

ИЗБРАННЫЕ МАТЕРИАЛЫ КОНФЕРЕНЦИИ “ПРОБЛЕМЫ
== РАЗРАБОТКИ ПЕРСПЕКТИВНЫХ МИКРО- И НАНОЭЛЕКТРОННЫХ СИСТЕМ” ==
ПРОЕКТИРОВАНИЕ ИНТЕГРАЛЬНЫХ СХЕМ

УДК 621.3.049.771.14

СИНТЕЗ СХЕМЫ ФУНКЦИОНАЛЬНОГО КОНТРОЛЯ
НА ОСНОВЕ СПЕКТРАЛЬНОГО R-КОДА
С РАЗБИЕНИЕМ ВЫХОДОВ НА ГРУППЫ

© 2019 г. А. Л. Стемпковский¹, Д. В. Тельпухов¹, Т. Д. Жукова¹,
А. И. Деменова¹, В. В. Надоленко¹, С. И. Гуров^{2, *}

¹Институт проблем проектирования в микроэлектронике Российской АН (ИППМ РАН)
Россия, 124365, г. Москва, г. Зеленоград, ул. Советская, д. 3

²Московский государственный университет имени М.В. Ломоносова (МГУ)
Россия, 119991, г. Москва, Ленинские горы, д. 1

*E-mail: sgur@cs.msu.ru

Поступила в редакцию 28.01.2019 г.

После доработки 29.01.2019 г.

Принята к публикации 29.01.2019 г.

Темпы развития микроэлектронной промышленности, а также прогресс в области разработки средств автоматизации проектирования актуализируют вопросы, связанные с разработкой комбинационных схем устойчивых к кратковременным самоустраняемым отказам. Данные отказы возникают вследствие совокупности множества различных факторов, таких как экстремальные условия эксплуатации, переход к нанометровым нормам проектирования и т.д. Зачастую для решения данной проблемы используются методы внедрения структурной избыточности, за счет применения принципов теории помехоустойчивого кодирования для защиты информации при ее передаче по каналам связи. Однако данные методы обладают существенным недостатком – большой структурной избыточностью. В данной работе для решения проблемы разработки сбоеустойчивых комбинационных схем предлагается использовать подход на основе синтеза схем функционального контроля с использованием спектрального R-кода. При защите определенной части кодера специальными технологическими средствами данный код способен исправлять однократную ошибку, а также обнаруживать двукратную ошибку. Полученная в результате схема при сравнении с традиционным методом тройного модульного резервирования обладает меньшей структурной избыточностью. Для сведения к минимуму вероятности возникновения многократных ошибок внутри комбинационной схемы также предлагается подход на основе разбиения выходов схемы на группы с последующим синтезом схем функционального контроля на основе R-кода, что приводит к увеличению вероятности обнаружения/исправления ошибок. В статье представлены результаты серии численных экспериментов, демонстрирующих высокую эффективность предлагаемых подходов.

Ключевые слова: сбоеустойчивость комбинационных схем, информационная избыточность, избыточное кодирование, R-код, кластеризация выходов схемы

DOI: 10.1134/S0544126919040094

1. ВВЕДЕНИЕ

Прогресс в области микроэлектронной промышленности помимо увеличения степени интеграции и улучшения параметров интегральных схем (ИС) приводит также к возникновению серьезных проблем, ведущих к уменьшению надежности функционирования схем [1, 2]. В свою очередь для некоторых практических приложений это может привести к существенному ослаблению преимуществ, полученных от увеличения степени интеграции. Более того, такие факторы, как увеличение тактовых частот и уменьшение напряжений питания также повышают уязвимость комбинаци-

онных участков ИС, работающих в условиях дестабилизирующих воздействий [3].

Это приводит к тому, что при проектировании комбинационных схем одной из основных задач является задача обеспечения устойчивости схемы к кратковременным обратимым нарушениям работы ИС – сбоям. Данные временные нарушения возникают без какого-либо повреждения схемы, и работоспособность схемы восстанавливается в течение короткого промежутка времени.

Необходимо отметить, что все основные подходы по борьбе со случайными сбоями основываются на различных механизмах маскирования, таких как [3, 4]:

- *логическое маскирование* (Logical Masking), позволяющее реализовывать такие схемы, в которых ошибки на вентилях не приводят к появлению ошибок на результирующем выходе схемы благодаря маскирующим свойствам самих вентилях;
- *электрическое маскирование* (Electrical Masking), обеспечивающее защиту путем ослабления электрического импульса с помощью прохождения через различные электронные компоненты;
- *временное маскирование* (Latching-Window Masking), основанное на том факте, что длительность дестабилизирующих эффектов обычно очень мала.

Существует множество разнообразных методов борьбы с последствиями сбоев, которые не зависят от причины и природы их возникновения. Одними из них являются традиционные подходы, связанные с использованием методов кратного резервирования [5, 6]. Данные методы реализуются с помощью N -кратного копирования основной схемы с последующим объединением полученных копий схем и проверки правильности функционирования результирующего выхода с помощью схемы голосования. Обычно используют метод тройного модульного резервирования (TMR, Triple Modular Redundancy), позволяющий в случае возникновения не более чем одиночной ошибки в одном из экземпляров резервируемого модуля и безошибочной работы схемы мажорирования исправить эту ошибку. Недостатком данного метода является большая структурная избыточность.

В настоящее время одним из основных методов повышения сбоеустойчивости комбинационных схем является подход, связанный с генерацией средств контроля. Как известно, эти средства делятся на следующие группы в зависимости от вида контроля [7]:

- *тестовый контроль* (Offline Testing) – проводится вне штатной работы устройства;
- *функциональный контроль* (Online Testing) – проводится непрерывно в рабочем режиме.

Реализация схем функционального контроля (СФК) на основе комбинационных схем производится с помощью добавления дополнительной подсхемы, способной сигнализировать о наличии ошибки, а также, в некоторых случаях, произвести ее исправление. В случае СФК, способной обнаружить и исправить ошибку, эффективность применения схем функционального контроля оценивается как отношение структурной избыточности

схемы к вероятности обнаружения и/или исправления ошибок.

Стоит также отметить, что на практике редко применяются схемы, способные исправлять ошибки. Происходит это по причине того, что помимо большой структурной избыточности, существует высокий риск возникновения сбоя в участках схемы, отвечающих именно за исправление.

Развитием данного направления служит предлагаемый в данной статье подход – синтез схемы функционального контроля на основе спектрального R -кода, способного исправлять однократные ошибки на выходе комбинационной схемы [8]. Для уменьшения вероятности возникновения двукратных ошибок предлагается использовать алгоритм кластеризации выходов комбинационной схемы. Данный подход по разбиению выходов по группам с последующим синтезом схемы функционального контроля для каждой из полученных групп позволяет повысить вероятность обнаружения/исправления ошибок кодами SEC/DED (Single Error Correction/Double Error Detection) [9].

2. ТЕОРЕТИЧЕСКИЙ АНАЛИЗ

На практике, для повышения сбоеустойчивости используют линейные коды, исправляющие однократную и обнаруживающие двукратную ошибки. Это связано с тем, что использование нелинейных кодов связано с трудоемкими исследованиями свойств конкретного синтезируемого устройства, а применение циклических кодов нецелесообразно, поскольку построение декодеров в комбинационном исполнении требует слишком больших аппаратных затрат.

При систематическом блоковом кодировании кодовые слова (n, k) -кода имеют длину $n = k + m$, где k – число *информационных разрядов*, повторяющих кодируемое слово, а m – число *проверочных разрядов*, которые являются суммой по модулю 2 различных подмножеств значений информационных разрядов. Минимальное расстояние кода d – минимум по числу несовпадающих координат среди всех пар кодовых слов.

Формально систематическим кодом с проверкой на четность является двоичный блоковый код, в котором сообщению $u = (u_1, \dots, u_k)^T$ сопоставлено кодовое слово $v = (v_1, \dots, v_n)^T, k \leq n$, определяемое соотношениями

$$v_i = \begin{cases} u_i, & 1 \leq i \leq k, \\ \sum_{j=1}^k u_j g_{i-k-1,j}, & k+1 \leq i \leq n, \end{cases}$$

где множество двоичных символов $\{g_{ij}\}$, $0 \leq i \leq m-1$, $1 \leq j \leq k$ фиксировано [7]. Таким образом предполагается размещение символов сообщения в первых позициях кодового слова. Кодирование в данном коде осуществляется умножением вектора сообщения на определяемую вышеприведенными соотношениями порождающую матрицу G : $v = Gu$. Искажение в результате сбоя описывается вектором $w = v + e$, где e – вектор ошибок (все векторы понимаются как вектор-столбцы).

Существует множество методов декодирования линейных кодов [10, 12], но наиболее удобными для решения проблемы сбоеустойчивости комбинационных схем являются синдромное и мажоритарное декодирование.

Синдромное декодирование основано на вычислении вектора s синдромов по проверочной матрице H : $s = Hw$, где w – выходной вектор, искаженный сбоями. Если ошибок нет, то $s \equiv 0$, иначе по заранее определенным наиболее вероятным позициям ошибки $e(s)$, соответствующей данному синдрому, вычисляется вектор $v = w + e(s)$, по которому производится восстановление сообщения u . Декодер при данном методе декодирования состоит из схемы обнаружения ошибки, включая подсхему вычисления синдрома, и схемы определения вектора ошибки с последующим ее исправлением.

Мажоритарное декодирование базируется на системе проверочных равенств для каждого информационного бита, в качестве которых используются линейные комбинации строк проверочной матрицы H . Данное декодирование допускает большое число кодов. При этом декодирующие устройства, построенное по мажоритарному принципу, часто оказываются наиболее простыми.

3. СПЕКТРАЛЬНЫЙ R-КОД НА ОСНОВЕ ФУНКЦИИ РАДЕМАХЕРА

Кратко опишем предлагаемый новый систематический код с проверкой на четность [13, 14].

Положим $g_{0,j} \equiv 1$, а значения $g_{i,j}$ для $i = 1, q$, $q = \lceil \log k \rceil$ (где $\lceil x \rceil$ – обозначает округление x до целого в большую сторону), $j = 1, k$ такими, чтобы вектор $(g_{1,j}, \dots, g_{q,j})$ был двоичным кодом числа $(j-1)$. Таким образом получены $m = q + 1$ проверочных разрядов и образован блоковый систематический код с проверкой на четность. Порождающей матрицей данного кода является единичная матрица k -го порядка с присоединенной транспонированной порождающей матрицей $(k, 1)$ кода Рида–Маллера первого порядка длины 2^k .

Значения $g_{i,j}$ можно получить линейным преобразованием аргумента и значений дискретных функций Радемахера. Как известно, система функций Радемахера ортогональна, ортонорми-

рована, но не полна и используется для построения функций Уолша, по которым, как по базису, осуществляют спектральное разложение кусочно-постоянных булевых функций [15]. В силу этого описанный код назван R спектральным.

Построенный код способен исправлять единичную ошибку инвертирования некоторого информационного разряда. Для ее парирования будем использовать синдромное декодирование.

При вычислении синдромов (суммирование по $\text{mod } 2$):

$$s_i = \sum_{j=1}^k w_j g_{i,j}, \quad 0 \leq i \leq q,$$

если $s_0 = 0$, то ошибки не произошло. В противном случае двоичный номер искаженного разряда есть число s_1, \dots, s_q и вектор ошибки e (веса 1) может быть получен его дешифрированием.

4. СИНТЕЗ СФК НА ОСНОВЕ СПЕКТРАЛЬНОГО R-КОДА

Схема функционального контроля R-кода состоит из исходной схемы, кодера, состоящего из копии исходной схемы и элементов “исключающее ИЛИ”, декодера и корректирующей схемы (рис. 1). Данная схема позволяет обнаружить однократные и двукратные ошибки, а также исправить однократные ошибки.

Кодер в СФК вычисляет m – число проверочных бит. При использовании кодов с проверкой на четность данные биты – биты четности некоторых подмножеств информационных бит выходного вектора основной схемы. Данные подмножества формируются с помощью порождающей матрицы.

В предлагаемом в данной статье R-коде для декодирования используется метод синдромного декодирования. Декодер для данного метода декодирования состоит из схемы обнаружения ошибки, включающей в себя подсхему вычисления синдрома, и схемы определения вектора ошибки. Последующее ее исправление выполняет корректирующая схема. Декодер определяет, произошла ли ошибка и в таком случае вычисляет некоторый синдром, по которому дешифратор вычисляет позицию ошибки. Рассмотрим в качестве примера синтез схемы функционального контроля для комбинационной схемы frg1_synth из набора тестовых схем LGSynth89 [16].

Данная схема обладает входами x_1, \dots, x_{28} и выходами y_1, y_2, y_3 . Для начала нам необходимо вычислить число проверочных разрядов:

$$m = \lceil \log k \rceil + 1 = \lceil \log 3 \rceil + 1 = 3,$$

где k – число информационных разрядов (число выходов исходной схемы).

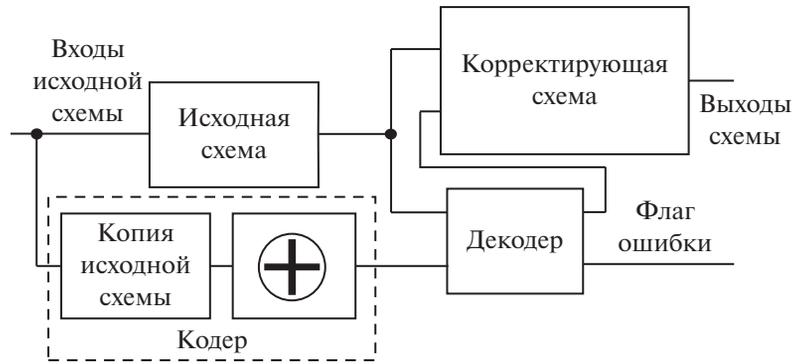


Рис. 1. Структура схемы функционального контроля.

После вычисления m , мы можем сгенерировать порождающую матрицу G , с помощью которой можно вычислить значения проверочных разрядов v_4, v_5 и v_6 для $u = (y_1, y_2, y_3)^T$:

$$G = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Значения проверочных разрядов (битов четности), благодаря полученной матрице G , определяются следующими соотношениями:

$$\begin{aligned} v_4 &= y_1 + y_2 + y_3, \\ v_5 &= y_3, \\ v_6 &= y_2. \end{aligned}$$

Полученные соотношения образуют части кодера и декодера, в которых вычисляются биты

четности для исходной схемы и ее копии, необходимые в дальнейшем для вычисления синдромов. Число синдромов совпадает с числом проверочных разрядов.

Допустим, на выходе исходной схемы при подаче входного сигнала были получены выходные значения с ошибкой:

$$\tilde{y} = (\tilde{y}_1, \tilde{y}_2, \tilde{y}_3) = (0, 1, 0).$$

Так как мы рассматриваем попадание однократной ошибки в исходную схему, из этого следует, что в кодере на выходе копии исходной схемы будут получены выходные значения без искажения:

$$y = (y_1, y_2, y_3) = (0, 0, 0).$$

Синдромы для представленной комбинационной схемы вычисляются следующим образом:

$$\begin{aligned} s_0 &= (\tilde{y}_1 + \tilde{y}_2 + \tilde{y}_3) + (y_1 + y_2 + y_3) = (0 + 1 + 0) + (0 + 0 + 0) = 1, \\ s_1 &= \tilde{y}_3 + y_3 = 0 + 0 = 0, \\ s_2 &= \tilde{y}_2 + y_2 = 1 + 0 = 1. \end{aligned}$$

Схема функционального контроля для рассматриваемой в данном примере схемы представлена на рис. 2.

Необходимо заметить, что корректное парирование сбоев схем с использованием спектрального R-кода будет происходить при выполнении следующих условий:

1. Считаем, что может быть искажен только один разряд как выходного вектора основной схемы $\tilde{y} = (\tilde{y}_1, \dots, \tilde{y}_k)^T = y + e$, так и вектора проверочных бит z , однако всех ошибок не должно быть более двух.

2. Значение z_0 (бит четности неискаженного вектора y) кодером должно всегда вычисляться верно. Для этого необходимо обеспечить специ-

альной технологической защитой соответствующие части кодера. При этом подсхемы вычисления остальных проверочных бит z_1, \dots, z_q могут оставаться незащищенными.

5. КЛАСТЕРИЗАЦИЯ ВЫХОДОВ СХЕМЫ

Для уменьшения вероятности возникновения многократных ошибок в данной статье предлагается использовать алгоритм разбиения выходов исходной схемы с последующим синтезом СФК для каждого из полученных групп. Использование данного подхода позволяет значительно повысить вероятность обнаружения/исправления ошибок кодами SEC/DED, но при этом приводя к дополнительным структурным затратам (рис. 3).

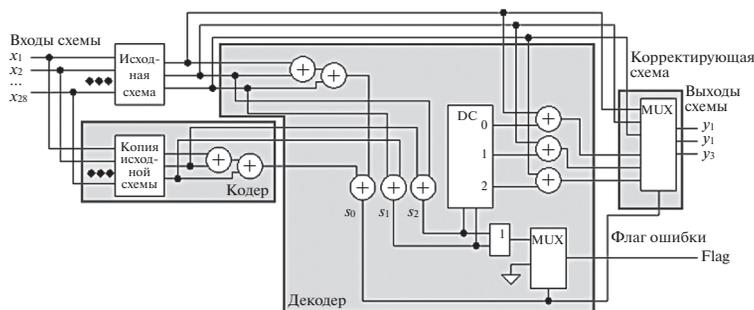


Рис. 2. Схема функционального контроля для схемы frg1_synth из тестового набора LGSynth89.

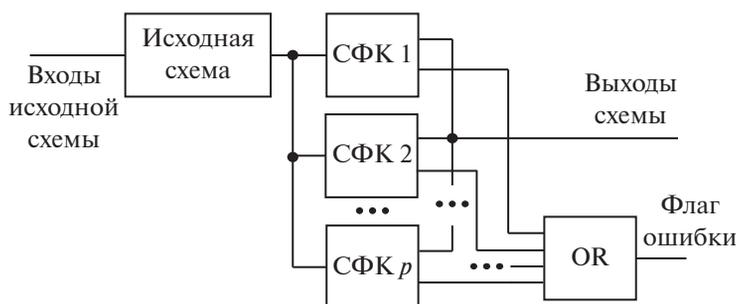


Рис. 3. Структура схемы с кластеризацией выходов и последующим синтезом СФК для каждой из полученных кластеров.

В статье предлагается алгоритм разбиения множества выходов схемы на две, необязательно равные группы. Данный алгоритм является детерминированным и жадным, и при необходимости может быть применен еще раз к одному из образовавшихся кластеров или к каждому из них.

Обозначим через $\{1, \dots, k\}$ множество всех k выходов исходной схемы, которые необходимо разбить на два кластера. Предполагаем, что известна логическая схема и алгоритм ее функционирования, позволяющий по входному набору определить ее выходной набор и производить моделирование возникающих в ней сбоев. Реальные схемы, для которых предполагается проводить подобную процедуру, должны иметь число выходов порядка нескольких десятков.

Исходной информацией для предлагаемого алгоритма является квадратная симметрическая матрица C порядка k взаимозависимости ошибок на выходах, получаемая следующим образом: на входы исходной схемы подается достаточно большая серия случайных входных наборов данных. Для каждого входного набора определяется выход

$y = (y_1, \dots, y_k)$ схемы. Затем согласно некоторой принятой модели возникновения сбоев в схему вносится та или иная неисправность и определяются искаженный выход $\tilde{y} = (\tilde{y}_1, \dots, \tilde{y}_k)$ и бинарный вектор ошибок $e = y + \tilde{y}$.

Элементами c_{ij} матрицы C являются количества (или величины им пропорциональные) векторов ошибок, имеющих 1 одновременно в i -м и j -м своих разрядах, $1 \leq i \leq k$, $i + 1 \leq j \leq k$. Ясно, что $c_{ii} = 0$ и $c_{ij} = c_{ji}$, $i, j = \overline{1, k}$.

Алгоритм начинается с определения множеств A и B , которые назовем *долями*, и множества V . В начале работы обе доли пусты, а V содержит номера всех выходов схемы. На каждом шаге алгоритма одна из долей будет пополняться некоторым выбранным элементом s из множества V , а сам элемент s — исключаться из V . Алгоритм закончит работу при полном опустошении множества V . В результате доли будут содержать номера выходов, образующие кластеризацию множества всех выходов $1, \dots, k$.

Шаг 0. Строим матрицу C , задаем множества $A = B = \emptyset$ и $V = \{1, \dots, k\}$.

Шаг 1. Выбираем ребро (i, j) с максимальным весом c_{ij} . Если таких ребер несколько, выбираем произвольно из них с равной вероятностью.

Произвольно пополняем множества A и B одним из элементов i и j каждое.

Удаляем из V элементы i и j .

Шаг 2. Для всех вершин $i \in V$ вычисляем суммы:

$$S_A(i) = \sum_{i \in A} c_{it}, \quad S_B(i) = \sum_{i \in B} c_{it}.$$

Выбираем вершину s , на которой достигается максимум, по какой-либо из подсчитанных сумм:

$$s = \arg \max_{i \in V} \{S_A(i), S_B(i)\}.$$

Шаг 3. Элемент s удаляем из V и пополняем им:

- множество A , если $S_A(s) < S_B(s)$,
- множество B , если $S_A(s) > S_B(s)$.

В случае $S_A(s) = S_B(s)$ пополняем элементом s множество наименьшей мощности; если же и $|A| = |B|$, множество для пополнения выбираем произвольно равновероятно.

Если множество V не пусто, переходим к **Шагу 2**, иначе **СТОП**. Относительное качество работы алгоритма может быть оценено величиной

$$Q = \frac{\sum_{i \in A, j \in B} c_{ij}}{\sum_{i, j \in A} c_{ij} + \sum_{i, j \in B} c_{ij}}.$$

Чем больше значение Q , тем лучше произведено разбиение. Отметим, что предлагаемый жадный алгоритм не гарантирует получения оптимального значения Q .

Рассмотрим кластеризацию выходов схемы на примере схемы `misex1` из набора тестовых схемы `LG Synth89`.

Шаг 0. Для начала необходимо определить матрицу C взаимозависимости ошибок на выходах схемы ($k = 7$). $A = B = \emptyset$, $V = \{1, 2, 3, 4, 5, 6, 7\}$.

$$C = \begin{pmatrix} 0.0000 & 0.0000 & 0.0041 & 0.0072 & 0.0113 & 0.0113 & 0.0113 \\ 0.0000 & 0.0000 & 0.0586 & 0.0132 & 0.0444 & 0.0642 & 0.0280 \\ 0.0041 & 0.0586 & 0.0000 & 0.0082 & 0.0352 & 0.0257 & 0.0302 \\ 0.0072 & 0.0132 & 0.0082 & 0.0000 & 0.0183 & 0.0183 & 0.0237 \\ 0.0113 & 0.0444 & 0.0352 & 0.0183 & 0.0000 & 0.0612 & 0.0396 \\ 0.0113 & 0.0642 & 0.0257 & 0.0183 & 0.0612 & 0.0000 & 0.0525 \\ 0.0113 & 0.0280 & 0.0302 & 0.0237 & 0.0396 & 0.0525 & 0.0000 \end{pmatrix}.$$

Шаг 1. Выберем ребро с максимальным весом $c_{26} = 0.0642$. Пополняем пустые множества A и B элементами i и j соответственно:

$$A = \{2\}, B = \{6\}, V = \{1, 3, 4, 5, 7\}.$$

Шаг 2.1. Для всех вершин $i \in V$ вычисляем суммы.

$$i = 1: S_A(1) = 0.0000, S_B(1) = 0.0113;$$

$$i = 3: S_A(3) = 0.0586, S_B(3) = 0.0257;$$

$$i = 4: S_A(4) = 0.0132, S_B(4) = 0.0183;$$

$$i = 5: S_A(5) = 0.0444, S_B(5) = 0.0612;$$

$$i = 7: S_A(7) = 0.0280, S_B(7) = 0.0525.$$

Вершина, на которой достигается максимум, по какой-либо из посчитанных сумм: $s = 5$.

Шаг 3.1. Добавляем s в множество A : $S_A(s) < S_B(s)$ и удаляем ее из множества V : $V = \{1, 3, 4, 7\}$.

$$A = \{2, 5\}, S_A = 0.0444, B = \{6\}, S_B = 0.$$

$V \neq \emptyset$ – переходим к Шагу 2.

Шаг 2.2.

$$i = 1: S_A(1) = 0.0557, S_B(1) = 0.0113;$$

$$i = 3: S_A(3) = 0.1383, S_B(3) = 0.0257;$$

$$i = 4: S_A(4) = 0.0760, S_B(4) = 0.0183;$$

$$i = 7: S_A(7) = 0.1120, S_B(7) = 0.0525.$$

Шаг 3.2.

$$A = \{2, 5\}, S_A = 0.0444; B = \{3, 6\}, S_B = 0.0257.$$

Шаг 2.3.

$$i = 1: S_A(1) = 0.0557, S_B(1) = 0.0410;$$

$$i = 4: S_A(4) = 0.0760, S_B(4) = 0.0522;$$

$$i = 7: S_A(7) = 0.1120, S_B(7) = 0.1084.$$

Шаг 3.3.

$$A = \{2, 5\}, S_A = 0.0444; B = \{3, 6, 7\}, S_B = 0.1084.$$

Шаг 2.4.

$$i = 1: S_A(1) = 0.0557, S_B(1) = 0.1350;$$

$$i = 4: S_A(4) = 0.0760, S_B(4) = 0.1586.$$

Шаг 3.4.

$$A = \{2, 5\}, S_A = 0.0760; B = \{3, 6, 7\}, S_B = 0.1084.$$

Шаг 2.5.

$$i = 1: S_A(1) = 0.0945, S_B(1) = 0.1350.$$

Шаг 3.4.

$$A = \{2, 5\}, S_A = 0.0945; B = \{3, 6, 7\}, S_B = 0.1084.$$

$V = \emptyset$, поэтому **СТОП**.

Вычисляем качество работы алгоритма:

$$Q = \frac{0.5666}{0.2029} = 2.7923.$$

6. РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Был проведен ряд вычислительных экспериментов на схемах из тестовых наборов ISCAS'85 [17] и LGSynth89 для оценки эффективности применения R-кода в целях повышения сбоеустойчивости комбинационных схем, а также выявления эффективности применения предлагаемого алгоритма кластеризации выход схемы. В качестве сбоя рассматривалась инверсия сигнала на выходе вентиля.

Во-первых, была проведена оценка структурных затрат для схемы функционального контроля на основе спектрального R-кода. В табл. 1 представлены результаты сравнения исходной схемы со схемой функционального контроля с и без применения оптимизации программой Yosys [18] (оптимизация кодера и декодера проводилась отдельно). При этом в столбце "Конус" представлено число элементов кодера, которые предстоит защитить специальными технологическими методами. В результате было получено, что благода-

ря оптимизации схема функционального контроля для представленных схем в среднем была уменьшена на 18.7%.

Во-вторых, была проведена оценка числа элементов схемы функционального контроля для спектрального R-кода и тройного модульного резервирования для схем из наборов ISCAS'85 и LGSynth89. В табл. 2 представлены результаты сравнения структурных затрат схем функционального контроля на основе спектрального R-кода и схем, полученных с помощью тройного модульного резервирования [19].

В-третьих, была проведена оценка предлагаемого алгоритма кластеризации. При проведении численных экспериментов рассматривалось разбиение на 2 и 4 кластера. В первом случае алгоритм кластеризации применялся ко всему множеству выходов, во втором случае – применялся к каждому из полученных в результате первого использования алгоритма кластеров.

Таблица 1. Оценка числа элементов схемы функционального контроля для R-кода до и после оптимизации для схем из тестовых наборов ISCAS'85 и LGSynth89

Benchmark	Число элементов исходной схемы	Конус	Число элементов схемы функционального контроля	Число элементов схемы функционального контроля (оптимизированная)	Соотношение числа элементов схемы функционального контроля с исходной схемой	Соотношение числа элементов схемы функционального контроля (оптимизированная) с исходной схемой
ISCAS						
c1355	590	1668	1169	312	2.83	1.98
c1908	1057	2480	1593	350	2.35	1.51
c499	246	980	806	307	3.98	3.28
c5315	2973	8268	5811	1684	2.78	1.95
c880	435	1252	1031	402	2.88	2.37
LGSynth89						
rot_synth	786	3552	2810	982	4.52	3.58
pcle_synth	83	280	228	91	3.37	2.75
misex2	134	526	464	177	3.93	3.46
C1908_synth	452	1270	1002	360	2.81	2.22
fig2_synth	733	4236	3365	1230	5.78	4.59

Таблица 2. Оценка числа элементов схемы функционального контроля для спектрального R-кода и тройного модульного резервирования для схем из наборов ISCAS'85 и LGSynth89

Benchmark	Число элементов исходной схемы	Число элементов схемы тройного модульного резервирования	Число элементов схемы функционального контроля (R-код)	Соотношение числа элементов схемы тройного модульного резервирования и исходной схемы	Соотношение числа элементов схемы функционального контроля (R-код) и исходной схемы
ISCAS					
c1355	590	1930	1169	3.27	1.98
c1908	1057	3296	1593	3.12	1.51
c499	246	898	806	3.65	3.28
c5315	2973	9534	5811	3.21	1.95
c880	435	1435	1031	3.30	2.37
LGSynth89					
rot_synth	786	2893	2810	3.68	3.58
pcle_synth	83	294	228	3.54	2.75
misex2	134	492	464	3.67	3.46
C1908_synth	452	1481	1002	3.28	2.22
fig2_synth	733	2894	3365	3.95	4.59

Таблица 3. Оценка предлагаемого алгоритма кластеризации для схем из наборов ISCAS'85 и LGSynth89

Benchmark	Число кластеров													
	2					4								
	A	B	SA	SB	Q	A	B	C	D	SA	SB	SC	SD	Q
ISCAS														
c1355	16	16	0.015	0.015	3.4	8	8	8	8	0.000	0.000	0.000	0.000	∞
c1908	11	14	0.131	0.132	2.5	4	7	6	8	0.022	0.020	0.024	0.025	7.3
c499	16	16	0.031	0.030	3.2	8	8	8	8	0.000	0.000	0.000	0.000	∞
c5315	42	81	0.190	0.190	2.6	22	20	38	43	0.033	0.033	0.035	0.035	7.3
c880	8	18	0.026	0.026	2.7	5	3	15	3	0.003	0.005	0.004	0.004	8.9
LGSynth89														
rot_synth	68	39	0.147	0.160	2.6	47	21	21	18	0.011	0.012	0.025	0.026	10.8
pair_synth	74	63	0.571	0.572	2.2	45	29	33	30	0.122	0.124	0.123	0.123	5.2
misex2	7	11	0.000	0.000	∞	4	3	6	5	0.000	0.000	0.000	0.000	∞
C1908_synth	17	8	0.247	0.247	2.6	8	9	5	3	0.038	0.038	0.032	0.031	9.3
frg2_synth	71	68	0.950	0.951	2.1	42	29	29	39	0.222	0.222	0.222	0.221	4.5

В табл. 3 представлены результаты проведенных численных экспериментов, где A, B, C, D – число выходов, попавших в кластер после применения предлагаемого алгоритма, SA, SB, SC, SD – результирующая стоимость кластера после применения предлагаемого алгоритма, Q – коэффициент качества разбиения. Как видно из представленной таблицы результирующая стоимость равномерно распределена между полученными кластерами, а коэффициент качества для некоторых схем уходит в бесконечность, т.е. достигает “идеального” значения (стоимость полученных кластеров равна 0).

В-четвертых, были проведены эксперименты, связанные с инжектированием однократных ошибок в полученные СФК. По результатам, представленным в табл. 4 под графой “Без группировки”, можно приблизительно оценить доли возможных исходов результатов сбоя:

- 65.3% сбоя маскируются с нулевым флагом ошибки;
- 17.5% сбоя исправляются схемой функционального контроля;
- 4.4% сбоя были пропущены;
- в 10.1% случаев получен сигнал об ошибке без ее наличия;
- в 2.6% случаев получен сигнал о наличии двукратной ошибки (однократная ошибка в каком-либо элементе схемы может привести к множественной ошибке на выходе [20]).

Заметим, что приведенные результаты говорят о том, что использование R-кода обеспечивает в среднем исправление или обнаружение прибли-

зительно 85.4% реальных ошибок на выходе схем без использования разбиений выходов на группы.

Также в табл. 4 представлены результаты численных экспериментов, связанных с инжектированием однократных ошибок в СФК на основе спектрального R-кода при применении предлагаемого алгоритма кластеризации на 2 и 4 группы. При сравнении полученных результатов с значениями без группировки следует, что применение предлагаемого алгоритма по разбиению выходов комбинационной схемы приводит к увеличению вероятности маскирования/исправления ошибки, а также к существенному снижению вероятности возникновения двукратной ошибки.

По результатам, представленным в табл. 4 под графой “Группировка”, можно приблизительно оценить доли возможных исходов результатов сбоя для группировки на 2 кластера:

- 56.3% сбоя маскируются;
- 29.8% сбоя исправляются;
- 11.3% сбоя были пропущены;
- 2.1% сбоя оказались ложной тревогой;
- 0.4% обнаружение двукратной ошибки.

При применении предлагаемого алгоритма для группировки на 4 кластера:

- 66.9% сбоя маскируются;
- 26.6% сбоя исправляются;
- 5.1% сбоя были пропущены;
- 0.9% сбоя оказались ложной тревогой;
- 0.5% обнаружение двукратной ошибки.

Применение алгоритма кластеризации с последующим синтезом схемы функционального кон-

Таблица 4. Результаты инъектирования однократных ошибок в схемы функционального контроля на основе спектрального R-кода без и с группировкой с помощью предлагаемого алгоритма кластеризации для схем из наборов ISCAS'85 и LGSynth89

Benchmark	Маскирование, %	Исправление, %	Ложная тревога, %	Пропущена ошибка, %	Обнаружена двукратная ошибка, %
ISCAS					
Без группировки					
c1355	56.080	21.286	18.477	1.069	3.088
c1908	61.051	25.081	6.161	4.143	3.564
c499	47.254	17.498	28.641	1.040	5.568
c5315	59.528	15.437	17.360	3.577	4.097
c880	53.126	27.983	13.937	1.411	3.544
Группировка (Число кластеров = 2)					
c1355	64.576	34.570	0.120	0.734	0
c1908	63.788	32.027	0	3.632	0.553
c499	62.997	35.361	0.129	1.513	0
c5315	70.630	25.963	0	3.028	0.379
c880	56.286	42.563	0	0.995	0.157
Группировка (Число кластеров = 4)					
c1355	65.629	34.371	0	0	0
c1908	65.018	32.277	0	2.663	0.042
c499	59.723	40.277	0	0	0
c5315	71.048	27.294	0	1.603	0.056
c880	55.468	43.905	0	0.618	0.009
LGSynth89					
Без группировки					
rot_synth	40.591	17.697	32.742	3.274	5.697
pcle_synth	44.111	35.422	13.597	1.054	5.817
misex2	55.673	14.108	22.375	0.152	7.692
C1908_synth	50.999	22.417	13.300	6.009	7.275
frg2_synth	31.951	21.210	39.517	1.365	5.957
Группировка (Число кластеров = 2)					
rot_synth	41.744	23.540	32.093	2.384	0.239
pcle_synth	52.281	35.604	9.524	1.162	1.430
misex2	62.844	16.672	20.379	0.019	0.086
C1908_synth	53.363	26.826	12.433	6.309	1.069
frg2_synth	34.167	25.313	38.440	1.558	0.521
Группировка (Число кластеров = 4)					
rot_synth	75.826	13.800	8.488	0.911	0.974
pcle_synth	68.686	25.140	3.766	0.775	1.634
misex2	64.104	14.471	21.048	0.046	0.331
C1908_synth	74.524	19.899	2.305	1.680	1.592
frg2_synth	68.480	14.441	15.763	0.804	0.513

троля на основе R-кода обеспечивает в среднем исправление или обнаружение приблизительно 86.1% реальных ошибок на выходе схем при разбиении выходов на 2 кластера, и приблизительно 93.4% ошибок – при разбиении на 4 кластера.

7. ЗАКЛЮЧЕНИЕ

В данной статье описан спектральный R-код с проверкой на четность с произвольным числом информационных символов. Код относится к классу SEC/SED, хотя и имеет кодовое расстояние равное 2, что является следствием безошибочности вычисления бита четности выходного вектора основной схемы. Важным свойством спектрального R-кода является возможность не обеспечивать технологической защитой всю корректирующую схему.

По результатам проведенных численных экспериментов было получено, что схема функционального контроля спектрального R-кода обладает меньшей структурной избыточностью, чем схема тройного модульного резервирования для схем из тестовых наборов ISCAS'85 и LGSynth89 в среднем на 32.7%.

Проведенные эксперименты по инъектированию однократных ошибок в СФК на основе R-кода позволили прийти к выводу об эффективности применения данного подхода для обнаружения двукратных и исправления однократных ошибок в комбинационных схемах.

Применение предлагаемого в данной статье алгоритма разбиения выходов комбинационной схемы на группы с последующим синтезом схемы функционального контроля на основе спектрального R-кода позволило существенно снизить вероятность возникновения многократных ошибок на выходе схемы, а также при этом повысить вероятность маскирования/исправления ошибки приблизительно на 8%.

СПИСОК ЛИТЕРАТУРЫ

1. *Shivakumar P., Kistler M., Keckler S.W., Burger D., Alvisi L.* Modeling the effect of technology trends on the soft error rate of combinational logic // Proceedings of international conference on dependable systems and networks, 2002 (DSN 2002), June 2002. P. 389–398.
2. *Narayanan V., Xie Y.* Reliability concerns in embedded system designs // Computer. 2006. V. 39(1). P. 118–120.
3. *Стемпковский А.Л., Тельпухов Д.В., Соловьев Р.А., Тельпухова Н.В.* Исследование вероятностных методов оценки логической уязвимости комбинационных схем. Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2016. № 4. С. 121–126.
4. Accurate reliability analysis of concurrent checking circuits employing an efficient analytical method / Stempkovsky A.L. et al. // Microelectronics Reliability. 2015. V. 55. P. 696–703.
5. *Стемпковский А.Л., Тельпухов Д.В., Соловьев Р.А., Мячиков М.В.* Повышение отказоустойчивости логических схем с использованием нестандартных мажоритарных элементов // Информационные технологии. 2015. Т. 21. № 10. С. 749–756.
6. *El-Maleha A.H., Oughalia F.C.* Microelectronics Reliability. 2014. V. 54. № 1. P. 316–326.
7. *Ефанов Д.В.* Три теоремы о кодах Бергера в схемах встроенного контроля / Д.В. Ефанов // Информатика и системы управления. 2013. № 1(35). С. 77–86.
8. *Гуров С.И.* Спектральный R-код с проверками на четность // Прикладная математика и информатика: Труды факультета Вычислительной математики и кибернетики. М.: МАКС Пресс, 2017. № 55. С. 91–96.
9. *Poolakkaparambil M., Mathew J.* BCH code based multiple bit error correction in finite field multiplier circuits // ISQED. 2011. P. 1–6.
10. *Галлагер Р.* Теория информации и надежная связь. М.: Сов. Радио, 1974.
11. *Савченко Ю.Г.* Цифровые устройства нечувствительные к неисправностям элементов. М.: Советское радио. 1977.
12. *Мак-Вильямс Дж.* Перестановочное декодирование системных кодов / Кибернетический сборник, № 1, (новая серия). М.: Мир. 1965. С. 35–57.
13. *Щербаков Н.С.* Самокорректирующиеся дискретные устройства. М.: Машиностроение, 1975.
14. *Stempkovskiy A.L., Telpukhov D.V., Gurov S.I., Zhukova T.D., Demeneva A.I.* R-code for concurrent error detection and correction in the logic circuits // Young Researchers in Electrical and Electronic Engineering (EIconRus), 2018 IEEE Conference of Russian. – IEEE. 2018. P. 1430–1433.
15. *Карповский М.Г., Москалев Э.С.* Спектральные методы анализа и синтеза дискретных устройств. Л.: Энергия, 1973.
16. URL: <https://people.engr.ncsu.edu/brglez/CBL/benchmarks/LGSynth89/> (дата обращения: 09.11.2018).
17. URL: http://icdm.ippm.ru/w/Схемы_ISCAS85/ (дата обращения: 09.11.2018).
18. URL: <http://www.clifford.at/yosys/> (дата обращения: 09.11.2018).
19. *Стемпковский А.Л., Тельпухов Д.В., Жукова Т.Д., Гуров С.И., Соловьев Р.А.* Методы синтеза боеустойчивых комбинационных КМОП схем, обеспечивающих автоматическое исправление ошибок // Известия ВУЗов. ЮФУ. 2017. № 7(192). С. 197–210.
20. *Gavrilov S.V., Gurov S.I., Zhukova T.D., Rukhlov V.S., Ryzhova D.I., Telpukhov D.V.* Methods to Increase Fault Tolerance of Combinational Integrated Microcircuits by Redundancy Coding. Computational Mathematics and Modeling. 2017. V. 28. № 3. P. 400–406.