

3. **Автолисп** – язык графического программирования / Г. А. Бугрименко. – Москва : Машиностроение, 1992.

4. **AutoCAD**: программирование и адаптация / Ю. А. Кречко. – Москва : Диалог – МИФИ, 1996.

УДК 681.518.5:004.052.32

**Вал. В. Сапожников, Вл. В. Сапожников, Д. В. Ефанов**

Петербургский государственный университет путей сообщения

## **СИНТЕЗ КОНТРОЛЕПРИГОДНЫХ ДИСКРЕТНЫХ УСТРОЙСТВ ПУТЕМ ПРИМЕНЕНИЯ МОДУЛЬНЫХ КОДОВ С СУММИРОВАНИЕМ**

При проектировании контролепригодных дискретных устройств часто используются коды с суммированием. Классический код с суммированием позволяет синтезировать дискретные устройства с обнаружением любых одиночных ошибок, возникающих в элементах внутренней структуры контролируемого устройства. При этом исходное устройство преобразуется и становится устройством с независимыми или монотонно-независимыми выходами. Показан подход, применение которого уменьшает сложность системы функционального контроля. Это достигается использованием в качестве основы системы контроля модульного кода с суммированием, а не кода Бергера.

функциональный контроль; код Бергера; модульный код с суммированием; информационные разряды; необнаруживаемая ошибка; однонаправленная необнаруживаемая ошибка; комбинационные схемы с монотонно-независимыми выходами.

### **Введение**

При проектировании контролепригодных дискретных устройств [1–5] используют различные способы достижения высокого уровня надежности, при этом несомненно важное значение имеет техническое диагностирование. Под техническим диагностированием понимается процесс определения технического состояния контролируемого объекта с заданной полнотой и глубиной поиска [6].

Процесс диагностирования может осуществляться в двух режимах: тестовом и функциональном. При тестовом диагностировании устройство кратковременно отключается и на его входы подается тест, включающий в себя совокупность проверок [6, 7]. По завершении диагностирования устройство вновь включается в работу. Такой подход, например, требует при построении систем управления 100%-го резервирования.

В отличие от тестового диагностирования, при функциональном, или рабочем, контроле определение состояния производится без отключения объекта [2]. В современных микропроцессорных системах управления оба метода диагностирования используются наравне.

Настоящая работа посвящена развитию теории функционального контроля комбинационных логических устройств, являющихся обязательными компонентами любой микропроцессорной и микроэлектронной системы управления.

### **1 Система функционального контроля**

В системе функционального контроля (рис. 1) выделяются три блока: исходное устройство  $f(x)$ , в функции которого входит вычисление системы булевых функций  $f_1(x)$ ,

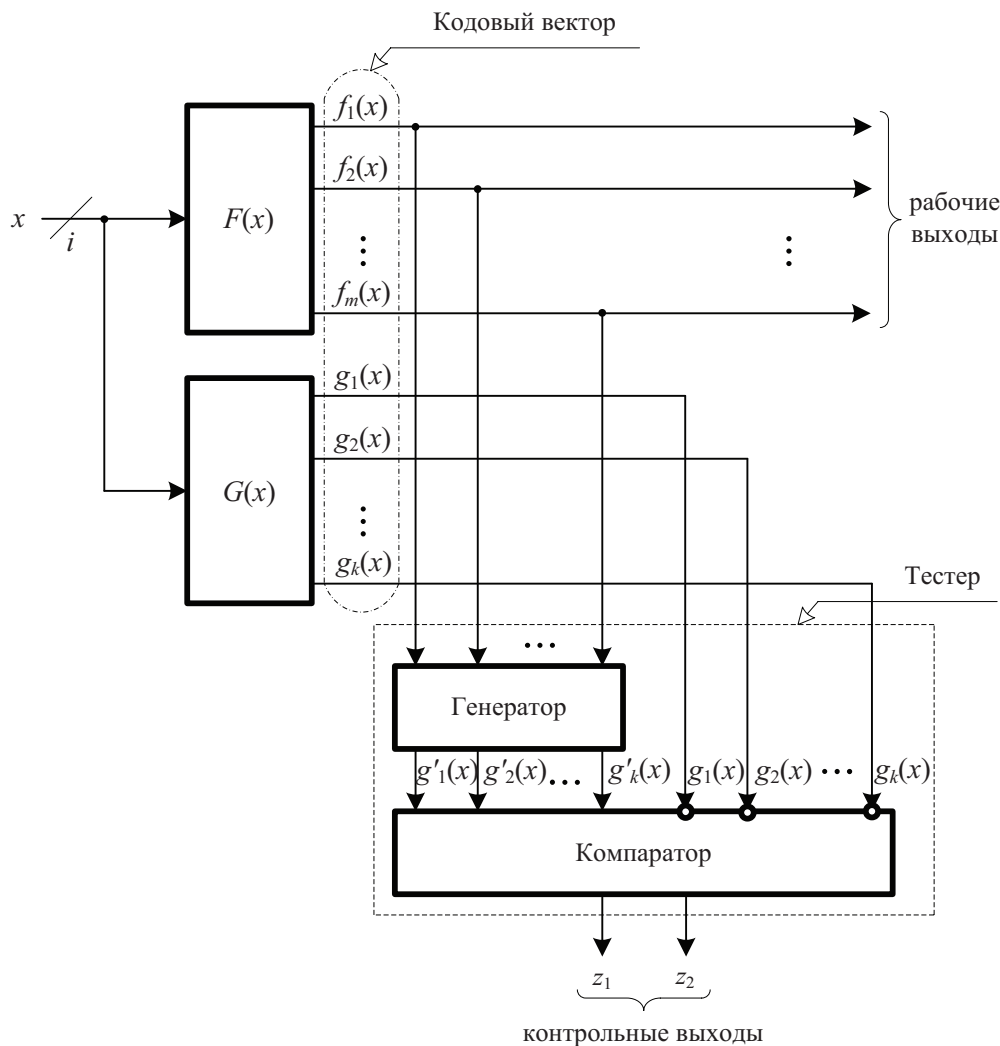


Рис. 1. Структура системы функционального контроля

$f_2(x), \dots, f_m(x)$ ; блок дополнительной логики  $g(x)$ , формирующий специальные контрольные функции  $g_1(x), g_2(x), \dots, g_k(x)$ ; блок самопроверяемого тестера – его задачей является контроль соответствия значений выходов предыдущих двух блоков [2, 6].

При построении системы, приведенной на рис. 1, используются свойства избыточных кодов [8–10]. Так, выходам контролируемой схемы  $F(x)$  ставят в соответствие информационный вектор длиной  $m$ , а выходам блока дополнительной логики  $G(x)$  – контрольный вектор длиной  $k$ , вычисляемый по правилам известного кода. От правил построения кода зависит сложность самой системы контроля и возможности обнаружения ошибок на выходах системы контроля. Ошибки в блоке

$G(x)$  искажают контрольные функции, следовательно, обнаруживаются схемой тестера всегда. Ошибки же в блоке  $F(x)$  могут не обнаруживаться.

Наиболее простым с позиции избыточности технической реализации является применение кодов паритета (в этом случае блок дополнительной логики имеет один выход), а наиболее сложным – применение кодов с повторением (в этом случае исходное устройство дублируется). Ясно, что коды паритета обнаруживают меньшее количество ошибок в своих информационных векторах, нежели коды с повторением (это следует из соотношения избыточностей кодов). Возникает задача построения системы функционального контроля с наиболее оптимальными показателями

телями сложности и контролепригодности. Эта задача решается путем применения кодов с суммированием [11].

## 2 Коды с суммированием в системах функционального контроля

Задачу идентификации дефектов внутренней структуры блока основной логики системы функционального контроля сводят к задаче сопоставления информационных и контрольных векторов используемого кода. Так, любая одиночная неисправность блока основной логики  $F(x)$  может привести к искажению некоторого количества информационных разрядов, при этом, в зависимости от логических связей внутри контролируемого устройства, возможны различные варианты искажений выходных функций. В зависимости от типа кода, выбранного на этапе проектирования, ошибка в информационном векторе может быть обнаружена или не обнаружена. Например, при использовании в основе системы контроля классического кода с суммированием, или кода Бергера (в нем контрольный вектор содержит сумму единичных информационных разрядов – вес информационного вектора  $r$ ) [11], любая разнонаправленная ошибка четной кратности, имеющая одинаковое число искажений  $0 \rightarrow 1$  и  $1 \rightarrow 0$ , не обнаруживается. Это вытекает из того факта, что информационные векторы кода Бергера, имеющие одинаковый вес, имеют и одинаковые контрольные векторы; сохранение веса информационного вектора при искажении соответствует необнаруживаемой ошибке. Классический код с суммированием

$$\beta_d = 2^{-d} C_d^{\frac{d}{2}} \quad (1)$$

не обнаруживает ошибок четной кратности  $d$  от общего числа ошибок той же кратности [12, 13]. Это, к примеру, половина всех двукратных искажений информационных разрядов.

Обозначим далее классический код с суммированием как  $S(n, m)$ -код, где  $m$  – дли-

на информационного вектора, а  $n$  – общая длина кодовой комбинации ( $n = m + k$ ,  $k$  – число контрольных разрядов). Причиной таких небольших возможностей обнаружения искажений информационных векторов является крайне неравномерное распределение информационных векторов между контрольными. Число информационных векторов, соответствующее одному контрольному вектору, определяется биномиальным коэффициентом  $C_m^r$ , где  $r = [0, 1, \dots, m]$ . Например, у кода Бергера  $S(12, 8)$  контрольному вектору  $\langle 0100 \rangle$  соответствует  $C_8^4 = 70$  информационных векторов, а контрольному вектору  $\langle 0001 \rangle$  – всего  $C_8^1 = 8$  информационных векторов. Свойства  $S(n, m)$ -кода можно улучшить. При этом наилучший результат по общему количеству необнаруживаемых искажений для кода с параметрами  $m$  и  $k$  будет получен при равномерном распределении  $2^m$  информационных векторов между  $2^k$  контрольными векторами [14]. Это так называемый оптимальный код с суммированием –  $O(n, m)$ -код. Данный код имеет минимальное количество необнаруживаемых искажений для заданных  $m$  и  $k$ . В сравнении с  $O(n, m)$ -кодом можно определить эффективность любого кода с суммированием при тех же параметрах. Обозначим как  $N_{m,k}^{\min}$  общее число необнаруживаемых ошибок в  $O(n, m)$ -коде и как  $N_{m,k}$  – общее число необнаруживаемых ошибок в коде с суммированием с теми же  $m$  и  $k$ . Тогда коэффициент эффективности можно определить отношением [14, 15]:

$$\xi = \frac{N_{m,k}^{\min}}{N_{m,k}}. \quad (2)$$

Знаменатель выражения (2) всегда больше или равен числителю, откуда следует, что чем ближе значение  $\xi$  к единице, тем ближе код с суммированием к  $O(n, m)$ -коду.

Величина  $N_{m,k}^{\min}$  вычисляется по формуле [14]:

$$N_{m,k}^{\min} = 2^m (2^{m-k} - 1). \quad (3)$$

Для классического кода Бергера величина  $N_{m,k}$  может быть получена так:

$$N_{m,k} = \sum_{p \in Q} 2C_p^2, \quad (4)$$

где  $Q = \{C_m^1, C_m^2, \dots, C_m^{m-1}\}$ .

Подсчитаем коэффициент эффективности кода  $S(12,8)$ -кода. По формуле (3) определим значение числителя:

$$N_{8,4}^{\min} = 2^8 (2^{8-4} - 1) = 3840.$$

Значение знаменателя определяем по формуле (4):

$$Q = \{C_8^1, C_8^2, C_8^3, C_8^4, C_8^5, C_8^6, C_8^7\} = \\ = \{8, 28, 56, 70, 56, 28, 8\};$$

$$N_{8,4} = 2(C_8^2 + C_{28}^2 + C_{56}^2 + C_{70}^2 + \\ + C_{28}^2 + C_{56}^2 + C_8^2) = 12\,614.$$

Подставляя значения, полученные по формулам (3) и (4), в (2) получим  $\xi = 0,629$ .

В целом коды Бергера имеют довольно низкое значение коэффициента эффективности – для большинства кодов Бергера значение величины  $\xi < 0,5$ . Значение  $\xi$  при данном  $k$  уменьшается с увеличением  $m$  от кода с числом информационных разрядов  $m = 2^{k-1}$  к коду с числом информационных разрядов  $m = 2^{k-1}$ , где  $k$  – число контрольных разрядов в коде Бергера,  $k = \lceil \log_2(m+1) \rceil$  – целое сверху от величины  $\log_2(m+1)$  [11].

Одним из способов повышения эффективности  $S(n, m)$ -кодов является модификация по следующему правилу [14]: контрольный вектор содержит в своих разрядах двоичный эквивалент значения величины  $W = V + \alpha M$ , где  $V$  – вес информационного вектора по модулю  $M = 2^{\lceil \log_2(m+1) \rceil - 1}$ , а  $\alpha = \bigoplus_{i \in F} f_i(x)$  ( $F$  – множество информационных разрядов) – сумма по модулю двух заранее выбранных информационных разрядов. Распределение необнаруживаемых ошибок по кратностям зависит только от количества информационных разрядов в линейной сумме поправоч-

ного коэффициента  $\alpha$  и никак не зависит от конкретно выбранных разрядов; кроме того, разные распределения присущи только модифицированным кодам с  $\alpha = 1, 2, \dots, \lfloor \frac{m}{2} \rfloor$ , где запись  $\lfloor \frac{m}{2} \rfloor$  обозначает целое снизу от  $\frac{m}{2}$ . Данные коды обозначаются как  $RS(n, m, i)$ -коды.

К классическим и модифицированным кодам с суммированием могут быть добавлены коды с уменьшенным количеством контрольных разрядов, что достигается путем уменьшения модуля счета. Это модульные коды с суммированием ( $SM(n, m)$ -коды) [16–18] и модульно-модифицированные коды с суммированием ( $RSM(n, m, i)$ -коды), где  $M \in 2^1, 2^2, \dots, 2^{\lceil \log_2(m+1) \rceil - 1}$  [19, 20].

Достоинством кодов с фиксированным количеством контрольных разрядов является возможность синтеза более простых, в сравнении с кодом Бергера, блоков дополнительной логики и тестера системы функционального контроля. Свойства модульных кодов с суммированием, а также их модификаций подробно изложены в работах [18, 19].

Для примера в табл. 1 приводятся характеристики всех кодов с суммированием единичных разрядов с длиной информационного вектора  $m = 8$ . При заданном  $m$  и различных  $k$  существует свой  $O(n, m)$ -код. Чем меньше модуль подсчета веса кода с суммированием, тем ближе данный код к оптимальному при тех же параметрах  $m$  и  $k$  (табл. 1). Например, код  $S2(9, 8)$  является оптимальным, так как все его информационные векторы равномерно распределены на две контрольные группы (этот код является кодом паритета), а все семейство модульно-модифицированных кодов  $RS2(10, 8, i)$  имеет четыре равномерно заполненные контрольные группы. Модификация кода позволяет увеличить возможности по обнаружению ошибок в информационных векторах кодов при любом  $k \neq 1$  (другими словами, код с суммированием с одним контрольным разрядом, или код паритета, является оптимальным в своем классе).

ТАБЛИЦА 1. Множество кодов с суммированием единичных разрядов с  $m = 8$ 

Код	Кратность необнаруживаемой ошибки, $d$				$N_{m,k}$	$\xi$
	2	4	6	8		
$k = 1$						
$S_2(9, 8)$	7168	17920	7168	256	32 512	1
$k = 2$						
$S_4(10, 8)$	3584	8960	3584	128	16 256	0,9921
$RS_2(10, 8, 1)$	5376	8960	1792	0	16 128	1
$RS_2(10, 8, 2)$	4096	7680	4096	256	16 128	1
$RS_2(10, 8, 3)$	3328	8960	3840	0	16 128	1
$RS_2(10, 8, 4)$	3072	9728	3072	256	16 128	1
$k = 3$						
$S_8(11, 8)$	3584	6720	2240	72	12 616	0,629
$RS_4(11, 8, 1)$	2688	4480	896	0	8064	0,9841
$RS_4(11, 8, 2)$	2048	3840	2048	128	8064	0,9841
$RS_4(11, 8, 3)$	1664	4480	1920	0	8064	0,9841
$RS_4(11, 8, 4)$	1536	4864	1536	128	8064	0,9841
$k = 4$						
$S(12, 8)$	3584	6720	2240	70	12 614	0,3044
$RS(12, 8, 1)$	2688	3360	560	0	6608	0,5811
$RS(12, 8, 2)$	2048	2880	1280	72	6280	0,6115
$RS(12, 8, 3)$	1664	3360	1200	0	6224	0,617
$RS(12, 8, 4)$	1536	3648	960	72	6216	0,6178

### 3 Особенности построения систем функционального контроля на основе модульных кодов с суммированием

Поскольку классический  $S(n, m)$ -код обнаруживает любые однонаправленные искажения информационных разрядов [11], он может эффективно применяться для построения схем с обнаружением любых одиночных неисправностей во внутренней структуре контролируемого блока  $F(x)$ , для чего структура блока  $F(x)$  преобразуется к схеме с

независимыми или монотонно-независимыми выходами. Теория построения схем, обладающих подобными свойствами, представлена в [21–24]. Само устройство никогда не превышает по сложности систему дублирования, а в большинстве случаев для набора контрольных примеров *MCNC Benchmark Circuits* [25], специально разработанных для проверки эффективности различных способов построения надежных логических устройств, блок  $F(x)$  усложняется примерно на 10%.

Таким образом, аппаратные затраты для реализации системы функционального кон-

троля связаны в основном с блоком  $G(x)$  и тестером. Классический код с суммированием имеет  $k = \lceil \log_2(m+1) \rceil$  контрольных разрядов. Например, при  $m = 15$  число контрольных разрядов  $k = 4$ . Использование модульных кодов с суммированием при построении систем функционального контроля позволяет уменьшить сложность блока  $G(x)$  и тестера за счет уменьшения числа контрольных разрядов кода. Например, код с суммированием с модулем счета  $M = 4$  имеет при длине информационного вектора  $m = 15$  всего  $k = 2$  контрольных разряда. Схема функционального контроля будет более простой, чем схема, построенная по коду Бергера с тем же значением  $m$ .

Модульные коды (так же как и модульно-модифицированные и модифицированные

коды, описанные в разделе 2) имеют существенный недостаток при построении схем с независимыми и монотонно-независимыми выходами – ими могут быть не обнаружены некоторые однонаправленные ошибки, что связано с операцией расчета по модулю.

Для примера остановимся на рассмотрении одного из классов модульных кодов –  $S4(n, m)$ -кодов.

Любой модульный код есть частный случай классического кода с суммированием. Пусть, к примеру, имеется код  $S(9, 6)$ . Для определения структуры необнаруживаемых ошибок его удобно представлять в табличной форме (табл. 2), где каждому контрольному вектору поставлены в соответствие все информационные векторы [12]. Все ложные переходы векторов внутри одной контрольной

ТАБЛИЦА 2. Код  $S(9, 6)$ 

Контрольные группы							
000	001	010	011	100	101	110	111
Информационные векторы							
000000	000001	000011	000111	001111	011111	111111	
	000010	000101	001011	010111	101111		
	000100	000110	001101	011011	110111		
	001000	001001	001110	011101	111011		
	010000	001010	010011	011110	111101		
	100000	001100	010101	100111	111110		
		010001	010110	101011			
		010010	011001	101101			
		010100	011010	101110			
		011000	011100	110011			
		100001	100011	110101			
		100010	100101	110110			
		100100	100110	111001			
		101000	101001	111010			
		110000	101010	111100			
			101100				
		110001					
		110010					
		110100					
		111000					



группы будут относиться к необнаруживаемым. Поскольку в каждой контрольной группе кода Бергера находятся информационные векторы с одинаковым весом  $r$ , все необнаруживаемые ошибки принадлежат к классу разнонаправленных.

Операция взятия модуля, например для кода  $S_4(8,6)$ , «уплотняет» таблицу распределения информационных векторов на контрольные группы: все информационные векторы группы  $\langle 100 \rangle$  перемещаются в группу  $\langle 000 \rangle$ ,  $\langle 101 \rangle$  – в группу  $\langle 001 \rangle$  и  $\langle 110 \rangle$  – в группу  $\langle 010 \rangle$ . В табл. 3 представлен  $S_4(8, 6)$ -код.

Число векторов в контрольных группах увеличивается, что приводит к увеличению числа необнаруживаемых ошибок в модульном коде в сравнении с классическим, и при

ТАБЛИЦА 3. Код  $S_4(8, 6)$ 

Контрольные группы			
00	01	10	11
Информационные векторы			
000000	000001	000011	000111
001111	000010	000101	001011
010111	000100	000110	001101
011011	001000	001001	001110
011101	010000	001010	010011
011110	100000	001100	010101
100111	011111	010001	010110
101011	101111	010010	011001
101101	110111	010100	011010
101110	111011	011000	011100
110011	111101	100001	100011
110101	111110	100010	100101
110110		100100	100110
111001		101000	101001
111010		110000	101010
111100		111111	101100
			110001
			110010
			110100
			111000

этом появляются однонаправленные необнаруживаемые ошибки, или монотонные ошибки. Так, любой переход информационных векторов в группе  $\langle 00 \rangle$  с истинным весом  $r = 4$  в вектор с истинным весом  $r = 0$  будет монотонным:  $\langle 000000 \rangle \rightarrow \langle 001111 \rangle$ . Общее число однонаправленных необнаруживаемых ошибок в  $S_4(8,6)$ -коде равно 120. Поскольку это переходы в одной группе с различным весом  $r$ , но одинаковым по модулю  $M = 4$ , все однонаправленные ошибки – это ошибки кратности  $d = 4$ .

С помощью специально разработанного программного обеспечения по расчету характеристик кодов с суммированием было установлено следующее свойство любого модульного кода с суммированием.

Модульные коды с суммированием имеют однонаправленные необнаруживаемые ошибки только кратностей  $M, 2M, \dots, iM (iM \leq m)$ ; однонаправленных необнаруживаемых ошибок других кратностей у модульных кодов нет.

Таким образом, в любом  $S_4(n, m)$ -коде однонаправленные необнаруживаемые ошибки – это ошибки, кратность которых модулю счета  $M = 4$ .

В табл. 4 представлены расчеты общего количества необнаруживаемых ошибок по кратностям и монотонных необнаруживаемых ошибок для некоторых  $S_4(n, m)$ -кодов. В верхней половине каждой клетки записано общее число необнаруживаемых ошибок, в нижней – число монотонных необнаруживаемых ошибок.

Анализ табл. 4 показывает, что число необнаруживаемых монотонных ошибок в  $S_4(n, m)$ -кодах кратности  $d = 4$  составляет 25% от общего числа необнаруживаемых ошибок, а кратности  $d = 8$  – 1,56% от общего числа необнаруживаемых ошибок.

На рис. 2 представлена зависимость отношения числа однонаправленных необнаруживаемых ошибок от общего числа ошибок в  $S_4(n, m)$ -коде – величина  $\zeta(m)$ . Данная величина имеет максимум при значении длины информационного вектора  $m = 8$ . Таким образом, для модульных кодов  $S_4(n, m)$  значение  $\zeta(m)$  не превосходит 7%. В диапазоне

ТАБЛИЦА 4. Характеристики  $S4(n, m)$ -кодов при  $m = 2 \div 12$ 

$m$	$k$	Кратность необнаруживаемой ошибки, $d$												Общее число необнаруживаемых ошибок			
		1	2	3	4	5	6	7	8	9	10	11	12				
2	2	0	2														2
	0	0	0														0
3	2	0	12	0													12
	0	0	0	0													0
4	2	0	48	0	8												56
	0	0	0	0	2												2
5	2	0	160	0	80	0											240
	0	0	0	0	20	0											20
6	2	0	480	0	480	0	32										992
	0	0	0	0	120	0	0										120
7	2	0	1344	0	2240	0	448	0									4032
	0	0	0	0	560	0	0	0									560
8	2	0	3584	0	8960	0	3584	0	128								16256
	0	0	0	0	2240	0	0	0	2								2242
9	2	0	9216	0	32256	0	21504	0	2304	0							65280
	0	0	0	0	8064	0	0	0	36	0							8100
10	2	0	23040	0	107520	0	107520	0	23040	0	512						261632
	0	0	0	0	26880	0	0	0	360	0	0						27240
11	2	0	56320	0	337920	0	473088	0	168960	0	11264	0					1047552
	0	0	0	0	84480	0	0	0	2640	0	0	0					87120
12	2	0	135168	0	1013760	0	1892352	0	1013760	0	135168	0	2048				4192256
	0	0	0	0	253440	0	0	0	15840	0	0	0	2				269282



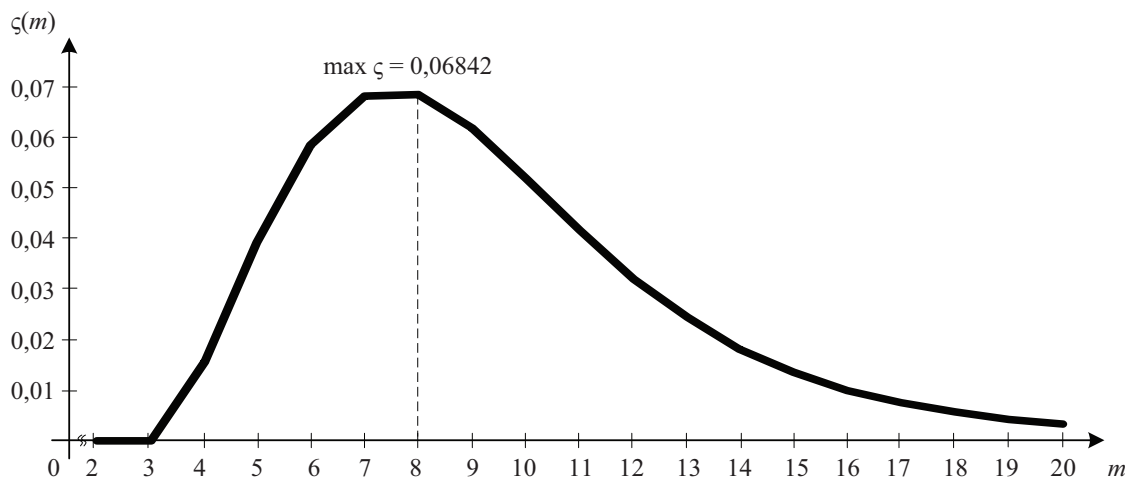


Рис. 2. Значение величины  $\zeta$  в зависимости от длины информационного вектора

$5 \div 13$  это  $0,07 < \zeta(m) < 0,02$ . Для остальных  $S4(n, m)$ -кодов доля однонаправленных необнаруживаемых ошибок составляет менее 2% от общего числа ошибок в информационных векторах кодов.

Свойство монотонных ошибок  $S4(n, m)$ -кодов быть кратными модулю счета  $M = 4$  можно использовать при построении систе-

мы функционального контроля схемы с независимыми или с монотонно-независимыми выходами. Например, очевидно, что схемы с  $m < 4$  легко контролируются с помощью  $S4(n, m)$ -кода. В случае контроля схемы с большим числом выходов можно осуществлять контроль по группам, имеющим  $m < 4$  выходов. На рис. 3 показана структура такого вариан-

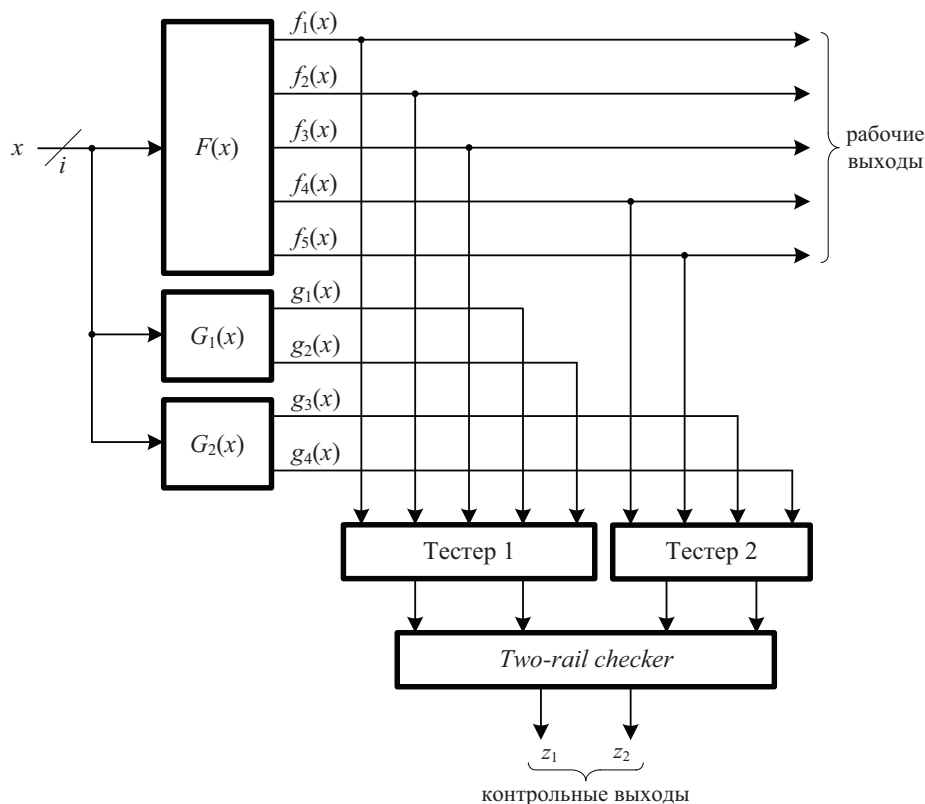


Рис. 3. Структура системы контроля по группам монотонно-независимых выходов

та контроля для произвольной схемы  $F(x)$ . В ней выделены два блока дополнительной логики для вычисления контрольных разрядов по группам выходов  $f_1(x), f_2(x), f_3(x)$  и  $f_4(x), f_5(x)$  – блоки  $G_1(x)$  и  $G_2(x)$ . Таким образом, тестер состоит из трех блоков – тестера первой группы, тестера второй группы и блока сравнения парафазных сигналов (*two-rail checker*) [9]. На выходах каждого тестера при соответствии информационных и контрольных выходов формируется парафазный сигнал  $\langle 10 \rangle$  или  $\langle 01 \rangle$ , соответственно парафазный сигнал формируется и на выходах блока сравнения парафазных сигналов. Любая ошибка в элементах всей системы контроля приведет к нарушению парафазности контрольных выходов.

Другим способом построения контролепригодной системы является следующий подход. Схема приводится к схеме с монотонно-независимыми выходами по алгоритму, изложенному в [24], после чего анализируются ошибки в структуре схемы  $F(x)$ . Если есть однократные искажения, влияющие на кратное четырем число выходов, рассматривается множество формируемых информационных векторов. В случае, если данное множество не дает возможности возникновения монотонных ошибок, система строится без дальнейшего преобразования, в противном случае – в исходной схеме вводится дополнительный выход, позволяющий увеличить кратность однонаправленной ошибки и обнаружить ее с помощью выбранного кода.

## Заключение

Рассмотрены некоторые особенности модульных кодов с суммированием (данные свойства присущи и модифицированным кодам с суммированием, описанным в разделе 2), позволяющие упрощать систему функционального контроля логического устройства с монотонными и монотонно-независимыми выходами.

Практическим результатом данной работы является предоставление разработчику более широкого выбора вариантов кодиро-

вания при построении системы диагностирования. Так, проведение эксперимента с блоком  $F(x)$  позволяет определить характер ошибок на выходах [25, 26] и соответственно выбрать оптимальный вариант кодирования с учетом свойств кодов по обнаружению искажений в информационных векторах.

## Библиографический список

1. **Goessel, M., Graf, S.** (1994). *Error Detection Circuits*, London, 261 p.
2. **Основы** технической диагностики / Вал. В. Сапожников, Вл. В. Сапожников. – Москва : Маршрут, 2004. – 316 с.
3. **Проектирование** и верификация цифровых систем на кристаллах. Verilog & System Verilog / В. И. Хаханов, И. В. Хаханова, Е. И. Литвинова, О. А. Гузь. – Харьков : Новое слово, 2010. – 528 с.
4. **Проектирование** и тестирование цифровых систем на кристаллах / В. И. Хаханов, Е. И. Литвинова, О. А. Гузь. – Харьков : ХНУРЭ, 2009. – 484 с.
5. **Göessel, M., Ocheretny, V., Sogomonyan, E., Marienfeld, D.** (2008). *New Methods of Concurrent Checking*, Dordrecht, 184 p.
6. **Основы** технической диагностики (оптимизация алгоритмов диагностирования, аппаратные средства) / П. П. Пархоменко, Е. С. Согомонян. – Москва : Энергоатомиздат, 1981. – 320 с.
7. **Обнаружение** несущественных путей логических схем на основе совместного анализа И-ИЛИ деревьев и SSBDD-графов / А. Ю. Матросова, С. А. Останин, В. Сингх // Автоматика и телемеханика. – 2013. – № 7. – С. 126–142.
8. **Hamming, R. W.** (1986). *Coding and Information Theory*, New-York, 272 p.
9. **Lala, P. K.** (2001). *Self-checking and Fault-tolerant Digital Design*, Arkansas, 216 p.
10. **Abramovici, M., Breuer, M. A., Friedman, A. D.** (1998). *Digital System Testing and Testable Design*, Washington, 652 p.
11. **Berger J. M.** (1961). *Information and Control*, 4 (3), 68–73.
12. **О свойствах** кода с суммированием в схемах функционального контроля / Д. В. Ефанов, Вал. В. Сапожников, Вл. В. Сапожников // Автоматика и телемеханика. – 2010. – № 6. – С. 155–162.

13. **Применение** кода с суммированием в системах технической диагностики и мониторинга устройств железнодорожной автоматики и телемеханики : дис. ... канд. техн. наук : 05.22.08 : защищена 10.11.10 : утв. 11.03.11 / Д. В. Ефанов. – Санкт-Петербург, 2010. – 192 с.
14. **Построение** модифицированного кода Бергера с минимальным числом необнаруживаемых ошибок информационных разрядов / А. А. Блюдов, Д. В. Ефанов, Вал. В. Сапожников, Вл. В. Сапожников // *Электронное моделирование*. – 2012. – Т. 34. – № 6. – С. 17–29.
15. **Sapozhnikov, V.**, Sapozhnikov, V., Efanov, D., Blyudov, A. (2013). *Proceedings of IEEE East-West Design&Test Symposium*, Rostov-on-Don, 200–207.
16. **Bose, B.**, Lin, D.J. (1985). *IEEE Transactions on Computers*, C-34, 1026–1032.
17. **Das, D.**, Toubia, N.A. (1999). *Journal of Electronic Testing: Theory and Applications*, 15 (1–2), 145–155.
18. **Blyudov, A.**, Efanov, D., Sapozhnikov, V., Sapozhnikov, V. (2012). *Proceedings of 10<sup>th</sup> IEEE East-West Design&Test Symposium*, Kharkiv, 114–117.
19. **Коды** с суммированием для организации контроля комбинационных схем / А. А. Блюдов, Д. В. Ефанов, Вал. В. Сапожников, Вл. В. Сапожников // *Автоматика и телемеханика*. – 2013. – № 6. – С. 153–164.
20. **Efanov, D.**, Sapozhnikov, V., Sapozhnikov, V., Blyudov, A. (2013). *Proceedings of IEEE East-West Design&Test Symposium*, Rostov-on-Don, 261–266.
21. **Построение** комбинационных самопроверяемых устройств с монотонно независимыми выходами / М. Гёссель, А. А. Морозов, Вал. В. Сапожников, Вл. В. Сапожников // *Автоматика и телемеханика*. – 1994. – № 7. – С. 148–160.
22. **Исследование** комбинационных самопроверяемых устройств с независимыми и монотонно независимыми выходами / М. Гёссель, А. А. Морозов, Вал. В. Сапожников, Вл. В. Сапожников // *Автоматика и телемеханика*. – 1997. – № 2. – С. 180–193.
23. **Morozov, A.**, Sapozhnikov, V., Sapozhnikov, V., Goessel, M. (1998). Self-Checking Combinational Circuits with Unidirectionally Independent Outputs. *VLSI Design*, 5 (4), 333–345.
24. **Метод** построения комбинационных самопроверяемых устройств с обнаружением всех одиночных неисправностей / Вал. В. Сапожников, Вл. В. Сапожников, М. Гёссель, А. А. Морозов // *Электронное моделирование*. – 1998. – Т. 20. – № 6. – С. 70–80.
25. **Yang, S.** (1991). *Logic Synthesis and Optimization Benchmarks User guide version 3.0*, North Carolina, 44 p.
26. **Moshanin, V.**, Ocheretnij, V., Dmitriev, A. (1998). *Proceedings of 4th IEEE International On-Line Testing Workshop*, Capri, 81–84.