_G) и числом выходов самого устройства (m). Количество операций первого подалгоритма определяется произведением mN_G . После выполнения операций первого подалгоритма в качестве выходных данных будет перечень ССЗ-групп выходов и логических элементов, наличие связей с которыми обуславливает такую зависимость. Максимальное количество вызывающих структурно симметричную зависимость определяется величиной N_G (все логические элементы), а количество выходов в каждой ССЗ-группе будет находиться в диапазоне от 2 до m .

Дальнейший функциональный анализ во втором подалгоритме будет связан с проверкой условия (6), где требуется вычислять булевы производные функций выходов схемы по функциям, реализуемым на выходах рассматриваемых логических элементов. Для каждого логического

элемента потребуется выполнить максимум $\sum_{d=2}^{d_{\max}} C_m^d$ (d — четное) проце-

дур вычисления. Общее количество операций — максимум $N_G \sum_{d=2}^{d_{\max}} C_m^d$.

Третий подалгоритм осуществляет процедуру выбора кода с учетом обнаружения симметричных ошибок на выходах. Требуется сформировать список входных комбинаций, на которых вызывается симметричные ошибки выходов устройства, а затем осуществить поиск кода с учетом наилучших характеристик обнаружения симметричных ошибок (аналог таблицы 2). Максимальное количество строк такого списка определяется числом входных комбинаций, на которых вызывается симметричная ошибка для каждого логического элемента, формирующего ФСЗ-группу выходов. Число входных комбинаций может быть различным для всех логических элементов. Максимум процедур перебора для

каждого логического элемента — 2^t (t — число входов устройства). Таким образом, для каждого кода $2^t N_G$ операций вычислений.

Четвертый подалгоритм связан с выполнением процедур поиска немонотонных элементов и их резервированием по алгоритму 1. Здесь основная процедура — для каждого логического элемента проверка путей, ведущих к выходам и соединение выходов. Максимальное число проверок mN_G .

Сложность конечного алгоритма определяется функцией нескольких переменных:

$$g = 2mN_G + N_G \sum_{d=2}^{d_{\max}} C_m^d + 2^t N_G, \quad (7)$$

где t — число входов устройства; m — число выходов устройства; N_G — число логических элементов.

Асимптотическая оценка трудоемкости алгоритма определяется по выражению:

$$Q = O \left(2mN_G + N_G \sum_{d=2}^{d_{\max}} C_m^d + 2^t N_G \right).$$

От числа логических элементов в структуре логического устройства сложность реализации алгоритма зависит линейно; от числа входов логического устройства зависимость является степенной (экспоненциальной) и от количества выходов устройства — факториальной. Другим словами, алгоритм наиболее эффективен для комбинационных устройств с небольшим (до 20-30) числом входов и выходов. При большем их числе потребуются декомпозиция устройства и отдельный контроль подсхем.

Вопрос выбора кода для контроля логического устройства может решаться двумя путями: последовательным перебором кодов из имеющегося множества, либо обоснованным анализом возможностей контроля в разных разрядах контрольного вектора значений рабочих функций объекта диагностирования в ФСЗ-группах.

8. Заключение. Сформулированные в представленной статье условия поиска структурно и функционально зависимых выходов позволяют на практике устанавливать множество тех логических элементов структур комбинационных логических устройств, неисправности которых будут вызывать симметричные ошибки на выхо-

дах самого объекта диагностирования. При этом установлена наиболее удобная последовательность анализа топологии объекта диагностирования (сокращается число вычислительных процедур), заключающаяся в том, что сначала находятся структурно симметрично зависимые группы выходов, а затем среди найденных групп с использованием приведенных в статье условий находятся группы функционально зависимых выходов. Финальным этапом (что и является принципиально новым по отношению к известным исследованиям), непосредственно определяющим множество резервируемых элементов, следует проверка для групп функционально зависимых выходов и конкретных элементов, вызывающих симметричные ошибки на выходах устройства, обнаруживается или нет конкретная ошибка на конкретном входном наборе выбранным (m, k) -кодом. За счет особенностей обнаружения ошибок в информационных векторах кодов удастся уменьшить число резервируемых элементов по отношению к тому, как это сделано для классических кодов Бергера в известном алгоритме [36, 37].

Представленный подход к синтезу самопроверяемых комбинационных устройств универсален и не ориентирован на конкретную элементную базу, он затрагивает только функциональное описание логических устройств и может быть легко адаптирован на использование любой элементной базы.

Литература

1. Сапожников В.В., Сапожников В.В., Христов Х.А., Гавзов Д.В. Методы построения безопасных микроэлектронных систем железнодорожной автоматики / под ред. Вл.В. Сапожникова // М.: Транспорт. 1995. 272 с.
2. Ubar R., Raik J., Vierhaus H.-T. Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source) // New York: IGI Global. 2011. 578 p.
3. Drozd A. et al. The use of natural resources for increasing a checkability of the digital components in safety-critical systems // Proceedings of 11th IEEE East-West Design & Test Symposium (EWDTS'2013). 2013. pp. 1–6.
4. Kharchenko V., Kondratenko Yu., Kacprzyk J. Green IT Engineering: Concepts, Models, Complex Systems Architectures // Springer Book series "Studies in Systems, Decision and Control". 2017. vol. 74. 305 p.
5. Sklyar V., Kharchenko V., Bardis N.G. Assurance case for green IT applications: proof of compliance with power consumption claims // Proceedings of 4th International Conference on Mathematics and Computers in Sciences and in Industry (MCSI). 2017. pp. 124–127.
6. Аксёнова Г.П. Локализация кратных неисправностей при групповом контроле в дискретном устройстве // Автоматика и телемеханика. 2017. № 12. С. 118–130.
7. Drozd O., Nikul V., Antoniuk V., Drozd M. Hidden faults in FPGA-built digital components of safety-related systems // Proceedings of 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET). 2018 pp. 805–809.

8. *Shah T., Matrosova A., Singh V.* Test pattern generation to detect multiple faults in ROBDD based combinational circuits // Proceedings of IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS). 2017. pp. 211–2012.
9. *Mosin S.* Automated simulation of faults in analog circuits based on parallel paradigm // Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017). 2017. pp. 73–78.
10. *Аксёнова Г.П.* Матричный способ локализации неисправностей в ПЛИС // Автоматика и телемеханика. 2013. № 9. С. 119–124.
11. *Dautov R., Mosin S.* A Technique to aggregate classes of analog fault diagnostic data based on association rule mining // Proceedings of 19th International Symposium on Quality Electronic Design (ISQED). 2018. pp. 238–243.
12. *Бибило П.Н. и др.* Автоматизация логического синтеза КМОП схем с пониженным энергопотреблением // Программная инженерия. 2013. № 8. С. 35–41.
13. *Черемисинова Л.Д.* Логический синтез комбинационных КМОП схем с учетом рассеивания мощности // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2014. № 3. С. 89–98.
14. *Поттосин Ю.В.* Энергосберегающее противогоночное кодирование состояний асинхронного автомата // Прикладная дискретная математика. Приложение. 2015. № 8. С. 120–123.
15. *Степченко Ю.А., Каменских А.Н., Тюрин С.Ф., Рождественский Ю.В.* Модели отказоустойчивых самосинхронных схем // Системы и средства информатики. 2016. Том 26. № 4. С. 19–30.
16. *Согомонян Е.С., Слабаков Е.В.* Самопроверяемые устройства и отказоустойчивые системы // М.: Радио и связь. 1989. 208 с.
17. *Nicolaidis M., Zorian Y.* On-Line Testing for VLSI – A Compendium of Approaches // Journal of Electronic Testing: Theory and Applications. 1998. vol. 12. pp. 7–20.
18. *Матросова А.Ю., Останин С.А., Паришина Н.А.* К синтезу контролепригодных комбинационных устройств // Автоматика и телемеханика. 1999. № 2. С. 129–137.
19. *Nicolaidis M.* On-Line Testing for VLSI: State of the Art and Trends // Integration, the VLSI Journal. 1998. vol. 26. Issue 1-2. pp. 197–209.
20. *Matrosova A.Yu., Levin I., Ostanin S.A.* Self-Checking Synchronous FSM Network Design with Low Overhead // VLSI Design. 2000. vol. 11. Issue 1. pp. 47–58.
21. *Mitra S., McCluskey E.J.* Which concurrent error detection scheme to choose? // Proceedings of International Test Conference. 2000. pp. 985–994.
22. *Kubalik P., Kubátová H.* Parity Codes Used for On-Line Testing in FPGA // Acta Polytechnica. 2005. vol. 45. no. 6. pp. 53–59.
23. *Butorina N.* Self-testing checker design for incomplete m-out-of-n codes // Proceedings of 12th IEEE East-West Design & Test Symposium (EWDTS'2014). 2014. pp. 258–261.
24. *Сапожников В.В., Сапожников В.В.* Самопроверяемые дискретные устройства // СПб: Энергоатомиздат. 1992. 224 с.
25. *Piestrak S.J.* Design of Self-Testing Checkers for Unidirectional Error Detecting Codes // Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej. 1995. 111 p.
26. *Das D., Toubá N.A.* Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes // Journal of Electronic Testing: Theory and Applications. 1999. vol. 15. Issue 1-2. pp. 145–155.
27. *Fujiwara E.* Code Design for Dependable Systems: Theory and Practical Applications // New Jersey: John Wiley & Sons. 2006. 720 p.
28. *Borecký J., Kohlík M., Kubátová H.* Parity Driven Reconfigurable Duplex System // Microprocessors and Microsystems. 2017. vol. 52. pp. 251–260.

29. *Бибило П.Н., Романов В.И.* Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний // Минск: Беларус. навука. 2011. 279 с.
30. *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. vol. 4. Issue 4. pp. 267–281.
31. *Гессель М., Морозов А.А., Сапожников В.В., Сапожников Вл.В.* Исследование комбинационных самопроверяемых устройств с независимыми и монотонно независимыми выходами // Автоматика и телемеханика. 1997. № 2. С. 180–193.
32. *Gössel M., Ocheretny V., Sogomonyan E., Marienfeld D.* New Methods of Concurrent Checking: Edition 1 // Springer Netherlands. 2008. 184 p.
33. *Berger J.M.* A Note on Error Detecting Codes for Asymmetric Channels // Information and Control. 1961. vol. 4. Issue 1. pp. 68–73.
34. *Freiman C.V.* Optimal Error Detection Codes for Completely Asymmetric Binary Channels // Information and Control. 1962. vol. 5. Issue 1. pp. 64–71.
35. *Busaba F.Y., Lala P.K.* Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors // Journal of Electronic Testing: Theory and Applications, 1994. vol. 5. Issue 1. pp. 19–28.
36. *Saposhnikov V.V., Morosov A., Sapozhnikov Vl.V., Gössel M.* A New Design Method for Self-Checking Unidirectional Combinational Circuits // Journal of Electronic Testing: Theory and Applications. 1998. vol. 12. Issue 1-2. pp. 41–53.
37. *Morosov A., Sapozhnikov V.V., Sapozhnikov Vl.V., Goessel M.* Self-Checking Combinational Circuits with Unidirectionally Independent Outputs // VLSI Design. 1998. vol. 5. Issue 4. pp. 333–345.
38. *Матросова А.Ю., Останин С.А., Синех В.* Обнаружение несущественных путей логических схем на основе совместного анализа И-ИЛИ деревьев и SSBDD-графов // Автоматика и телемеханика. 2013. № 7. С. 126–142.
39. *Ostanin S.* Self-checking synchronous FSM network design for path delay faults // Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017). 2017. pp. 696–699.
40. *Matrosova A., Andreeva V., Tomkov V.* Fully delay and multiple stuck-at faults testable FSM design // Proceedings of 13th IEEE East-West Design & Test Symposium (EWDTS'2015). 2015. pp. 212–215.
41. *Сапожников В.В., Сапожников Вл.В., Ефанов Д.В.* Классификация ошибок в информационных векторах систематических кодов // Известия вузов. Приборостроение. 2015. Том 58. № 5. С. 333–343.
42. *Efanov D., Sapozhnikov V., Sapozhnikov Vl.* Generalized algorithm of building summation codes for the tasks of technical diagnostics of discrete systems // Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017). 2017. pp. 365–371.
43. *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.
44. *Das D., Toubia N.A.* Weight-based codes and their application to concurrent error detection of multilevel circuits // Proceedings of 17th IEEE Test Symposium. 1999. pp. 370–376.
45. *Das D., Toubia N.A., Seuring M., Goessel M.* Low cost concurrent error detection based on modulo weight-based codes // Proceedings of IEEE 6th International On-Line Testing Workshop (IOLTW). 2000. pp. 171–176.

46. *Сапожников В.В., Сапожников Вл.В., Ефанов Д.В.* Коды с суммированием, обнаруживающие любые симметричные ошибки // *Электронное моделирование*. 2017. Том 39. № 3. С. 47–60.
47. *Efanov D.V., Sapozhnikov V.V., Sapozhnikov Vl.V.* Two-Modulus Codes with Summation of One-Data Bits for Technical Diagnostics of Discrete Systems // *Automatic Control and Computer Sciences*. 2018. vol. 52. Issue 1. pp. 1–12.
48. *Efanov D., Sapozhnikov V., Sapozhnikov Vl.* Generic two-modulus sum codes for technical diagnostics of discrete systems problems // *Proceedings of 14th IEEE East-West Design & Test Symposium (EWDTS'2016)*. 2016. pp. 256–260.
49. *Блодов А.А., Ефанов Д.В., Сапожников В.В., Сапожников Вл.В.* О кодах с суммированием единичных разрядов в системах функционального контроля // *Автоматика и телемеханика*. 2014. № 8. С. 131–145.
50. *Сапожников В.В., Сапожников Вл.В., Ефанов Д.В.* Взвешенные коды с суммированием для организации контроля логических устройств // *Электронное моделирование*. 2014. Том 36. № 1. С. 59–80.

Ефанов Дмитрий Викторович — д-р техн. наук, доцент, профессор кафедры автоматики, телемеханики и связи на железнодорожном транспорте, Российский университет транспорта (МИИТ), руководитель направления систем мониторинга и диагностики, ООО «ЛокоТех-Сигнал». Область научных интересов: дискретная математика, надежность и техническая диагностика дискретных систем. Число научных публикаций — 300. TrES-4b@yandex.ru; ул. Образцова, 9, Москва, 127994; р.т.: +7(911)709-2164.

D.V. EFANOV
**THE SYNTHESIS OF SELF-CHECKING COMBINATIONAL
DEVICES ON THE BASIS OF CODES WITH THE EFFECTIVE
SYMMETRICAL ERROR DETECTION**

Efanov D.V. The Synthesis of Self-Checking Combinational Devices on the Basis of Codes with the Effective Symmetrical Error Detection.

Abstract. The methods of fault-tolerant coding are often used in the designing of reliable and safety components of automatic control systems: both in the data transmission between system nodes, and at the level of hardware and software architecture.

The redundant coding is widely used in the management of combinational logic devices control. In this case, codes, which are oriented to the error detection rather than correction of this, are in use. Such features of codes make it possible to implement the checkable automation systems with acceptable redundancy, which does not exceed the redundancy in the situation of duplication using.

The paper highlights the method of the synthesis of self-checking combinational devices, which makes it possible to take into account the features of the source devices architecture, as well as the properties of error detection by redundant codes in solving the problem of the synthesis of technical means for diagnosis. The paper gives the basic information on the theory of the checkable digital systems synthesis on the basis of redundant codes with summation.

The basic stages of the analysis of the diagnosis objects topologies are determined with the selection of groups of outputs — groups of structurally and functionally symmetrically independent devices outputs. The formulas are given to determine the presence or the absence of a symmetrical dependence of the diagnosis object outputs. The example illustrating the calculation process is given. The main stages of the analysis of the redundant codes application in the error detection on the functionally symmetric dependent outputs are formulated. The algorithm of the synthesis of the self-checking combinational devices with taking into account the object of diagnosis structure features and the redundant codes properties is proposed.

Keywords: logic devices in automation, checkable structure, technical diagnostics, diagnosis, technical condition monitoring, uniform block code, Berger code, sum codes, error detection.

Efanov Dmitry Viktorovich — Ph.D., Dr. Sci., associate professor, professor of automation, remote control and communication on railway transport department, Russian University of Transport, head of monitoring and diagnostic systems direction, «LocoTech-Signal» LCC. Research interests: discrete mathematics, reliability and technical diagnostics of discrete devices. The number of publications — 300. TrES-4b@yandex.ru; 9, Obraztsova str., Moscow, 127994, Russia; office phone: +7(911)709-2164.

References

1. Sapozhnikov V.V., Sapozhnikov V.I., Hristov H.A., Gavzov D.V. *Metody postroeniya bezopasnyh mikroelektronnyh sistem zheleznodorozhnoj avtomatiki. Pod red. V.I.V. Sapozhnikova* [Methods for constructing safe microelectronic systems for railway automation. Edited by V.I.V. Sapozhnikov]. M.: Transport. 1995. 272 p. (In Russ.).
2. Ubar R., Raik J., Vierhaus H.-T. Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source). New York: IGI Global. 2011. 578 p.
3. Drozd A., Kharchenko V., Antoshchuk S., Drozd J., Lobachev M., Sulima J. The use of natural resources for increasing a checkability of the digital components in safety-

- critical systems. Proceedings of 11th IEEE East-West Design & Test Symposium (EWDTS'2013). 2013. pp. 1–6.
4. Kharchenko V., Kondratenko Yu., Kacprzyk J. Green IT Engineering: Concepts, Models, Complex Systems Architectures. Springer Book series "Studies in Systems, Decision and Control". 2017. vol. 74. 305 p.
 5. Sklyar V., Kharchenko V., Bardis N.G. Assurance case for green IT applications: proof of compliance with power consumption claims. Proceedings of 4th International Conference on Mathematics and Computers in Sciences and in Industry (MCSI). 2017. pp. 124–127.
 6. Aksyonova G.P. [Localization of multiple faults with group control on a discrete device]. *Avtomatika i telemekhanika – Automation and remote control*. 2017. vol. 12. pp. 118–130. (In Russ.).
 7. Drozd O., Nikul V., Antoniuk V., Drozd M. Hidden faults in FPGA-built digital components of safety-related systems. Proceedings of 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET). 2018 pp. 805–809.
 8. Shah T., Matrosova A., Singh V. Test pattern generation to detect multiple faults in ROBDD based combinational circuits. Proceedings of IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS). 2017. pp. 211–2012.
 9. Mosin S. Automated simulation of faults in analog circuits based on parallel paradigm. Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017). 2017. pp. 73–78.
 10. Aksyonova G.P. [A matrix method for PLD failure localization]. *Avtomatika i telemekhanika – Automation and remote control*. 2013. vol. 9. pp. 119–124. (In Russ.).
 11. Dautov R., Mosin S. A Technique to aggregate classes of analog fault diagnostic data based on association rule mining. Proceedings of 19th International Symposium on Quality Electronic Design (ISQED). 2018. pp. 238–243.
 12. Bibilo P.N. et al. [Low-power logical synthesis of CMOS circuits automation]. *Programmnaya inzheneriya – Software Engineering*. 2013. vol. 8. pp. 35–41. (In Russ.).
 13. Cheremisinova L.D. [Lower-power logic synthesis of combinational CMOS circuits]. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk state university journal of control and computer science*. 2014. vol. 3. pp. 89–98. (In Russ.).
 14. Pottosin Yu.V. [Low power race-free state assignment of an asynchronous automaton]. *Prikladnaya diskretnaya matematika. Prilozhenie – Applied Discrete Mathematics. Supplement*. 2015. vol. 8. pp. 120–123. (In Russ.).
 15. Stephenkov Yu.A., Kamenskih A.N., Tyurin S.F., Rozhdvestvenskij Yu.V. [Models of fault-tolerant self-timed circuits]. *Sistemy i sredstva informatiki – Systems and means of informatics*. 2016. Issue 26. vol. 4. pp. 19–30. (In Russ.).
 16. Sogomonyan E.S., Slabakov E.V. *Samoproveryaemye ustroystva i otkazoustoychivyye sistemy* [Self-Checking and Fail-Safety Systems]. M.: Radio and Communication. 1989. 207 p. (In Russ.).
 17. Nicolaidis M., Zorian Y. On-Line Testing for VLSI – A Compendium of Approaches. *Journal of Electronic Testing: Theory and Applications*. 1998. vol. 12. pp. 7–20.
 18. Matrosova A.Yu., Ostanin S.A., Parshina N.A. [Synthesizing testable combinational circuits]. *Avtomatika i telemekhanika – Automation and remote control*. 1999. vol. 2. pp. 129–137. (In Russ.).
 19. Nicolaidis M. On-Line Testing for VLSI: State of the Art and Trends. *Integration, the VLSI Journal*. 1998. vol. 26. Issues 1-2. pp. 197–209.

20. Matrosova A.Yu., Levin I., Ostanin S.A. Self-Checking Synchronous FSM Network Design with Low Overhead. *VLSI Design*. 2000. vol. 11. Issue 1. pp. 47–58.
21. Mitra S., McCluskey E.J. Which concurrent error detection scheme to choose? Proceedings of International Test Conference. 2000. pp. 985–994.
22. Kubalík P., Kubátová H. Parity Codes Used for On-Line Testing in FPGA. *Acta Polytechnica*. 2005. vol. 45. no. 6. pp. 53–59.
23. Butorina N. Self-testing checker design for incomplete m-out-of-n codes. Proceedings of 12th IEEE East-West Design & Test Symposium (EWDTS'2014). 2014. pp. 258–261.
24. Sapozhnikov V.V., Sapozhnikov V.I.V. *Samoproveryaemye diskretnye ustrojstva* [Self-checking discrete devices]. St. Petersburg: Energoatomizdat. 1992. 224 p. (In Russ.).
25. Piestrak S.J. Design of self-testing checkers for unidirectional error detecting codes. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
26. Das D., Toubá N.A. Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes. *Journal of Electronic Testing: Theory and Applications*. 1999. vol. 15. Issue 1-2. pp. 145–155.
27. Fujiwara E. Code Design for Dependable Systems: Theory and Practical Applications. New Jersey: John Wiley & Sons. 2006. 720 p.
28. Borecký J., Kohlík M., Kubátová H. Parity Driven Reconfigurable Duplex System. *Microprocessors and Microsystems*. 2017. vol. 52. pp. 251–260.
29. Bibilo P.N., Romanov V.I. *Logicheskoe proektirovanie diskretnyh ustrojstv s ispol'zovaniem produkcionno-frejmovej modeli predstavleniya znanij* [Logical design of discrete devices using the product-frame model of knowledge representation]. Minsk: Belarus. navuka. 2011. 279 p. (In Russ.).
30. 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. vol. 4. Issue 4. pp. 267–281.
31. Gessel' M., Morozov A.A., Sapozhnikov V.V., Sapozhnikov V.I.V. [Investigation of combination self-testing devices having independent and monotone independent outputs]. *Avtomatika i telemekhanika – Automation and remote control*. 1997. vol. 2. pp. 180–193. (In Russ.).
32. Gössel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1. Springer Netherlands. 2008. 184 p.
33. Berger J.M. A Note on Error Detecting Codes for Asymmetric Channels. *Information and Control*. 1961. vol. 4. Issue 1. pp. 68–73.
34. Freiman C.V. Optimal Error Detection Codes for Completely Asymmetric Binary Channels. *Information and Control*. 1962. vol. 5. Issue 1. pp. 64–71.
35. Busaba F.Y., Lala P.K. Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors. *Journal of Electronic Testing: Theory and Applications*. 1994. vol. 5. Issue 1. pp. 19–28.
36. Saposhnikov V.V., Morosov A., Saposhnikov V.I.V., Gössel M. A New Design Method for Self-Checking Unidirectional Combinational Circuits. *Journal of Electronic Testing: Theory and Applications*. 1998. vol. 12. Issue 1-2. pp. 41–53.
37. Morosow A., Sapozhnikov V.V., Sapozhnikov V.I.V., Goessel M. Self-Checking Combinational Circuits with Unidirectionally Independent Outputs. *VLSI Design*. 1998. vol. 5. Issue 4. pp. 333–345.
38. Matrosova A.Yu., Ostanin S.A., Singh V. [Detection of false paths in logical circuits by joint analysis of the AND/OR trees and SSBDD-graphs]. *Avtomatika i telemekhanika – Automation and remote control*. 2013. vol. 7. pp. 126–142. (In Russ.).

39. Ostanin S. Self-checking synchronous FSM network design for path delay faults. Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017). 2017. pp. 696–699.
40. Matrosova A., Andreeva V., Tomkov V. Fully delay and multiple stuck-at faults testable FSM design. Proceedings of 13th IEEE East-West Design & Test Symposium (EWDTS'2015). 2015. pp. 212–215.
41. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V. [Errors classification in information vectors of systematic codes]. *Izvestiya vysshih uchebnyh zavedenij. Priborostroenie – Journal of Instrument Engineering*. 2015. Issue 58. vol. 5. pp. 333–343. (In Russ.).
42. Efanov D., Sapozhnikov V., Sapozhnikov VI. Generalized algorithm of building summation codes for the tasks of technical diagnostics of discrete systems. Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017). 2017. pp. 365–371.
43. Efanov D.V., Sapozhnikov V.V., Sapozhnikov VI.V. [Conditions for Detecting a Logical Element Fault in a Combination Device under Concurrent Checking Based on Berger's Code]. *Avtomatika i telemekhanika – Automation and remote control*. 2017. vol. 5. pp. 152–165. (In Russ.).
44. Das D., Touba N.A. Weight-based codes and their application to concurrent error detection of multilevel circuits. Proceedings of 17th IEEE Test Symposium. 1999. pp. 370–376.
45. Das D., Touba N.A., Seuring M., Goessel M. Low cost concurrent error detection based on modulo weight-based codes. Proceedings of IEEE 6th International On-Line Testing Workshop (IOLTW). 2000. pp. 171–176.
46. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V. [On the class of codes with summation with all symmetric errors detection]. *Elektronnoje Modelirovanije – Electronic modeling*. 2017. Issue 39. vol. 3. pp. 47–60. (In Russ.).
47. Efanov D.V., Sapozhnikov V.V., Sapozhnikov VI.V. Two-Modulus Codes with Summation of One-Data Bits for Technical Diagnostics of Discrete Systems. *Automatic Control and Computer Sciences*. 2018. vol. 52. Issue 1. pp. 1–12.
48. Efanov D., Sapozhnikov V., Sapozhnikov VI. Generic two-modulus sum codes for technical diagnostics of discrete systems problems. Proceedings of 14th IEEE East-West Design & Test Symposium (EWDTS'2016). 2016. pp. 256–260.
49. Blyudov A.A., Efanov D.V., Sapozhnikov V.V., Sapozhnikov VI.V. [On Codes with Summation of Unit Bits in Concurrent Error Detection Systems]. *Avtomatika i telemekhanika – Automation and remote control*. 2014. vol. 8. pp. 131–145. (In Russ.).
50. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V. [Weight-based sum codes for logical devices checking organization]. *Elektronnoje Modelirovanije – Electronic modeling*. 2014. Issue 36. vol. 1. pp. 59–80. (In Russ.).