

ОШИБКИ В ТРОИЧНЫХ КОДОВЫХ ВЕКТОРАХ, ИХ КЛАССИФИКАЦИЯ И ОБНАРУЖЕНИЕ С ПОМОЩЬЮ ПОМЕХОЗАЩИЩЕННОГО КОДИРОВАНИЯ

Д. В. ЕФАНОВ

*Российский университет транспорта, 127994, Москва, Россия
E-mail: TrES-4b@yandex.ru*

Рассматриваются особенности ошибок, возникающих в троичных кодовых векторах вследствие искажений одного или нескольких разрядов и характеризующихся кратностью и видом. Предложена классификация ошибок в троичных кодовых векторах, а также проведено сравнение с ошибками, возникающими в двоичных кодовых векторах. Даны определения различных видов ошибок в двоичных и троичных кодовых векторах и приведены примеры, иллюстрирующие особенности таких ошибок. Классификация ошибок в троичных кодовых векторах позволяет разрабатывать принципы кодирования данных для обнаружения ошибок определенного вида и использовать их как при передаче данных, так и при синтезе функционирующих в троичной логике устройств, обладающих свойством обнаружения неисправностей. Приведены некоторые способы построения троичных кодов, ориентированных на обнаружение ошибок в информационных векторах или в кодовых словах в целом. Результаты могут быть использованы при разработке основ синтеза надежных и безопасных устройств автоматики, функционирующих в троичной логике.

Ключевые слова: *ошибки в кодовых векторах, двоичные кодовые векторы, троичные кодовые векторы, классификация ошибок в кодовых векторах, коды с обнаружением ошибок, коды с обнаружением монотонных ошибок*

Введение. Несмотря на повсеместное использование двоичной логики в качестве основы цифровой техники, троичная логика и троичная техника представляет особый интерес для инженеров и ученых всего мира [1—7]. Известны попытки реализации устройств, функционирующих в троичной логике, как на основе традиционных устройств двоичной логики, так и с применением троичных логических элементов [8—10]. Обсуждаются вопросы использования троичной логики и при разработке квантовых компьютеров, а применение кутритов вместо кубитов позволяет существенно уменьшить количество квантовых вентилях [11]. Важной проблемой является исследование способов построения устройств и систем автоматики, наделенных свойством обнаружения неисправностей, что связано с использованием помехозащищенных и помехоустойчивых троичных кодов.

При разработке самопроверяемых двоичных устройств автоматики и вычислительной техники широко используются методы помехоустойчивого кодирования [12—15]. При построении дискретных устройств и систем кодирование применяется как на уровне элементарных компонентов, так и на более сложном уровне. Например, при синтезе устройств на уровне

моделей конечных автоматов их состояния зачастую кодируют определенным избыточным кодом, что позволяет обнаруживать и парировать неисправности в процессе функционирования устройств, а также обеспечивать их штатную работу [16, 17]. Для защиты запоминающих устройств применяются коды с коррекцией ошибок [18, 19]. Кроме того, избыточное кодирование характерно для синтеза средств диагностического обеспечения [20], а также при передаче данных между узлами различных систем автоматизации [21]. Аналогичные принципы могут быть реализованы при разработке самопроверяемых цифровых устройств, функционирующих в троичной логике.

Фундаментальное значение в процессе разработки самопроверяемых устройств автоматики на основе помехозащищенных или помехоустойчивых кодов имеет выбор кода с определенными характеристиками обнаружения ошибок разного вида и различной кратностью [22]. С этой точки зрения, ошибки в двоичных кодовых векторах принято классифицировать на монотонные, симметричные и асимметричные, а также выделять особые классы избыточных кодов, обладающие возможностями обнаружения любых ошибок определенного вида и любых ошибок определенного вида кратностью до конкретного значения [23, 24].

В настоящей статье рассматриваются особенности классификации ошибок в троичных кодовых векторах по сравнению с особенностями ошибок в двоичных кодовых векторах. Установлены типичные виды ошибок, а также введены классы троичных избыточных кодов, использование которых позволяет синтезировать самопроверяемые цифровые устройства, функционирующие в троичной логике.

Ошибки в двоичных кодовых векторах. Рассмотрим особенности ошибок, возникающих в двоичных кодовых векторах.

Проведем аналогию между кодовым вектором и устройством, имеющим определенное количество выходов: каждый выход соответствует определенному разряду кодового вектора, а значение этого разряда формируется путем вычисления функции выхода. В результате штатной работы устройства формируется корректный кодовый вектор. Возникновение неисправности в структуре устройства при определенных условиях приводит к появлению ошибки на его выходах (при соблюдении условий наблюдаемости ошибки). К примеру, если на пути к выходу устройства от места возникновения неисправности имеются инверсии, то ошибка при трансляции к выходу будет изменять свой вид. Таким образом, ошибка на выходе какого-либо элемента структуры цифрового устройства может привести к ошибке в кодовом векторе, причем эти ошибки могут быть различными по составу искажаемых значений. Например, устройство, функционирующее в двоичной логике и не имеющее в своем составе инверторов, будет обладать свойством трансляции вида ошибки на его выходы по каждому пути. Такая особенность устройства связана с тем, что реализуемые им функции являются монотонными, а любое искажение на линии схемы будет приводить только к монотонным проявлениям ошибок. Другими словами, свойство монотонности функций напрямую связано и со свойством монотонности возникающей в кодовом векторе ошибки.

Класс монотонных функций является одним из основных в алгебре логики [2], а сами свойства основных классов функций алгебры логики широко используются при разработке методов технического диагностирования. Например, при контроле вычислений широко применяется линейная функция паритета [25], при функциональном контроле логических схем используются коды с обнаружением любых монотонных ошибок [26, 27], а также может быть установлена принадлежность функции к классу самодвойственных [28]. Ключевое значение при организации диагностического обеспечения имеет вид возможных ошибок на выходах объектов диагностирования. При этом зачастую проводится аналогия (как показано выше) между выходами устройств автоматики и кодовыми векторами, а ошибка на выходах устройств сопоставляется с ошибкой в кодовом векторе [23].

Определение 1. Под *ошибкой* в кодовом векторе понимается совокупность искажений его разрядов.

Пусть число разрядов кодового вектора, формируемого на выходах некоторого объекта диагностирования, равно m . Тогда ошибка в кодовом векторе может быть связана с искажением от 1 до m разрядов.

Определение 2. Число искажаемых при возникновении ошибки разрядов называется *кратностью d ошибки*.

Общее число ошибок в двоичных кодовых векторах (Binary Code Vector) определяется удвоенным числом переходов каждого из 2^m кодовых векторов в каждый:

$$N_m^{\text{Bin}} = 2C_{2^m}^2 = 2 \frac{2^m}{2! \cdot (2^m - 2)!} = \frac{2 \cdot 2^m (2^m - 1)(2^m - 2)!}{2! \cdot (2^m - 2)!} = 2^m (2^m - 1). \quad (1)$$

Например, общее количество искажений трехразрядных двоичных кодовых векторов составляет $N_3^{\text{Bin}} = 2C_{2^3}^2 = 2^3 (2^3 - 1) = 56$.

С другой стороны, общее число необнаруживаемых ошибок в двоичных кодовых векторах рассчитывается как сумма всех ошибок каждой из кратностей $d \in \{1, 2, \dots, m\}$:

$$N_m^{\text{Bin}} = \sum_{d=1}^m 2^m \cdot 1^d \cdot C_m^d = \sum_{d=1}^m 2^m C_m^d = 2^m \sum_{d=1}^m C_m^d, \quad (2)$$

где множитель 1^d характеризует число вариантов искажений d разрядов, множитель C_m^d — число искажений d разрядов из m .

Например, для вектора с $m=3$ формула (2) дает следующий результат:

$$N_3^{\text{Bin}} = 2^3 \sum_{d=1}^{m=3} C_3^d = 2^3 \cdot (C_3^1 + C_3^2 + C_3^3) = 8 \cdot (3 + 3 + 1) = 56.$$

В двоичной логике каждый разряд кодового вектора может содержать два вида искажений: $0 \rightarrow 1$ или $1 \rightarrow 0$. По совокупности искажений различных разрядов устанавливается вид возникающей ошибки.

Ошибки (ERR) в кодовых векторах могут быть однократными (одинокными) (SIN) и многократными (MULT).

Определение 3. Если при возникновении ошибки в двоичном кодовом векторе присутствуют искажения только вида $0 \rightarrow 1$ или только вида $1 \rightarrow 0$, то ошибка называется *монотонной (однонаправленной)*.

Определение 4. Если при возникновении ошибки в двоичном кодовом векторе имеются искажения различных видов, то ошибка называется *немонотонной (разнонаправленной)*.

Одинокные (SIN), монотонные (MON) и немонотонные (NMON) ошибки образуют полное множество ошибок в двоичных кодовых векторах. Немонотонные ошибки принято делить на два неравнозначных класса — симметричные (SYM) и асимметричные (ASYM).

Определение 5. Ошибка называется *симметричной*, если при ее возникновении в двоичном кодовом векторе присутствует одинаковое количество искажений вида $0 \rightarrow 1$ и вида $1 \rightarrow 0$.

Определение 6. Ошибка называется *асимметричной*, если при ее возникновении в двоичном кодовом векторе присутствует различное количество искажений вида $0 \rightarrow 1$ и вида $1 \rightarrow 0$.

На рис. 1 приведены примеры ошибок разного вида в двоичных кодовых векторах, где a — одиночная (SIN), b — монотонная (MON), v — симметричная (SYM), z — асимметричная (ASYM); на рис. 2 дана их полная классификация. Следует отметить, что монотонная

ошибка может иметь кратность $d \geq 2$, симметричная ошибка всегда имеет четное значение кратности, а асимметричная — кратность $d \geq 3$.

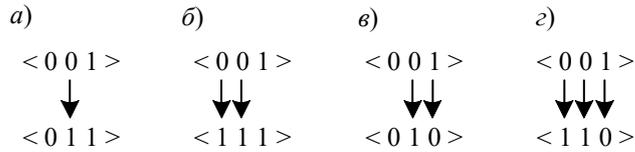


Рис. 1

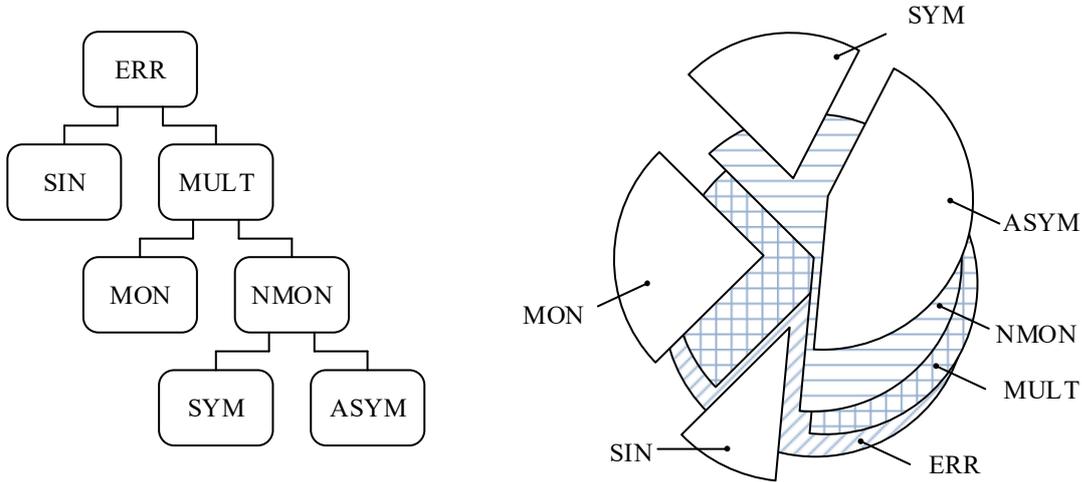


Рис. 2

В табл. 1 для примера приведены характеристики ошибок в двоичных кодовых векторах с $m=3$, где указано количество ошибок разного вида и различной кратностью. Кроме того, в последней графе таблицы приведен относительный показатель τ , характеризующий долю числа ошибок конкретного вида от общего количества ошибок. Число монотонных и симметричных ошибок примерно одинаково, тогда как асимметричных примерно вдвое меньше.

Таблица 1

Вид ошибки	Кратность ошибки			Общее число ошибок данного вида	$\tau, \%$
	1	2	3		
SIN	24	—	—	24	42,857
MON	—	12	2	14	25
SYM	—	12	0	12	21,429
ASYM	—	—	6	6	10,714
Общее число ошибок данной кратностью	24	24	8	56	100

Для различных длин кодовых векторов распределения ошибок по видам и кратностям различны, различны и значения показателя τ .

Ошибки в троичных кодовых векторах. Рассмотрим кодовые векторы троичной логики.

В математике троичной логики используется большее число значений, чем в двоичной (будем использовать их следующие обозначения: 0, 1 и 2). Соответственно число специальных классов функций троичной логики больше, чем двоичной (известно 18 основных классов функций троичной логики и 5 основных классов двоичной [2]). При этом важным при классификации ошибок является определение классов монотонных функций, которых в троичной логике три в зависимости от порядка следования аргументов.

1. Класс M_1 — монотонные функции, для которых при сравнении аргументов принят порядок $0 < 1 < 2$.

2. Класс M_2 — монотонные функции, для которых при сравнении аргументов принят порядок $1 < 2 < 0$.

3. Класс M_3 — монотонные функции, для которых при сравнении аргументов принят порядок $2 < 0 < 1$.

Базируясь на классификации ошибок в двоичных кодовых векторах, классифицируем ошибки в троичных кодовых векторах с учетом особенностей сочетаний искажений видов $0 \rightarrow 1, 1 \rightarrow 0, 0 \rightarrow 2, 2 \rightarrow 0, 1 \rightarrow 2, 2 \rightarrow 1$.

Общее число ошибок в троичных кодовых векторах (Ternary Code Vector) определяется удвоенным числом переходов каждого из 3^m кодовых векторов в каждый:

$$N_m^{\text{Ter}} = 2C_{3^m}^2 = 2 \frac{3^m}{2!(3^m - 2)!} = \frac{2 \cdot 3^m (3^m - 1)(3^m - 2)!}{2!(3^m - 2)!} = 3^m (3^m - 1). \quad (3)$$

Например, общее количество искажений трехразрядных троичных кодовых векторов составляет $N_3^{\text{Ter}} = 2C_{3^3}^2 = 3^3 (3^3 - 1) = 702$. По сравнению с количеством ошибок в двоичных векторах это число в 12,54 раза больше. С увеличением длины кодового вектора соотношение количества ошибок в троичных и двоичных векторах стремительно возрастает (табл. 2). В пределе при $m \rightarrow \infty$ это соотношение определяется как

$$\lim_{m \rightarrow \infty} \delta_m = \lim_{m \rightarrow \infty} \frac{N_m^{\text{Ter}}}{N_m^{\text{Bin}}} = \lim_{m \rightarrow \infty} \frac{3^m (3^m - 1)}{2^m (2^m - 1)} = \lim_{m \rightarrow \infty} \left(\frac{3}{2}\right)^m \lim_{m \rightarrow \infty} \frac{3^m - 1}{2^m - 1} \Rightarrow \infty. \quad (4)$$

Таблица 2

m	N_m^{Bin}	N_m^{Ter}	δ_m
2	12	72	6
3	56	702	12,54
4	240	6480	27
5	992	58806	59,28
6	4032	530712	131,63
7	16256	4780782	294,09
8	65280	43040160	659,32
9	261632	387400806	1480,71
10	1047552	3486725352	3328,45
...
20	$1,09951 \cdot 10^{12}$	$1,21577 \cdot 10^{19}$	11057343
...
50	$1,26765 \cdot 10^{30}$	$5,15378 \cdot 10^{47}$	$4,066 \cdot 10^{17}$
...
100	$1,60694 \cdot 10^{60}$	$2,65614 \cdot 10^{95}$	$1,653 \cdot 10^{35}$

Количество необнаруживаемых ошибок в троичных кодовых векторах может быть определено и как сумма всех ошибок каждой из кратностей $d \in \{1, 2, \dots, m\}$:

$$N_m^{\text{Ter}} = \sum_{d=1}^m 3^m 2^d C_m^d = 3^m \sum_{d=1}^m 2^d C_m^d, \quad (5)$$

где множитель 2^d характеризует число вариантов искажений d разрядов (в отличие от двоичных разрядов каждый троичный разряд может искажаться двумя вариантами).

Для вектора с $m=3$ формула (5) дает следующий результат:

$$N_3^{\text{Ter}} = 3^3 \sum_{d=1}^{m=3} 2^d C_3^d = 3^3 (2^1 C_3^1 + 2^2 C_3^2 + 2^3 C_3^3) = 27 \cdot (6 + 12 + 6) = 702.$$

Отметим также, что формула (5) может быть обобщена для кодовых векторов q -значной логики:

$$N_m^q = q^m \sum_{d=1}^m (q-1)^d C_m^d.$$

Ошибки в троичных кодовых векторах, кроме значительного их числа, гораздо разнообразнее ошибок в двоичных кодовых векторах.

Как и ошибки в двоичных векторах, ошибки в троичных векторах разделяются на однократные (одионочные) и многократные. Многократные ошибки делятся на два класса — монотонные и немонотонные.

Определение 7. Ошибка в троичном кодовом векторе является *монотонной* в случае, если при ее возникновении сохраняется приоритет значений, определенный в каждом из классов монотонных функций M_1 , M_2 и M_3 .

Монотонные ошибки делятся на однонаправленные (UNI) и двунаправленные (BIDI).

Определение 8. Монотонная ошибка в троичном кодовом векторе является *однонаправленной*, если при ее возникновении все искажения осуществляются только в нуль, либо только в единицу, либо только в двойку.

Следует отметить, что монотонные однонаправленные ошибки могут быть разделены на два класса — двухпозиционные (DS) и трехпозиционные (TS).

Определение 9. Монотонная однонаправленная ошибка в троичном кодовом векторе называется *двухпозиционной*, если при ее возникновении искажаются одинаковые значения разрядов.

Определение 10. Монотонная однонаправленная ошибка в троичном кодовом векторе называется *трехпозиционной*, если при ее возникновении искажаются различные значения разрядов.

Определение 11. Монотонная ошибка в троичном кодовом векторе является *двунаправленной*, если при ее возникновении сохраняется приоритет значений, определенный в каждом из классов монотонных функций M_1 , M_2 и M_3 , и возникают искажения, которые приводят к установлению нуля, единицы и двойки.

Определение 12. Ошибка в троичном кодовом векторе является *немонотонной* в случае, если при ее возникновении возможны любые виды искажений и нарушается приоритет значений, определенный в каждом из классов монотонных функций M_1 , M_2 и M_3 .

Немонотонные ошибки делятся на композиционные (COMP) и асимметричные (ASYM).

Определение 13. Немонотонная ошибка в троичном кодовом векторе является *композиционной* в случае, если при ее возникновении сохраняется число нулевых разрядов, единичных разрядов и разрядов, равных двум.

Немонотонная композиционная ошибка в троичном кодовом векторе является аналогом немонотонной симметричной ошибки в двоичном кодовом векторе.

Определение 14. Немонотонная ошибка в троичном кодовом векторе является *асимметричной* в случае, если при ее возникновении не сохраняется число нулевых разрядов, единичных разрядов и разрядов, равных двум.

Примеры ошибок разного вида в троичных кодовых векторах приведены на рис. 3, где a — одиночная (SIN), b — однонаправленная двухпозиционная (UNI DS), v — однонаправленная трехпозиционная (UNI TS), z — двунаправленная (BIDI), d — композиционная (COMP), e — асимметричная (ASYM); на рис. 4 приведена их классификация.

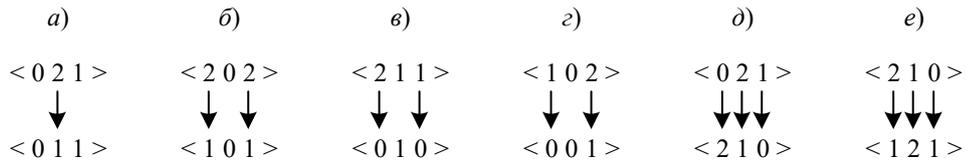


Рис. 3

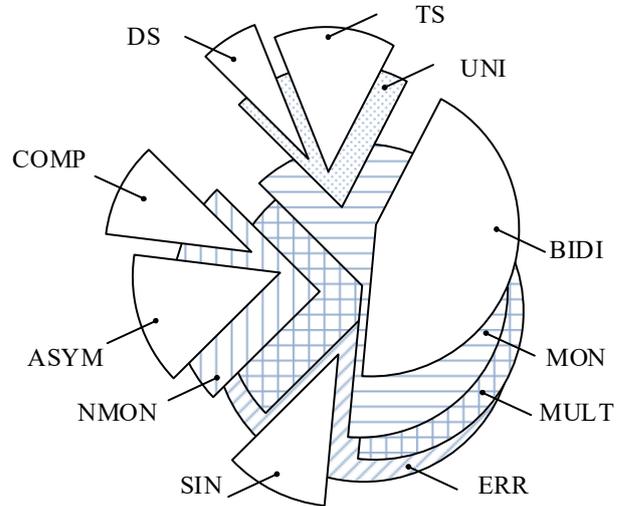
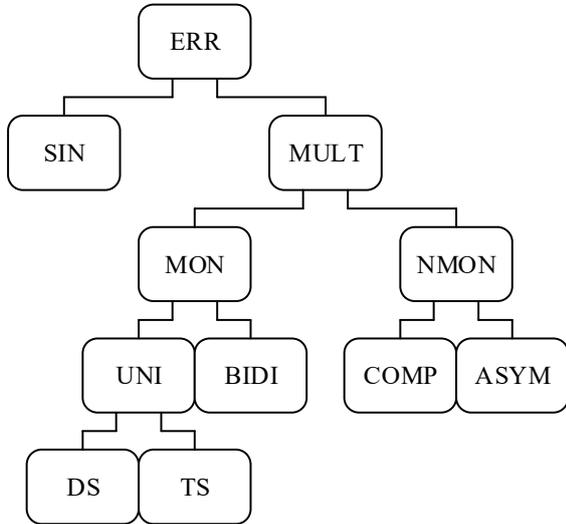


Рис. 4

В табл. 3 дана характеристика ошибок в троичных кодовых векторах с $m=3$, где также указано количество ошибок разного вида и различной кратностью, а также рассчитан показатель τ . Существенную долю ошибок (54,701 %) составляют различные монотонные ошибки, тогда как композиционные ошибки составляют лишь 9,402 %, а асимметричные — 12,82 %.

Таблица 3

Вид ошибки	Кратность ошибки			Общее число ошибок данного вида	τ , %
	1	2	3		
SIN	162	—	—	162	23,077
UNI DS	—	51	24	75	10,684
UNI TS	—	54	0	54	7,692
BIDI	—	165	90	255	36,325
COMP	—	36	30	66	9,402
ASYM	—	—	90	90	12,82
Общее число ошибок данной кратностью	162	306	234	702	100

Распределение ошибок по видам в троичных кодовых векторах существенно отличается от распределения ошибок по видам в двоичных кодовых векторах. С увеличением значения m распределения ошибок по видам и кратностям изменяются.

Коды с обнаружением ошибок определенного вида. Для решения задач синтеза устройств автоматики, обладающих свойством обнаружения неисправностей, могут использоваться избыточные коды, ориентированные на обнаружение ошибок конкретного вида в кодовых векторах.

На рис. 5 представлена классификация двоичных избыточных кодов, ориентированных на обнаружение ошибок и используемых при построении цифровых устройств. Среди множества двоичных избыточных кодов выделяются специальные классы — коды с обнаружением любых монотонных и асимметричных ошибок (UAED-коды) и коды с обнаружением любых монотонных ошибок (UED-коды). Кроме того, зачастую выделяют коды, которые обнаруживают любые монотонные ошибки кратностью до определенного значения d_0 (d_0 -UED-коды),

и коды, которые обнаруживают любые монотонные и асимметричные ошибки кратностью до определенных значений d_v и d_a (d_v, d_a -UAED-коды). Также существуют более узкие классы кодов с обнаружением любых симметричных ошибок (SED-коды) и обнаружением любых симметричных ошибок кратностью до определенного значения d_σ (d_σ -SED-коды), а также с обнаружением любых асимметричных ошибок (AED-коды) и обнаружением любых асимметричных ошибок кратностью до определенного значения d_a (d_a -AED-коды) [29].

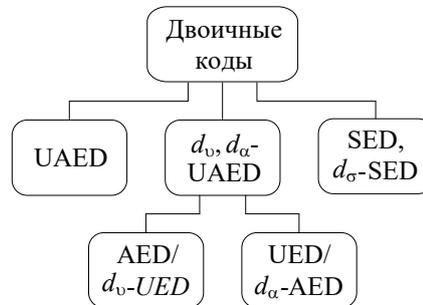


Рис. 5

Представленная классификация обусловлена особенностями реализации устройств автоматике с контролепригодными структурами. Так, например, UAED-коды эффективно применяются при контроле устройств, выходы которых образуют монотонно и асимметрично независимые группы (МАН-группы), а UED-коды — при контроле устройств, выходы которых образуют монотонно независимые группы (МН-группы). Известны методы преобразования структур устройств автоматике в устройства со структурами, выходы которых образуют МН- и МАН-группы [30]. Коды же типа d_v -UED и d_v, d_a -UAED применяются при контроле устройств автоматике по нескольким МН- и МАН-группам.

Аналогичным образом могут быть классифицированы и троичные избыточные коды. При этом возможно построение аналогов двоичных избыточных кодов с определенными свойствами (рис. 6).

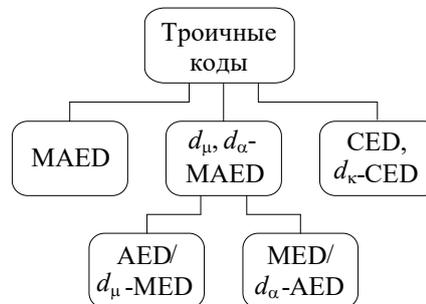


Рис. 6

Так, во множестве троичных избыточных кодов можно выделить коды, обнаруживающие любые монотонные и асимметричные ошибки. Обозначим их как MAED-коды (отличие в обозначении обусловлено более широким понятием „монотонности“ в троичной логике и выделением как однонаправленных, так и двунаправленных монотонных ошибок). Помимо MAED-кодов, могут быть выделены коды с обнаружением любых монотонных ошибок (MED-коды), эффективно применяемые при контроле устройств троичной логики, выходы которых образуют МН- и МАН-группы. Для контроля устройств по нескольким группам монотонно и асимметрично независимых выходов могут применяться коды с обнаружением монотонных и асимметричных ошибок кратностью до определенных значений d_μ и d_α (d_μ, d_α -MAED-коды / d_μ -MED-коды). Можно также выделить троичные коды, обнаруживающие любые композиционные ошибки (CED-коды), и коды, обнаруживающие любые композиционные ошибки кратностью до определенного значения d_k (d_k -CED-коды), а также коды, обнаруживающие любые асимметричные ошибки (AED-коды) и обнаруживающие любые

асимметричные ошибки кратностью до определенного значения d_a (d_a -AED-коды). Кроме того, следует отметить, что возможно построение кодов, ориентированных на обнаружение только одного из подклассов монотонных ошибок.

Троичные коды, обладающие свойствами обнаружения ошибок определенного вида, могут быть построены по аналогии с двоичными избыточными кодами.

MAED-кодом является троичный код, составленный из кодовых слов, имеющих одинаковое число разрядов, равных 1 и 2. Такой код называется *кодом с постоянной композицией значений* (constant-composition code) [31, 32]. Его аналогом в двоичной логике является равновесный код [33]. Приведем пример композиционного кода ($C(r_1, r_2)$ -кода, где r_1 и r_2 — число разрядов, равных 1, и число разрядов, равных 2) с числом разрядов $m=4$. Например, $C(1,2)$ -код образуется кодовыми векторами из множества {0122, 1022, 0212, 1202, 0221, 1220, 2012, 2102, 2021, 2120, 2201, 2210}.

Еще один пример MAED-кода — троичный код с суммированием, который строится по следующим правилам. В информационном векторе определяются числа r_1 и r_2 , которые представляются в троичном виде и записываются соответственно в k_1 старших и k_2 младших разрядов контрольных векторов длиной $k = k_1 + k_2$. Число r_1 и число r_2 в информационном векторе длиной m может быть равным от 0 до m . Отсюда следует, что $k_1 = k_2 = \lceil \log_3(m+1) \rceil$ и $k = 2 \lceil \log_3(m+1) \rceil$. В табл. 4 приведен расчет общего числа контрольных разрядов в троичных кодах с суммированием при различной длине информационного вектора.

Таблица 4

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

Рассмотрим пример определения значений разрядов контрольного вектора для информационного вектора <011120222> троичного кода с суммированием: $r_1=3$, $r_2=4$, $[r_1]_3=010$, $[r_2]_3=011$. Контрольный вектор будет иметь следующие разряды: <010011>.

Троичный код с суммированием, построенный по приведенным выше правилам, для каждого информационного вектора с одинаковой композицией значений 1 и 2 будет иметь один и тот же контрольный вектор. Таким образом, троичный код с суммированием по своим особенностям является аналогом классического двоичного кода Бергера [34].

Отметим, что все MAED-коды в классе необнаруживаемых будут иметь только композиционные ошибки, а остальные виды ошибок будут обнаруживаться. Если в двоичной логике это свойство говорит о значительной доле необнаруживаемых ошибок от их общего числа [23], то в троичной логике их доля будет меньшей (так как монотонных ошибок в троичных векторах существенно больше, чем в двоичных).

d_μ -MED-коды — это коды с суммированием, для которых числа r_1 и r_2 рассчитываются в кольце вычетов по модулю $M \in \{3^1, 3^2, \dots, 3^{\lceil \log_3(m+1) \rceil - 1}\}$ (аналоги двоичных модульных кодов с суммированием [35]). Одним из таких кодов будет код, для которого числа r_1 и r_2 рассчитываются в кольце вычетов по модулю $M=9$. Число контрольных разрядов в таких троичных модульных кодах всегда равно $k=4$. Определим контрольный вектор для информационного вектора $\langle 011120222 \rangle$ троичного кода с суммированием в кольце вычетов по модулю $M=9$: $r_1(\bmod 9)=3$, $r_2(\bmod 9)=4$, $[r_1]_3=10$, $[r_2]_3=11$. Контрольный вектор будет иметь следующие разряды: $\langle 1011 \rangle$. Рассматриваемым кодом будут обнаруживаться любые монотонные ошибки кратностью $d_\mu < 9$ в информационных векторах. По этой причине их применение весьма перспективно.

Еще один пример d_μ -MED-кодов — коды, получаемые путем объединения кодовых слов кодов с постоянной композицией значений:

$$C(r_1, r_2, d_\mu) = \bigcup_{r_1, r_2 = \lfloor m/2 \rfloor \pmod{d_\mu}} C(r_1, r_2), \quad r_1 + r_2 \leq m.$$

Положим, к примеру, $m=8$ и $d_\mu=4$, тогда $r_1, r_2 = \lfloor 8/2 \rfloor \pmod{4} = 4 \pmod{4} = 0$. Таким образом, для построения d_μ -MED-кода с указанными параметрами необходимо взять все кодовые векторы, принадлежащие $C(r_1, r_2)$ -кодам, для которых $r_1, r_2 \in \{0, 1, 2\}$, при этом $r_1 + r_2 \leq 8$. Указанным значениям соответствуют коды с постоянной композицией значений $C(0,0)$, $C(0,4)$, $C(0,8)$, $C(4,0)$, $C(8,0)$, $C(4,4)$.

Построенный по представленным выше правилам троичный код является аналогом двоичного кода Бордена [36].

Заключение. Представленные классификации ошибок в двоичных и троичных кодовых векторах позволяют установить и основные классы двоичных и троичных избыточных кодов, ориентированных на обнаружение ошибок определенного вида и кратности. Это, в свою очередь, позволяет обоснованно выбирать тот или иной способ кодирования при разработке самопроверяемых цифровых устройств и выборе их диагностического обеспечения.

В отличие от ошибок в двоичных кодовых векторах, ошибки в троичных кодовых векторах существенно разнообразнее, а преимущественным видом ошибок являются монотонные, которые также делятся на несколько классов в соответствии с тем, в какой вид сигналов осуществляются искажения (однонаправленные и двунаправленные).

Приведены примеры троичных кодов, относящихся к введенным классам MAED- и d_μ -MED-кодов. Данные коды являются аналогами двоичных кодов с суммированием (кодов Бергера и модульных кодов с суммированием). Их использование перспективно при синтезе контролепригодных цифровых устройств троичной логики, а также при разработке технических средств их диагностического обеспечения.

СПИСОК ЛИТЕРАТУРЫ

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.
2. Поспелов Д. А. Логические методы анализа и синтеза схем. М.: Энергия, 1974. 368 с.
3. Connely J. Ternary Computing Testbed 3-Trit Computer Architecture / California Polytechnic State University of San Luis Obispo, 29 Aug., 2008. 184 p.
4. Петров А. Троичный компьютер: да, нет, может быть // Популярная механика. 2011. № 9. С. 72—76.
5. Ahmad S., Alam M. Balanced-ternary logic for improved and advanced computing // Intern. Journal of Computer Science and Information Technologies (IJCSIT). 2014. Vol. 5, iss. 4. P. 5157—5160.

6. Cambou B., Flikkema P. G., Palmer J., Telesca D., Philabaum C. Can ternary computing improve information assurance? // *Cryptography*. 2018. Vol. 2, iss. 1. P. 1—16. DOI: 10.3390/cryptography2010006.
7. Efanov D. V. Ternary parity codes: features // *Proc. of the 17th IEEE East-West Design & Test Symposium (EWDTS'2019)*, Batumi, Georgia, 13—16 Sept., 2019. P. 315—319. DOI: 10.1109/EWDTS.2019.8884414.
8. Wu J. Ternary logic circuit for error detection and error correction // *Proc. of 19th Intern. Symp. on Multiple-Valued Logic*, Guangzhou, China, 29—31 May 1989. P. 94—99. DOI: 10.1109/ISMVL.1989.37766.
9. Kim S., Lim T., Kang S. An Optimal Gate Design For The Synthesis Of Ternary Logic Circuits // *Proc. of the 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jeju, South Korea, 22—25 Jan. 2018. P. 476—481. DOI: 10.1109/ASPDAC.2018.8297369.
10. Vudadha C., Rajagopalan S., Dusi A., Phaneendra P. S., Srinivas M. B. Encoder-based optimization of CNFET-based ternary logic circuits // *IEEE Transact. on Nanotechnology*. 2018. Vol. 17, iss. 2. P. 299—310. DOI: 10.1109/TNANO.2018.2800015.
11. Lanyon B. P., Barbieri M., Almeida M. P., Jennewein T., Ralph T. C., Resch K. J., Pryde G. J., O'Brien J. L., Gilchrist A., White A. G. Simplifying quantum logic using higher-dimensional hilbert spaces // *Nature Physics*. 2009. Vol. 5, iss. 2. P. 134—140. DOI: 10.1038/nphys1150.
12. Согомонян Е. С., Слабаков Е. В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
13. Сапожников В. В., Сапожников Вл. В. Самопроверяемые дискретные устройства. СПб: Энергоатомиздат, 1992. 224 с.
14. Piestrak S. J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław: Офіуна Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
15. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Коды Хэмминга в системах функционального контроля логических устройств. СПб: Наука, 2018. 151 с.
16. Сагалович Ю. Л. Последовательности максимальной длины как коды состояний автомата // *Проблемы передачи информации*. 1976. Т. 12, № 4. С. 296—299.
17. Matrosova A. Yu., Levin I., Ostanin S. A. Self-checking synchronous FSM network design with low overhead // *VLSI Design*. 2000. Vol. 11, iss. 1. P. 47—58. DOI: 10.1155/2000/46578.
18. Сагалович Ю. Л., Щербаков Н. С. Выбор системы кодирования для защиты запоминающих устройств от ошибок // *Проблемы передачи информации*. 1984. Т. 20, № 1. С. 14—20.
19. Tshagharyan G., Harutyunyan G., Shoukourian S., Zorian Y. Overview study on fault modeling and test methodology development for FinFET-based memories // *Proc. of the 13th IEEE East-West Design & Test Symposium (EWDTS'2015)*, Batumi, Georgia, 26—29 Sept., 2015. P. 19—22. DOI: 10.1109/EWDTS.2015.7493149.
20. Goessel M., Graf S. Error Detection Circuits. London: McGraw-Hill, 1994. 261 p.
21. Fujiwara E. Code Design for Dependable Systems: Theory and Practical Applications. John Wiley & Sons, 2006. 720 p.
22. Efanov D., Sapozhnikov V., Sapozhnikov Vl. Generalized algorithm of building summation codes for the tasks of technical diagnostics of discrete systems // *Proc. of the 15th IEEE East-West Design & Test Symposium (EWDTS'2017)*, Novi Sad, Serbia, 29 Sept.—2 Oct., 2017. P. 365—371. DOI: 10.1109/EWDTS.2017.8110126.
23. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Классификация ошибок в информационных векторах систематических кодов // *Изв. вузов. Приборостроение*. 2015. Т. 58, № 5. С. 333—343. DOI: 10.17586/0021-3454-2015-58-5-333-343.
24. Ефанов Д. В., Сапожников В. В., Сапожников Вл. В. Коды с суммированием с фиксированными значениями кратностей обнаруживаемых монотонных и асимметричных ошибок для систем технического диагностирования // *Автоматика и телемеханика*. 2019. № 6. С. 121—141.
25. Kubalik P., Kubátová H. Parity codes used for on-line testing in FPGA // *Acta Polytechnica*. 2005. Vol. 45, N 6. P. 53—59.
26. Busaba F. Y., Lala P. K. Self-checking combinational circuit design for single and unidirectional multibit errors // *J. of Electronic Testing: Theory and Applications*. 1994. Vol. 5, iss. 5. P. 19—28. DOI: 10.1007/BF00971960.

27. Saposhnikov V. V., Morosov A., Saposhnikov V. V., Göessel M. A New design method for self-checking unidirectional combinational circuits // J. of Electronic Testing: Theory and Applications. 1998. Vol. 12, iss. 1—2. P. 41—53. DOI: 10.1023/A:1008257118423.
28. Сапожников В. В., Сапожников Вл. В., Гессель М. Самодвойственные дискретные устройства. СПб: Энергоатомиздат, 2001. 331 с.
29. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Коды с суммированием, обнаруживающие любые симметричные ошибки // Электронное моделирование. 2017. Т. 39, № 3. С. 47—60.
30. Ефанов Д. В., Сапожников В. В., Сапожников Вл. В. Синтез самопроверяемых комбинационных устройств на основе выделения специальных групп выходов // Автоматика и телемеханика. 2018. № 9. С. 79—94.
31. Svanström M. A lower bound for ternary constant weight codes // IEEE Transact. on Information Theory. 1997. Vol. 43. P. 1630—1632.
32. Svanström M., Östergård P. R. J., Bogdanova G. T. Bounds and constructions for ternary constant-composition codes // IEEE Transact. on Information Theory. 2002. Vol. 48. P. 101—111.
33. Freiman C. V. Optimal error detection codes for completely asymmetric binary channels // Information and Control. 1962. Vol. 5, iss. 1. P. 64—71. DOI: 10.1016/S0019-9958(62)90223-1.
34. 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.
35. Ефанов Д. В., Сапожников В. В., Сапожников Вл. В. Применение модульных кодов с суммированием для построения систем функционального контроля комбинационных логических схем // Автоматика и телемеханика. 2015. № 10. С. 152—169.
36. Borden J. M. Optimal asymmetric error detecting codes // Information and Control. 1982. Vol. 53, iss. 1—2. P. 66—73. DOI: 10.1016/S0019-9958(82)91125-1

Сведения об авторе

Дмитрий Викторович Ефанов — д-р техн. наук, доцент; Российский университет транспорта, кафедра автоматизации, телемеханики и связи на железнодорожном транспорте;
E-mail: TrES-4b@yandex.ru

Поступила в редакцию
31.01.2020 г.

Ссылка для цитирования: Ефанов Д. В. Ошибки в троичных кодовых векторах, их классификация и обнаружение с помощью помехозащищенного кодирования // Изв. вузов. Приборостроение. 2020. Т. 63, № 5. С. 391—404.

ERRORS IN TERNARY CODE VECTORS, THEIR CLASSIFICATION AND DETECTION USING NOISE-RESISTANT CODING

D. V. Efanov

Russian University of Transport, 127994, Moscow, Russia
E-mail: TrES-4b@yandex.ru

Specific of errors that occur in ternary code vectors due to distortions of one or more digits and are characterized by multiplicity and type, are considered. A classification of errors in ternary code vectors is proposed, and comparison with errors occurring in binary code vectors is performed. Definitions of various types of errors in binary and ternary code vectors are given, and examples are presented to illustrate features of such errors. The classification of errors in ternary code vectors allows to develop data encoding principles for detecting errors of a certain type and use them in both data transmission and synthesis of devices functioning in ternary logic that have the property of detecting faults. Some methods of constructing ternary codes aimed at detecting errors in information vectors or in code words in general are described. It is stated that the results can be used for developing a basis for synthesis of reliable and safe automation devices operating in ternary logic.

Keywords: errors in code vectors, binary code vectors, ternary code vectors, code vector errors classification, codes with error detection, codes with monotonous error detection

REFERENCES

1. Brusentsov N.P., Maslov S.P., Rozin V.P., Tishulina A.M. *Malaya tsifrovaya vychislitel'naya mashina "Setun"* (Small Digital "Setun" Computer), Moscow, 1962, 140 p. (in Russ.)
2. Pospelov D.A. *Logicheskiye metody analiza i sinteza skhem* (Logical Methods of Analysis and Synthesis of Circuits), Moscow, 1974, 368 p. (in Russ.)
3. Connely J. *Ternary Computing Tested 3-Trit Computer Architecture*, California Polytechnic State University of San Luis Obispo, August 29, 2008, 184 p.
4. Petrov A. *Populyarnaya mekhanika*, 2011, no. 9, pp. 72–76. (in Russ.)
5. Ahmad S., Alam M. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 2014, no. 4(5), pp. 5157–5160.
6. Cambou B., Flikkema P.G., Palmer J., Telesca D., Philabaum C. *Cryptography*, 2018, no. 1(2), pp. 1–16, DOI: 10.3390/cryptography2010006.
7. Efanov D.V. *Proceedings of 17th IEEE East-West Design & Test Symposium (EWDTS'2019)*, Batumi, Georgia, September 13–16, 2019, pp. 315–319, DOI: 10.1109/EWDTS.2019.8884414.
8. Wu J. *Proceedings of 19th International Symposium on Multiple-Valued Logic*, May 29–31 1989, Guangzhou, China, pp. 94–99, DOI: 10.1109/ISMVL.1989.37766.
9. Kim S., Lim T., Kang S. *23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 22–25 January 2018, Jeju, South Korea, pp. 476–481, DOI: 10.1109/ASPDAC.2018.8297369.
10. Vudadha C., Rajagopalan S., Dusi A., Phaneendra P.S., Srinivas M.B. *IEEE Transactions on Nanotechnology*, 2018, no. 2(17), pp. 299–310, DOI: 10.1109/TNANO.2018.2800015.
11. Lanyon B.P., Barbieri M., Almeida M.P., Jennewein T., Ralph T.C., Resch K.J., Pryde G.J., O'Brien J.L., Gilchrist A., White A.G. *Nature Physics*, 2009, no. 2(5), pp. 134–140, DOI: 10.1038/nphys1150.
12. 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.)
13. Sapozhnikov V.V., Sapozhnikov V.I. *Samoproveryaemye diskretnyye ustroystva* (Self-Checked Discrete Devices), St. Petersburg, 1992, 224 p. (in Russ.)
14. Piestrak S.J. *Design of Self-Testing Checkers for Unidirectional Error Detecting Codes*, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 1995, 111 p.
15. Sapozhnikov V.V., Sapozhnikov V.I., Efanov D.V. *Kody Khemminga v sistemakh funktsional'nogo kontrolya logicheskikh ustroystv* (Hamming Codes in Logic Devices Functional Control Systems), St. Petersburg, 2018, 151 p. (in Russ.)
16. Sagalovich Yu.L. *Problemy peredachi informatsii*, 1976, no. 4(12), pp. 296–299. (in Russ.)
17. Matrosova A.Yu., Levin I., Ostanin S.A. *VLSI Design*, 2000, no. 1(11), pp. 47–58, DOI: 10.1155/2000/46578.
18. Sagalovich Yu.L., Shcherbakov N.S. *Problemy peredachi informatsii*, 1984, no. 1(20), pp. 14–20. (in Russ.)
19. Tshagharyan G., Harutyunyan G., Shoukourian S., Zorian Y. *Proceedings of 13th IEEE East-West Design & Test Symposium (EWDTS'2015)*, Batumi, Georgia, September 26–29, 2015, pp. 19–22, DOI: 10.1109/EWDTS.2015.7493149.
20. Goessel M., Graf S. *Error Detection Circuits*, London, McGraw-Hill, 1994, 261 p.
21. Fujiwara E. *Code Design for Dependable Systems: Theory and Practical Applications*, John Wiley & Sons, 2006, 720 p.
22. Efanov D., Sapozhnikov V., Sapozhnikov V.I. *Proceedings 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.
23. Sapozhnikov V.V., Sapozhnikov V.I., Efanov D.V. *Journal of Instrument Engineering*, 2015, no. 5(58), pp. 333–343, DOI: 10.17586/0021-3454-2015-58-5-333-343. (in Russ.)
24. Efanov D.V., Sapozhnikov V.V. *Automation and Remote Control*, 2019, no. 6(80), pp. 1082–1097. (in Russ.)
25. Kubalík P., Kubátová H. *Acta Polytechnica*, 2005, no. 6(45), pp. 53–59.
26. Busaba F.Y., Lala P.K. *Journal of Electronic Testing: Theory and Applications*, 1994, no. 5(5), pp. 19–28, DOI: 10.1007/BF00971960.
27. Saposhnikov V.V., Morosov A., Saposhnikov V.I., Göessel M. *Journal of Electronic Testing: Theory and Applications*, 1998, no. 1–2(12), pp. 41–53, DOI: 10.1023/A:1008257118423.
28. Saposhnikov V.V., Saposhnikov V.I., Göessel M. *Samodvoystvennyye diskretnyye ustroystva* (Self-Dual Discrete Devices), St. Petersburg, 2001, 331 p. (in Russ.)
29. Sapozhnikov V.V., Sapozhnikov V.I., Efanov D.V., Cherepanova M.R. *Electronic Modeling*, 2016, no. 3(38), pp. 47–61. (in Russ.)
30. Efanov D.V., Sapozhnikov V.V. *Automation and Remote Control*, 2018, no. 9(79), pp. 1609–1620. (in Russ.)
31. Svanström M. *IEEE Transactions on Information Theory*, 1997, vol. 43, pp. 1630–1632.
32. Svanström M., Östergård P.R.J., Bogdanova G.T. *IEEE Transactions on Information Theory*, 2002, vol. 48, pp. 101–111.
33. Freiman C.V. *Information and Control*, 1962, no. 1(5), pp. 64–71, DOI: 10.1016/S0019-9958(62)90223-1.
34. Berger J.M. *Information and Control*, 1961, no. 1(4), pp. 68–73, DOI: 10.1016/S0019-9958(61)80037-5.
35. Efanov D.V., Sapozhnikov V.V. *Automation and Remote Control*, 2015, no. 10(76), pp. 1834–1848. (in Russ.)

36. Borden J.M. *Information and Control*, 1982, no. 1–2(53), pp. 66–73. DOI: 10.1016/S0019-9958(82)91125-1.

Data on author

Dmitry V. Efanov

— Dr. Sci., Associate Professor; Russian University of Transport, Department of Automation and Telemechanics on Railway Transport; E-mail: TrES-4b@yandex.ru

For citation: Efanov D. V. Errors in ternary code vectors, their classification and detection using noise-resistant coding. *Journal of Instrument Engineering*. 2020. Vol. 63, N 5. P. 391—404 (in Russian).

DOI: 10.17586/0021-3454-2020-63-5-391-404