УДК 681.518.5+004.052.32

© 2021 г. **Д.В. Ефанов**, д-р техн. наук

(Российский университет транспорта, Москва),

Д.В. Пивоваров, канд. техн. наук

(Петербургский государственный университет путей сообщения

Императора Александра I)

ФУНКЦИОНАЛЬНЫЙ ПОДХОД К СИНТЕЗУ СХЕМ ВСТРОЕННОГО КОНТРОЛЯ НА ОСНОВЕ ЛОГИЧЕСКОГО ДОПОЛНЕНИЯ И ИСПОЛЬЗОВАНИЯ РАВНОВЕСНОГО КОДА «2 ИЗ 5»

Описан метод синтеза самопроверяемых схем встроенного контроля на основе логического дополнения и использования равновесного кода «2 из 5». Метод отличается от известных тем, что подразумевает не анализ работы устройства на каждом входном наборе, а использование функциональной зависимости между сигналами с рабочих и контрольных выходов. Функциональная зависимость устанавливается с учетом требований по формированию кодовых слов кода «2 из 5» и полной проверки тестера, а также с учетом формирования полного множества тестовых комбинаций для элементов преобразования в схеме встроенного контроля. Представлена методика получения функциональной зависимости между контрольными выходами и рабочими выходами объекта диагностирования. Дан один из вариантов доопределения значений контрольных функций дополнения, а также сама функциональная зависимость. Отмечается, что число вариантов доопределения значений контрольных функций дополнения велико, что позволяет для любого объекта диагностирования иметь множество вариантов синтеза самопроверяемой схемы встроенного контроля по представленному методу. Использование кода «2 из 5» при синтезе схем встроенного контроля по методу логического дополнения является эффективным вариантом организации самопроверяемых цифровых устройств автоматики и вычислительной техники.

Ключевые слова: схема встроенного контроля, контроль вычислений, контроль принадлежности заданному избыточному коду, равновесный код «2 из 5», полностью самопроверяемая структура, условия синтеза полностью самопроверяемой структуры.

Введение

Методы теории информации и кодирования широко используются при построении самопроверяемых и отказоустойчивых устройств автоматики и вычислительной техники [1 – 5]. Одним из их приложений является организация схем встроенного контроля (СВК) комбинационных логических устройств [6].

Устройство автоматики снабжается СВК для определения фактов возникновения ошибок на его выходах в процессе функционирования. Другими словами, СВК предназначена для контроля корректности вычислений объектом диагностирования. Универсальным подходом к синтезу СВК считается использование двойной модульной избыточности (дублирования) со сравнением результатов вычислений самопроверяемым компаратором [7]. Дублирование позволяет обнаруживать любые сочетания искажений на выходах объекта диагностирования. Однако данный подход не позволяет учесть индивидуальные особенности объекта диагностирования, – например, допустимые режимы функционирования или же внутреннюю конфигурацию элементов и связей между ними, входами устройства и его выходами. Учет особенностей объекта диагностирования позволяет сокращать вносимую структурную избыточность в конечное самопроверяемое цифровое устройство. Для этого применяются методы, основанные на использовании равномерных избыточных кодов – как разделимых, так и неразделимых [8 – 10].

Существуют два основных подхода к организации СВК на основе избыточного кодирования. Первый состоит в том, что в СВК кодовый вектор, формируемый на выходах объекта диагностирования, дополняется вектором контрольных функций таким образом, чтобы совместно они давали кодовое слово, принадлежащее заранее выбранному равномерному разделимому или неразделимому коду [11]. Контроль соответствия формируемого кодового слова в СВК заранее выбранному коду осуществляется с помощью самопроверяемого тестера [12]. Контрольные функции вычисляются специальным блоком контрольной логики. Для конкретного избыточного кода данный блок имеет единственную реализацию. Второй подход к организации СВК связан с применением в ней блока коррекции значений рабочих функций. В литературе он получил название логического дополнения* [13]. Значения рабочих функций в блоке коррекции преобразуются таким образом, чтобы на

* Строго говоря, правильнее его называть *методом логической коррекции*, а не *дополнения*, однако будем далее использовать устоявшееся название.

его выходах формировалось кодовое слово заранее выбранного кода. Для этого также используется блок контрольной логики, вычисляющий специальные функции дополнения. Контроль принадлежности кодового слова выбранному коду также осуществляется с помощью самопроверяемого тестера. В отличие от первого подхода использование логического дополнения позволяет даже для одного выбранного кода синтезировать большое количество блоков контрольной логики и выбрать наименее сложный из них. Этим можно добиться и значительного сокращения структурной избыточности конечного устройства и обеспечения свойства его самопроверяемости даже тогда, когда этого невозможно добиться при использовании дублирования и первого подхода на основе избыточного кодирования [14, 15]. При синтезе СВК по методу логического дополнения может быть использован не только диагностический признак принадлежности формируемого кодового вектора заранее выбранному коду, но и признак соответствия вычисляемых функций заранее определенному классу функций алгебры логики, – например, классу самодвойственных функций [16].

Одной из первых работ, в которых описан метод логического дополнения, является статья [17], где предложено использовать для организации СВК метод самодвойственного паритета. Данная структура является развитием структуры, предложенной в [18]. В дальнейшем метод логического дополнения развивался в двух направлениях: при организации самодвойственных цифровых вычислительных устройств с импульсным представлением сигналов и при синтезе структур с традиционным представлением сигналов [19]. Основное внимание было уделено применению неразделимых кодов для организации СВК по методу логического дополнения [14, 15, 19 – 21]. Гораздо меньшее внимание было уделено для этих целей [22] применению делимых кодов. Такая особенность связана с работой блока коррекции: в нем преобразуются значения со всех выходов объекта диагностирования, а значит, ошибка на выходе объекта диагностирования может привести к искажению значений разрядов и в информационном и в контрольном векторах на выходах блока коррекции. При традиционном подходе к организации СВК ошибки на выходах объекта диагностирования могут приводить только к ошибкам в информационном векторе, что определяет преимущества применения данного подхода при использовании делимых кодов [23].

Наибольшее внимание в исследованиях метода логического дополнения для синтеза СВК, как отмечено выше, уделено неразделимым равновесным кодам. Гораздо меньшее внимание уделено другим неразделимым кодам для синтеза СВК по этому методу [24]. Задачей абстрактного синтеза

СВК является получение выражений, описывающих функции выходов блока контрольной логики. При использовании равновесных кодов также существует два основных подхода к синтезу СВК – эвристический и функциональный. Эвристический подход связан с анализом работы СВК на каждой входной комбинации и доопределением значений контрольных функций дополнения с проверкой формирования тестовых комбинаций для элементов преобразования и тестера выбранного кода [14, 15]. Функциональный подход связан с установлением функциональной зависимости между контрольными функциями дополнения и рабочими функциями объекта диагностирования [25, 26].

При синтезе СВК по методу логического дополнения до разделимых кодов можно применять как контроль всех выходов объекта диагностирования по одному коду, так и контроль групп выходов. При этом, как показывают исследования, наиболее эффективным является использование как раз группового контроля, поскольку при этом можно использовать коды с меньшей мощностью кодовых слов и имеющих простые структуры тестеров, требующие относительно небольшого числа комбинаций для полной проверки. К таким кодам относятся равновесные коды «1 из 3», «1 из 4», «2 из 4», «1 из 5», «2 из 5». Использование первых четырех кодов при синтезе СВК по методу логического дополнения хорошо проработано [14, 15, 19 – 21, 25 – 27], а по использованию последнего кода имеется только работа [28] с эвристическим подходом к получению функций логического дополнения.

Постановка задачи

Целью настоящей работы является исследование особенностей синтеза СВК по методу логического дополнения до равновесного кода «2 из 5» (2/5-кода) с применением функционального подхода к получению значений на выходах блока контрольной логики.

Ставится задача разработки метода синтеза СВК с применением логического дополнения рабочих функций объекта диагностирования до кодовых слов, принадлежащих 2/5-коду, для группы из пяти выходов устройства (для базовой структуры), в которой между выходами блока контрольной логики и основной логики будет установлена функциональная зависимость.

Особенности организации схемы встроенного контроля по методу логического дополнения до 2/5-кода

При синтезе СВК по методу логического дополнения применяется базовая структура, приведенная на рис. 1.

Рис. 1. Структура СВК на основе метода логического дополнения до 2/5-кода.

Данная структура строится для группы из пяти выходов объекта диагностирования. Соответственно все выходы объекта диагностирования при синтезе СВК покрываются группами по пять выходов в каждой. Для каждой группы строится своя СВК, а затем производится их совместная оптимизация и сжатие отдельных выходов каждой СВК на входах специализированного компаратора.

В базовой структуре блок $F(x)$ является объектом диагностирования (подсхемой исходного устройства) и вычисляет рабочие функции f_1, f_2, f_3, f_4 и f_5 . Он снабжается специализированной СВК в составе блока контрольной логики $G(x)$, блока коррекции сигналов (БКС) и самопроверяемого тестера. БКС предназначен для корректировки значений функций, формирующихся на рабочих выходах объекта диагностирования. В нем преобразуются любые три из пяти выходов объекта диагностирования, этого достаточно для преобразования любого пятиразрядного кодового вектора в кодовое слово равновесного 2/5-кода. Отметим, что преобразовываться могут и все пять выходов, однако в представленной на рисунке структуре минимизировано число элементов преобразования (меньшее число элементов преобразования позволяет упростить задачу обеспечения их тестирования в процессе функционирования СВК). Для преобразования используются двухвходовые элементы сложения по модулю $M = 2$ (элементы XOR), на первые выходы которых поступают сигналы от блока $F(x)$, а на вторые – от блока $G(x)$. Преобразования осуществляются по формулам: $h_3 = f_3 \oplus g_3$, $h_4 = f_4 \oplus g_4$, $h_5 = f_5 \oplus g_5$. Блок

$G(x)$ соответственно вычисляет значения контрольных функций g_3 , g_4 и g_5 . На выходах блока коррекции сигналов формируются функции h_1 , h_2 , h_3 , h_4 и h_5 . Принадлежность данного кодового слова равновесному 2/5-коду контролируется схемой самопроверяемого тестера [12]. Для фиксации корректности вычислений он снабжается двумя выходами z^0 и z^1 , на которых формируется парафазный сигнал $\langle 01 \rangle$ или $\langle 10 \rangle$ при исправной работе тестера и поступлении кодового слова 2/5-кода на входы и непарафазный сигнал $\langle 00 \rangle$ или $\langle 11 \rangle$ – в случае, если присутствуют ошибки в вычислениях самого тестера или на его входах.

Равновесный 2/5-код образован десятью комбинациями из множества $\{00011, 00101, 00110, 01001, 01010, 01100, 10001, 10010, 10100, 11000\}$. Для полной проверки тестера в общем случае потребуется подача на его входы всех комбинаций из данного множества. Кроме того, чтобы СВК была самопроверяемой, требуется, чтобы на входы элементов XOR_3 , XOR_4 и XOR_5 поступали хотя бы по разу тестовые комбинации из множества $\{00, 01, 10, 11\}$ [29]. Кроме обозначенных условий для обеспечения полной самопроверяемости структуры рисунка, необходимо обеспечить проверяемость блоков $F(x)$ и $G(x)$, а также выполнить условия по невозможности компенсации ошибок на входах элементов XOR_3 , XOR_4 и XOR_5 , приведенные в [30].

С учетом сформулированных условий и строится полностью самопроверяемая структура, изображенная на рис. 1. Это можно сделать эвристически, путем доопределения значений контрольных функций g_3 , g_4 и g_5 на каждой входной комбинации по методике, описанной в [28], либо установив функциональную зависимость между контрольными и рабочими функциями [25, 26]. Рассмотрим методику получения функциональной зависимости между контрольными и рабочими функциями при организации СВК по логическому дополнению с использованием 2/5-кода.

Методика получения функциональной зависимости между выходами объекта диагностирования и блока логического дополнения

Для установления функциональной зависимости между выходами блока контрольной логики и блока основной логики необходимо доопределить все значения функций g_3 , g_4 и g_5 на всех возможных значениях функций f_1, f_2, f_3, f_4, f_5 . Это удобно сделать с помощью табличной формы задания. Функции g_3 , g_4 и g_5 предназначены для трансформации функций f_1, f_2, f_3, f_4, f_5 в функции h_1, h_2, h_3, h_4, h_5 , формирующие кодовые слова 2/5-кода.

В табл. 1 представлены частично определенные функции с учетом условия принадлежности формируемых векторов 2/5-коду.

Таблица 1

№	$f_1f_2f_3f_4f_5$	$g_3g_4g_5$	$h_1h_2h_3h_4h_5$
0	00000		
1	00001		
2	00010		
3	00011		
4	00100		
5	00101		
6	00110		
7	00111		
8	01000		
9	01001		
10	01010		
11	01011		
12	01100		
13	01101		
14	01110		
15	01111		
16	10000		
17	10001		
18	10010		
19	10011		
20	10100		
21	10101		
22	10110		
23	10111		
24	11000	000	11000
25	11001	001	11000
26	11010	010	11000
27	11011	011	11000
28	11100	100	11000
29	11101	101	11000
30	11110	110	11000
31	11111	111	11000

На номерах рабочих векторов № 24 – 31 (они соответствуют двоичным представлениям выходных векторов) зависимость определена однозначно, поскольку только на данных наборах при отсутствии преобразования значений функций f_1 и f_2 возможно формирование кодового слова 2/5-кода <11000>. На остальных наборах требуется доопределение значений функций g_3 , g_4 и g_5 . Это делается с учетом двух требований:

1) для каждого номера в таблице должно осуществляться преобразование вектора рабочих функций в вектор равновесного 2/5-кода;

2) должна обеспечиваться подача на входы элементов преобразования хотя бы по одному разу каждой тестовой комбинации.

С учетом этого в табл. 2 приведены все возможные варианты доопре-

деления значений функций дополнения.

Таблица 2

№	$f_1f_2f_3f_4f_5$	$g_3g_4g_5$	f_3g_3	f_4g_4	f_5g_5	$h_1h_2h_3h_4h_5$
0	00000	011/101/110	00/01/01	01/00/01	01/01/00	00011/00101/00110
1	00001	010/100/111	00/01/01	01/00/01	10/10/11	00011/00101/00110
2	00010	001/111/100	00/01/01	10/11/10	01/01/00	00011/00101/00110
3	00011	000/110/101	00/01/01	10/11/10	10/10/11	00011/00101/00110
4	00100	111/001/010	11/10/10	01/00/01	01/01/00	00011/00101/00110
5	00101	110/000/011	11/10/10	01/00/01	11/10/11	00011/00101/00110
6	00110	101/011/000	11/10/10	10/11/10	01/01/00	00011/00101/00110
7	00111	100/010/001	11/10/10	10/11/10	10/10/11	00011/00101/00110
8	01000	001/010/100	00/01/01	00/01/00	01/01/00	01001/01010/01100
9	01001	000/011/101	00/01/01	00/01/00	10/10/11	01001/01010/01100
10	01010	011/000/110	00/01/01	11/10/11	01/01/00	01001/01010/01100
11	01011	010/001/111	00/01/01	11/10/11	10/10/11	01001/01010/01100
12	01100	101/110/000	11/10/10	00/01/00	01/01/00	01001/01010/01100
13	01101	100/111/001	11/10/10	00/01/00	11/10/11	01001/01010/01100
14	01110	111/100/010	11/10/10	11/10/11	01/01/00	01001/01010/01100
15	01111	110/101/011	11/10/10	11/10/11	10/10/11	01001/01010/01100
16	10000	001/010/100	00/00/01	00/01/00	01/01/00	10001/10010/10100
17	10001	000/011/101	00/00/01	00/01/00	10/10/11	10001/10010/10100
18	10010	011/000/110	00/00/01	11/10/11	01/01/00	10001/10010/10100
19	10011	010/001/111	00/00/01	11/10/11	10/10/11	10001/10010/10100
20	10100	101/110/000	11/00/10	00/01/00	01/01/00	10001/10010/10100
21	10101	100/111/001	11/00/10	00/01/00	10/10/11	10001/10010/10100
22	10110	111/100/010	11/00/10	11/10/11	01/01/00	10001/10010/10100
23	10111	110/101/011	11/00/10	11/10/11	10/10/11	10001/10010/10100

На каждом наборе возможно требуемое преобразование тремя различными способами. Всего нужно доопределить значения функций g_3 , g_4 и g_5 на 24 номерах. Число вариантов доопределения весома и составляет $3^{24}=282\,429\,536\,481$. Поэтому просто поясним методику заполнения исходной таблицы.

На наборах № 0 – 7 могут быть сформированы только три кодовых слова 2/5-кода: <00011>, <00101> и <00110>.

На наборах № 8 – 15 аналогично могут быть сформированы только три кодовых слова 2/5-кода: <01001>, <01010> и <01100>.

На наборах № 16 – 23 – кодовые слова <10001>, <10010> и <10100>. Соответственно на каждой «восьмерке» рабочих комбинаций требуется такое доопределение, которое хотя бы по разу дает три соответствующих кодовых слова 2/5-кода.

Кроме того, на всех 24 комбинациях нужно, чтобы хотя бы по разу сформировались тестовые комбинации <01> и <10> для элементов XOR_3 , XOR_4 и XOR_5 (тестовые комбинации <00> и <11> формируются на наборах с № 24 – 31 для каждого элемента преобразования).

Сформулированные выше условия не являются столь жесткими. В

табл. 3 выделен один из вариантов заполнения табл. 1 (доопределения значений функций дополнения), а также указаны тестовые комбинации для всех элементов СВК.

Таблица 3

№	$f_1f_2f_3f_4f_5$	$g_3g_4g_5$			f_3g_3			f_4g_4			f_5g_5			$h_1h_2h_3h_4h_5$		
		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
0	00000	011	101	110	00	01	01	01	00	01	01	01	00	00011	00101	00110
1	00001	010	100	111	00	01	01	01	00	01	10	10	11	00011	00101	00110
2	00010	001	111	100	00	01	01	10	11	10	01	01	00	00011	00101	00110
3	00011	000	110	101	00	01	01	10	11	10	10	10	11	00011	00101	00110
4	00100	111	001	010	11	10	10	01	00	01	01	01	00	00011	00101	00110
5	00101	110	000	011	11	10	10	01	00	01	11	10	11	00011	00101	00110
6	00110	101	011	000	11	10	10	10	11	10	01	01	00	00011	00101	00110
7	00111	100	010	001	11	10	10	10	11	10	10	10	11	00011	00101	00110
8	01000	001	010	100	00	01	01	00	01	00	01	01	00	01001	01010	01100
9	01001	000	011	101	00	01	01	00	01	00	10	10	11	01001	01010	01100
10	01010	011	000	110	00	01	01	11	10	11	01	01	00	01001	01010	01100
11	01011	010	001	111	00	01	01	11	10	11	10	10	11	01001	01010	01100
12	01100	101	110	000	11	10	10	00	01	00	01	01	00	01001	01010	01100
13	01101	100	111	001	11	10	10	00	01	00	11	10	11	01001	01010	01100
14	01110	111	100	010	11	10	10	11	10	11	01	01	00	01001	01010	01100
15	01111	110	101	011	11	10	10	11	10	11	10	10	11	01001	01010	01100
16	10000	001	010	100	00	00	01	00	01	00	01	01	00	10001	10010	10100
17	10001	000	011	101	00	00	01	00	01	00	10	10	11	10001	10010	10100
18	10010	011	000	110	00	00	01	11	10	11	01	01	00	10001	10010	10100
19	10011	010	001	111	00	00	01	11	10	11	10	10	11	10001	10010	10100
20	10100	101	110	000	11	00	10	00	01	00	01	01	00	10001	10010	10100
21	10101	100	111	001	11	00	10	00	01	00	10	10	11	10001	10010	10100
22	10110	111	100	010	11	00	10	11	10	11	01	01	00	10001	10010	10100
23	10111	110	101	011	11	00	10	11	10	11	10	10	11	10001	10010	10100

Отбрасывая лишние значения из табл. 3 и дополняя ее недостающими строками с номерами № 24 – 31, получаем таблицу значений функций дополнения для одного из вариантов функциональной зависимости между рабочими и контрольными выходами самопроверяемого устройства (табл. 4).

Таблица 4

№	$f_1f_2f_3f_4f_5$	$g_3g_4g_5$	$h_1h_2h_3h_4h_5$	f_3g_3	f_4g_4	f_5g_5
0	00000	011	00011	00	01	01
1	00001	010	00011	00	01	10
2	00010	100	00110	01	10	00
3	00011	101	00110	01	10	11
4	00100	001	00101	10	00	01
5	00101	000	00101	10	00	10
6	00110	011	00101	10	11	01
7	00111	010	00101	10	11	10
8	01000	001	01001	00	00	01
9	01001	000	01001	00	00	10
10	01010	110	01100	01	11	00

№	$f_1f_2f_3f_4f_5$	$g_3g_4g_5$	$h_1h_2h_3h_4h_5$	f_3g_3	f_4g_4	f_5g_5
11	01011	111	01100	01	11	11
12	01100	110	01010	10	01	01
13	01101	111	01010	10	01	10
14	01110	100	01010	10	10	01
15	01111	101	01010	10	10	10
16	10000	001	10001	00	00	01
17	10001	000	10001	00	00	10
18	10010	110	10100	01	11	00
19	10011	111	10100	01	11	11
20	10100	110	10010	00	01	01
21	10101	111	10010	00	01	10
22	10110	100	10010	00	10	01
23	10111	101	10010	00	10	10
24	11000	000	11000	00	00	00
25	11001	001	11000	00	00	11
26	11010	010	11000	00	11	00
27	11011	011	11000	00	11	11
28	11100	100	11000	11	00	00
29	11101	101	11000	11	00	11
30	11110	110	11000	11	11	00
31	11111	111	11000	11	11	11

Минимизируя функции g_3 , g_4 и g_5 с использованием метода Карно, получаем формулы, устанавливающие зависимость между функциями логического дополнения и рабочими функциями объекта диагностирования:

$$\begin{aligned}
 g_3 &= f_1f_3 \vee f_2f_3 \vee \overline{f_2f_3f_4} \vee \overline{f_1f_2f_4} = f_3(f_1 \vee f_2) \vee f_4(\overline{f_2f_3} \vee \overline{f_1f_2}); \\
 g_4 &= f_1f_2f_3f_4 \vee f_1f_2f_3\overline{f_4} \vee f_1f_2f_3\overline{f_4} \vee \overline{f_1f_2f_3f_4} \vee f_2f_3f_4 \vee f_1f_3f_4 = \\
 &= \overline{f_1f_2}(\overline{f_3f_4} \vee f_3f_4) \vee f_3\overline{f_4}(\overline{f_1f_2} \vee f_1\overline{f_2}) \vee \overline{f_3f_4}(f_2 \vee f_1); \\
 g_5 &= f_1f_3f_4f_5 \vee f_2f_3f_4f_5 \vee f_1f_2f_3f_5 \vee f_2f_3f_5 \vee f_1f_3f_5 \vee f_1f_2f_5 \vee \overline{f_3f_4f_5} = \\
 &= \overline{f_5}(f_4(\overline{f_1f_3} \vee \overline{f_2f_3}) \vee \overline{f_1f_2f_3}) \vee f_5(f_3(f_2 \vee f_1) \vee (f_1f_2 \vee \overline{f_3f_4})).
 \end{aligned} \tag{1}$$

Число вариантов доопределения контрольных функций дополнения весьма велико. Для сравнения: в структуре логического дополнения с использованием 1/3-кода можно получить только 2 варианта функциональной зависимости [25], 1/4-кода – 72 варианта [26], 2/4-кода – 1 вариант [27].

Из анализа табл. 3 следует, что для полной проверки СВК необходимо, как минимум, сформировать 10 рабочих векторов из 32 (31,25 % от полного множества рабочих комбинаций).

Выпишем условия полной проверки 2/5-TSC. Для этого введем обозначение n_i – логическая переменная, соответствующая десятичному номеру $i = \overline{0,31}$ в таблице соответствия значений рабочих и контрольных функций (табл. 1 – 4). Следующее логическое выражение охватывает все случаи формирования рабочих комбинаций, подача которых на входы 2/5-TSC обеспе-

чивает его полную проверку:

$$\gamma = (n_0 \vee n_1)(n_2 \vee n_3)(n_4 \vee n_5 \vee n_6 \vee n_7)(n_8 \vee n_9)(n_{10} \vee n_{11})(n_{12} \vee n_{13} \vee n_{14} \vee n_{15}) \& \\ \& (n_{16} \vee n_{17})(n_{18} \vee n_{19})(n_{20} \vee n_{21} \vee n_{22} \vee n_{23})(n_{24} \vee n_{25} \vee n_{26} \vee n_{27} \vee n_{28} \vee n_{29} \vee n_{30} \vee n_{31}). \quad (2)$$

В выражении (2) раскрываются скобки, оно представляется в дизъюнктивной нормальной форме (ДНФ), где каждая конъюнкция соответствует набору тестовых комбинаций, необходимых для полной проверки 2/5-TSC.

Аналогично получаем выражения для поиска всех тестовых комбинаций для элементов XOR_3 , XOR_4 и XOR_5 :

$$\delta_3 = (n_0 \vee n_1 \vee n_8 \vee n_9 \vee n_{16} \vee n_{17} \vee n_{20} \vee n_{21} \vee n_{22} \vee n_{23} \vee n_{24} \vee n_{25} \vee n_{26} \vee n_{27}) \& \\ \& (n_2 \vee n_3 \vee n_{10} \vee n_{11} \vee n_{18} \vee n_{19})(n_4 \vee n_5 \vee n_6 \vee n_7 \vee n_{12} \vee n_{13} \vee n_{14} \vee n_{15}) \& \\ \& (n_{28} \vee n_{29} \vee n_{30} \vee n_{31}). \quad (3)$$

$$\delta_4 = (n_4 \vee n_5 \vee n_8 \vee n_9 \vee n_{16} \vee n_{17} \vee n_{24} \vee n_{25} \vee n_{28} \vee n_{29}) \& \\ \& (n_0 \vee n_1 \vee n_{12} \vee n_{13} \vee n_{20} \vee n_{21})(n_2 \vee n_3 \vee n_{14} \vee n_{15} \vee n_{22} \vee n_{23}) \& \\ \& (n_6 \vee n_7 \vee n_{10} \vee n_{11} \vee n_{18} \vee n_{19} \vee n_{26} \vee n_{27} \vee n_{30} \vee n_{31}). \quad (4)$$

$$\delta_5 = (n_2 \vee n_{10} \vee n_{18} \vee n_{24} \vee n_{26} \vee n_{28} \vee n_{30}) \& \\ \& (n_0 \vee n_4 \vee n_6 \vee n_8 \vee n_{12} \vee n_{14} \vee n_{16} \vee n_{20} \vee n_{22})(n_1 \vee n_5 \vee n_7 \vee n_9 \vee n_{13} \vee n_{15} \vee n_{17} \vee n_{21} \vee n_{23}) \& \\ \& (n_3 \vee n_{11} \vee n_{19} \vee n_{25} \vee n_{27} \vee n_{29} \vee n_{31}). \quad (5)$$

Учитывая, что для полной проверки СВК должны подаваться все тестовые комбинации на входы элементов XOR_3 , XOR_4 и XOR_5 и 2/5-TSC, можно записать следующее условие формирования полного множества вариантов, пригодных для полной проверки:

$$\Phi = \gamma \delta_3 \delta_4 \delta_5. \quad (6)$$

Выражение (6) также представляется в ДНФ, где каждая конъюнкция будет соответствовать одному из способов тестирования СВК.

Одним из способов полной проверки тестера является подача на входы СВК следующих десяти комбинаций из множества $\{n_0, n_2, n_5, n_8, n_{11}, n_{12}, n_{16}, n_{18}, n_{20}, n_{28}\}$. В табл. 5 приведены формируемые на каждой из представленных рабочих комбинаций тестовые комбинации для элементов СВК.

Таблица 5

Рабочая комбинация	XOR_3	XOR_4	XOR_5	2/5-TSC
n_0	00	01	01	00011
n_2	01	10	00	00110
n_5	10	00	10	00101
n_8	00	00	01	01001
n_{11}	01	11	11	01100
n_{12}	10	01	01	01010
n_{16}	00	00	01	10001
n_{18}	01	11	00	10100
n_{20}	00	01	01	10010
n_{28}	11	00	11	11000

Заключение

Установление зависимости между функциями, описывающими выходы блока контрольной логики в СВК, и функциями, описывающими выходы объекта диагностирования, позволяет на практике не анализировать и не определять значения контрольных функций на каждой входной комбинации устройства. Наличие функциональной зависимости между выходами объекта диагностирования и блока контрольной логики дает возможность значительно сократить число операций при синтезе СВК.

Число вариантов установления функциональной зависимости между выходами объекта диагностирования и блока контрольной логики описывается астрономическим числом. Соответственно имеется огромное количество вариантов формирования функций блока контрольной логики. Это обстоятельство позволяет для каждого объекта диагностирования выбирать вариант синтеза СВК с наилучшими показателями структурной избыточности путем сравнения показателей сложности технической реализации при различных вариантах дополнения.

Использование 2/5-кода по методу логического дополнения расширяет число известных подходов к синтезу самопроверяемых цифровых устройств и вычислительных систем.

ЛИТЕРАТУРА

1. Щербаков Н.С. Достоверность работы цифровых устройств. – М.: Машиностроение, 1989.
2. Piestrak S.J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. – Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995.
3. Lala P.K. Principles of Modern Digital Design. – New-Jersey: John Wiley & Sons, 2007.
4. Fujiwara E. Code Design for Dependable Systems: Theory and Practical Applications. – John Wiley & Sons, 2006.
5. Дрозд А.В., Харченко В.С., Антощук С.Г., Дрозд Ю.В., Дрозд М.А., Сулима Ю.Ю. Рабочее диагностирование безопасных информационно-управляющих систем / под ред. А.В. Дрозда и В.С. Харченко. – Харьков: Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», 2012.
6. Согомонян Е.С., Слабаков Е.В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989.
7. Goessel M., Graf S. Error Detection Circuits. – London: McGraw-Hill, 1994.
8. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Коды Хэмминга в системах функционального контроля логических устройств. – СПб.: Наука, 2018.
9. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Коды с суммированием для систем технического диагностирования. Т. 1: Классические коды Бергера и их модификации. – М.: Наука, 2020.
10. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Коды с суммированием для систем

- технического диагностирования. Т. 2: Взвешенные коды с суммированием. – М.: Наука, 2021.
11. *Mitra S., McCluskey E.J.* Which Concurrent Error Detection Scheme to Choose? // *Proceedings of International Test Conference, 2000, USA, Atlantic City, NJ, 3-5 October 2000.* – P. 85-994. DOI:10.1109/TEST.2000.894311.
 12. *Сапожников В.В., Сапожников В.В.* Самопроверяемые дискретные устройства. – СПб: Энергоатомиздат, 1992.
 13. *Сапожников В.В., Сапожников Вл.В., Дмитриев А.В., Морозов А.В., Гессель М.* Организация функционального контроля комбинационных схем методом логического дополнения // *Электронное моделирование.* – 2002. – Т. 24, №6. – С. 52-66.
 14. *Гессель М., Морозов А.В., Сапожников В.В., Сапожников Вл.В.* Логическое дополнение – новый метод контроля комбинационных схем // *Автоматика и телемеханика.* – 2003. – №1. – С. 167-176.
 15. *Гессель М., Морозов А.В., Сапожников В.В., Сапожников Вл.В.* Контроль комбинационных схем методом логического дополнения // *Автоматика и телемеханика.* – 2005. – №8. – С. 161-172.
 16. *Сапожников В.В., Сапожников Вл.В., Гессель М.* Самодвойственные дискретные устройства. – СПб: Энергоатомиздат (Санкт-Петербургское отделение), 2001.
 17. *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.
 18. *Аксенова Г.П.* Метод синтеза схем встроенного контроля для автоматов с памятью // *Автоматика и телемеханика.* – 1973. – №2. – С. 109-116.
 19. *Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D.* New Methods of Concurrent Checking: Edition 1. – Dordrecht: Springer Science+Business Media B.V., 2008.
 20. *Sen S.K.* A Self-Checking Circuit for Concurrent Checking by 1-out-of-4 code with Design Optimization using Constraint Don't Cares // *National Conference on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEEERE 2010), Sikkim Manipal Institute of Technology, Sikkim, held during 22-24 December, 2010.*
 21. *Das D.K., Roy S.S., Dmitriev A., Morozov A., Gössel M.* Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes // *Proceedings of the 10th International Workshops on Boolean Problems, Freiberg, Germany, September, 2012.* – P. 33-40.
 22. *Morozov M., Saposhnikov V.V., Saposhnikov Vl.V., Goessel M.* New Self-Checking Circuits by Use of Berger-Codes // *Proceedings of 6th IEEE International On-Line Testing Workshop, Palma de Mallorca, Spain, 3-5 July 2000.* – P. 171-176.
 23. *Ефанов Д.В., Сапожников В.В., Сапожников Вл.В.* О свойствах кода с суммированием в схемах функционального контроля // *Автоматика и телемеханика.* – 2010. – №6. – С. 155-162.
 24. *Ефанов Д.В.* Некоторые особенности обнаружения ошибок равномерными неразделимыми кодами // *Известия вузов. Приборостроение.* – 2019. – Т. 62, №7. – С. 621-631. DOI:10.17586/0021-3454-2019-62-7-621-631.
 25. *Сапожников В.В., Сапожников Вл.В., Ефанов Д.В.* Построение полностью самопроверяемых структур систем функционального контроля с использованием равновесного кода «1 из 3» // *Электронное моделирование.* – 2016. – Т. 38, №6. – С. 25-43.

26. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Пивоваров Д.В. Метод логического дополнения на основе равновесного кода «1 из 4» для построения полностью самопроверяемых структур систем функционального контроля // Электронное моделирование. – 2017. – Т. 39, №2. – С. 15-34.
27. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Построение самопроверяемых структур систем функционального контроля на основе равновесного кода «2 из 4» // Проблемы управления. – 2017. – №1. – С. 57-64.
28. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Синтез самопроверяемых схем встроенного контроля на основе логического дополнения до равновесного кода «2 из 5» // Известия вузов. Приборостроение. – 2021. – Т. 64, №3. – С. 163-175. DOI: 10.17586/0021-3454-2021-64-3-163-175.
29. Аксенова Г.П. Необходимые и достаточные условия построения полностью проверяемых схем свертки по модулю 2 // Автоматика и телемеханика. – 1979. – №9. – С. 126-135.
30. Efanov D.V., Sapozhnikov V.V., Sapozhnikov Vl.V., Pivovarov D.V. The Synthesis Conditions of Completely Self-Testing Embedded-Control Circuits Based on the Boolean Complement Method to the «1-out-of-m» Constant-Weight Code // Automatic Control and Computer Sciences. – 2020. – Vol. 54, Issue 2. – P. 89-99. – DOI: 10.3103/S0146411620020042.

Статья представлена к публикации членом редколлегии С.В. Шалобановым.

E-mail:

Ефанов Дмитрий Викторович – TrES-4b@yandex.ru;

Пивоваров Дмитрий Вячеславович – pivovarov.d.v.spb@gmail.com.