

ТРОИЧНЫЕ КОДЫ С СУММИРОВАНИЕМ ДЛЯ КОНТРОЛЯ ЦИФРОВЫХ СХЕМ

Д.В. Ефанов

Аннотация. Обсуждены особенности защиты данных, представленных в троичной форме. Подчеркнута актуальность применения троичной логики в будущем для решения задач разработки надежных цифровых устройств и систем. Описан троичный код с суммированием, обнаруживающий любые монотонные и асимметричные ошибки в информационных векторах, применение которого перспективно при построении устройств и систем с обнаружением неисправностей. Отмечено, что принципы построения троичного кода с суммированием аналогичны принципам построения классических двоичных кодов с суммированием (кодов Бергера). Установлены ранее неизвестные свойства троичных кодов с суммированием на случай возникновения ошибок только в информационных векторах при безошибочности контрольных разрядов. Показано, что такая задача актуальна в практических приложениях, в которых разряды контрольных и информационных векторов вычисляются физически различными устройствами. Указано, что учет установленных свойств троичных кодов с суммированием может быть полезен при выборе способа защиты как самих данных, так и парирования возникающих в устройствах автоматики неисправностей.

Ключевые слова: троичная логика, троичная техника, контроль вычислений, защита данных, троичный код с суммированием, код Бергера, необнаруживаемая ошибка, характеристики обнаружения ошибок.

ВВЕДЕНИЕ

Современные цифровые устройства и системы работают в двоичной логике. Тем не менее, во всем мире велись и ведутся исследования в области построения и применения устройств и систем, работающих в троичной логике [1–7]. Во многих источниках отмечаются преимущества троичной логики в более плотной записи чисел и возможностях ее применения.

Устройства, работающие в троичной логике, были построены еще во второй половине прошлого столетия [1]. Известны и современные реализации устройств, работающие на аналогичных принципах [8]. Некоторые исследователи отмечают преимущества троичной логики по сравнению с двоичной при реализации квантовых компьютеров [9]. Указанные примеры подчеркивают актуальность исследований в области применения троичной логики при построении цифровых устройств и систем.

Широко известно, что при построении современных надежных и безопасных устройств и систем автоматизации применяются методы избыточного кодирования, резервирования, диверсифицирования как аппаратных, так и программных средств и многие другие [10–14]. Все эти методы применяются на различных уровнях архитектуры устройств и систем (и на микроуровне при использовании высоконадежной элементной базы, и на макроуровне при обеспечении надежности систем при передаче данных между компонентами и объектами управления). Применяют избыточное кодирование и при решении задач синтеза контролепригодных цифровых устройств и систем и технических средств их диагностирования [15–17]. В подобных приложениях часто применяются коды, ориентированные на обнаружение ошибок, а не на их исправление. Прежде всего, коды с исправлением ошибок имеют несколько большую избыточность, чем коды с обнаружением ошибок [18]. Кроме того, исправление ошибок в аппаратных средствах может приводить к их накоплению и



последующему критическому отказу системы. Повсеместно применяются равновесные коды, разнообразие коды с суммированием и коды с проверками четности значений разрядов [19, 20].

В настоящей статье внимание сконцентрировано на изложении результатов исследования автора в области построения троичных избыточных кодов, предназначенных для решения задач синтеза контролепригодных цифровых устройств и систем и изучения их характеристик. Автором поставлена задача построения троичных кодов с суммированием, обнаруживающих любые монотонные и асимметричные ошибки в информационных векторах, а также установления их основных характеристик, учет которых целесообразно вести при синтезе надежных и безопасных устройств и систем, функционирующих в троичной логике. Особенности же процедур синтеза самих цифровых устройств и систем с применением избыточных кодов (и, в частности, кодов с суммированием), также как и вопросы правильности обработки информации, в настоящей работе обойдены. Данные аспекты требуют глубоких исследований и отдельных публикаций, освещающих их результаты.

1. ПРИНЦИПЫ ПОСТРОЕНИЯ ТРОИЧНОГО КОДА С СУММИРОВАНИЕМ

Существует большое разнообразие способов построения троичных помехоустойчивых кодов [21–25].

При построении кода можно пойти несколькими путями. Один из них состоит в эвристическом составлении правил получения контрольных разрядов. Другой — в установлении первоначальных основных свойств «будущего» кода.

В троичной логике для представления сигналов используются три символа. Обозначим их как 0, 1 и 2 без привязки к виду троичной системы счисления (симметричная или несимметричная). Дальнейшие результаты связаны только с троичным представлением сигналов, но не с троичной логикой как таковой. Построим такой троичный код, который будет обнаруживать любые ошибки, связанные с нарушением числа разрядов, равных 1 и 2, в информационных векторах.

Ошибки в троичных кодовых векторах могут быть классифицированы в общем случае таким образом. Ошибки делятся на одиночные и многократные. Первый вид ошибок — это ошибки, связанные с одиночными искажениями в кодовых словах, второй — с одновременным возникновением искажений двух и более разрядов. В свою очередь, многократные ошибки в кодовых векторах троичной логики делятся на композиционные,

монотонные и асимметричные ошибки. *Композиционной* называется такая ошибка, при которой в искаженном кодовом векторе сохраняется число разрядов, равное 1 и 2 в искажаемом кодовом векторе (сохраняется «композиция» разрядов). Остальные искажения приводят к возникновению либо монотонных, либо асимметричных ошибок. Под *монотонной* ошибкой в троичном кодовом векторе понимается такая ошибка, при которой сохраняется приоритет значений, определенный в каждом из известных классов троичных монотонных функций [3]. Напомним читателю, что в троичной логике известно три класса монотонных функций (классы M_1 , M_2 и M_3). Класс M_1 — монотонные функции, для которых при сравнении аргументов принят порядок $0 < 1 < 2$. Класс M_2 — монотонные функции, для которых при сравнении аргументов принят порядок $1 < 2 < 0$. Класс M_3 — монотонные функции, для которых при сравнении аргументов принят порядок $2 < 0 < 1$. Монотонные ошибки в троичной логике могут быть классифицированы на подвиды (однаправленная, двунаправленная и др.). Ошибки с любыми другими видами искажений принадлежат к виду *асимметричных*. Такое разделение ошибок связано именно с особенностями их возникновения на выходах цифровых устройств и обнаружения с помощью избыточных кодов.

Построим такой троичный код, который будет обнаруживать любые монотонные и асимметричные ошибки. Такой код будет принадлежать к классу *MAED*-кодов (monotonous & asymmetrical error-detection codes). Коды из данного класса можно использовать при синтезе контролепригодных структур устройств, работающих в троичной логике, а также технических средств их диагностирования.

Троичный код, относящийся к классу кодов с обнаружением любых монотонных и асимметричных ошибок, строится следующим образом.

Алгоритм. *Правила определения значений разрядов контрольных векторов троичных кодов с суммированием:*

1. В информационном векторе длиной m определяется число разрядов, равных 1, и число разрядов, равных 2, — числа r_1 и r_2 соответственно.

2. Число r_1 представляется в троичном виде и записывается в $k_1 = \lceil \log_3(m + 1) \rceil$ старших разрядах контрольного вектора (запись $\lceil \dots \rceil$ обозначает целое сверху от вычисляемого значения).

3. Число r_2 представляется в троичном виде и записывается в $k_2 = \lceil \log_3(m + 1) \rceil$ младших разрядах контрольного вектора.

Так как числа r_1 и r_2 определяются путем применения операций суммирования, назовем строящийся по представленному алгоритму код кодом с

Таблица 1

Число контрольных разрядов в $\Sigma(m, k)$ -кодах

m	k
4	4
5	4
6	4
7	4
8	4
9	6
10	6
...	...
20	6
...	...
50	8
...	...
100	10

суммированием и обозначим как $\Sigma(m, k)$ -код, где $k = k_1 + k_2 = 2\lceil \log_3(m + 1) \rceil$ — число разрядов в контрольных векторах. В табл. 1 приведены значения числа k для $\Sigma(m, k)$ -кодов с различной длиной информационного вектора. Из таблицы следует, что имеет смысл строить $\Sigma(m, k)$ -коды для случаев $m > 4$ (в противном случае можно ограничиться применением троичных кодов с повторением, обнаруживающих любые ошибки в информационных векторах).

Приведем пример получения значений разрядов контрольного вектора для информационного вектора $\langle 01012121222 \rangle$. Длина информационного вектора $m = 11$. Отсюда следует, что числа $k_1 = k_2 = \lceil \log_3 12 \rceil = 3$, а $k = 6$. Числу $r_1 = 4$ соответствует троичный вектор $\langle 011 \rangle$. Числу $r_2 = 5$ соответствует троичный вектор $\langle 012 \rangle$. Таким образом, контрольный вектор $\Sigma(12, 6)$ -кода будет иметь вид $\langle 011012 \rangle$.

2. НЕКОТОРЫЕ СВОЙСТВА ТРОИЧНЫХ КОДОВ С СУММИРОВАНИЕМ

Рассмотрим особенности обнаружения ошибок в информационных векторах $\Sigma(m, k)$ -кодами.

Прежде всего, отметим, что $\Sigma(m, k)$ -кодами, за исключением кодов с длинами информационных векторов $m = 3^p - 1$, $p \in \{2, 3, 4, \dots\}$, не используются все возможные комбинации разрядов в контрольных векторах. Назовем $\Sigma(m, k)$ -коды с длинами информационных векторов $m = 3^p - 1$, $p \in \{2, 3, 4, \dots\}$, совершенными троичными кодами с суммированием.

Утверждение 1. $\Sigma(m, k)$ -коды обнаруживают любые монотонные и асимметричные ошибки в информационных векторах и не обнаруживают любые композиционные ошибки. ♦

Справедливость утверждения 1 вытекает из принципов построения $\Sigma(m, k)$ -кода. Фактически информационные векторы при построении кода классифицируются на контрольные группы, соответствующие числам r_1 и r_2 ($r_1 - r_2$ группам), определяющие все информационные векторы с одной и той же композицией [26]. Ошибка будет необнаруживаемой в том случае, если она исказит информационный вектор, принадлежащий одной контрольной группе, в информационный вектор, принадлежащий той же контрольной группе. Такая ошибка не нарушает композиции значений и, как отмечено выше, является композиционной. Отсюда следует, что ошибки композиционного типа не могут быть обнаружены $\Sigma(m, k)$ -кодами. Если же ошибка переведет информационный вектор одной контрольной группы в информационный вектор другой контрольной группы, то она будет связана с нарушением композиции значений в информационном векторе. Такая ошибка будет обнаружена и будет принадлежать к типу монотонных или асимметричных ошибок.

Для определения характеристик обнаружения ошибок $\Sigma(m, k)$ -кодами целесообразно воспользоваться табличной формой задания кода, в которой все информационные векторы классифицируются по всем возможным $r_1 - r_2$ группам. Анализируя такие $r_1 - r_2$ группы, можно установить особенности обнаружения ошибок $\Sigma(m, k)$ -кодами. Для примера в табл. 2 задан $\Sigma(4, 4)$ -код (данный код, хоть и имеет двойную избыточность, выбран для иллюстрации особенностей троичных кодов с суммированием как код с наименьшей мощностью множества кодовых слов).

Таблица задания $\Sigma(4, 4)$ -кода включает в себя информационные векторы, разбитые на три категории. Первая из них включает в себя те, для которых разряды принимают только значения 0 и 1. Такие векторы помещены в группы $r_1 - 0$. Вторая категория информационных векторов включает в себя те из них, для которых разряды принимают только значения 0 и 2 и занимают группы $0 - r_2$. Третья категория включает в себя информационные векторы, значения разрядов в которых равны 0, 1 и 2. При этом в таблице для наглядности и выделения категорий векторов потребовалось сдвинуть все векторы, образованные из всех сочетаний значений 0 и 2, а также все векторы, образованные из всех сочетаний 0, 1 и 2 вниз внутри каждой контрольной группы. В результате такого сдвига и в результате того, что распределение векторов между контрольными группами неравномерно, в таблице появляется большое число пустых ячеек. Такое представление кода не только удобно с пози-

Распределение информационных векторов в контрольные группы $\Sigma(4, 4)$ -кода

Контрольные группы (r_1-r_2 группы)														
00-00	00-01	00-02	00-10	00-11	01-00	01-01	01-02	01-10	02-00	02-01	02-02	10-00	10-01	11-00
Информационные векторы														
0000					0001 0010 0100 1000				0011 0101 0110 1001 1010 1100			0111 1011 1101 1110		1111
	0002 0020 0200 2000	0022 0202 0220 2002 2020 2200	0222 2022 2202 2220	2222										
						0012 0021 0102 0120 0201 0210 1002 1020 1200 2001 2010 2100	0122 0212 0221 1022 1202 1220 2012 2021 2102 2120 2201 2210	1222 2122 2212 2221		0112 0121 0211 1012 1021 1102 1120 1201 1210 2011 2101 2110	1122 1212 1221 2112 2121 2211		1112 1121 1211 2111	

ций чтения таблицы, но и позволяет установить основные характеристики обнаружения ошибок $\Sigma(m, k)$ -кодами.

Отметим также, что можно отдельно рассматривать коды с суммированием, для которых используются только информационные векторы, входящие в одну из трех категорий информационных векторов — $\Sigma^1(m, k)$, $\Sigma^2(m, k)$ и $\Sigma^3(m, k)$ коды. Если, например, рассматривать $\Sigma^1(m, k)$ -код, соответствующий распределению информационных векторов только первой категории, то можно сказать, что построен классический двоичный код с суммированием (код Бергера) [27].

Утверждение 2. *Необнаруживаемые ошибки, происходящие в информационных векторах, входящих в r_1-0 и $0-r_2$ группы, могут иметь только четную кратность. ♦*

Формулировка утверждения 2 обосновывается следующими соображениями. Каждая r_1-r_2 группа для рассматриваемых случаев определяется только числом r_1 (или только числом r_2). Чтобы ошибка

была необнаруживаемой, она должна перевести информационные векторы конкретной контрольной группы друг в друга. В этом случае не должно нарушиться общее число разрядов, равных 1 (или равных 2). Это возможно только в том случае, если любое искажение 1 (или 2) будет скомпенсировано противоположным искажением нулевого разряда. Отсюда следует, что ошибка будет иметь только четную кратность.

Необнаруживаемые ошибки для контрольных групп, характеризующих информационные векторы третьей категории, могут иметь любую кратность $d \geq 2$.

Из табл. 2 следует способ подсчета числа необнаруживаемых $\Sigma(m, k)$ -кодом ошибок: необходимо определить для заданного значения m общее число информационных векторов, соответствующих каждой r_1-r_2 группе. Обозначим это число как Q_{r_0, r_1, r_2} , где r_0, r_1 и r_2 — число разрядов, равных 0, 1 и 2 соответственно. В табл. 3 приведены представители контрольных групп $\Sigma(4, 4)$ -кода. Число

информационных векторов, соответствующих каждой контрольной группе, можно определить по формуле:

$$Q_{r_0, r_1, r_2} = C_m^{r_1} C_{m-r_1}^{r_2} C_{m-(r_1+r_2)}^{r_0}, \quad (1)$$

где $C_m^{r_1}$ — число вариантов расположения разрядов, равных 1, в информационном векторе длины m ; $C_{m-r_1}^{r_2}$ — число вариантов расположения разрядов, равных 2, в оставшихся $m - r_1$ информационных разрядах; $C_{m-(r_1+r_2)}^{r_0}$ — число вариантов расположения разрядов, равных 0, в оставшихся $m - (r_1 + r_2)$ информационных разрядах.

Учитывая, что $m - (r_1 + r_2) = r_0$ и принимая во внимание, что $C_{r_0}^{r_0} = 1$, формулу (1) перепишем в виде $Q_{r_0, r_1, r_2} = C_m^{r_1} C_{m-r_1}^{r_2}$.

Таблица 3

 Представители контрольных групп $\Sigma(4,4)$ -кода

r_1-r_2 группа	Представители контрольных групп	Общее число представителей	Формула подсчета
00-00	0000	1	$C_4^0 C_4^0$
00-01	0002	4	$C_4^0 C_4^1$
00-02	0022	6	$C_4^0 C_4^2$
00-10	0222	4	$C_4^0 C_4^3$
00-11	2222	1	$C_4^0 C_4^4$
01-00	0001	4	$C_4^1 C_3^0$
01-01	0012	12	$C_4^1 C_3^1$
01-02	0122	12	$C_4^1 C_3^2$
01-10	1222	4	$C_4^1 C_3^3$
02-00	0011	6	$C_4^2 C_2^0$
02-01	0112	12	$C_4^2 C_2^1$
02-02	1122	6	$C_4^2 C_2^2$
10-00	0111	4	$C_4^3 C_1^0$
10-01	1112	4	$C_4^3 C_1^1$
11-00	1111	1	$C_4^4 C_0^0$

Тогда число необнаруживаемых ошибок в каждой контрольной группе будет определяться величиной $N_{r_0, r_1, r_2} = C_m^{r_1} C_{m-r_1}^{r_2} (C_m^{r_1} C_{m-r_1}^{r_2} - 1)$.

Общее же число необнаруживаемых ошибок равно сумме всех необнаруживаемых ошибок, «даваемых» каждой контрольной группой:

$$N_{m,k} = \sum_{r_1, r_2=0}^{r_1, r_2=m} N_{r_0, r_1, r_2} = \sum_{r_1=0}^{r_1=m} \left(\sum_{r_2=0}^{r_2=m} C_m^{r_1} C_{m-r_1}^{r_2} (C_m^{r_1} C_{m-r_1}^{r_2} - 1) \right). \quad (2)$$

Например, формула (2) для подсчета общего числа необнаруживаемых $\Sigma(4, 4)$ -кодом ошибок, дает результат: $N_{4,0,0} = 0$, $N_{3,0,1} = 12$, $N_{2,0,2} = 30$, $N_{1,0,3} = 12$, $N_{0,0,4} = 0$, $N_{3,1,0} = 12$, $N_{2,1,1} = 132$, $N_{1,1,2} = 132$, $N_{0,1,3} = 12$, $N_{2,2,0} = 30$, $N_{1,2,1} = 132$, $N_{0,2,2} = 30$, $N_{1,3,0} = 12$, $N_{0,3,1} = 12$, $N_{0,4,0} = 0$. Суммируя полученные числа, получаем: $N_{4,4} = 558$.

С увеличением длины информационного вектора на единицу увеличивается и число контрольных групп (представителей контрольных групп). Причем прослеживается такая закономерность: в группе 00—00 всегда присутствует один вектор, в группах 00— $r_2 - m$, 01— $r_2 - m$, 02— $r_2 - m - 1$, 11— $r_2 - m - 2$, ..., $r_1 - 00 - 1$ векторов. Исходя из этого, общее число контрольных групп (и различных композиций) определяется по формуле:

$$R_m = 1 + m + m + (m - 1) + (m - 2) + \dots + 2 + 1 = m + 1 + \frac{m(m+1)}{2} = \frac{m^2 + 3m + 2}{2}.$$

Например, для рассматриваемого $\Sigma(4,4)$ -кода $R_4 = \frac{4^2 + 3 \cdot 4 + 2}{2} = 15$.

В табл. 4 приведены рассчитанные значения числа R_m для различных $\Sigma(m, k)$ -кодов.

Введем в рассмотрение относительные показатели обнаружения ошибок $\Sigma(m, k)$ -кодами.

Общее число ошибок, которые могут возникнуть в информационных векторах троичных кодов, равно [28]: $N_m = 3^m(3^m - 1)$.

Показатель $\gamma_{m,k}$ позволяет сравнивать число необнаруживаемых кодами ошибок с общим возможным их числом:

$$\gamma_{m,k} = \frac{N_{m,k}}{N_m} \cdot 100 \%. \quad (3)$$



Таблица 4

Числа R_m для различных $\Sigma(m, k)$ -кодов

m	R_m
4	15
5	21
6	28
7	36
8	45
9	55
10	66
...	...
20	231
...	...
50	1326
...	...
100	5151

Чем ближе значение $\gamma_{m,k}$ к значению 0 %, тем больше ошибок обнаруживает рассматриваемый код.

Как показано в работе [28], существует такой троичный разделимый код, который обладает равномерным распределением информационных векторов на контрольные группы. Такой код обладает очень важной особенностью.

Теорема 1. *Троичный код с параметрами m и k обладает минимальным общим количеством необнаруживаемых ошибок при условии, что все 3^m информационных вектора равномерно распределены между всеми 3^k контрольными векторами, а общее число необнаруживаемых ошибок в таком коде определяется по формуле*

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

Таким образом, любой разделимый код, в том числе $\Sigma(m, k)$ -код, можно сравнивать с кодом с равномерным распределением информационных векторов на контрольные группы:

$$\xi_{m,k} = \frac{N_{m,k}}{N_{m,k}^{\min}} \cdot 100 \%. \quad (4)$$

Показатель $\xi_{m,k}$ характеризует эффективность использования контрольных разрядов троичным разделимым кодом, чем ближе он к 100 %, тем эффективнее код обнаруживает ошибки.

Из сравнения формул (3) и (4) следует, что между коэффициентами $\gamma_{m,k}$ и $\xi_{m,k}$ имеется зависимость:

$$\xi_{m,k} = \gamma_{m,k} \frac{N_m}{N_{m,k}^{\min}} = \gamma_{m,k} \frac{3^m(3^m - 1)}{3^m(3^{m-k} - 1)} = \gamma_{m,k} \frac{3^m - 1}{3^{m-k} - 1}.$$

Тем не менее, далее обоими коэффициентами будем пользоваться наравне, так как они дают возможность наглядного представления о характеристиках $\Sigma(m, k)$ -кодов.

В табл. 5 представлены рассчитанные по приведенным выше формулам показатели $\Sigma(m, k)$ -кодов с малыми значениями длин информационных векторов. Графики зависимостей величин $\gamma_{m,k}$ и $\xi_{m,k}$ от m дополняют приведенные рассчитанные числа.

С увеличением длины информационного вектора доля необнаруживаемых ошибок от общего их числа уменьшается (рис. 1). Что касается эффективности использования контрольных разрядов (рис. 2), то с увеличением длины информационного вектора до величины $m = 3^p - 1$, $p \in \{2, 3, 4, \dots\}$, соответствующей значению длины информационного вектора совершенного $\Sigma(m, k)$ -кода,

Таблица 5

Характеристики обнаружения ошибок $\Sigma(m, k)$ -кодами

m	k	$N_{m,k}$	N_m	$N_{m,k}^{\min}$	$\gamma_{m,k}$	$\xi_{m,k}$
4	4	558	6 480	0	8,611	0
5	4	4 410	58 806	486	7,499	11,02
6	4	34 440	530 712	5 832	6,489	16,934
7	4	270 648	4 780 782	56 862	5,661	21,01
8	4	2 151 198	43 040 160	524 880	4,998	24,399
9	6	17 300 154	387 400 806	511 758	4,466	2,958
10	6	140 609 016	3 486 725 352	4 723 920	4,033	3,36
11	6	1 153 285 848	31 380 882 462	42 869 574	3,675	3,717
12	6	9 533 107 584	282 429 005 040	386 889 048	3,375	4,058
13	6	79 324 972 272	2 541 864 234 006	3 485 190 078	3,121	4,394
14	6	663 830 247 366	22 876 787 671 992	31 376 276 640	2,902	4,727
15	6	5 582 710 119 186	205 891 117 745 742	282 415 187 574	2,711	5,059

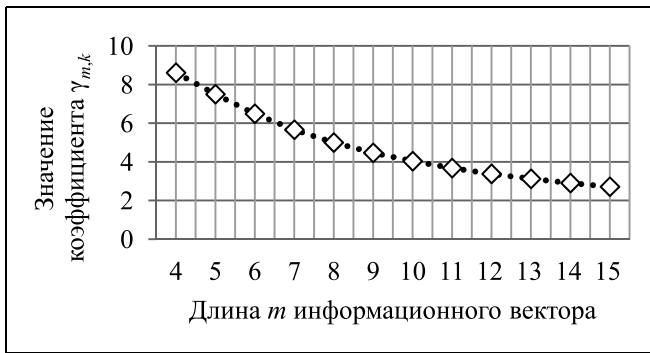

 Рис. 1. Зависимость коэффициента $\gamma_{m,k}$ от значения m

 Рис. 2. Зависимость коэффициента $\xi_{m,k}$ от значения m

происходит ее повышение. При достижении величины $m = 3^p$, $p \in \{2, 3, 4, \dots\}$, происходит резкое уменьшение значения коэффициента $\xi_{m,k}$, что объясняется увеличением числа контрольных разрядов на 2 и крайне большим числом неиспользуемых контрольных групп. Затем постепенно с увеличением длины информационного вектора растет число информационных векторов, заполняется большее число контрольных групп, значение коэффициента $\xi_{m,k}$ повышается.

Из рис. 1 и 2 следует, что $\Sigma(m, k)$ -коды не обнаруживают менее 10 % ошибок в информационных векторах при любой их длине. Более того, при $m \geq 8$ доля необнаруживаемых ошибок от общего их количества составляет менее 5 %. Тем не менее, контрольные разряды используются ими весьма неэффективно, о чем говорит график величины $\xi_{m,k}$: значения данного коэффициента не превышают 25 % для кодов с любыми значениями m .

Приведенные выше формулы, к сожалению, не дают возможности подсчета числа необнару-

живаемых $\Sigma(m, k)$ -кодами ошибок каждой кратностью. Для этого требуется детальный анализ задающих коды таблиц. Результаты такого анализа для $\Sigma(m, k)$ -кодов с малыми значениями длин информационных векторов представлены в табл. 6. Для каждого значения m приведены данные о числе необнаруживаемых ошибок данной кратностью d (верхнее число), общем числе ошибок данной кратностью (среднее число) и доле необнаруживаемых ошибок данной кратностью от общего количества ошибок данной кратностью — величине β_d , % (нижнее число).

Расчеты показывают, что $\Sigma(m, k)$ -коды (как и их двоичные аналоги [29]) обладают интересной особенностью.

Теорема 2. Доля необнаруживаемых троичными кодами с суммированием ошибок кратностью d от общего числа ошибок данной кратностью не зависит от длины информационного вектора и является постоянной величиной. ♦

Таким образом, любые $\Sigma(m, k)$ -коды не обнаруживают 16,667 % двукратных ошибок, 5,556 % трехкратных ошибок, 6,944 % четырехкратных ошибок и т. д. Однако, следует отметить, что положение теоремы 2 является гипотезой, которая подтверждается расчетами, но к настоящему времени математически не доказана.

Сравнивая между собой двоичные и троичные коды с суммированием, отметим, что у троичных кодов с суммированием в классе необнаруживаемых имеются ошибки как четной, так и нечетной кратностью, а доли необнаруживаемых ошибок каждой кратностью d от общего числа ошибок

Таблица 6

 Характеристики обнаружения $\Sigma(m, k)$ -кодами ошибок по кратностям

m	d					
	1	2	3	4	5	6
3	0	54	12	—	—	—
	162	324	216	—	—	—
	0	16,667	5,556	—	—	—
4	0	324	144	90	—	—
	648	1 944	2592	296	—	—
	0	16,667	5,556	6,944	—	—
5	0	1 620	1080	1 350	360	—
	2430	9 720	19 440	19 440	7776	—
	0	16,667	5,556	6,944	4,63	—
6	0	7290	6480	12 150	6480	2040
	8748	43 740	116 640	174 960	139 968	466 56
	0	16,667	5,556	6,944	4,63	4,372



данной кратностью значительно ниже, чем у двоичных кодов с суммированием (к примеру, двоичные коды с суммированием не обнаруживают 50 % двукратных ошибок [29], а троичные — 16,667 %).

ЗАКЛЮЧЕНИЕ

Троичные коды с суммированием, обнаруживающие любые монотонные и асимметричные ошибки, могут применяться при построении цифровых устройств и систем, работающих в троичной логике. Ими не обнаруживаются любые композиционные ошибки, которые, как установлено в настоящей статье, составляют для всех $\Sigma(m, k)$ -кодов с $m \geq 8$ менее 5 % от общего числа ошибок в информационных векторах. Если сравнивать троичный код с суммированием с двоичным [29], то можно отметить куда большее значение доли обнаруживаемых им ошибок.

Отметим некоторую особенность $\Sigma(m, k)$ -кода. Все возможные контрольные векторы ими используются только в частных случаях: для каждого значения k существует только одно значение длины информационного вектора $m = 3^p - 1$, $p \in \{2, 3, 4, \dots\}$, при которой формируются все возможные контрольные векторы. Все остальные значения длин информационных векторов не позволяют строить коды, для которых формируются все возможные контрольные векторы. Это усложняет задачу синтеза контролепригодных устройств на их основе, но может учитываться при построении иных троичных кодов, ориентированных на обнаружение ошибок в информационных векторах. Один из способов построения последних состоит в подсчете чисел r_1 и r_2 в кольце вычетов по модулю

$M \in \{3^1, 3^2, \dots, 3^{\lceil \log_3(m+1) \rceil - 1}\}$ и записи получаемых значений в разряды троичного контрольного вектора (аналог двоичных модульных кодов с суммированием [30, 31]). Например, перспективным является использование модульных кодов, для которых числа r_1 и r_2 подсчитываются в кольце вычетов по модулю $M = 9$. Число контрольных разрядов в таких троичных модульных кодах будет всегда равно $k = 4$, для них гораздо проще сформировать полное множество контрольных векторов, при этом будут обнаруживаться любые монотонные ошибки в информационных векторах кратностью $d_{\mu} < 9$. По этой причине их применение может оказаться также перспективным.

Дальнейшие исследования могут быть направлены на изучение особенностей различных модификаций $\Sigma(m, k)$ -кода, а также установление основных характеристик обнаружения ими ошибок в информационных векторах.

ЛИТЕРАТУРА

1. Брусенцов Н.П., Маслов С.П., Розин В.П., Тишулина А.М. Малая цифровая вычислительная машина «Сетунь». — М.: Изд-во МГУ, 1962. — 140 с. [Brusentsov, N.P., Maslov, S.P., Rozin, V.P., Tishulina, A.M. Small digital computer «Setun». — Moscow: Publishing House of Moscow State University, 1962. — 140 p. (in Russian)].
2. Merrill, R.D. Ternary Logic in Digital Computers // Proceedings of the SHARE design automation project (DAC'65). — ACM New York, NY, USA. — P. 6.1–6.17. — DOI: 10.1145/800266.810759.
3. Поспелов Д.А. Логические методы анализа и синтеза схем. — М.: Энергия, 1974. — 368 с. [Pospelov, D.A. Logical Methods of Analysis and Synthesis of Circuits. — Moscow: Energy, 1974. — 368 p. (in Russian)].
4. Rahman, Md.R., Rise, J.E. On Designing a Ternary Reversible Circuit for Online Testability // IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. — 2011. — P. 1–7. — DOI: 10.1109/PACRIM.2011.6032878.
5. Nayeem, N.M., Rice, J.E. Design of an Online Testable Ternary Circuit from the Truth Table // Lecture Notes in Computer Science book series (LNCS, vol. 7581): Reversible Computation, 4th International Workshop on Reversible Computation (RC 2012), Copenhagen, Denmark, July 2–3. — 2012. — P. 152–159.
6. Vudadha, C., Katragadda, S., Phaneendra, P.S. 2:1 Multiplexer Based Design for Ternary Logic Circuits // IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), 19–21 December 2013, Visakhapatnam, India. — P. 46–51. — DOI: 10.1109/PrimeAsia.2013.6731176.
7. Nair, R.S.P., Smith, S.C., Di, J. Delay Insensitive Ternary CMOS Logic for Secure Hardware // Journal of Low Power Electronics and Applications. — 2015. — Iss. 5. — P. 183–215. — DOI: 10.3390/jlpea5030183.
8. Connely, J. Ternary Computing Testbed 3-Trit Computer Architecture. — California Polytechnic State University of San Luis Obispo, August 29th, 2008. — 184 p.
9. Lanyon, B.P., Barbieri, M., Almeida, M.P., et al. Simplifying Quantum Logic Using Higher-Dimensional Hilbert Spaces // Nature Physics. — 2009 — Vol. 5, iss. 2. — P. 134–140. — DOI: 10.1038/nphys1150.
10. Гавзов Д.В., Сапожников В.В., Сапожников Вл.В. Методы обеспечения безопасности дискретных систем // Автоматика и телемеханика. — 1994. — № 8. — С. 3–50. [Gavzov, D.V., Sapozhnikov, V.V., Sapozhnikov, Vl.V. Methods for Providing Safety in Discrete Systems // Automation and Remote Control. — 1994. — Vol. 55, iss. 8. — P. 1085–1122].
11. Smith, D.J., Simpson, K.G.L. Functional Safety: A Straightforward Guide to IEC 61508 and Related Standards. — Butterworth-Heinemann, 1st edition (June 26, 2001). — 208 p.
12. Дрозд А.В., Харченко В.С., Антошук С.Г., и др. Рабочее диагностирование безопасных информационно-управляющих систем / под ред. А.В. Дрозда и В.С. Харченко. — Харьков: Нац. аэрокосм. ун-тет им. Н.Е. Жуковского, 2012. — 614 с. [Drozd, A.V., Kharchenko, V.S., Antoshchuk, S.G., et al. Objects and Methods of On-Line Testing for Safe Instrumentation and Control Systems / ed. A.V. Drozd and V.S. Kharchenko. — Kharkov: Nat. aerospace. University named after N.E. Zhukovsky, 2012. — 614 p. (in Russian)].
13. Скляр В.В. Обеспечение безопасности АСУТП в соответствии с современными стандартами. — М.: Инфра-Инженерия, 2018. — 384 с. [Sklyar, V.V. Ensuring the Safety of Process Control Systems in Accordance with Modern Standards. — Moscow: Infra-Engineering, 2018. — 384 p. (in Russian)].
14. Sogomonyan, E.S. Self-Correction Fault-Tolerant Systems. — Preprint, October 2018. — 30 p.

15. *Согомонян Е.С., Слабаков Е.В.* Самопроверяемые устройства и отказоустойчивые системы. — М.: Радио и связь, 1989. — 208 с. [*Sogomonyan, E.S., Slabakov, E.V.* Self-Checking Devices and Fault-Tolerant Systems. — Moscow: Radio and Communications, 1989. — 208 p. (in Russian)]
 16. *Piestrak, S.J.* Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. — Wrocław: Ofiyna Wydawnicza Politechniki Wrocławskiej, 1995. — 111 p.
 17. *Сапожников В.В., Сапожников Вл.В., Ефанов Д.В.* Коды Хэмминга в системах функционального контроля логических устройств: монография. — СПб.: Наука, 2018. — 151 с. [*Sapozhnikov, V.V., Sapozhnikov, Vl.V., Efanov, D.V.* Hamming Codes in Concurrent Error Detection Systems of Logic Devices. — St. Petersburg: Nauka, 2018. — 151 p. (in Russian)]
 18. *Петерсон У.* Коды, исправляющие ошибки. — М.: Мир, 1964. — 338 с. [*Peterson, W.* Codes for Correcting Errors. — Moscow: Mir, 1964. — 338 p. (in Russian)]
 19. *Fujiwara, E.* Code Design for Dependable Systems: Theory and Practical Applications. — John Wiley & Sons, 2006. — 720 p.
 20. *Lala, P.K.* Principles of Modern Digital Design. — New-Jersey: John Wiley & Sons, 2007. — 436 p.
 21. *Brouwer, A.E., Hamalainen, H.O., Ostergard, P.R.J., Sloane, N.J.A.* Bounds on Mixed Binary/Ternary Codes // IEEE Transactions on Information Theory. — 1988. — Vol. 44, iss. 1. — P. 140—161. — DOI: 10.1109/18.651001.
 22. *Gulliver, T.A., Ostergard, P.R.J.* Improved Bounds for Ternary Linear Codes of Dimension 7 // IEEE Transactions on Information Theory. — 1997. — Vol 43, iss. 4. — P. 1377—1381. — DOI: 10.1109/18.605613.
 23. *Mirzaee, R.F., Daliri, M.S., Navi, K., Bagherzadeh, N.* A Single Parity-Check Digit for One Trist Error Detection in Ternary Communication Systems: Gate-Level and Transistor-Level Designs // Journal of Multiple-Valued Logic and Soft Computing — 2017. — No. 29 (3—4). — P. 303—326.
 24. *Bitouze, N., Graell i Amat, A., Rosnes, E.* Error Correcting Coding for a Nonsymmetric Ternary Channel // IEEE Transactions on Information Theory — 2010, Vol. 56, iss. 11. — P. 5715—5729. — DOI: 10.1109/TIT.2010.2069211.
 25. *Laaksonen, A., Ostergard, P.R.J.* New Lower Bounds on Error-Correcting Ternary, Quaternary and Quinary Codes // Lecture Notes in Computer Science 10495, Springer: Coding Theory and Applications, 5th International Castle Meeting, ICMCTA 2017, Vihula, Estonia, August 28—31. — 2017. — P. 228—237.
 26. *Svanström, M., Östergård, P.R.J., Bogdanova, G.T.* Bounds and Constructions for Ternary Constant-Composition Codes // IEEE Transactions on Information Theory. — 2002. — Vol. 48. — P. 101—111.
 27. *Berger, J.M.* A Note on Error Detecting Codes for Asymmetric Channels // Information and Control. — 1961. — Vol. 4, iss. 1. — P. 68—73. — DOI: 10.1016/S0019-9958(61)80037-5.
 28. *Efanov, D.V.* Ternary Parity Codes: Features // Proceedings of 17th IEEE East-West Design & Test Symposium (EWDTS' 2019), Batumi, Georgia, September 13—16. — 2019. — P. 315—319. — DOI: 10.1109/EWDTS.2019.8884414.
 29. *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. [*Efanov, D.V., Sapozhnikov, V.V., Sapozhnikov, Vl.V.* On Summation Code Properties in Functional Control Circuits // Automation and Remote Control. — 2010. — Vol. 71, iss. 6. — P. 1117—1123. — DOI: 10.1134/S0005117910060123]
 30. *Das, D., Toubia, N.A.* Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes // Journal of Electronic Testing: Theory and Applications. — 1999. — Vol. 15, iss. 1—2. — P. 145—155. — DOI: 10.1023/A:1008344603814.
 31. *Sapozhnikov, V., Sapozhnikov, Vl., Efanov, D.* Modular Sum Code in Building Testable Discrete Systems // Proceedings of 13th IEEE East-West Design & Test Symposium (EWDTS' 2015), Batumi, Georgia, September 26—29. — 2015. — P. 181—187. — DOI: 10.1109/EWDTS.2015.7493133.
- Статья представлена к публикации членом редколлегии В.Г. Лебедевым.*
- Поступила в редакцию 3.02.2020, после доработки 03.05.2020.
Принята к публикации 3.06.2020.*
- Ефанов Дмитрий Викторович** — д-р техн. наук, Российский университет транспорта, г. Москва, ☒ TrES-4b@yandex.ru.

TERNARY SUM CODES FOR THE DIGITAL CIRCUIT TESTING

D.V. Efanov

Russian University of Transport, Moscow, Russia

☒ TrES-4b@yandex.ru

Abstract. The specifics of protecting data presented in the ternary form are discussed. Future relevance of the ternary logic is emphasized, and the main features of reliable digital devices and systems development are identified. A ternary sum code is described that detects any monotonous and asymmetrical errors in data vectors, the use of which is promising for constructing devices and systems with fault detection. The principles of building a ternary sum code are similar to those of classical binary sum codes (Berger codes). Previously unknown ternary sum codes properties are established only for the event of errors in data vectors with error-free check bits. Such task is relevant for practical applications in which the check bits and data vectors are calculated by physically different devices. Taking into account the established ternary sum codes properties can be useful when choosing a method of protecting both the data itself and parrying faults arising in automation devices.

Keywords: ternary logic, ternary logic device, fault detection in automation devices, calculation control, data protection, ternary sum code, Berger's code, undetectable error, error detection characteristics.