

ОРГАНИЗАЦИЯ КОНТРОЛЯ КОМБИНАЦИОННЫХ СХЕМ НА ОСНОВЕ МЕТОДА ЛОГИЧЕСКОГО ДОПОЛНЕНИЯ ДО РАВНОВЕСНОГО КОДА „1 ИЗ 4“

В. В. САПОЖНИКОВ¹, Вл. В. САПОЖНИКОВ¹, Д. В. ЕФАНОВ², Д. В. ПИВОВАРОВ¹

¹Петербургский государственный университет путей сообщения Императора Александра I,
190031, Санкт-Петербург, Россия

²ООО „ЛокоТех-Сигнал“, 107113, Москва, Россия
E-mail: TrES-4b@yandex.ru

Предложен вариант развития метода логического дополнения по равновесным кодам. Представлен способ организации контроля комбинационных логических устройств на основе метода логического дополнения до равновесного кода „1 из 4“, не требующий доопределения значений одной из рабочих функций объекта диагностирования. Он основан на алгоритме пошагового доопределения контрольных функций с учетом формирования хотя бы по одному разу всех тестовых комбинаций для элементов сложения по модулю два в блоке логического дополнения и тестера. Возникающие в процессе получения значений контрольных функций неопределенные значения позволяют оптимизировать структуру блока контрольной логики и минимизировать сложность конечного устройства. На сложность технической реализации конечного устройства также влияют выбор непреобразуемой функции и последовательность доопределения функций. Замена функций в последовательность доопределения позволяет выбрать наилучший способ построения схемы контроля по показателю структурной избыточности конечного устройства. Рассматриваемый способ, однако, имеет недостаток: необходимо обеспечить появление на входах логического устройства требуемого множества тестовых комбинаций, что может оказаться затруднительным при малом количестве входов.

Ключевые слова: комбинационное логическое устройство, система контроля, логическое дополнение, равновесный код, код „1 из 4“, оптимизация структурной избыточности

Введение. При организации контроля комбинационных логических устройств на любой элементной базе, включая современную программируемую логику, повсеместно применяются равномерные двоичные коды [1—6]. Свойства кода напрямую определяют особенности синтеза системы встроенного контроля, а также непосредственно влияют на характеристики получаемого дискретного устройства, такие как сложность технической реализации, контролепригодность технических средств диагностирования, энергопотребление, быстродействие и т.д. [7, 8]. Таким образом, от выбора кода на этапе проектирования дискретного устройства с встроенной схемой контроля зависят не только технические характеристики конечного устройства, но и экономические показатели (стоимость проектирования и реализации, а также последующие затраты на эксплуатацию).

Известны две основные структурные схемы реализации технических средств диагностирования при контроле комбинационных логических устройств [9]. Первая реализуется с использованием метода вычисления контрольных разрядов [10], согласно которому с помощью блока контрольной логики формируется контрольный вектор для каждого информационного вектора, соответствующего вектору рабочих функций объекта диагностирования (информационному вектору). Соответствие разрядов контрольных и информационных векторов определяется на этапе проектирования дискретного устройства, оно зачастую продиктовано

правилами вычисления контрольных разрядов заранее выбранного делимого кода. К таким кодам относятся классические коды Бергера (коды с суммированием) и их разнообразные модификации [11]. Для контроля установленного соответствия объект диагностирования и блок контрольной логики дополняются самопроверяемой схемой тестера [1, 8].

Вторая структурная схема реализуется по методу логического дополнения [12]. Этот метод, как и метод вычисления контрольных разрядов, подразумевает наличие в схеме контроля блока контрольной логики и тестера выбранного кода. Кроме того, требуется специальный блок логического дополнения, образованный каскадом сумматоров по модулю два. Данный блок позволяет преобразовывать любой вектор рабочих функций в кодовое слово заранее установленного равновесного кода [13].

Метод логического дополнения позволяет реализовывать схемы контроля для заданных комбинационных логических устройств по одному и тому же равновесному коду. Число вариантов построения велико, оно зависит от конкретных значений контрольных функций. Подбирая значения контрольных функций на этапе проектирования схемы контроля, можно выбирать наилучший способ ее реализации исходя из требуемых характеристик. В этом состоит основное преимущество метода логического дополнения перед методом вычисления контрольных разрядов.

В качестве „основы“ при организации контроля комбинационных логических устройств целесообразно выбирать равновесные коды с небольшой длиной кодовых слов (не более шести разрядов), так как их тестеры имеют простую структуру, а для полной проверки требуется формирование множества кодовых слов малой мощности [14].

Настоящая статья дополняет множество известных работ, посвященных синтезу самопроверяемых схем встроенного контроля на основе метода логического дополнения [9, 12—23], и предлагает еще один способ реализации подобных структур. Он основан на введении неопределенностей при формировании значений контрольных функций с учетом обеспечения полной проверяемости элементов блока логического дополнения и тестера. Введение неопределенностей позволяет упрощать структуру блока контрольной логики, снижая тем самым структурную избыточность конечного устройства.

Структурная схема системы контроля. На рис. 1 приведена структурная схема системы контроля, реализованной на основе метода логического дополнения до равновесного кода „1 из 4“ (1/4-кода). Эта схема исследовалась, например, в работах [13, 15, 17, 19, 22]. В ней на множестве выходов комбинационного логического устройства $F(x)$ выделяются группы по четыре выхода в каждой и организуются отдельные схемы контроля по методу логического дополнения, это позволяет организовывать более простые, чем при дублировании, схемы контроля. Каждый вектор $\langle f_1 f_2 f_3 f_4 \rangle$ с помощью блока логического дополнения преобразуется в кодовое слово 1/4-кода $\langle h_1 h_2 h_3 h_4 \rangle$. Преобразования осуществляются по формуле $f_i \oplus g_i = h_i$ ($i = \overline{1, 4}$), где f_i и g_i — i -е рабочая и контрольная функции, а h_i — соответствующий разряд получаемого на выходах блока логического дополнения кодового вектора. Для контроля принадлежности вектора $\langle h_1 h_2 h_3 h_4 \rangle$ 1/4-коду в схеме контроля устанавливается тестер 1/4-кода (1/4-TSC), наиболее простая функциональная схема которого изображена на рис. 2 [24].

Блок логического дополнения в структурной схеме системы контроля имеет четыре элемента сложения по модулю два (XOR), однако для преобразования любого вектора $\langle f_1 f_2 f_3 f_4 \rangle$ в вектор $\langle h_1 h_2 h_3 h_4 \rangle$, принадлежащий 1/4-коду, требуется максимум три таких элемента. Это обстоятельство может быть использовано для снижения структурной избыточности конечного дискретного устройства.

Основной задачей при реализации схемы контроля является обеспечение тестируемости всех элементов сложения по модулю два в составе блока логического дополнения и схемы тестера 1/4-кода. Для ее решения необходимо доопределить контрольные функции таким об-

разом, чтобы для каждого элемента *XOR* в блоке логического дополнения хотя бы по разу формировалась каждая из тестовых комбинаций {00;01;10;11} [25], а для 1/4-*TSC* — хотя бы по разу каждое кодовое слово {0001; 0010; 0100; 1000} [8].

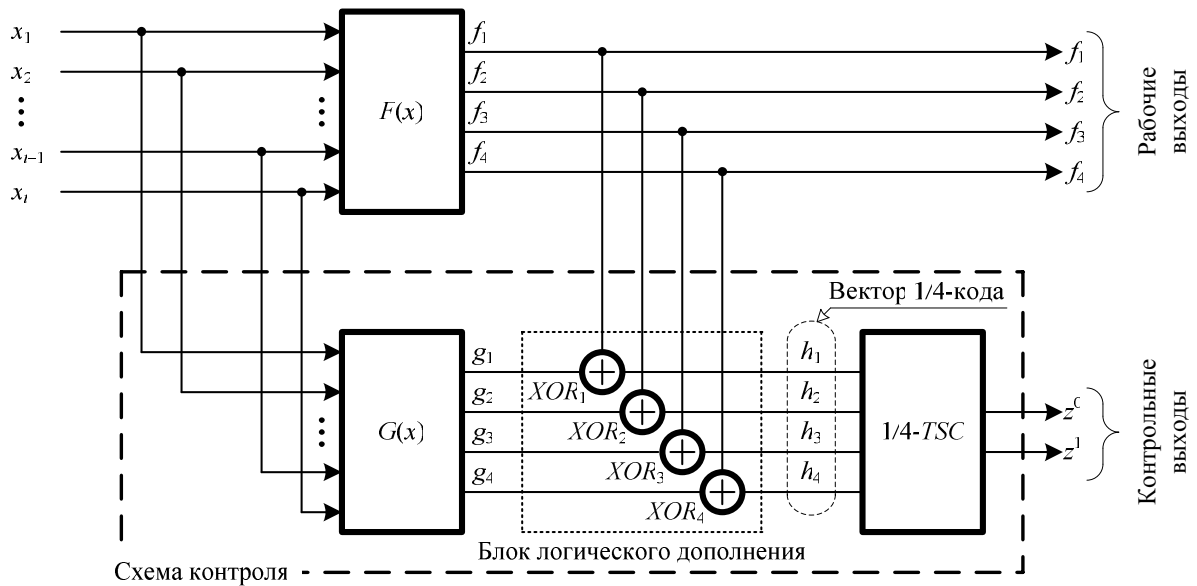


Рис. 1

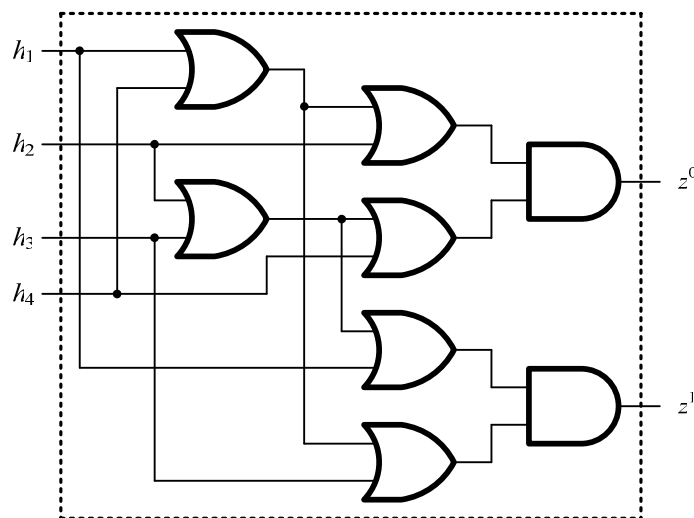


Рис. 2

Способ построения полностью самопроверяемой схемы контроля. Рассмотрим способ построения полностью самопроверяемой схемы контроля, позволяющий пошагово формировать полное множество тестовых комбинаций для каждого элемента *XOR* в структуре блока логического дополнения. Будем рассматривать способ на примере комбинационного логического устройства *F(x)*, заданного таблицей истинности (табл. 1). Устройство имеет четыре входа x_1, x_2, x_3 и x_4 , а также четыре выхода f_1, f_2, f_3 и f_4 .

Шаг 1. Функция h_1 приравнивается к функции f_1 : $h_1 = f_1$ (см. табл. 1). В качестве первой может быть выбрана любая из рабочих функций блока *F(x)*, удовлетворяющая следующему условию: хотя бы на одном наборе она должна принимать единичное значение и хотя бы на одном — нулевое. В противном случае на входе 1/4-*TSC* не будет формироваться тестовая комбинация 1000.

Таблица 1

№	x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4	h_1	h_2	h_3	h_4	g_2	g_3	g_4
0	0	0	0	0	0	1	1	0	0	~			~		
1	0	0	0	1	0	0	0	1	0	~			~		
2	0	0	1	0	0	1	0	1	0	~			~		
3	0	0	1	1	1	0	1	1	1	0			0		
4	0	1	0	0	0	0	1	0	0	~			~		
5	0	1	0	1	0	0	0	0	0	~			~		
6	0	1	1	0	0	1	1	0	0	~			~		
7	0	1	1	1	0	0	1	0	0	~			~		
8	1	0	0	0	1	0	1	1	1	0			0		
9	1	0	0	1	1	1	0	1	1	0			1		
10	1	0	1	0	0	0	0	0	0	~			~		
11	1	0	1	1	1	0	0	1	1	0			0		
12	1	1	0	0	0	1	1	1	0	~			~		
13	1	1	0	1	0	0	0	0	0	~			~		
14	1	1	1	0	0	0	1	1	0	~			~		
15	1	1	1	1	0	0	1	0	0	~			~		

Шаг 2. Определяется значение функции h_2 , при этом подразумевается введение неопределенности (~) для некоторых ее значений

$$h_2 = \begin{cases} 0, & \text{если } h_1 = 1; \\ \sim, & \text{если } h_1 = 0. \end{cases}$$

Шаг 3. Вычисляется значение функции g_2 (см. табл. 1)

$$g_2 = \begin{cases} f_2, & \text{если } h_2 = 0; \\ \sim, & \text{если } h_2 = \sim. \end{cases}$$

Шаг 4. Выполняется минимизация частично определенной функции g_2 , полученной на третьем шаге (рис. 3). Ее значения вносятся в общую таблицу (табл. 2).

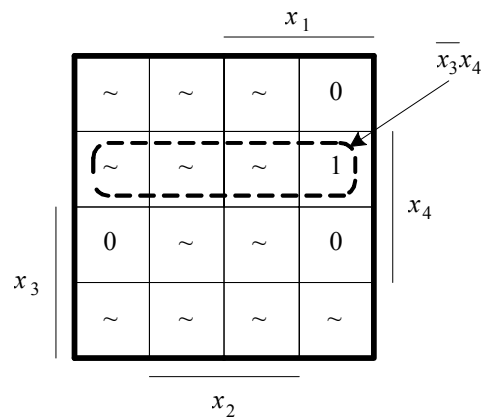


Рис. 3

Шаг 5. Вычисляется функция h_2 по формуле $h_2 = f_2 \oplus g_2$, ее значения вносятся в табл. 2.

Шаг 6. Анализируется наличие четырех проверяющих наборов для элемента XOR₂. Для этого вычисляются функции

$$p_1^2 = f_2 g_2 = x_1 \bar{x}_2 \bar{x}_3 x_4 \neq 0;$$

$$p_2^2 = f_2 \bar{g}_2 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 \bar{x}_4 \neq 0;$$

$$p_3^2 \equiv f_2 g_2 = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 x_2 \bar{x}_3 x_4 \neq 0;$$

$$p_4^2 = \bar{f}_2 \bar{g}_2 = \bar{x}_1 \bar{x}_2 x_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_3 x_4 \vee x_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4 \neq 0.$$

Если все четыре функции не равны нулю, то элемент XOR₂ тестируется. В данном случае на входы XOR₂ подаются все проверяющие наборы.

Таблица 2

№	x ₁	x ₂	x ₃	x ₄	f ₁	f ₂	f ₃	f ₄	h ₁	h ₂	h ₃	h ₄	g ₂	g ₃	g ₄
0	0	0	0	0	0	1	1	0	0	1			0	1	
1	0	0	0	1	0	0	0	1	0	1			1	0	
2	0	0	1	0	0	1	0	1	0	1			0	0	
3	0	0	1	1	1	0	1	1	1	0			0	1	
4	0	1	0	0	0	0	1	0	0	0			0	~	
5	0	1	0	1	0	0	0	0	0	1			1	0	
6	0	1	1	0	0	1	1	0	0	1			0	1	
7	0	1	1	1	0	0	1	0	0	0			0	~	
8	1	0	0	0	1	0	1	1	1	0			0	1	
9	1	0	0	1	1	1	0	1	1	0			1	0	
10	1	0	1	0	0	0	0	0	0	0			0	~	
11	1	0	1	1	1	0	0	1	1	0			0	0	
12	1	1	0	0	0	1	1	1	0	1			0	1	
13	1	1	0	1	0	0	0	0	0	1			1	0	
14	1	1	1	0	0	0	1	1	0	0			0	~	
15	1	1	1	1	0	0	1	0	0	0			0	~	

Шаг 7. Вычисляется функция g₃:

$$g_3 = \begin{cases} f_3, & \text{если } h_1 \vee h_2 = 1; \\ \sim, & \text{если } \overline{h_1 h_2} = 0. \end{cases}$$

Значения функции заносятся в табл. 2.

Шаг 8. Осуществляется минимизация частично заданной функции g₃ (рис. 4), и ее значения вносятся в общую таблицу (табл. 2): $g_3 = \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_3 x_4 \vee x_2 x_3$.

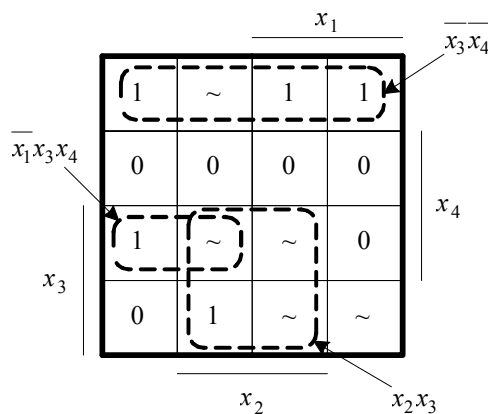


Рис. 4

Шаг 9. Анализируется наличие четырех проверяющих наборов для элемента XOR₃. Вычисляются функции

$$p_1^3 = f_3 g_3 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 x_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4 \neq 0;$$

$$p_2^3 = f_3 \bar{g}_3 = 0;$$

$$p_3^3 = \bar{f}_3 g_3 = 0;$$

$$p_4^3 = \bar{f}_3 \bar{g}_3 = x_1 x_2 \bar{x}_3 x_4 \neq 0.$$

Так как $p_2^3 = 0$ и $p_3^3 = 0$, то на XOR_3 не поступают наборы $f_3 \bar{g}_3 = 10$ и $\bar{f}_3 g_3 = 01$.

Производится корректировка функции g_3 , она определяет наборы, на которых $g_3 = \sim$.

Вычисляется функция корректировки $g_3^k = \bar{h}_1 \bar{h}_2$, задающая наборы, на которых $g_3 = \sim$ (табл. 3).

$$g_3^k = \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4.$$

Исключается случай, когда $p_2^3 = 0$. Для этого вычисляется функция

$$k_2^3 = f_3 g_3^k = \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee x_1 x_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4.$$

Необходимо на любом из этих наборов зафиксировать значение $g_3 = 0$ (например, на $x_1 x_2 x_3 x_4$).

Исключается случай, когда $p_3^3 = 0$. Для этого вычисляется функция $k_3^3 = \bar{f}_3 g_3^k = x_1 \bar{x}_2 x_3 \bar{x}_4$.

На данном наборе необходимо зафиксировать значение $g_3 = 1$.

Задается скорректированная функция g'_3 (табл. 3).

Функция g'_3 минимизируется (рис. 5): $g'_3 = \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_4 \vee \bar{x}_1 x_3 x_4 \vee x_2 \bar{x}_4$.

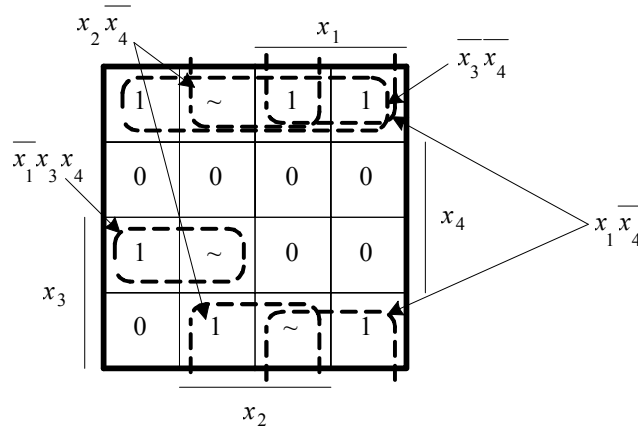


Рис. 5

Шаг 10. Вычисляется функция h_3 по формуле $h_3 = f_3 \oplus g_3$, и ее значения вносятся в табл. 3.

Таблица 3

№	x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4	h_1	h_2	h_3	h_4	g_2	g_3	g'_3	g_4
0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1
2	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0
3	0	0	1	1	1	0	1	1	1	0	0	0	0	1	1	1
4	0	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1
5	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0
6	0	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0
7	0	1	1	1	0	0	1	0	0	0	0	1	0	1	1	1
8	1	0	0	0	1	0	1	1	1	0	0	0	0	1	1	1
9	1	0	0	1	1	1	0	1	1	0	0	0	1	0	0	1
10	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0
11	1	0	1	1	1	0	0	1	1	0	0	0	0	0	0	1
12	1	1	0	0	0	1	1	1	0	1	0	0	0	1	1	1
13	1	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0
14	1	1	1	0	0	0	1	1	0	0	0	1	0	1	1	0
15	1	1	1	1	0	0	1	0	0	0	1	0	0	1	0	0

Шаг 11. Вычисляется функция g_4 (см. табл. 3):

$$g_4 = \begin{cases} f_4, & \text{если } h_1 \vee h_2 \vee h_3 = 1; \\ \overline{f_4}, & \text{если } \overline{h_1} \overline{h_2} \overline{h_3} = 0. \end{cases}$$

Шаг 12. Анализируется получение четырех проверяющих наборов для элемента g_4 XOR_4 . Вычисляются функции

$$\begin{aligned} p_1^4 &= f_4 g_4 = \overline{x_1} \overline{x_2} \overline{x_3} x_4 \vee \overline{x_1} \overline{x_2} x_3 x_4 \vee x_1 \overline{x_2} \overline{x_3} \overline{x_4} \vee \\ &\vee x_1 \overline{x_2} \overline{x_3} x_4 \vee x_1 \overline{x_2} x_3 x_4 \vee x_1 x_2 \overline{x_3} \overline{x_4} \neq 0; \\ p_2^4 &= f_4 \overline{g_4} = \overline{x_1} \overline{x_2} x_3 \overline{x_4} \vee x_1 x_2 x_3 \overline{x_4} \neq 0; \\ p_3^4 &= \overline{f_4} g_4 = \overline{x_1} x_2 \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 x_3 x_4 \neq 0; \\ p_4^4 &= \overline{f_4} \overline{g_4} = \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 \overline{x_3} x_4 \vee \overline{x_1} x_2 x_3 \overline{x_4} \vee \\ &\vee x_1 \overline{x_2} x_3 \overline{x_4} \vee x_1 x_2 \overline{x_3} x_4 \vee x_1 x_2 x_3 x_4 \neq 0. \end{aligned}$$

Так как все функции не равны 0, элемент XOR_4 тестируется.

Шаг 13. Вычисляется функция h_4 по формуле $h_4 = f_4 \oplus g_4$, и ее значения вносятся в табл. 3.

Если на каком-то шаге не удастся обеспечить тестируемость элементов XOR , то доопределяемая функция заменяется и операции алгоритма повторяются.

Приведем формулы доопределения контрольных функций:

$$\begin{aligned} h_1 &= f_1; \\ h_2 &= \begin{cases} 0, & \text{если } h_1 = 1; \\ \sim, & \text{если } h_1 = 0; \end{cases} \\ g_2 &= \begin{cases} f_2, & \text{если } h_2 = 0; \\ \sim, & \text{если } h_2 = \sim; \end{cases} \\ g_3 &= \begin{cases} f_3, & \text{если } h_1 \vee h_2 = 1; \\ \sim, & \text{если } \overline{h_1} \overline{h_2} = 0; \end{cases} \\ g_4 &= \begin{cases} f_4, & \text{если } h_1 \vee h_2 \vee h_3 = 1; \\ \overline{f_4}, & \text{если } \overline{h_1} \overline{h_2} \overline{h_3} = 0. \end{cases} \end{aligned}$$

Отметим также, что в заключение процедуры доопределения значений контрольных функций не требуется проверка формирования контрольного множества тестовых комбинаций для $1/4-TSC$, поскольку представленный способ доопределения значений контрольных функций на этапе проверки наличия тестовых комбинаций элементов XOR обеспечивает и автоматическое формирование соответствующих комбинаций $1/4$ -кода: наличие тестовых комбинаций $\langle 01 \rangle$ и $\langle 10 \rangle$ на XOR гарантирует формирование единицы на его выходе, тогда как все остальные значения функций h_i будут равны нулю.

Заключение. Представленный способ организации контроля комбинационных логических устройств позволяет обеспечивать тестируемость всех элементов XOR в составе блока логического дополнения, а также формировать контрольное множество тестовых комбинаций для $1/4-TSC$. Следует отметить, что алгоритм доопределения функций является универсальным и позволяет синтезировать схемы контроля для логических комбинационных устройств, реализованных на любой элементной базе. Для многовыходных логических устройств при организации контроля выделяются группы по четыре выхода в каждой, а затем синтезируется отдельная схема контроля. Выходы получаемых подсхем контроля объединяются на входах самопроверяемой схемы сжатия парафазных сигналов для получения одного контрольного

парафазного выхода. При этом блоки контрольной логики отдельных подсхем контроля могут быть реализованы совместно, что позволит уменьшить структурную избыточность конечного устройства.

СПИСОК ЛИТЕРАТУРЫ

1. *Piestrak S. J.* Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
2. *Nicolaidis M.* On-Line Testing for VLSI: State of the Art and Trends // *Integration. The VLSI Journal*. 1998. Vol. 26, is. 1—2. P. 197—209. DOI: 10.1016/S0167-9260(98)00028-5.
3. *Das D., Toubia N. A.* Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes // *J. of Electronic Testing: Theory and Applications*. 1999. Vol. 15, is. 1—2. P. 145—155. DOI: 10.1023/A:1008344603814.
4. *Mitra S., McCluskey E. J.* Which Concurrent Error Detection Scheme to Choose? // *Proc. of Intern. Test Conf. USA, Atlantic City, NJ, 03—05 October 2000*. P. 985—994. DOI: 10.1109/TEST.2000.894311.
5. *Lala P. K.* Self-Checking and Fault-Tolerant Digital Design. San Francisco: Morgan Kaufmann Publishers, 2001. 216 p.
6. *Fujiwara E.* Code Design for Dependable Systems: Theory and Practical Applications. John Wiley & Sons, 2006. 720 p.
7. *Согомонян Е. С., Слабаков Е. В.* Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
8. *Сапожников В. В., Сапожников Вл. В.* Самопроверяемые дискретные устройства. СПб: Энергоатомиздат, 1992. 224 с.
9. *Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D.* New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science+Business Media B.V., 2008. 184 p.
10. *Nicolaidis M., Zorian Y.* On-Line Testing for VLSI — A Compendium of Approaches // *J. of Electronic Testing: Theory and Applications*. 1998. N 12. P. 7—20. DOI: 10.1023/A:1008244815697.
11. *Efanov D., Sapozhnikov V., Sapozhnikov Vl.* Generalized Algorithm of Building Summation Codes for the Tasks of Technical Diagnostics of Discrete Systems // *Proc. of 15th IEEE East-West Design & Test Symposium (EWDTs'2017)*. Novi Sad, Serbia, September 29—October 2, 2017. P. 365—371. DOI: 10.1109/EWDTs.2017.8110126.
12. *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.
13. *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.
14. *Сапожников В. В., Сапожников Вл. В., Ефанов Д. В., Пивоваров Д. В.* Синтез систем функционального контроля многовыходных комбинационных схем на основе метода логического дополнения // *Вестн. Томского гос. ун-та. Управление, вычислительная техника и информатика*. 2017. № 4. С. 69—80. DOI: 10.17223/19988605/41/9.
15. *Goessel M., Saposhnikov Vl., Saposhnikov V., Dmitriev A.* A New Method for Concurrent Checking by Use of a 1-out-of-4 Code // *Proc. of the 6th IEEE Intern. On-line Testing Workshop*. Palma de Mallorca, Spain. 3—5 July 2000. P. 147—152.
16. *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.
17. *Saposhnikov V. V., Saposhnikov Vl. V., Morozov A., Osadtchi G., Gossel M.* Design of Totally Self-Checking Combinational Circuits by Use of Complementary Circuits // *Proc. of East-West Design & Test Workshop*. Yalta, Ukraine, 2004. P. 83—87.

18. Das D. K., Roy S. S., Dmitiriev A., Morozov A., Gössel M. Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes // Proc. of the 10th Intern. Workshop on Boolean Problems. Freiburg, Germany, September, 2012. P. 33—40.
19. 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 Conf. on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEEERE 2010). Sikkim Manipal Institute of Technology, Sikkim, 22—24 December, 2010.
20. Sapozhnikov V., Sapozhnikov Vl., Efanov D. Concurrent Error Detection of Combinational Circuits by the Method of Boolean Complement on the Base of „2-out-of-4“ Code // Proc. of 14th IEEE East-West Design & Test Symposium (EWDTS'2016). Yerevan, Armenia, 14—17 October 2016. P. 126—133. DOI: 10.1109/EWDTS.2016.7807677.
21. 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.
22. 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.
23. 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.
24. Сапожников В. В., Рабара В. Универсальный алгоритм синтеза 1/n-тестеров // Проблемы передачи информации. 1982. Т. 18, № 3. С. 62—73.
25. Аксёнова Г. П. Необходимые и достаточные условия построения полностью проверяемых схем свертки по модулю 2 // Автоматика и телемеханика. 1979. № 9. С. 126—135.

Сведения об авторах

- Валерий Владимирович Сапожников** — д-р техн. наук, профессор; Петербургский государственный университет путей сообщения Императора Александра I, кафедра автоматки и телемеханики на железных дорогах;
E-mail: port.at.pgups@gmail.com
- Владимир Владимирович Сапожников** — д-р техн. наук, профессор; Петербургский государственный университет путей сообщения Императора Александра I, кафедра автоматки и телемеханики на железных дорогах;
E-mail: at.pgups@gmail.com
- Дмитрий Викторович Ефанов** — д-р техн. наук, доцент; ООО „ЛокоТех-Сигнал“; Российский университет транспорта (МИИТ), кафедра автоматки, телемеханики и связи на железнодорожном транспорте;
E-mail: TrES-4b@yandex.ru
- Дмитрий Вячеславович Пивоваров** — Петербургский государственный университет путей сообщения Императора Александра I, кафедра автоматки и телемеханики на железных дорогах; ассистент; E-mail: pivovarov.d.v.spb@gmail.com

Поступила в редакцию
29.05.18 г.

Ссылка для цитирования: Сапожников В. В., Сапожников Вл. В., Ефанов Д. В., Пивоваров Д. В. Организация контроля комбинационных схем на основе метода логического дополнения до равновесного кода „1 из 4“ // Изв. вузов. Приборостроение. 2018. Т. 61, № 12. С. 1025—1035.

ORGANIZATION OF CONTROL OF COMBINATIONAL CIRCUITS BASED ON THE METHOD OF LOGICAL COMPLEMENT TO EQUILIBRIUM 1-OUT-OF-4 CODE

V. V. Sapozhnikov¹, VI. V. Sapozhnikov¹, D. V. Efanov², D. V. Pivovarov¹

¹ Emperor Alexander I St. Petersburg State Transport University,
190031, St. Petersburg, Russia

² LocoTech-Signal Ltd., 107113, Moscow, Russia
E-mail: TrES-4b@yandex.ru

A variant of the development of the method of logical complement by the equilibrium code is proposed. A method of organization of control of combinational logic devices based on the method of logical complement to the equilibrium 1-out-of-4 code is presented, which does not require additional determination of the values of one of the working functions of the object of diagnosis. The method employs a step-by-step algorithm to determine the control functions, considering the formation of at least once all test combinations for the elements of addition modulo two in the logical complement block and the tester. Uncertainties arising in the process of obtaining the values of control functions allow optimizing the structure of the control logic block and minimizing the complexity of the final device. Complexity of technical implementation of the end device is also affected by the choice of non-convertible function and the sequence of supplementing function definition. Replacing functions in the sequence of supplementing definitions allows to choose the best way to build a control scheme by the index of structural redundancy of the end device. The disadvantage of the considered method is the necessity to ensure the appearance of the required set of test combinations at the inputs of the logic device, which can be difficult with a small number of inputs.

Keywords: combinational logical device, control system, logical complement, equilibrium 1-out-of-4 code, structural redundancy optimization

REFERENCES

1. Piestrak S.J. *Design of Self-Testing Checkers for Unidirectional Error Detecting Codes*, Wroclaw, Oficyna Wydawnicza Politechniki Wroclawskiej, 1995, 111 p.
2. Nicolaidis M. *Integration, the VLSI Journal*, 1998, no. 1–2(26), pp. 197–209, DOI: 10.1016/S0167-9260(98)00028-5.
3. Das D., Toubia N.A. *Journal of Electronic Testing: Theory and Applications*, 1999, no. 1–2(15), pp. 145–155. DOI: 10.1023/A:1008344603814.
4. Mitra S., McCluskey E.J. *Proceedings of International Test Conference*, 2000, USA, Atlantic City, NJ, 03–05 October 2000, pp. 985–994. DOI: 10.1109/TEST.2000.894311.
5. Lala P.K. *Self-Checking and Fault-Tolerant Digital Design*, San Francisco, Morgan Kaufmann Publishers, 2001, 216 p.
6. Fujiwara E. *Code Design for Dependable Systems: Theory and Practical Applications*, John Wiley & Sons, 2006, 720 p.
7. Sogomonyan E.S., Slabakov E.V. *Samoproveryaemye ustroystva i otkazoustoychivyye sistemy* (The Self-Checked Devices and Failure-Safe Systems), Moscow, 1989, 208 p. (in Russ.)
8. Sapozhnikov V.V., Sapozhnikov VI.V. *Samoproveryayemyye diskretnyye ustroystva* (The Self-Checked discrete devices), St. Petersburg, 1992, 224 p. (in Russ.)
9. Gössel M., Ocheretny V., Sogomonyan E., Marienfeld D. *New Methods of Concurrent Checking: Edition 1*, Dordrecht: Springer Science+Business Media B.V., 2008, 184 p.
10. Nicolaidis M., Zorian Y. *Journal of Electronic Testing: Theory and Applications*, 1998, no. 12, pp. 7–20. DOI: 10.1023/A:1008244815697.
11. Efanov D., Sapozhnikov V., Sapozhnikov VI. *Proc. of 15th IEEE East-West Design & Test Symposium (EWDTs'2017)*, Novi Sad, Serbia, September 29–October 2, 2017, pp. 365–371. DOI: 10.1109/EWDTs.2017.8110126.
12. Sapozhnikov V.V., Sapozhnikov VI.V., Dmitriyev A.V., Morozov A.V., Goessel M. *Modelling and simulation in engineering*, 2002, no. 6(24), pp. 52–66.
13. Goessel M., Morozov A.V., Sapozhnikov V.V., Sapozhnikov VI.V. *Automation and Remote Control*, 2005, no. 8(66), pp. 1336–1346.
14. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V., Pivovarov D.V. *Tomsk State University Journal. Control and Computer Science*, 2017, no. 4, pp. 69–80. DOI: 10.17223/19988605/41/9. (in Russ.)
15. Goessel M., Saposhnikov VI., Saposhnikov V., Dmitriev A. *Proc. of the 6th IEEE International On-line Testing Workshop*, 3–5 July 2000, Palma de Mallorca, Spain, pp. 147–152.
16. Gessel M., Morozov A.V., Sapozhnikov V.V., Sapozhnikov VI.V. *Autom. Remote Control*, 2003, no. 1(64), pp. 153–161.
17. Saposhnikov V.V., Saposhnikov VI.V., Morozov A., Osadtschi G., Gossel M. *Proc. of East-West Design & Test Workshop*, Yalta, Ukraine, 2004, pp. 83–87.
18. Das D.K., Roy S.S., Dmitriev A., Morozov A., Gössel M. *Proc. of the 10th Intern. Workshops on Boolean Problems*, Freiberg, Germany, September, 2012, pp. 33–40.
19. Sen S.K. *National Conf. on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEERE 2010)*, Sikkim Manipal Institute of Technology, Sikkim, 22–24 December, 2010.

20. Sapozhnikov V., Sapozhnikov VI., Efanov D. *Proc. of 14th IEEE East-West Design & Test Symposium (EWDTs'2016)*, Yerevan, Armenia, 14–17 October, 2016, pp. 126–133. DOI: 10.1109/EWDTs.2016.7807677.
21. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V. *Modelling and simulation in engineering*, 2016, no. 6(38), pp. 25–43. (in Russ.)
22. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V., Pivovarov D.V. *Modelling and simulation in engineering*, 2017, no. 2(39), pp. 15–34. (in Russ.)
23. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V. *Control Sciences*, 2017, no. 1, pp. 57–64. (in Russ.)
24. Sapozhnikov V.V., Rabara V. *Problems of Information Transmission*, 1982, no. 3(18), pp. 62–73. (in Russ.)
25. Aksenova G.P. *Automation and Remote Control*, 1979, no. 9, pp. 126–135. (in Russ.)

Data on authors

- | | |
|--------------------------------|---|
| Valery V. Sapozhnikov | — Dr. Sci., Professor; Emperor Alexander I St. Petersburg State Transport University, Department of Automation and Remote Control on Railways; E-mail: port.at.pgups@gmail.com |
| Vladimir V. Sapozhnikov | — Dr. Sci., Professor; Emperor Alexander I St. Petersburg State Transport University, Department of Automation and Remote Control on Railways; E-mail: at.pgups@gmail.com |
| Dmitry V. Efanov | — Dr. Sci., Associate Professor; LocoTech-Signal Ltd.; Russian University of Transport; Department of Automation, Remote Control, and Communication in Railway Transport; E-mail: TrES-4b@yandex.ru |
| Dmitry V. Pivovarov | — Emperor Alexander I St. Petersburg State Transport University, Department of Automation and Remote Control on Railways; Assistant; E-mail: pivovarov.d.v.spb@gmail.com |

For citation: Sapozhnikov V. V., Sapozhnikov VI. V., Efanov D. V., Pivovarov D. V. Organization of control of combinational circuits based on the method of logical complement to equilibrium 1-out-of-4 code. *Journal of Instrument Engineering*. 2018. Vol. 61, N 12. P. 1025—1035 (in Russian).

DOI: 10.17586/0021-3454-2018-61-12-1025-1035