

Управление в технических системах

© 2025 г. Д.В. ЕФАНОВ, д-р техн. наук (TrES-4b@yandex.ru)
(Санкт-Петербургский политехнический университет Петра Великого;
Российский университет транспорта, Москва),
Д.В. ПИВОВАРОВ, канд. техн. наук (pivovarov.d.v.spb@gmail.com)
(Петербургский государственный университет путей сообщения
Императора Александра I, Санкт-Петербург)

СИНТЕЗ САМОПРОВЕРЯЕМЫХ ДИСКРЕТНЫХ УСТРОЙСТВ НА ОСНОВЕ ПОЛИНОМИАЛЬНЫХ КОДОВ С КОНТРОЛЕМ ВЫЧИСЛЕНИЙ ПО НЕСКОЛЬКИМ ДИАГНОСТИЧЕСКИМ ПРИЗНАКАМ

Предложено при синтезе самопроверяемых дискретных устройств с контролем вычислений по нескольким диагностическим признакам использовать полиномиальные коды. Разработан алгоритм быстрого получения функций, описывающих проверочные символы полиномиальных кодов в виде логических выражений. Показано, что кодеры полиномиальных кодов могут быть отнесены к устройствам трех видов: 1) на выходах которых реализуются исключительно самодвойственные булевы функции, 2) на выходах которых реализуются исключительно «близкие» к самодвойственным (самоквазидвойственным) булевы функции и 3) на выходах которых реализуются и самодвойственные, и самоквазидвойственные булевы функции. Разработана классификация полиномиальных кодов, учитывающая эту особенность. Описана структура организации контроля вычислений на выходах самодвойственных дискретных устройств по нескольким диагностическим признакам. Предложен алгоритм синтеза полностью самопроверяемого дискретного устройства с контролем вычислений по нескольким диагностическим признакам, отличающийся от известных тем, что учитывает характер возникающих на выходах дискретных устройств ошибок и предварительное покрытие их с помощью тестеров самодвойственных и/или самоквазидвойственных сигналов. Результаты могут быть использованы при разработке автоматизированных средств проектирования дискретных устройств для широкого круга приложений.

Ключевые слова: самопроверяемые дискретные устройства, контроль вычислений по нескольким диагностическим признакам, контроль самодвойственности сигналов, контроль самоквазидвойственности сигналов, полиномиальные коды.

DOI: 10.31857/S0005231025050036, EDN: AXPIOQ

1. Введение

Рост сложности разрабатываемых и внедряемых технических систем, обеспечивающих протекание ответственных технологических процессов, повышение их производительности, наделение развитым функционалом, в том числе искусственным интеллектом, требуют особого внимания к надежности и безопасности работы. В этой связи важнейшим аспектом является наделение устройств возможностью идентификации неисправностей в процессе выполнения ими своих функций.

При разработке дискретных систем и устройств управления их структуры наделяют различными свойствами, позволяющими с легкостью обеспечивать обнаружение возникающих неисправностей. Существуют методы контролепригодного проектирования устройств [1], а также разработки для них самопроверяемых структур [2]. Это для любых систем критического применения, в том числе для таких, входные данные в которых меняются редко (к ним можно отнести системы управления в атомной промышленности, установки электрической централизации стрелок и сигналов на железнодорожном транспорте, комплексы противовоздушной обороны и т.д.). В таких системах не должно возникать «скрытых» неисправностей и накопления ошибок, поскольку так или иначе это с течением времени приведет к нарушению свойства готовности реализации алгоритма управления [3, 4].

Для своевременного обнаружения неисправностей в дискретных устройствах требуется их самопроверяемая реализация. Это делается различными методами с внесением избыточности в их структуры по определенным алгоритмам, подразумевающим, например, кодирование состояний и использование самопроверяемых схем встроенного контроля (СВК) [5].

Одним из действенных подходов к реализации самопроверяемых СВК является использование так называемого инвертирования данных, когда все входные и выходные сигналы представляются не в виде постоянных значений 0 и 1, а в виде последовательностей 0101...01 и 1010...10 [6]. Такой подход к реализации самопроверяемых дискретных устройств связан с использованием не только аппаратной избыточности, но и временной. Он подразумевает организацию контроля вычислений по определенному диагностическому признаку, учитывающему постоянную смену входных сигналов и подачу наборов значений аргументов парами – рабочего и контрольного наборов. Особенности, которые можно учесть при организации диагностического обеспечения в таком режиме функционирования, обладают самодвойственными и «близкие» к ним булевы функции [7]. Вопросам реализации самодвойственных дискретных устройств посвящено некоторое (небольшое) количество научных работ, например [6, 8–11]. Также известны три монографии, освещающие базовые сведения из теории синтеза самодвойственных дискретных устройств [12–14].

Для контроля вычислений по признаку самодвойственности требуется самодвойственная реализация дискретных устройств, подразумевающая такую организацию их структур, которая позволяет описать все их выходы самодвойственными булевыми функциями. Далеко не все дискретные устройства являются самодвойственными. Можно привести здесь несколько примеров: полный сумматор, мажоритарный элемент, кодеры некоторых разделимых блоковых кодов и др. [6]. Однако учитывая то, что число самодвойственных функций от t переменных равно 2^{2^t-1} , существует возможность построения $C_{2^{2^t-1}}^m$ комбинационных устройств с $m \leq 2^{2^t-1}$ выходами. Например, при $t = 4$ и $m = 4$ можно построить 174 792 640 различных самодвойственных комбинационных дискретных устройств. Кроме того, любую структуру комбинационного дискретного устройства можно преобразовать в самодвойственную с использованием только одной переменной на основании известного разложения Шеннона по одной переменной [15]. Для самодвойственной реализации устройств с памятью достаточно комбинационные части сделать самодвойственными [6]. Особенности построения самодвойственных дискретных устройств хорошо изучены, например, в [9].

В настоящей статье сфокусируем внимание читателя на вопросах организации контроля вычислений на выходах самодвойственных дискретных устройств, применяя не только диагностический признак самодвойственности сигналов, но и ряд дополнительных признаков. Такой подход к построению самопроверяемого устройства, как показывают многочисленные эксперименты, позволяет повышать показатели контролепригодности в части наблюдаемости ошибок при использовании аппаратной избыточности меньшей, чем при известном методе дублирования [16]. Это достигается за счет использования свойств некоторых помехозащищенных кодов, обладающих самодвойственными и «близкими» к ним функциями, описывающими их проверочные символы [17]. В качестве таких кодов в статье рассматривается широкий класс полиномиальных, или алгебраических, кодов [18].

2. Полиномиальные коды и их особенности

2.1. Принципы построения полиномиальных кодов

Полиномиальные коды строятся на основании деления двоичных полиномов, соответствующих информационным сообщениям, с остатком. При делении с остатком различных полиномов степени q на порождающий полином $G(y)$ получают 2^q остатков от деления. Каждый проверочный символ полиномиального кода, таким образом, описывается логическим выражением, в которое входит только операция суммирования по модулю $M = 2(\text{XOR})$ со значениями определенных информационных символов. Далее, для простоты изложения, назовем функции, описывающие проверочные символы кодовых слов полиномиальных кодов, *контрольными функциями*.

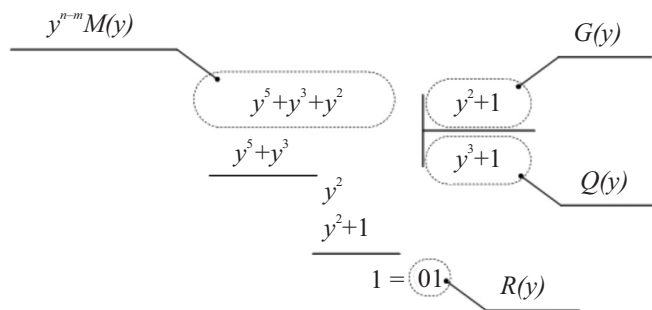


Рис. 1. Пример получения остатка от деления полиномов.

При формировании кодовых слов полиномиальных кодов каждому разряду кодового вектора присваивается переменная y со степенью j , которая соответствует расположению разряда: младшему разряду присваивается значение $j = 0$, следующему – $j = 1$ и т.д. Алгебраическая запись кодового вектора получается путем умножения величины y^j на значение соответствующего разряда. Например, кодовое слово $\langle 1011 \rangle$ можно представить в виде полинома следующим образом: $1 \times y^3 + 0 \times y^2 + 1 \times y^1 + 1 \times y^0$. Удалив нулевые слагаемые, имеем полином вида: $y^3 + y + 1$.

Для образования кодового слова $V(y)$ полиномиального кода определяют остаточный полином $R(y)$, получаемый при делении полинома $M(y)$, умноженного на величину y^{n-m} , на порождающий полином [18]: $y^{n-m}M(y) = G(y)Q(y) + R(y)$, где $Q(y)$ – частное, а $R(y)$ – остаток от деления $y^{n-m}M(y)$ на $G(y)$. В этом случае найденный при делении остаток $R(y)$ будет представлять контрольный вектор, занимающий k младших разрядов кодового слова длиной n . Старшие же t разрядов будут соответствовать информационному вектору.

Закодируем информационный вектор $\langle 1011 \rangle$ полиномиальным кодом с порождающим полиномом $y^2 + 1$. Степень порождающего полинома равна двум (она всегда должна быть меньше или равна степени информационного полинома). При выполнении операции умножения полином $y^{n-m}M(y)$ будет иметь следующий вид: $y^2(y^3 + y + 1) = y^5 + y^3 + y^2$.

Разделим его на порождающий полином $y^2 + 1$ (рис. 1).

Полином, соответствующий кодовому слову, получается в виде суммы полиномов $y^{n-m}M(y)$ и $R(y)$: $y^5 + y^3 + y^2 + 1$. Таким образом, зашифрованное сообщение примет такой вид: $\langle 101101 \rangle$ (вместо исходной комбинации $\langle 1011 \rangle$). По сути, информационный вектор был дополнен контрольным $\langle 01 \rangle$.

Так как порождающему полиному можно сопоставить двоичное число, будем для сокращения записи использовать десятичный эквивалент N этого двоичного числа для указания порождающего полинома. Некоторые свойства обнаружения ошибок в СВК на основе полиномиальных кодов с различными значениями N изучены в [19, 20]. Следует отметить, что порождающий

полином $G(y)$ выбирается исходя из условия обнаружения максимального количества ошибок. Например, нет смысла в выборе в качестве порождающего полинома y^0 , так как при делении любого полинома с произвольной степенью остаток всегда будет равным нулю (имеется еще ряд полиномов даже с высшей степенью, при делении на которые формируется нулевой остаток, например y^2 или y^3). Такие полиномы не позволяют обнаруживать ошибки в кодовом векторе. Далее исключим из рассмотрения порождающие полиномы, не имеющие свободного члена y^0 (числа N , соответствующие таким полиномам, являются четными). Далее будем рассматривать полиномиальные коды, порождающим полиномам которых соответствуют нечетные числа N . Только для таких полиномов возможно получение полного множества контрольных векторов с k разрядами и, соответственно, для них более просто обеспечить полную самопроверяемость кодеров и тестеров данных кодов в СВК [20].

2.2. Способ получения логических выражений, описывающих контрольные функции для полиномиальных кодов

Значения проверочных символов для полиномиальных кодов получают, как правило, с использованием сдвиговых регистров [18], что обуславливается спецификой их применения. Однако существуют некоторые приложения полиномиальных кодов, для которых требуется использование кодеров полиномиальных кодов, являющихся комбинационными схемами. Одним из таких приложений является синтез СВК для дискретных устройств. При построении кодера требуется знать формульную запись, описывающую каждый проверочный символ полиномиального кода, поскольку данное устройство должно быть универсальным и в один такт генерировать по поступающему информационному вектору контрольный вектор.

Чтобы получить формулы, описывающие проверочные символы полиномиального кода, можно использовать тривиальный способ. Он подразумевает для полного множества информационных векторов с числом разрядов m расчет значений проверочных символов и формирование контрольных векторов, а затем для каждого проверочного символа получение логических выражений. Далее по таким функциям легко синтезируется кодер рассматриваемого полиномиального кода, являющийся комбинационной схемой.

Для получения логических выражений, описывающих контрольные функции, необходимо определить значения всех проверочных символов при подстановке в качестве аргументов булевых векторов, соответствующих информационным векторам кода, из полного множества 2^m . Здесь m – число аргументов. Однако уже при $m \geq 6$ это становится достаточно трудоемким. Сократить число информационных векторов при рассмотрении до величины m можно с использованием следующей теоремы.

Теорема 1. Для получения функций, описывающих проверочные символы полиномиальных кодов, необходимо и достаточно определить значения

проверочных символов при подстановке аргументов из информационных векторов с весом $r = 1$.

Доказательство теоремы приводится в Приложении.

Из теоремы 1 следует алгоритм определения логических выражений, описывающих контрольные функции полиномиального кода.

Алгоритм 1. Правила формирования контрольных функций для полиномиальных кодов:

1. Задаются порождающий полином N и количество m разрядов информационного вектора.
2. Определяется число разрядов в контрольных векторах $k = \lfloor \log_2 N \rfloor$.
3. Формируется множество Q информационных векторов с весом $r = 1$ с мощностью $|Q| = m$. Информационные векторы ранжируются по старшинству значащего разряда, и образуется их список.
4. Полагается $i = 1$.
5. Рассматривается i -ый информационный вектор, для которого путем деления определяются значения разрядов контрольного вектора.
6. Составляются формулы для контрольных функций: если значение проверочного символа равно 1, то в формулу включается i -ый информационный разряд, иначе разряд не включается.
7. Разряды в формулах записываются через знак операции XOR.
8. Полагается $i := i + 1$.
9. $i + 1 > m$? Если «да», то переходят к п. 10, если «нет», то переходят к п. 5.
10. Формулы получены.

Пользуясь алгоритмом 1, получим формулы, описывающие контрольные функции рассматриваемого выше полиномиального кода с $N = 13$ и $m = 6$. В табл. 1 приведены все информационные векторы с весом $r = 1$ для рассматриваемого случая.

По табл. 1 нетрудно для каждой функции g_j получить формульную запись. Для этого можно поступить следующим образом.

Таблица 1. Таблица кодовых слов полиномиального кода с порождающим полиномом, соответствующим числу $N = 13$ при $m = 3$, для информационных векторов с весом $r = 1$

m	y_6	y_5	y_4	y_3	y_2	y_1	g_3	g_2	g_1
6	1	0	0	0	0	0	1	0	1
5	0	1	0	0	0	0	1	1	1
4	0	0	1	0	0	0	1	1	0
3	0	0	0	1	0	0	0	1	1
2	0	0	0	0	1	0	1	0	0
1	0	0	0	0	0	1	0	1	0

Алгоритм 2. Правила формирования контрольных функций для полиномиальных кодов по таблице кодовых слов для информационных векторов с весом $r = 1$:

1. Определяются m кодовых слов для информационных векторов с весом $r = 1$.
2. Полагается $j = 1$.
3. Рассматривается j -ый столбец, соответствующий j -ой контрольной функции.
4. Записывается формула, в которую включаются те информационные символы f_i , для которых в j -ом столбце записана 1.
5. Полагается $j := j + 1$.
6. $j + 1 > k$? Если «да», то переходят к п. 7, если «нет», то переходят к п. 3.
7. Формулы получены.

Получаем следующую систему булевых функций:

$$\begin{cases} g_1 = y_3 \oplus y_5 \oplus y_6; \\ g_2 = y_1 \oplus y_3 \oplus y_4 \oplus y_5; \\ g_3 = y_2 \oplus y_4 \oplus y_5 \oplus y_6. \end{cases}$$

Использование алгоритмов 1 и 2 позволяет получить формульные записи для любого полиномиального кода с линейной трудоемкостью. Их можно использовать при синтезе кодеров полиномиальных кодов для применения в составе тестеров в СВК.

3. Классификация полиномиальных кодов по виду контрольных функций

3.1. Особые свойства полиномиальных кодов

Контрольные функции полиномиальных кодов являются линейными. Хорошо известны следующие особенности линейных булевых функций [17].

Теорема 2. Линейная булева функция будет самодвойственной только в том случае, если имеет нечетное количество аргументов, от которых она зависит существенно.

Теорема 3. Линейная булева функция будет самоквазидвойственной только в том случае, если имеет четное количество аргументов, от которых она зависит существенно.

Здесь следует напомнить читателю несколько известных определений [7].

Определение 1. *Функция принадлежит классу самодвойственных булевых функций, если при инвертировании всех ее аргументов она принимает противоположные значения:*

$$SD = \{f(x_1, x_2, \dots, x_t) | f(\overline{x_1}, \overline{x_2}, \dots, \overline{x_t}) = \overline{f(x_1, x_2, \dots, x_t)}\}.$$

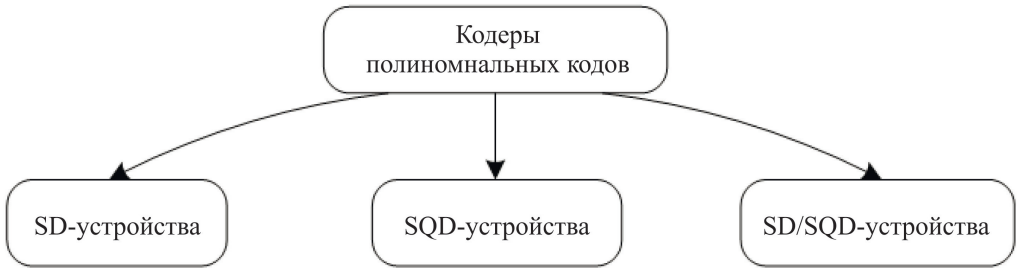


Рис. 2. Классификация кодеров полиномиальных кодов.

Определение 2. Функция принадлежит классу самоквазидвойственных¹ булевых функций, если при инвертировании всех ее аргументов она принимает такие же значения:

$$SQD = \{f(x_1, x_2, \dots, x_t) | f(\overline{x_1}, \overline{x_2}, \dots, \overline{x_t}) = f(x_1, x_2, \dots, x_t)\}.$$

Теоремы 2 и 3 позволяют говорить о том, что полиномиальные коды с различными порождающими полиномами N и с различным количеством m информационных символов могут принадлежать к трем классам: I – коды, кодеры которых описываются исключительно самодвойственными булевыми функциями, II – коды, кодеры которых описываются исключительно самоквазидвойственными булевыми функциями, III – коды, кодеры которых описываются самодвойственными и самоквазидвойственными булевыми функциями.

Определение 3. Дискретные устройства, выходы которых описываются исключительно самодвойственными функциями, называются самодвойственными устройствами, или SD-устройствами.

Определение 4. Дискретные устройства, выходы которых описываются исключительно самоквазидвойственными функциями, называются самоквазидвойственными устройствами, или SQD-устройствами.

Определение 5. Дискретные устройства, часть выходов которых описывается самодвойственными булевыми функциями, а часть – самоквазидвойственными, называются самодвойственно-самоквазидвойственными устройствами, или SD/SQD-устройствами.

Таким образом, кодеры полиномиальных кодов могут оказаться устройствами только трех видов (рис. 2).

Необходимо отметить следующие важные особенности, присущие кодерам полиномиальных кодов [17].

Теорема 4. В схемной реализации линейной булевой функции от нечетного числа аргументов, от которых она зависит существенно, при контро-

¹ Автор [7] использовал для обозначения самоквазидвойственных функций не вполне удачный термин «самоантидвойственные» функции.

ле вычислений по признаку самодвойственности не тестируются неисправности логических элементов, которые связаны путями с четным числом входов.

Теорема 5. В схемной реализации линейной булевой функции от четного числа аргументов, от которых она зависит существенно, при контроле вычислений по признаку самоквазидвойственности не тестируются неисправности логических элементов, которые связаны путями с четным числом входов.

Теоремы 4 и 5 говорят о том, что невозможно синтезировать полностью самопроверяемую СВК по признаку самодвойственности и/или самоквазидвойственности контрольных функций, поскольку невозможно обеспечить покрытие всех неисправностей кодеров полиномиальных кодов при контроле только таких признаков. В самом деле, ошибки, вызываемые неисправностями сумматоров по модулю два в линейных схемах, всегда транслируются на их выходы, а при контроле вычислений по признаку самодвойственности или самоквазидвойственности проверяется парафазность пространственного сигнала, а не параллельно поступающего на входы тестера. Поэтому ошибка при соблюдении условий теорем 4 и 5 проявляется при подаче на входы устройства на обоих наборах значений аргументов и не фиксируется тестером самодвойственного или самоквазидвойственного сигнала. Требуется дополнительный контроль по еще одному диагностическому признаку. В качестве такого признака может выступать контроль принадлежности кодовых слов выбранному полиномиальному коду. Это действительно так, потому что на каждом наборе значений аргументов при наличии ошибки будет либо вычислено правильное значение каждого разряда контрольного вектора, либо произойдет его искажение хотя бы в одном разряде. Тестер полиномиального кода зафиксирует это событие. Таким образом, полиномиальные коды можно эффективно использовать в структуре организации контроля вычислений по нескольким диагностическим признакам, о которой речь пойдет далее.

3.2. Классы полиномиальных кодов по виду кодера

Использование свойств линейных функций, а также способа простого получения логических выражений для функций, описывающих проверочные символы полиномиальных кодов, позволило сформировать классификацию кодов по виду их кодеров, используемых в составах тестеров в СВК [21]. Ее фрагмент приведен в табл. 2.

В табл. 2 в первой графе записаны числа N , соответствующие порождающим полиномам. В таблице приведены все порождающие полиномы для построения полиномиальных кодов с числом проверочных символов $k = 1, \dots, 6$.

Для практического использования удобной формой представления различных классов полиномиальных кодов является матричная форма, где отмечается принадлежность кодера конкретного кода с заданным порождающим

Таблица 2. Детализированная классификация кодов по видам кодеров

N	SD-	SQD-	SD/SQD-
$k = 1$			
3	$m \equiv 1(\text{mod}2)$	$m \equiv 0(\text{mod}2)$	–
$k = 2$			
5	$m \equiv 2(\text{mod}4)$	$m \equiv 0(\text{mod}4)$	$m \equiv 1(\text{mod}2)$
7	$m \equiv 2(\text{mod}3)$	$m \equiv 0(\text{mod}3)$	$m \equiv 1(\text{mod}3)$
$k = 3$			
9	$m \equiv 3(\text{mod}6)$	$m \equiv 0(\text{mod}6)$	$m \equiv \alpha(\text{mod}8), \alpha \in \{0, 1, \dots, 5\} \setminus \{0, 3\}$
11	$m \equiv 2(\text{mod}7)$	$m \equiv 0(\text{mod}7)$	$m \equiv \alpha(\text{mod}7), \alpha \in \{0, 1, \dots, 6\} \setminus \{0, 2\}$
13	$m \equiv 6(\text{mod}7)$	$m \equiv 0(\text{mod}7)$	$m \equiv \alpha(\text{mod}7), \alpha \in \{0, 1, \dots, 6\} \setminus \{0, 6\}$
15	–	$m \equiv 0(\text{mod}4)$	$m \equiv \alpha(\text{mod}7), \alpha \in \{1, 2, 3\}$
$k = 4$			
17	$m \equiv 4(\text{mod}8)$	$m \equiv 0(\text{mod}8)$	$m \equiv \alpha(\text{mod}8), \alpha \in \{0, 1, \dots, 7\} \setminus \{0, 4\}$
19	$m \equiv 3(\text{mod}15)$	$m \equiv 0(\text{mod}15)$	$m \equiv \alpha(\text{mod}15), \alpha \in \{0, 1, \dots, 14\} \setminus \{0, 3\}$
21	–	$m \equiv 0(\text{mod}6)$	$m \equiv \alpha(\text{mod}6), \alpha \in \{1, 2, 3, 4, 5\}$
23	$m \equiv 2(\text{mod}14)$	$m \equiv 0(\text{mod}14)$	$m \equiv \alpha(\text{mod}14), \alpha \in \{0, 1, \dots, 13\} \setminus \{0, 2\}$
25	$m \equiv 8(\text{mod}15)$	$m \equiv 0(\text{mod}15)$	$m \equiv \alpha(\text{mod}15), \alpha \in \{0, 1, \dots, 14\} \setminus \{0, 8\}$
27	$m \equiv 4(\text{mod}12)$	$m \equiv 0(\text{mod}12)$	$m \equiv \alpha(\text{mod}12), \alpha \in \{0, 1, \dots, 11\} \setminus \{0, 4\}$
29	$m \equiv 6(\text{mod}14)$	$m \equiv 0(\text{mod}14)$	$m \equiv \alpha(\text{mod}14), \alpha \in \{0, 1, \dots, 13\} \setminus \{0, 6\}$
31	$m \equiv 4(\text{mod}5)$	$m \equiv 0(\text{mod}5)$	$m \equiv \alpha(\text{mod}5), \alpha \in \{1, 2, 3\}$
$k = 5$			
33	$m \equiv 5(\text{mod}10)$	$m \equiv 0(\text{mod}10)$	$m \equiv \alpha(\text{mod}10), \alpha \in \{0, 1, \dots, 9\} \setminus \{0, 5\}$
35	$m \equiv 2(\text{mod}42)$	$m \equiv \alpha(\text{mod}42),$ $\alpha \in \{0, 20, 31\}$	$m \equiv \alpha(\text{mod}42),$ $\alpha \in \{0, 1, \dots, 41\} \setminus \{0, 2, 20, 31\}$
37	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{40, 56\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 38\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 1, \dots, 61\} \setminus \{0, 38, 40, 56\}$
39	$m \equiv 1(\text{mod}28)$	$m \equiv 0(\text{mod}28)$	$m \equiv \alpha(\text{mod}28), \alpha \in \{0, 1, \dots, 27\} \setminus \{0, 1\}$
41	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{18, 60\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 39, 62\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 1, \dots, 61\} \setminus \{0, 18, 39, 60, 62\}$
43	$m \equiv 11(\text{mod}30)$	$m \equiv 0(\text{mod}30)$	$m \equiv \alpha(\text{mod}30), \alpha \in \{0, 1, \dots, 29\} \setminus \{0, 11\}$
45	–	$m \equiv 0(\text{mod}12)$	$m \equiv \alpha(\text{mod}12), \alpha \in \{1, 2, \dots, 11\}$
47	$m \equiv 2(\text{mod}31)$	$m \equiv 0(\text{mod}31)$	$m \equiv \alpha(\text{mod}31), \alpha \in \{0, 1, \dots, 30\} \setminus \{0, 2\}$
49	–	$m \equiv 0(\text{mod}21)$	$m \equiv \alpha(\text{mod}21), \alpha \in \{1, 2, \dots, 20\}$
51	$m \equiv 5(\text{mod}8)$	$m \equiv 0(\text{mod}8)$	$m \equiv \alpha(\text{mod}8), \alpha \in \{0, 1, \dots, 7\} \setminus \{0, 5\}$
53	$m \equiv 1(\text{mod}30)$	$m \equiv 0(\text{mod}30)$	$m \equiv \alpha(\text{mod}30), \alpha \in \{0, 1, \dots, 29\} \setminus \{0, 1\}$
55	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{34, 42\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 10\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 1, \dots, 62\} \setminus \{0, 10, 34, 42\}$
57	$m \equiv 17(\text{mod}28)$	$m \equiv 0(\text{mod}28)$	$m \equiv \alpha(\text{mod}28), \alpha \in \{0, 1, \dots, 27\} \setminus \{0, 17\}$
59	$m \equiv 18(\text{mod}31)$	$m \equiv 0(\text{mod}31)$	$m \equiv \alpha(\text{mod}31), \alpha \in \{0, 1, \dots, 30\} \setminus \{0, 18\}$
61	$m \equiv 22(\text{mod}31)$	$m \equiv 0(\text{mod}31)$	$m \equiv \alpha(\text{mod}31), \alpha \in \{0, 1, \dots, 30\} \setminus \{0, 22\}$
63	–	$m \equiv 0(\text{mod}6)$	$m \equiv \alpha(\text{mod}6), \alpha \in \{1, 2, \dots, 5\}$

Таблица 3. Матрица полиномиальных кодов с числом проверочных символов $k = 1, \dots, 4$

m	N														
	$k = 1$		$k = 2$		$k = 3$				$k = 4$						
	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															

полиномом тому или иному классу при заданном значении числа информационных символов. Такая матрица для кодов с числом проверочных символов $k \leq 4$ представлена в табл. 3. В ней клетки на пересечении конкретной строки и столбца окрашены в три различных цвета: темно-серый цвет использован для указания на принадлежность кодера полиномиального кода классу SD-устройств, светло-серый цвет говорит о принадлежности кодера полиномиального кода классу SQD-устройств, белый цвет говорит о том, что кодер является SD/SQD-устройством.

4. Использование полиномиальных кодов при синтезе самопроверяемых дискретных устройств с контролем вычислений по нескольким диагностическим признакам

4.1. Структура организации схемы встроенного контроля

Рассмотрим особенности использования полиномиальных кодов при синтезе самопроверяемых дискретных устройств с контролем вычислений по нескольким диагностическим признакам. Подход к организации СВК сразу же по нескольким диагностическим признакам использовался ранее в [22] при организации контроля вычислений устройств с применением принципа логической коррекции сигналов и свойств равновесных кодов вида « r из $2r$ », где r – вес кодового слова. Также он исследовался в [16] при контроле вычислений по нескольким диагностическим признакам с применением хорошо известных кодов Хэмминга. Структура организации контроля вычислений по нескольким диагностическим признакам изображена на рис. 3.

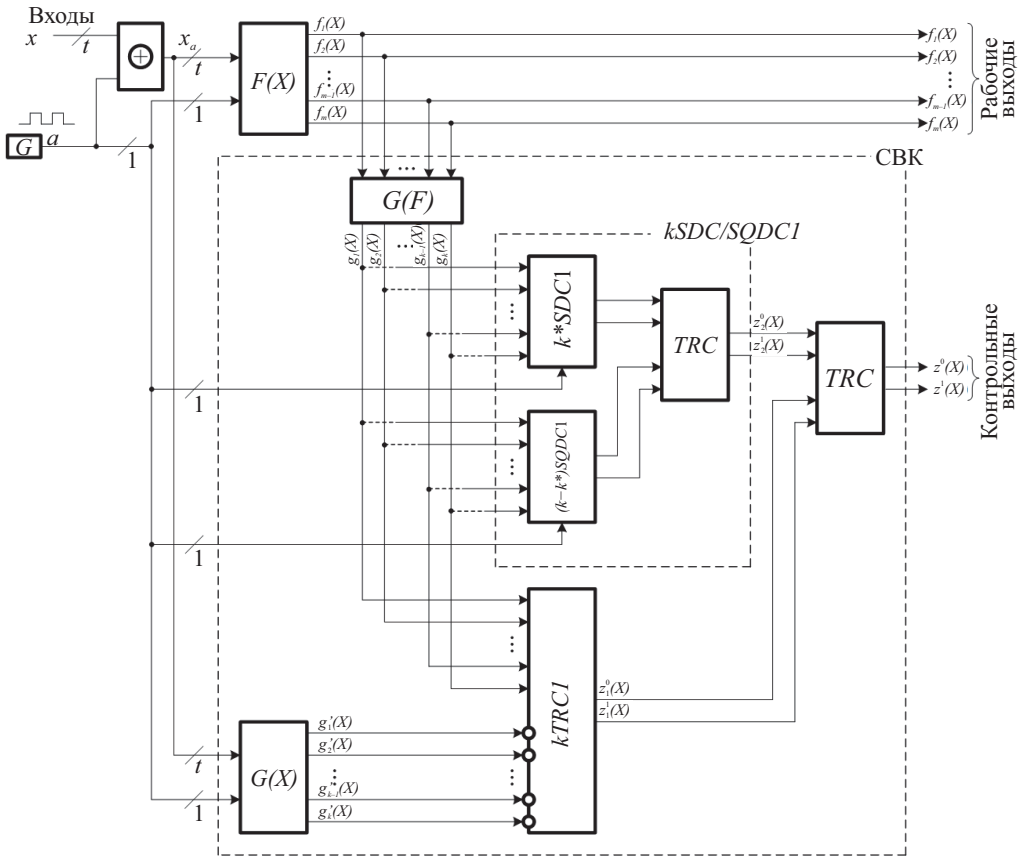


Рис. 3. Структура организации СВК для самодвойственных дискретных устройств по нескольким диагностическим признакам.

В этой структуре исходным дискретным устройством (объектом диагностирования) является блок $F(X)$, вычисляющий значения булевых функций $f_1(X), f_2(X), \dots, f_{m-1}(X), f_m(X)$ при подаче на входы наборов значений аргументов $\langle X \rangle = \langle x_t x_{t-1} \dots x_2 x_1 \rangle$. Объект диагностирования является SD-устройством. Для контроля вычислений на выходах блока $F(X)$ используется специализированная СВК по нескольким диагностическим признакам. СВК организуется на основе полиномиальных кодов. Выходы объекта диагностирования отождествляются с информационным вектором с m разрядами. Блок $G(F)$ по значениям функций $f_1(X), f_2(X), \dots, f_{m-1}(X), f_m(X)$ формирует значения проверочных символов $g_1(X), g_2(X), \dots, g_{k-1}(X), g_k(X)$ кодовых слов для заданного полиномиального кода, генерируя тем самым конкретный контрольный вектор. С другой стороны, блок $G(X)$ формирует значения проверочных символов $g'_1(X), g'_2(X), \dots, g'_{k-1}(X), g'_k(X)$ кодовых слов этого же полиномиального кода при подаче на входы наборов значений аргументов $\langle X \rangle = \langle x_t x_{t-1} \dots x_2 x_1 \rangle$. Это позволяет сформировать под схему контроля вычислений по признаку принадлежности формируемых в СВК кодовых слов выбранному полиномиальному коду. Так, одноименные выходы блоков $G(F)$ и $G(X)$ подключены ко входам самопроверяемого компаратора $kTRC1$, сравнивающего k пар сигналов на входах и формирующей единственную пару контрольных сигналов $z_1^0(X)$ и $z_1^1(X)$. Самопроверяемый компаратор реализуется в парафазной логике и является схемой сжатия парафазных сигналов, синтезируемой на основе $k - 1$ элементарных модулей сжатия парафазных сигналов TRC (*two-rail checker*) [23]. Поэтому сигналы от одного из блоков $G(F)$ или $G(X)$ предварительно инвертируются (в структуре рис. 3 инвертированы сигналы от блока $G(X)$). Вообще при синтезе блока $G(X)$ можно сразу же получать инвертированные сигналы $g_1(X), g_2(X), \dots, g_{k-1}(X), g_k(X)$, тогда инверторы на входах компаратора не потребуются. Ошибки в вычислениях на выходах блоков $G(F)$ и $G(X)$, а также в структуре самого компаратора приводят к нарушению парафазности сигнала на выходах $z_1^0(X)$ и $z_1^1(X)$, что косвенно свидетельствует о наличии неисправностей в структуре всей системы. Структура организации контроля вычислений именно таким образом – с дополнением информационных символов контрольными – довольно хорошо изучена [24].

Так как выходы блока $G(F)$ описываются самодвойственными и/или самоквазидвойственными функциями, то для контроля вычислений используется и под схема контроля самодвойственности/самоквазидвойственности сигналов $g_1(X), g_2(X), \dots, g_{k-1}(X), g_k(X)$. Для контроля самодвойственности сигналов устанавливается блок k^*SDC1 , осуществляющий контроль k^* самодвойственных сигналов и формирующий одну пару парафазных выходов. Аналогично для контроля самоквазидвойственности сигналов устанавливается блок $(k - k^*)SQDC1$, осуществляющий контроль $k - k^*$ самоквазидвойственных сигналов и формирующий одну пару парафазных выходов. Данные блоки строятся на основе схем сжатия самодвойственных и самоквазидвойственных сигналов [11] и тестеров самодвойственных и самоквазидвойствен-

ных сигналов [25]. Выходы обоих блоков k^*SDC1 и $(k - k^*)SQDC1$ подключены ко входам элементарного модуля сжатия парафазных сигналов TRC . Его выходы являются контрольными выходами $z_2^0(X)$ и $z_2^1(X)$ подсхемы контроля вычислений по признакам самодвойственности/самоквазидвойственности. Выходы подсхем контроля вычислений по различным диагностическим признакам подключаются ко входам одного модуля TRC . Его выходы $z^0(X)$ и $z^1(X)$ являются контрольными выходами СВК.

Принципы работы и настройки устройств контроля самодвойственных и самоквазидвойственных сигналов описаны авторами настоящей статьи в [25]. Для работы всей структуры используется временная избыточность и режим инвертирования данных [6]. Он подразумевает представление сигналов в виде импульсных последовательностей: логический ноль кодируется последовательностью 0101...01, а логическая единица – 1010...10. Это позволяет реализовать работу устройств при подаче на входы наборов значений аргументов парами: первый набор является рабочим, а второй – контрольным. При этом в паре наборы значений аргументов оказываются ортогональными по всем переменным (инверсными по всем переменным). Данная особенность подхода к организации контроля вычислений требует наличия генератора прямоугольных импульсов G , формирующего сигнал a со скважностью 2. Для формирования последовательностей сигналов, соответствующих входным переменным, используют каскад двухвходовых элементов сложения по модулю 2, на первые входы которых подают сигналы от конкретных входов, а на вторые – сигнал a . Этот же сигнал может подаваться на входы блоков $F(X)$ и $G(X)$ при реализации их из несамодвойственных структур с использованием разложения Шеннона по одной переменной. Сигнал a требуется и для работы тестеров самодвойственности/самоквазидвойственности.

В [16] показано, что структура контроля вычислений по нескольким диагностическим признакам обладает улучшенными показателями контролепригодности в части наблюдаемости ошибок на выходах объектов диагностирования за счет используемого режима работы и возможности разностороннего контроля вычислений.

Отметим следующую важную особенность структуры контроля вычислений по нескольким диагностическим признакам.

Теорема 6. В структуре СВК с контролем вычислений по нескольким диагностическим признакам обнаруживаются любые одиночные ошибки на выходах подсхемы контроля вычислений по признаку самодвойственности и/или самоквазидвойственности формируемых сигналов.

Доказательство теоремы приводится в Приложении.

Теорема 6 позволяет на практике упростить задачу построения СВК с покрытием любых ошибок на выходах объекта диагностирования за счет исключения из рассмотрения тех неисправностей объекта диагностирования, которые приводят к возникновению одиночных ошибок на их выходах.

4.2. Алгоритм выбора порождающего полинома

Общий алгоритм построения самопроверяемой СВК по нескольким диагностическим признакам с использованием полиномиальных кодов содержит следующие шаги.

Алгоритм 3. Правила синтеза схемы встроенного контроля по нескольким диагностическим признакам с использованием полиномиальных кодов:

1. Рассматривается структура комбинационного объекта диагностирования с m выходами и t входами.

2. Из полного множества Ω неисправностей из выбранной модели, например одиночных константных неисправностей выходов логических элементов, выделяется подмножество $\Omega' \subset \Omega$ только тех, которые возникают на выходах логических элементов, связанных путями с двумя и более выходами объекта диагностирования.

3. Если $\Omega' = \emptyset$ то контроль вычислений организуют с помощью полиномиального кода с порождающим полиномом $N = 3$, синтезируя СВК по нескольким диагностическим признакам. Если $\Omega' \neq \emptyset$, то переходят к рассмотрению возможностей покрытия ошибок с помощью полиномиальных кодов с порождающими полиномами $N = 3 + 2i$, $i \in \mathbb{N}$. При этом рассматриваются только такие полиномы, для которых число $k = \lfloor \log_2 N \rfloor < m$. Другими словами, максимальное значение числа равно $N = 2^{m-1} - 1$, а список содержит числа $N = 3 + 2i, i = \{1, \dots, 2^{m-2} - 2\}$.

4. Моделируется работа самодвойственного устройства при подаче на его входы всех пар $(A_i^j, B_{2^n-1-i}^j)$, где A_i^j и $B_{2^n-1-i}^j$ – значения j -ой функции, формируемой на j -ом выходе объекта диагностирования при поступлении входной комбинации, соответствующей десятичному эквиваленту i и $2^n - 1 - i$, наборов значений аргументов и действию одиночных константных неисправностей из множества Ω' , а также фиксируются вызываемые ими ошибки на выходах.

5. Образуется список порождающих полиномов $N = 3 + 2i$, $i = \{1, 2, \dots, 2^{m-2} - 2\}$, в котором они упорядочиваются по мере возрастания чисел N . Такое упорядочение необходимо для рассмотрения в первую очередь порождающих полиномов, позволяющих строить коды с наименьшим числом проверочных символов.

6. Выбирается первый член из образованного списка. Он удаляется из списка.

7. На полном множестве наборов значений аргументов определяются значения функций, задающих проверочные символы кодовых слов полиномиального кода с выбранным порождающим полиномом N .

8. Проверяется покрытие ошибок с помощью контроля вычислений по признаку самодвойственности и/или самоквазидвойственности вычисляемых контрольных функций. Из полного множества Ψ ошибок, формируемых на выходах схем, исключаются однократные ошибки, а также все ошибки с

кратностями $d \geq 2$, покрываемые при контроле вычислений по признаку самодвойственности и/или самоквазидвойственности. Образуется множество $\Psi' \subset \Psi$ необнаруживаемых по данному признаку ошибок.

9. Если $\Psi' = \emptyset$, то контроль вычислений организуют с помощью полиномиального кода с порождающим полиномом, соответствующим числу N . Если $\Psi' \neq \emptyset$, то осуществляют проверку покрытия ошибок из множества Ψ' с помощью выбранного полиномиального кода.

10. Если ошибки покрываются все, то синтезируют СВК по выбранному полиномиальному коду, если нет, то выбирают следующий порождающий полином из образованного списка, если он не пустой, и переходят к выполнению п. 4 настоящего алгоритма. Если список пустой, то оказывается невозможным синтез полностью самопроверяемой структуры на основе контроля вычислений по нескольким диагностическим признакам.

Рассмотрим пример организации контроля вычислений с использованием представленного алгоритма для самодвойственного дискретного устройства, описываемого следующей системой булевых функций:

$$\begin{aligned}
 f_1 &= \underbrace{x_1 x_2}_1 \vee \underbrace{x_1 \bar{x}_4}_2 \vee \underbrace{x_2 \bar{x}_4}_3; \\
 f_2 &= \underbrace{x_3 x_4}_4 \vee \underbrace{x_1 x_4}_5 \vee \underbrace{x_1 x_3}_6; \\
 f_3 &= \underbrace{x_1 \bar{x}_2 \bar{x}_3}_7 \vee \underbrace{\bar{x}_2 x_3 x_4}_8 \vee \underbrace{x_2 \bar{x}_3 x_4}_9 \vee \underbrace{x_1 x_2 x_3}_{10}; \\
 f_4 &= \underbrace{\bar{x}_2 x_3 x_4}_8 \vee \underbrace{x_2 \bar{x}_3 x_4}_9 \vee \underbrace{x_2 x_3 \bar{x}_4}_{11} \vee \underbrace{x_1 x_3 x_4}_{12} \vee \underbrace{x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_{13}; \\
 f_5 &= \underbrace{x_1 \bar{x}_4}_2 \vee \underbrace{x_1 \bar{x}_3}_{14} \vee \underbrace{x_1 \bar{x}_2}_{15} \vee \underbrace{\bar{x}_2 \bar{x}_3 \bar{x}_4}_{16}; \\
 f_6 &= \underbrace{x_1 \bar{x}_3 x_4}_{17} \vee \underbrace{x_1 x_3 \bar{x}_4}_{18} \vee \underbrace{x_1 \bar{x}_2 x_3}_{19} \vee \underbrace{\bar{x}_1 x_3 x_4}_{20} \vee \underbrace{x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_{21}.
 \end{aligned}$$

На первом шаге алгоритма рассмотрим конкретную реализацию устройства по приведенным выше формулам в виде двухуровневой структуры с учетом того, что инверсии аргументов будем переносить на входы элементов первого каскада схемы. При этом пронумеруем все неповторяющиеся конъюнкции в выражениях. Саму схему здесь приводить не будем ввиду ее громоздкости. Она содержит 21 многоходовой логический элемент И на первом каскаде и 6 многоходовых элементов ИЛИ на втором каскаде.

На втором и третьем шагах определим подмножество одиночных константных неисправностей логических элементов, вызывающих кратные ошибки на выходах устройства, а также возможности покрытия ошибок с помощью полиномиального кода с порождающим полиномом $N = 3$. Кратные ошибки на выходах устройства могут вызывать только неисправности элементов, реализующих конъюнкции с номерами 2 (могут одновременно искажаться значе-

ния функций f_1 и f_5), 8 (могут одновременно искажаться значения функций f_3 и f_4) и 9 (могут одновременно искажаться значения функций f_3 и f_4). Полиномиальный код с порождающим полиномом $N = 3$ не подходит для контроля вычислений в рассматриваемом случае.

На четвертом шаге моделируем работу устройства в условиях действия одиночных константных неисправностей и запоминаем полученные результаты, храня их в табличной форме.

Образуем список порождающих полиномов $N = 3 + 2i$, $i = \{1, \dots, 14\}$.

Выбираем полином, соответствующий номеру $N = 5$. Удаляем его из списка. Проверяем возможности покрытия ошибок на выходах устройства.

Расчеты показали, что полиномиальным кодом с выбранным порождающим полиномом покрываются все кратные ошибки, вызываемые неисправностями элементов, реализующих конъюнкции с номерами 2, 8 и 9, кроме пяти двукратных ошибок, вызываемых неисправностью вида константа 1 элемента, реализующего конъюнкцию с номером 2 (эти пять неисправностей не обнаруживаются ни при контроле самодвойственности функций (см. матрицу в табл. 3), описывающих проверочные символы кодовых слов, ни при контроле по принадлежности контрольного вектора данному полиномиальному коду). Мощность множества $|\Psi'| \neq 5$. А это значит, что требуется выбрать следующий полином из сокращенного списка. Этот полином соответствует числу $N = 7$.

Выполняем те же действия, что и ранее. Здесь уже, судя по матрице в табл. 3, осуществляется контроль вычислений по признаку принадлежности функций, описывающих проверочные символы кодовых слов, классу самоквазидвойственных, а также контроль принадлежности контрольного вектора кодовым словам выбранного полиномиального кода. Расчеты показали, что все кратные ошибки покрываются с помощью контроля вычислений только по одному из диагностических признаков. Далее синтезируем СВК по двум диагностическим признакам с использованием выбранного полиномиального кода.

Следует отметить, что контроль вычислений по признакам самодвойственности и самоквазидвойственности не приводит к существенному усложнению СВК относительно СВК с контролем вычислений по полиномиальному коду, так как структуры схем сжатия самодвойственных/самоквазидвойственных сигналов, а также тестеры данных сигналов довольно просты [11, 25]. Поэтому удается синтезировать менее избыточные структуры, чем, например, при дублировании, при этом получая улучшенные показатели контролепригодности в части наблюдаемости ошибок на выходах СВК.

В качестве достоинств алгоритма 3 следует выделить то, что за счет выбора порождающего полинома от малых значений $N = 3$ до $N = 2^{m-2} - 2$ можно добиться полного покрытия ошибок с помощью кодов с малыми значениями k , что влияет и на показатели эффективности реализации СВК. Недостатком же является необходимость моделирования неисправностей и

фиксации ошибок на выходах объектов диагностирования, что ограничивает возможности применения метода при достаточно большом числе входов. При этом применимость метода для организации СВК для тех или иных дискретных устройств также определяется и сложностью вычисляемых на их выходах булевых функций [26]. Трудоемкость алгоритма построения самопроверяемого устройства может быть асимптотически оценена величиной $2^{O(t)}$, т.е. задача решается за экспоненциальное время с линейной экспонентой. Тем не менее всегда можно использовать методы декомпозиции при синтезе СВК. Кроме того, учет структур объектов диагностирования является одним из действенных методов синтеза СВК, когда не используются стандартные методы, основанные на внесении модульной избыточности.

Также следует отметить, что полиномиальные коды обнаруживают большое количество ошибок при высоких степенях полиномов $M(y)$ и $G(y)$, что широко используется при контроле вычислений и передаче данных. Однако в рассматриваемом приложении полиномиальных кодов степень полинома $M(y)$ определяется числом выходов объекта диагностирования, а вот степени образующих полиномов подбираются с учетом покрытия ошибок, вызываемых неисправностями в структурах объектов диагностирования. По сути, чем раньше будет найдено число N по алгоритму 3, тем меньше будет степень у остатка $R(y)$ и тем меньшую структурную избыточность будет иметь конечное самопроверяемое дискретное устройство.

5. Заключение

Предложенный в статье метод синтеза полностью самопроверяемых дискретных устройств позволяет наделять их не только повышенными показателями контролепригодности относительно широко известных методов, но и обеспечивать обнаружение некоторого подмножества ошибок в вычислениях при условии отсутствия изменения входных воздействий. Это связано с представлением сигналов в дискретных устройствах в виде двоичных последовательностей.

Разработанный алгоритм синтеза СВК на основе свойств полиномиальных кодов подразумевает последовательный перебор порождающих полиномов от самого малого, соответствующего числу $N = 3$, к соответствующему числу $N = 2^{m-2} - 2$. При малых значениях $m < 6$ количество порождающих полиномов, удовлетворяющих этому условию, будет небольшим, что снизит потенциальные возможности построения полностью самопроверяемого устройства. Поэтому наиболее эффективным рассматриваемый подход должен оказаться для устройств с относительно большим числом выходов $m \geq 6$. С ростом значения N до предельного числа количество проверяющих символов в кодовых словах меняется от величины $k = 1$ к величине $k = m - 1$, что обеспечивает и увеличение мощности множества покрываемых полиномиальными кодами ошибок. Поэтому алгоритм будет эффективным при построении самопроверяемых устройств практически для любых структур объектов диагностиро-

вания с $m \geq 6$. Тем не менее для каждого конкретного дискретного устройства потребуется сравнение показателей их самопроверяемой реализации с показателями самопроверяемых реализаций, получаемых по иным методам, например по методу дублирования.

Возможные направления дальнейших исследований могут быть связаны с постановкой разнообразных экспериментов с тестовыми дискретными устройствами и оценкой показателей эффективности их самопроверяемых реализаций, разработкой иных алгоритмов выбора порождающих полиномов с учетом особенностей структур объектов диагностирования, учетом особенностей иных линейных кодов при синтезе СВК на основе нескольких диагностических признаков, а также с изучением особенностей построения подобных устройств на различной, в том числе программируемой, элементной базе.

Использование при построении самопроверяемых дискретных устройств контроля вычислений по нескольким диагностическим признакам представляется авторам перспективным и до конца не исследованным подходом к реализации составляющих высоконадежных систем управления.

ПРИЛОЖЕНИЕ

Доказательство теоремы 1.

Необходимость. Каждая контрольная функция описывается выражением вида $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}$, $j \in \{1, k\}$, $i_1, i_2, \dots, i_q \in \{1, \dots, m\}$. Для получения выражений, описывающих каждую контрольную функцию, необходимо установить, какие из информационных символов являются значимыми. При рассмотрении всех информационных векторов с весом $r = 1$ последовательно определяются значения проверочных символов при подстановке единственного значимого информационного символа. Если значение функции $g_j = 0$, то соответствующий информационный символ не влияет на конкретную контрольную функцию, иначе, при $g_j = 1$, он включается в логическое выражение, описывающее контрольную функцию.

Достаточность. При подстановке информационного вектора с весом $r = 1$ значение функции g_j будет определяться только значением значащего информационного символа, следовательно, он будет входить в формулу описания соответствующей контрольной функции. При рассмотрении всех информационных векторов с весом $r = 1$ будут определены все суммируемые в каждой функции g_j аргументы. Это позволяет определить формулы, описывающие проверочные символы полиномиальных кодов.

Доказательство теоремы 6.

Все функции кодера полиномиального кода являются линейными: $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}$, $j \in \{1, k\}$, $i_1, i_2, \dots, i_q \in \{1, \dots, m\}$. Рассмотрим одну из них. Положим, происходит одиночная ошибка на выходе $f_q \in \{1, \dots, m\}$. Тогда значение функции f_p искажается, т.е. становится равным $f_p = 0$ или $f_p = 1$. Выражение для контрольной функции выглядело так: $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_p \oplus \dots \oplus f_{i_q}$. При искажении оно получило вид: $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus$

$\oplus(f_p = 0) \oplus \dots \oplus f_{i_q} = \frac{f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}}{\oplus(f_p = 1) \oplus \dots \oplus f_{i_q} = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}}$ или же $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus$
 $\oplus(f_p = 1) \oplus \dots \oplus f_{i_q} = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}$. В первом случае имеем изменение четности числа аргументов исходной функции и зависимости ее от всех переменных, кроме f_p . Во втором случае четность числа аргументов также меняется, но при этом еще реализуется инверсия функции сложения по модулю 2, сохраняющая все признаки принадлежности к классам самодвойственных и/или самоквазидвойственных функций. Другими словами, функция поменяла свой вид с самоквазидвойственной/самодвойственной на самодвойственную/самоквазидвойственную. Раз вид функции поменялся, то рассматриваемая ошибка обнаружится в подсхеме контроля вычислений по выбранному признаку, так как он изменится при одиночной ошибке.

СПИСОК ЛИТЕРАТУРЫ

1. *Bennetts R.G.* Design of Testable Logic Circuits. Addison-Wesley Publishing Company, 1984. 164 p.
2. *Согомонян Е.С., Слабаков Е.В.* Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
3. *Drozd A., Kharchenko V., Antoshchuk S., et. al.* Checkability of the Digital Components in Safety-Critical Systems: Problems and Solutions // Proceedings of 9th IEEE East-West Design and Test Symposium (EWDTS'2011), Sevastopol, Ukraine, 2011. P. 411-416. <https://doi.org/10.1109/EWDTS.2011.6116606>
4. *Дрозд А.В., Харченко В.С., Антощук С.Г. и др.* Рабочее диагностирование безопасных информационно-управляющих систем. Под ред. А.В. Дрозда, В.С. Харченко. Харьков: Нац. аэрокосм. ун-т им. Н.Е. Жуковского «ХАИ», 2012. 614 с.
5. *Сапожников В.В., Сапожников Вл.В.* Самопроверяемые дискретные устройства. СПб.: Энергоатомиздат, 1992. 224 с.
6. *Аксёнова Г.П.* Восстановление в дублированных устройствах методом инвертирования данных // АИТ. 1987. № 10. С. 144–153.
7. *Шальто А.А.* Модули, универсальные в классе самодвойственных функций и в «близких» к ним классах // Известия Российской Академии наук наук. Теория и системы управления. 2001. № 5. С. 110–120.
8. *Saposhnikov Vl.V., Dmitriev A., Goessel M., Saposhnikov V.V.* Self-Dual Parity Checking a New Method for on Line Testing // Proceedings of 14th IEEE VLSI Test Symposium, USA, Princeton, 1996. P. 162–168. <https://doi.org/10.1109/VTEST.1996.510852>
9. *Гессель М., Мошанин В.И., Сапожников В.В., Сапожников Вл.В.* Обнаружение неисправностей в самопроверяемых комбинационных схемах с использованием свойств самодвойственных функций // АИТ. 1997. № 12. С. 193–200.
10. *Гессель М., Дмитриев А.В., Сапожников В.В., Сапожников Вл.В.* Самотестируемая структура для функционального обнаружения отказов в комбинационных схемах // АИТ. 1999. № 11. С. 162–174.
11. *Dmitriev A., Saposhnikov V., Saposhnikov V., Goessel M.* New Self-Dual Circuits for Error Detection and Testing // VLSI Design. 2000. Vol. 11. Issue 1. P. 1–21. <https://doi.org/10.1155/2000/84720>

12. Сапожников В.В., Сапожников Вл.В., Гессель М. Самодвойственные дискретные устройства. СПб.: Энергоатомиздат (Санкт-Петербургское отделение), 2001. 331 с.
13. Сапожников В.В., Сапожников Вл.В., Валиев Р.Ш. Синтез самодвойственных дискретных систем. СПб.: Элмор, 2006, 220 с.
14. Goessel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science+Business Media B.V., 2008. 184 p.
15. Яблонский С.В. Введение в дискретную математику. Под ред. В.А. Садовничева. М.: Высш. шк. 2003. 384 с.
16. Ефанов Д.В., Погодина Т.С. Исследование свойств самодвойственных комбинационных устройств с контролем вычислений на основе кодов Хэмминга // Информатика и автоматизация. 2023. Том 22. № 2. С. 349–392. <https://doi.org/10.15622/ia.22.2.5>
17. Ефанов Д.В. Особенности реализации самопроверяемых структур на основе метода инвертирования данных и линейных кодов // Вест. Том. гос. ун-та. Управление, вычислительная техника и информатика. 2023. № 65. С. 126–138. <https://doi.org/10.17223/19988605/65/13>
18. Sellers F.F., Hsiao M.-Y., Bearson L.W. Error Detecting Logic for Digital Computers. New York: McGraw-Hill, 1968. XXI+295 p.
19. Efanov D., Plotnikov D., Sapozhnikov V., Sapozhnikov Vl., Abdullaev R. Experimental Studies of Polynomial Codes in Concurrent Error Detection Systems of Combinational Logical Circuits // Proceedings of 16th IEEE East-West Design and Test Symposium (EWDTS'2018), Kazan, Russia, September 14–17, 2018. P. 184–190. <https://doi.org/10.1109/EWDTS.2018.8524684>
20. Abdullaev R., Efanov D. Polynomial Codes Properties Application in Concurrent Error-Detection Systems of Combinational Logic Devices // Proceedings of 19th IEEE East-West Design and Test Symposium (EWDTS'2021), Batumi, Georgia, September 10–13, 2021, pp. 40–46. <https://doi.org/10.1109/EWDTS52692.2021.9580992>
21. Ефанов Д.В., Пивоваров Д.В. Организация контроля вычислений на выходах самодвойственных цифровых устройств с применением циклических избыточных кодов // Матер. XIV Всероссийского совещания по проблемам управления (ВСПУ-2024) под ред. акад. Д.А. Новикова, М., 17–20 июня 2024 г., с. 2395–2399.
22. Efanov D.V., Pivovarov D.V. The Hybrid Structure of a Self-Dual Built-In Control Circuit for Combinational Devices with Pre-Compression of Signals and Checking of Calculations by Two Diagnostic Parameters // Proceedings of 19th IEEE East-West Design and Test Symposium (EWDTS'2021), Batumi, Georgia, September 10–13, 2021, с. 200–206. <https://doi.org/10.1109/EWDTS52692.2021.9581019>
23. Lala P.K. Self-Checking and Fault-Tolerant Digital Design. San Francisco: Morgan Kaufmann Publishers, 2001. 216 p.
24. Sahana A.R., Chiraag V., Suresh G., Thejaswini P., Nandi S. Application of Error Detection and Correction Techniques to Self-Checking VLSI Systems: An Overview // Proceedings of 2023 IEEE Guwahati Subsection Conference (GCON), Guwahati, 2023. <https://doi.org/10.1109/GCON58516.2023.10183449>

25. Ефанов Д.В., Пивоваров Д.В. Тестеры самодвойственных и «близких» к ним сигналов // Известия высших учебных заведений. Приборостроение. 2024. Т. 67. № 1. С. 5–19. <https://doi.org/10.17586/0021-3454-2024-67-1-5-19>
26. Коршунов А.Д. Сложность вычислений булевых функций // Успехи математических наук. 2012. Том 67. № 1 (403). С. 97–168. <https://doi.org/10.4213/rm9459>

Статья представлена к публикации членом редколлегии Л.Ю. Филимонюком.

Поступила в редакцию 28.11.2024

После доработки 17.03.2025

Принята к публикации 27.03.2025