

ПРИМЕНЕНИЕ КОДОВ ВАРШАМОВА—ТЕНЕНГОЛЬЦА ПРИ СИНТЕЗЕ УСТРОЙСТВ АВТОМАТИКИ

Д. В. ЕФАНОВ

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

Рассматриваются особенности применения кодов Варшамова—Тененгольца для задач построения систем автоматики с учетом возможностей как обнаружения неисправностей, так и коррекции их опасных проявлений. Описаны принципы построения кодов Варшамова—Тененгольца, дается краткий анализ характеристик обнаруживаемых ими ошибок в кодовых словах. Представлена структурная схема кодирующего оборудования для рассматриваемых кодов, основанная на использовании сумматоров двоичных чисел в кольце вычетов по произвольному, но заранее известному, модулю и компаратора. Кодирующее оборудование позволяет из полного множества двоичных кодовых векторов заданной длины выделить те, которые образуют соответствующий код Варшамова—Тененгольца. Приводятся примеры использования рассматриваемых кодов при построении устройств и систем автоматики с различными свойствами архитектуры (обнаружением неисправностей, коррекцией ошибок). Предложены алгоритмы синтеза отказоустойчивых комбинационных и последовательностных устройств автоматики на основе применения кодов Варшамова—Тененгольца.

Ключевые слова: *системы автоматического управления, помехоустойчивое кодирование, коды с коррекцией ошибок, код Варшамова—Тененгольца, контролепригодная архитектура, диагностическое обеспечение, обнаружение ошибок, исправление ошибок*

Введение. При построении систем автоматического управления и контроля как в транспортной отрасли, так и в промышленности широко применяются методы помехоустойчивого кодирования [1—6]. Кодирование используют при выборе архитектуры управляющих комплексов и способов внесения избыточности в аппаратные и программные средства для достижения высоких показателей надежности [7—9]. Кроме того, помехоустойчивое кодирование применяют при передаче данных между управляющими вычислительными комплексами и периферийными объектами автоматики [10]. Кодирование используют и при выборе способов синтеза контролепригодных структур компонентов систем управления, а также при выборе средств диагностического обеспечения отдельных блоков и устройств [11—15].

Применяемые способы кодирования крайне разнообразны, а выбор того или иного определяется „исходными данными“ — набором информации о том, от каких видов неисправностей или ошибок защищают технический объект и каковы при этом ограничения по сложности аппаратных средств и времени на выполнение вычислительных процедур.

Часто при построении систем автоматического управления и контроля используются блочные равномерные коды. Среди них применяются как коды, ориентированные на коррекцию ошибок (коды Хэмминга, Рида—Соломона и др.), так и коды, ориентированные только на их обнаружение (коды Бергера, равновесные коды и др.). Для коррекции ошибок в кодовых векторах в общем случае требуется бóльшая избыточность, чем для их обнаружения, а значит, бóльшие аппаратные или временные затраты на реализацию „самой защиты“. Кроме того, в некоторых технических задачах требуется использование помехоустойчивого кодирования для исправления ошибки (например, при передаче данных к периферийным объектам), а в некоторых такое свойство кода может привести к накоплению отказов и к снижению надежности (например, при организации диагностического обеспечения системы или устройства).

Кодирование особо важно в решении современных задач построения систем автоматического управления и контроля. В настоящей статье исследуется возможность применения при построении надежных и отказоустойчивых систем автоматики кодов Варшавова—Тененгольца [16]. Данные коды предназначены для обнаружения и коррекции однократных ошибок в кодовых словах при условии однонаправленного (монотонного) проявления искажений. Их можно использовать как на этапе проектирования устройств и систем автоматики при выборе контролепригодных архитектур и способов реализации технических средств диагностирования, так и при защите данных на различных уровнях организации систем (в том числе при взаимодействии разнородных систем, например, стационарной системы управления движением поездов и бортовых систем локомотивов).

Коды Варшавова—Тененгольца. Рассмотрим следующий способ построения двоичного помехозащищенного кода, описанный Р. Р. Варшавовым и Г. М. Тененгольцем [16]:

$$W = \sum_{i=1}^n i\alpha_i \equiv \beta \pmod{(n+1)}, \quad (1)$$

где $\alpha_i = \langle \alpha_1 \alpha_2 \dots \alpha_{n-1} \alpha_n \rangle$ — двоичные кодовые векторы длиной n , β — произвольное целое число, удовлетворяющее соотношению $0 \leq \beta \leq n$.

Код Варшавова—Тененгольца представляет собой множество всех кодовых векторов, для которых удовлетворяется соотношение (1).

Проиллюстрируем принцип сравнения чисел W и β для кода Варшавова—Тененгольца при $n=12$ и $\beta=3$. Покажем, что кодовый вектор $\langle 111001100100 \rangle$ принадлежит рассматриваемому коду.

Для этого вначале определим значение числа W :

$$\begin{aligned} W &= \sum_{i=1}^{12} i\alpha_i = 1 \cdot \alpha_1 + 2 \cdot \alpha_2 + \dots + 11 \cdot \alpha_{11} + 12 \cdot \alpha_{12} = \\ &= 1 \cdot 1 + 2 \cdot 1 + 3 \cdot 1 + 4 \cdot 0 + 5 \cdot 0 + 6 \cdot 1 + 7 \cdot 1 + 8 \cdot 0 + 9 \cdot 0 + 10 \cdot 1 + 11 \cdot 0 + 12 \cdot 0 = \\ &= 1 + 2 + 3 + 6 + 7 + 10 = 29. \end{aligned}$$

Полученное значение $W=29$ необходимо сравнить с $\beta=3$ по модулю $M = n+1 = 12+1 = 13$:

$$W \pmod{(n+1)} = 29 \pmod{13} = 3 \equiv \beta \pmod{(n+1)} = 3 \pmod{13} = 3.$$

Таким образом, представленный кодовый вектор $\langle 111001100100 \rangle$ принадлежит рассматриваемому коду.

Совокупность всех кодовых векторов, удовлетворяющих (1) при $n=12$ и $\beta=3$, и образует код Варшавова—Тененгольца.

Очевидно, что для $n=12$ может быть построено $n+1$ различных кодов.

Код Варшавова—Тененгольца является блочным, равномерным и может быть использован при построении систем автоматического управления с обнаружением неисправностей, он корректирует одиночные ошибки в так называемых асимметричных каналах передачи

данных. В таких каналах вероятность одного из видов искажений доминирует над другой: вероятность искажения $0 \rightarrow 1$ существенно выше вероятности искажения $1 \rightarrow 0$, или наоборот.

Код Варшавова—Тененгольца может быть использован при организации систем автоматики с обнаружением неисправностей. В том числе перспективна возможность применения данного кода при синтезе отказоустойчивых безопасных систем, реализованных на безопасных логических элементах [7]. Последние обладают тем же свойством, что и асимметричные каналы передачи данных, но на их выходах допустимы только монотонные искажения $1 \rightarrow 0$ (что является более жестким ограничением).

Принципы выделения кодовых векторов из множества 2^n кодовых векторов длиной n аналогичны принципам построения взвешенных кодов с суммированием в кодовом векторе и подсчета суммарного веса единичных разрядов с последующим определением наименьшего неотрицательного вычета по модулю $M = n + 1$ и записью полученного числа в контрольный вектор [17—21].

Рассмотрим особенности формирования множества кодовых векторов, принадлежащих различным кодам Варшавова—Тененгольца с $n=4$ и 5 (табл. 1 и 2). Для получения всех кодов, соответствующих $0 \leq \beta \leq n$, все множество кодовых векторов разбивается на n непересекающихся подмножеств (по числу возможных кодов). Множества являются непересекающимися, так как результат сравнения (1) — единственный для каждого кодового вектора длиной n .

Таблица 1

[N]	α_1	α_2	α_3	α_4	W	W(mod5)	$\beta=0$	$\beta=1$	$\beta=2$	$\beta=3$	$\beta=4$
0	0	0	0	0	0	0	×				
1	0	0	0	1	4	4					×
2	0	0	1	0	3	3				×	
3	0	0	1	1	7	2			×		
4	0	1	0	0	2	2			×		
5	0	1	0	1	6	1		×			
6	0	1	1	0	5	0	×				
7	0	1	1	1	9	4					×
8	1	0	0	0	1	1		×			
9	1	0	0	1	5	0	×				
10	1	0	1	0	4	4					×
11	1	0	1	1	8	3				×	
12	1	1	0	0	3	3				×	
13	1	1	0	1	7	2			×		
14	1	1	1	0	6	1		×			
15	1	1	1	1	10	0	×				

Примечание. [N] — десятичный эквивалент кодового вектора. Например, числу [13] в десятичном виде соответствует двоичный вектор <1101>.

В табл. 1 и 2 знаком „×“ отмечены кодовые векторы, входящие в соответствующее подмножество кодовых векторов, принадлежащих конкретному коду Варшавова—Тененгольца. В табл. 3 и 4 приведены различные коды Варшавова—Тененгольца, построенные для кодовых векторов с $n=4$ и 5.

Описанный способ похож на способ построения классических равновесных кодов [22]. Равновесные коды широко применяются при построении контролепригодных устройств автоматики и средств технического диагностирования благодаря возможности обнаружения любых однонаправленных (монотонных) искажений в кодовых векторах [11, 23].

Следует отметить, что анализ табл. 3 и 4 показывает наличие в классе необнаруживаемых кодами Варшавова—Тененгольца однонаправленных и разнонаправленных ошибок. Таким образом, при использовании данных кодов в задачах построения систем автоматики следует учитывать другие свойства обнаружения ошибок.

Таблица 2

[M]	α_1	α_2	α_3	α_4	α_5	W	W(mod6)	$\beta=0$	$\beta=1$	$\beta=2$	$\beta=3$	$\beta=4$	$\beta=5$
0	0	0	0	0	0	0	0	×					
1	0	0	0	0	1	5	5						×
2	0	0	0	1	0	4	4					×	
3	0	0	0	1	1	9	3				×		
4	0	0	1	0	0	3	3				×		
5	0	0	1	0	1	8	2			×			
6	0	0	1	1	0	7	1		×				
7	0	0	1	1	1	12	0	×					
8	0	1	0	0	0	2	2			×			
9	0	1	0	0	1	7	1		×				
10	0	1	0	1	0	6	0	×					
11	0	1	0	1	1	11	5						×
12	0	1	1	0	0	5	5						×
13	0	1	1	0	1	10	4					×	
14	0	1	1	1	0	9	3				×		
15	0	1	1	1	1	14	2			×			
16	1	0	0	0	0	1	1		×				
17	1	0	0	0	1	6	0	×					
18	1	0	0	1	0	5	5						×
19	1	0	0	1	1	10	4					×	
20	1	0	1	0	0	4	4					×	
21	1	0	1	0	1	9	3				×		
22	1	0	1	1	0	8	2			×			
23	1	0	1	1	1	13	1		×				
24	1	1	0	0	0	3	3				×		
25	1	1	0	0	1	8	2			×			
26	1	1	0	1	0	7	1		×				
27	1	1	0	1	1	12	0	×					
28	1	1	1	0	0	6	0	×					
29	1	1	1	0	1	11	5						×
30	1	1	1	1	0	10	4					×	
31	1	1	1	1	1	15	3				×		

Таблица 3

$\beta=0$	$\beta=1$	$\beta=2$	$\beta=3$	$\beta=4$
0000	0101	0011	0010	0001
0110	1000	0100	1011	0111
1001	1110	1101	1100	1010
1111	—	—	—	—

Таблица 4

$\beta=0$	$\beta=1$	$\beta=2$	$\beta=3$	$\beta=4$	$\beta=5$
00000	00110	00101	00011	00010	00001
00111	01001	01000	00100	01101	01011
01010	10000	01111	01110	10011	01100
10001	10111	10110	10101	10100	10010
11011	11010	11001	11000	11110	11101
11100	—	—	11111	—	—

Кодирующее оборудование. Удобная для использования структура детектора кода Варшавова—Тенегольца приведена на рис. 1, она включает устройство определения принадлежности поступающего кодового вектора $\langle \alpha_1 \alpha_2 \dots \alpha_{n-1} \alpha_n \rangle$ выбранному коду с установленным значением β . Устройство фильтрации кодовых векторов, разделяющих их на „разрешен-

ные“ и „запрещенные“ для заданного кода (его можно реализовать на основе мультиплексов), на рис. 1 не приводится.

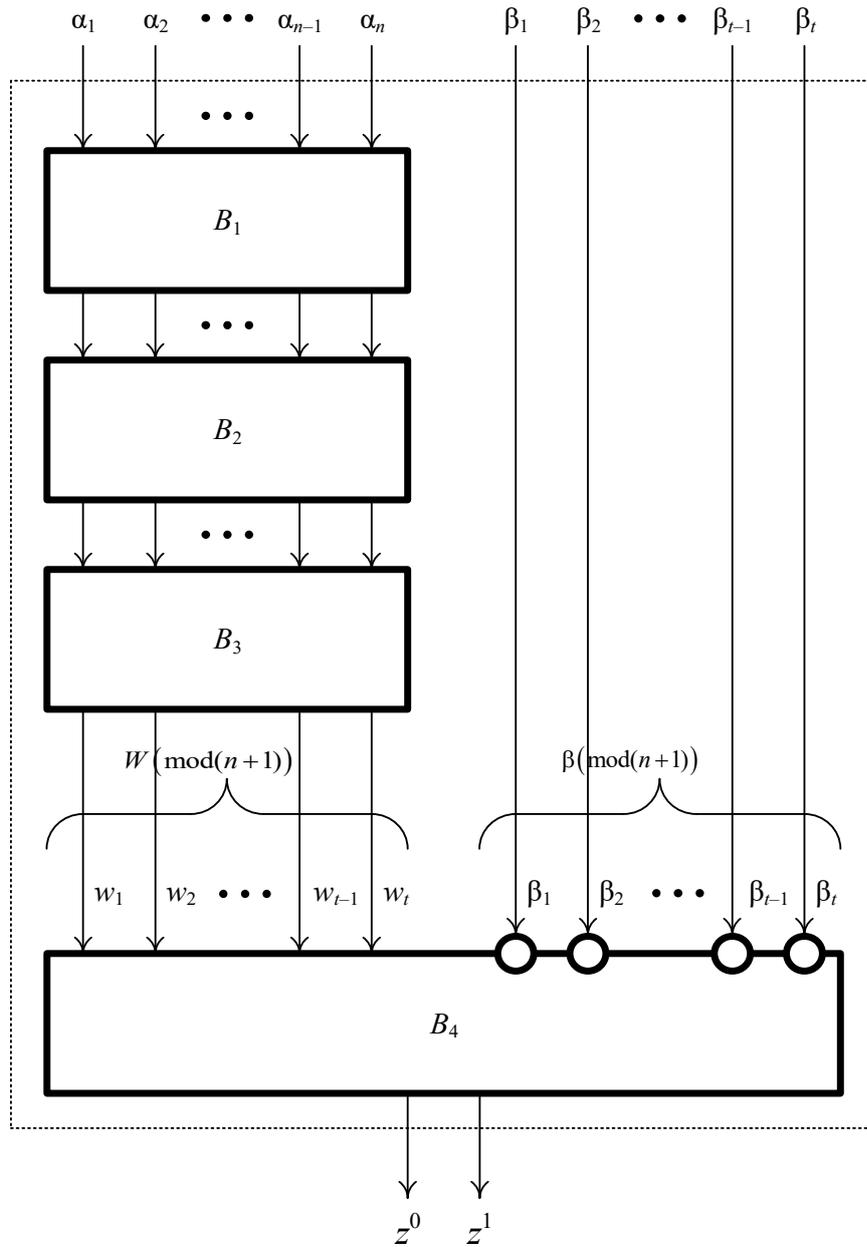


Рис. 1

Детектор состоит из двух основных модулей: формирователя числа $W(\text{mod}(n+1))$ и устройства сравнения его с заранее выбранным числом $\beta(\text{mod}(n+1))$.

Формирователь числа включает в себя: B_1 — блок взвешивания разрядов кодового слова; B_2 — блок суммирования весовых коэффициентов значащих разрядов; B_3 — блок определения наименьшего неотрицательного вычета по модулю $M = n + 1$. Блок B_1 может быть реализован в виде коммутатора сигналов для формирования двоичных чисел, соответствующих весовым коэффициентам разрядов. B_2 , представляющий собой сумматор сформированных в B_1 двоичных чисел, синтезируется известными методами [23]. Блоки B_1 и B_2 могут быть реализованы совместно, например, по методу разложения весовых коэффициентов на степени числа 2 [21]. Блок B_3 является преобразователем числа W в число $W(\text{mod}(n+1))$. Методика синтеза таких преобразователей описана, например, в [24, 25].

Устройство сравнения является классическим компаратором для двух двоичных чисел. Первое число $\langle w_1 w_2 \dots w_{t-1} w_t \rangle$ представляет собой полученный на выходах формирователя кодовый вектор $W \pmod{(n+1)}$, а второе $\langle \beta_1 \beta_2 \dots \beta_{t-1} \beta_t \rangle$ — поданный на входы компаратора кодовый вектор, соответствующий константе $\beta \pmod{(n+1)}$. Оба двоичных вектора сравниваются поразрядно в блоке B_4 . Один из вариантов его реализации — на основе модулей сжатия парафазных сигналов (*two-rail checkers*) [26], для использования которых разряды вектора, соответствующего числу $\beta \pmod{n+1}$, предварительно должны быть инвертированы.

В случае принадлежности поступающего на входы детектора кодового вектора заранее выбранному коду Варшамова—Тененгольца на выходах компаратора формируется парафазный сигнал $\langle z^0 z^1 \rangle = \langle 01 \rangle$ или $\langle z^0 z^1 \rangle = \langle 10 \rangle$, служащий сигналом управления для установления принадлежности кодового вектора $\langle \alpha_1 \alpha_2 \dots \alpha_{n-1} \alpha_n \rangle$ множеству разрешенных слов выбранного кода.

Практические приложения. Как уже отмечалось ранее, коды Варшамова—Тененгольца можно эффективно использовать при построении устройств автоматики на безопасных логических элементах. Кроме того, данные коды могут быть использованы при построении самопроверяемых систем.

Рассмотрим некоторые приложения кодов Варшамова—Тененгольца в задачах построения устройств автоматики и управления.

Синтез устройств с обнаружением неисправностей. Структурная схема конечного автомата с обнаружением неисправностей изображена на рис. 2 [27]. Для обнаружения неисправностей внутренние состояния закодированы кодом Варшамова—Тененгольца. На выходах блока памяти и выходного преобразователя устанавливаются тестеры данных кодов — TSC_1 и TSC_2 (см. рис. 2). Выходы тестеров объединяются на входах самопроверяемого компаратора TRC , на выходах которого формируется сигнал контроля $\langle z^0 z^1 \rangle$.

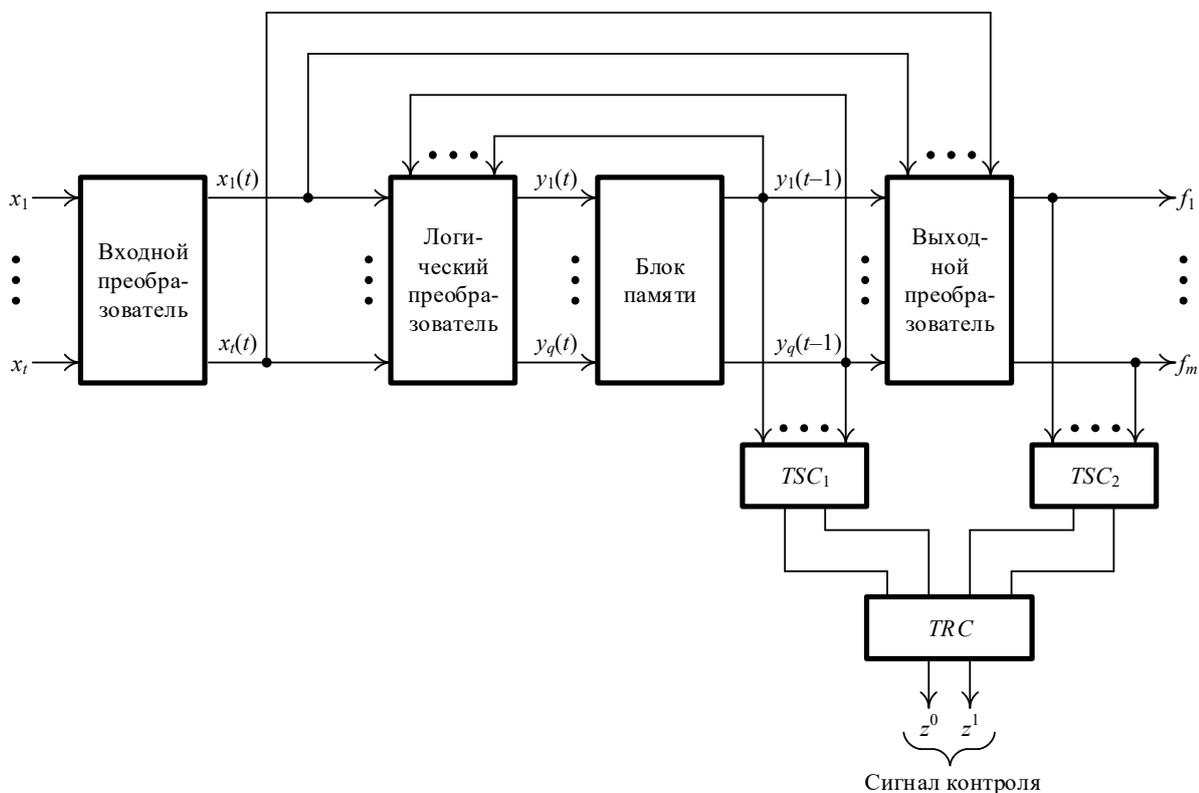


Рис. 2

Для наделения синтезируемого устройства свойством самопроверяемости требуется выполнение следующих условий.

1. При кодировании внутренних и выходных состояний автомата необходимо обеспечить наличие двух непересекающихся подмножеств внутренних рабочих и нерабочих защитных состояний.

2. Значения функций переходов автомата должны быть определены таким образом, чтобы при наличии неисправности осуществлялся переход либо в правильное состояние автомата, либо в одно из защитных состояний.

3. Значения функций выходов автомата должны быть определены таким образом, чтобы при наличии неисправности вычислялись корректные выходные значения, либо, при наличии ошибочных состояний, формировались защитные выходные состояния.

Все эти условия в полной мере выполняются при использовании рассматриваемых в данной статье кодов.

Синтез систем рабочего диагностирования. В таких системах исходное устройство дополняется техническими средствами, позволяющими в процессе функционирования осуществлять техническое диагностирование, определять его состояние косвенным образом, фиксируя неисправности по результатам вычислений рабочих функций объекта [28]. При построении систем рабочего диагностирования часто используют дублирование с последующим сравнением результатов вычислений или синтез самопроверяемых схем встроенного контроля, реализуемых с уменьшенной избыточностью, чем при дублировании. Такие схемы синтезируются при использовании блочных равномерных кодов [29]. Возможно применение при синтезе схем контроля и рассматриваемых в данной статье кодов.

При реализации систем рабочего диагностирования с использованием кодов Варшавова—Тененгольца могут быть использованы как классическая структурная схема, реализующая принцип доопределения формирующегося на выходах объекта диагностирования кодового вектора до кодового вектора, принадлежащего заданному коду, так и модифицированная структурная схема, включающая в себя блок логического дополнения рабочих функций и позволяющая преобразовать их в кодовый вектор выбранного кода [30]. В первом случае контрольное оборудование системы диагностирования строится так, чтобы рабочие функции дополнялись контрольными таким образом, чтобы формируемый кодовый вектор принадлежал коду Варшавова—Тененгольца. Второй способ построения системы диагностирования универсален и позволяет преобразовывать любой вектор рабочих функций в кодовый вектор кода Варшавова—Тененгольца.

Синтез отказоустойчивых устройств автоматики. В [30] описан подход к синтезу отказоустойчивых устройств автоматики, основанный на внесении значительной избыточности в структуру исходного устройства путем первоначального помехоустойчивого кодирования его состояний. Число разрядов в кодовых векторах, необходимых для защиты от δ одновременно возникающих неисправностей, определяется с учетом минимального расстояния Хэмминга:

$$d_{\min} \geq 2\delta + 1. \quad (2)$$

Например, если $\delta=1$, $d_{\min} \geq 3$.

Если устройство является комбинационным, т.е. не обладает памятью, то фактически реализуются три его идентичные копии с мажоритарной коррекцией сигналов. Несколько сложнее реализуется отказоустойчивое устройство с памятью.

Если устройство с памятью имеет всего два состояния S_1 и S_2 , то для его синтеза каждое состояние кодируется одним символом (0 или 1). При этом для защиты от однократных неисправностей требуется закодировать каждое из состояний трехбитными векторами: соответственно $\langle 000 \rangle$ и $\langle 111 \rangle$. Если состояний три (S_1, S_2, S_3) или четыре (S_1, S_2, S_3, S_4), то при синтезе

устройства каждое из состояний кодируется уже двухбитными векторами, а для защиты от однократных неисправностей потребуется кодирование каждого бита тремя битами. Избыточность, необходимая для защиты устройства от однократных неисправностей, существенно возрастает с увеличением числа состояний. С практической точки зрения наделение исходного устройства свойством отказоустойчивости оказывается весьма затратным. Стоит только обратить внимание на избыточность устройства из примера в [30, рис. 6.29].

Если устройство автоматики реализуется на безопасных логических элементах, то в нем допускаются неисправности, приводящие только к искажениям вида $1 \rightarrow 0$. С учетом данного обстоятельства можно синтезировать отказоустойчивое устройство автоматики с уменьшенной структурной избыточностью, чем при описанном в [30] подходе. Для этих целей эффективным может оказаться использование рассматриваемых кодов.

Обратимся для примера к табл. 3. В ней представлены все кодовые векторы кодов Варшавова—Тененгольца с длиной $n=4$. Если речь идет о безопасной схеме, то в ней возможно возникновение только искажений вида $1 \rightarrow 0$. Любая однократная неисправность приводит к искажению только одного из разрядов кодового слова при так называемой защитной ошибке [31]. На рис. 3—7 рассмотрены все вероятные переходы из основных состояний в защитные при использовании кодов Варшавова—Тененгольца с $n=4$ (рис. 3 — $\beta=0$, 4 — 1, 5 — 2, 6 — 3, 7 — 4). Как следует из примеров, подмножества защитных состояний, соответствующие различным основным состояниям, не пересекаются, а значит, могут быть однозначно отождествлены с ними.

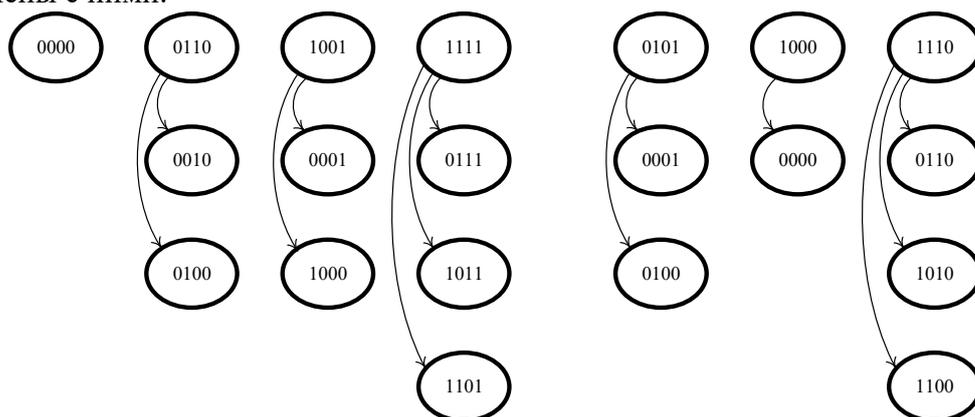


Рис. 3

Рис. 4

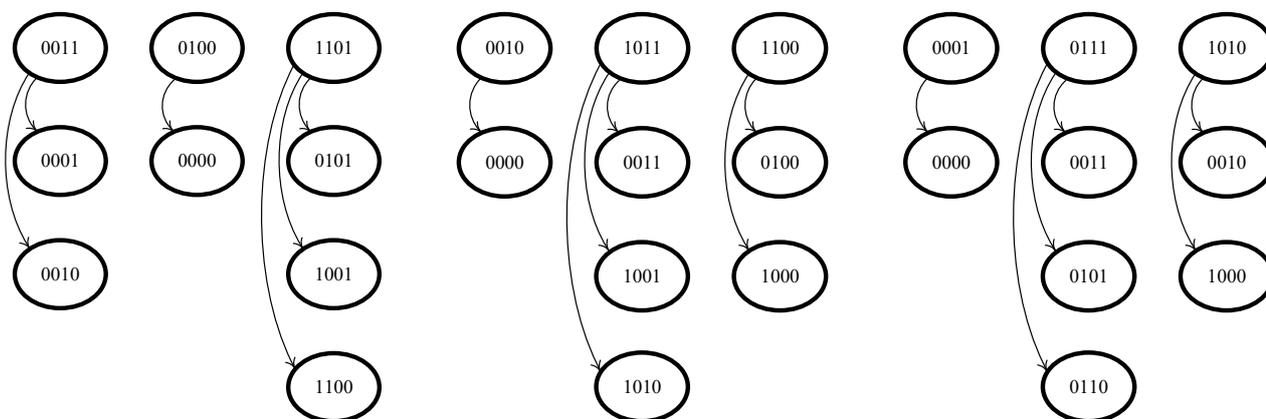


Рис. 5

Рис. 6

Рис. 7

Для сравнения на рис. 8 представлены те же множества разрешенных и защитных кодовых векторов для равновесных кодов с $n=4$ (a — код „1 из 4“; b — код „2 из 4“; v — код „3 из 4“). Защитные состояния при кодировании равновесными кодами пересекаются, а значит, не могут

быть однозначно отождествлены с основными состояниями. Равновесные коды не подходят для решения задачи синтеза отказоустойчивого устройства.

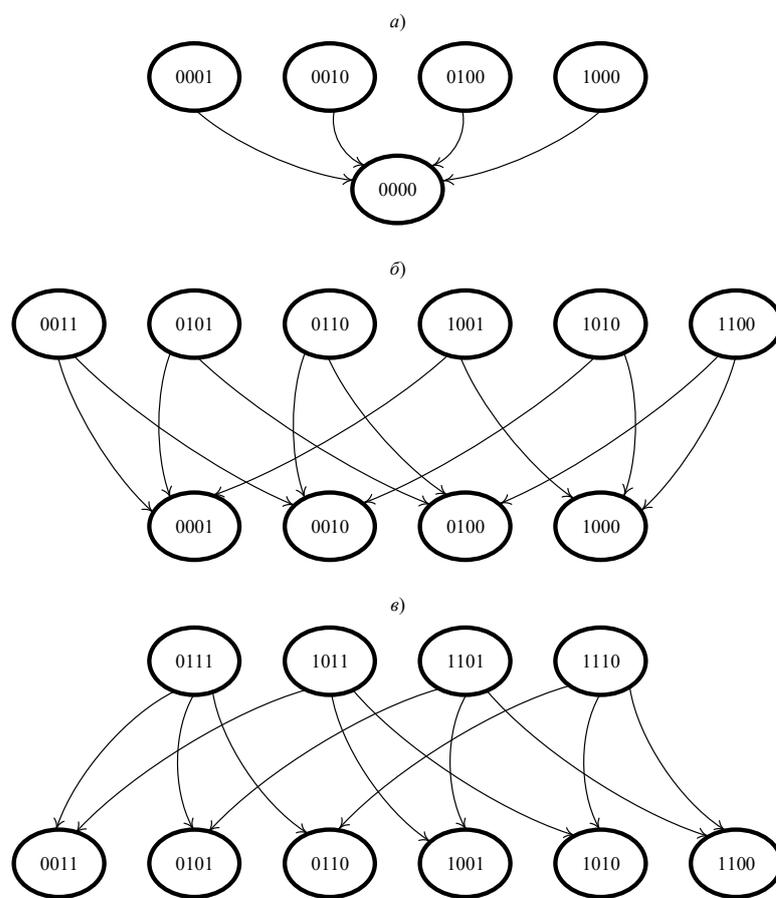


Рис. 8

Наиболее простыми устройствами автоматики являются комбинационные схемы. Для таких устройств входные кодовые векторы однозначно определяют значения разрядов выходных векторов. Методика синтеза отказоустойчивых комбинационных схем следующая.

Алгоритм 1. Синтез отказоустойчивых комбинационных схем:

1. Выбирается базис, включающий в себя безопасные логические элементы или блоки.
2. Имитируется работа схемы на всех входных наборах и определяется множество рабочих векторов.
3. В соответствии с мощностью множества рабочих векторов выбирается код Варшамова—Тененгольца, имеющий столько же или больше кодовых векторов.
4. Все рабочие векторы кодируются комбинациями кода Варшамова—Тененгольца.
5. Осуществляется перепроектирование схемы, выходные цепи которой синтезируются независимыми.

Полученное устройство будет нечувствительным к одиночным неисправностям и сбоям. Для обеспечения свойства коррекции ошибок, вызванных сбоями, в процессе функционирования требуется определенная избыточность.

Более сложными являются устройства, обладающие памятью, — последовательные схемы, или конечные автоматы. Их также можно наделять свойством отказоустойчивости.

Использование на этапе абстрактного синтеза кодов Варшамова—Тененгольца позволяет синтезировать отказоустойчивые устройства автоматики. При этом большое разнообразие способов кодирования для заданного числа n позволяет выбирать наиболее простые в технической реализации конечные устройства. Эта задача решается непосредственно при анализе работы исходного устройства.

Методика синтеза отказоустойчивых последовательностных схем следующая.

Алгоритм 2. Синтез отказоустойчивых последовательностных схем:

1. Выбирается базис, включающий в себя безопасные логические элементы или блоки, включая безопасные элементы памяти.
2. Составляется исходная таблица переходов или исходный граф переходов.
3. Состояния в таблице переходов кодируются кодом Варшавова—Тененгольца с соответствующим множеством разрешенных кодовых векторов.
4. Формируются защитные состояния, в которые устройство может перейти при наличии ошибок, вызванных защитными отказами.
5. Осуществляется доопределение всех защитных состояний до соответствующих основных состояний.
6. Неиспользованные кодовые векторы соответствуют тем защитным состояниям, которые в работе устройства недостижимы, так как оно реализуется на безопасных логических элементах, по этой причине при данных состояниях таблица переходов заполняется неопределенными (безразличными) состояниями.
7. Выбирается способ реализации устройства в виде конечного автомата (синхронный или асинхронный, Мили или Мура), а затем устройства по соответствующему алгоритму [30].

Исследования показывают, что указанный алгоритм синтеза позволяет построить отказоустойчивые устройства только при соблюдении ряда ограничений. При этом обеспечивается защита от ошибок элементов памяти в регистре памяти. Однако требуется ограничить пространство кратных неисправностей в блоке логического преобразователя, которые могут быть следствиями одиночных неисправностей в регистре памяти, проявляющихся путем трансляции ошибок через обратные связи. Это же касается входной схемы синхронизации. Требуется резервирование элементов памяти и реализация независимых схем включения элементов памяти. Кратность резервирования каждого элемента памяти определяется числом сигналов соответствующего элемента, используемого в комбинационной составляющей устройства. В ряде случаев такое резервирование может оказаться нецелесообразным.

Заключение. Возможности применения кодов Варшавова—Тененгольца при построении систем автоматики достаточно разнообразны: могут использоваться как свойство обнаружения ими ошибок в кодовых векторах, так и свойство коррекции однократных искажений. Это позволяет применять данные коды как при решении задачи организации диагностического обеспечения устройств, так и при реализации контролепригодных архитектур на этапе проектирования устройств.

Ограничением на применение кодов Варшавова—Тененгольца является то, что они корректируют только однократные однонаправленные ошибки в кодовых векторах. Тем не менее такие ошибки являются более вероятными, чем ошибки бóльших кратностей. В приложениях к синтезу устройств с отказоустойчивыми архитектурами на безопасных логических элементах эта особенность кодов может быть эффективно использована.

В качестве альтернативы использованию рассматриваемых кодов в задачах реализации устройств автоматики можно рассматривать коды Хэмминга [32], также широко применяемые для решения схожих задач [15]. При этом в самой реализации устройства можно применять и логический базис с симметричными отказами. Однако и в этом случае остается ограничение на структуру блока логического преобразователя.

Принципы построения кодов Варшавова—Тененгольца таковы, что фактически при использовании (1) в процессе выбора кодовых векторов применяется взвешивание разрядов весовыми коэффициентами из натурального ряда чисел (аналогично тому, как это проделано в [33] при построении кодов с суммированием). Такой подход позволяет строить коды с коррекцией одиночных ошибок в кодовых векторах при ограничении видов ошибок (только монотонное их проявление). Интересным может оказаться использование при формировании

кодовых слов других соотношений весовых коэффициентов (по аналогии с выбором последовательности при построении кода с суммированием в [34]). При этом сам формируемый код может быть ориентирован только на обнаружение ошибок, а не на их коррекцию, что актуально при решении задач синтеза диагностического обеспечения.

СПИСОК ЛИТЕРАТУРЫ

1. Pradhan D. K. Fault-Tolerant Computer System Design. NY: Prentice Hall, 1996. 560 p.
2. Дрозд А. В., Харченко В. С., Антощук С. Г., Дрозд Ю. В., Дрозд М. А., Сулима Ю. Ю. Рабочее диагностирование безопасных информационно-управляющих систем. Харьков: Национальный аэрокосмический университет им. Н. Е. Жуковского „ХАИ“, 2012. 614 с.
3. Глебова Е. С., Блинные А. А. Модификация меток стелеразливочных ковшей в задаче автоматизации их оборота // Изв. вузов. Приборостроение. 2015. Т. 58, № 9. С. 765—769.
4. Borecký J., Kohlík M., Kubátová H. Parity Driven Reconfigurable Duplex System // Microprocessors and Microsystems. 2017. Vol. 52. P. 251—260. DOI: 10.1016/j.micpro.2017.06.015.
5. Nahanov V. Cyber Physical Computing for IoT-driven Services. NY: Springer International Publishing AG, 2018. 279 p.
6. Шаманов В. И. Системы интервального регулирования движения поездов с цифровыми радиоканалами // Автоматика на транспорте. 2018. Т. 4, № 2. С. 223—240.
7. Сапожников В. В., Сапожников Вл. В., Христов Х. А., Гавзов Д. В. Методы построения безопасных микроэлектронных систем железнодорожной автоматики. М.: Транспорт, 1995. 272 с.
8. Волковой А. В., Лысенко И. В., Харченко В. С., Шурыгин О. В. Многоверсионные системы и технологии для критических приложений. Харьков: Нац. аэрокосм. ун-т им. Н. Е. Жуковского „ХАИ“, 2008. 224 с.
9. Kharchenko V., Kondratenko Yu., Kasprzyk J. Green IT Engineering: Concepts, Models, Complex Systems Architectures. Springer Book series "Studies in Systems, Decision and Control". 2017. Vol. 74. 305 p.
10. Fujiwara E. Code Design for Dependable Systems: Theory and Practical Applications. John Wiley & Sons, 2006. 720 p.
11. Согомонян Е. С., Слабаков Е. В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
12. Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science+Business Media B.V., 2008. 184 p.
13. Тельпухов Д. В., Деменева А. И., Жукова Т. Д., Хрущев Н. С. Исследование и разработка систем автоматизированного проектирования схем функционального контроля комбинационных логических устройств // Электронная техника. Сер. 3. Микроэлектроника. 2018. № 1. С. 15—22.
14. 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.
15. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Коды Хэмминга в системах функционального контроля логических устройств. СПб: Наука, 2018. 151 с.
16. Варшавов Р. Р., Тененголец Г. М. Код, исправляющий одиночные несимметричные ошибки // Автоматика и телемеханика. 1965. Т. XXVI, № 2. С. 288—292.
17. Berger J. M. A Note on Burst Detection Sum Codes // Information and Control. 1961. Vol. 4, is. 2—3. P. 297—299. DOI: 10.1016/S0019-9958(61)80024-7.
18. Das D., Toubia N. A. Weight-Based Codes and Their Application to Concurrent Error Detection of Multilevel Circuits // Proc. of the 17th IEEE VLSI Test Symposium. USA, CA, Dana Point, 25—29 April 1999. P. 370—376.
19. Das D., Toubia N. A., Seuring M., Gossel M. Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes // Proc. of IEEE 6th Intern. On-Line Testing Workshop (IOLTW). Spain, Palma de Mallorca, 3—5 July 2000. P. 171—176. DOI: 10.1109/OLT.2000.856633.

20. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Взвешенные коды с перестановками — новый класс кодов с суммированием для технической диагностики дискретных систем // Электронное моделирование. 2018. Т. 40, № 2. С. 43—70. DOI: 10.15407/emodel.40.02.043.
21. Efanov D. V., Sapozhnikov V. V., Sapozhnikov Vl. V. Using Codes with Summation of Weighted Bits to Organize Checking of Combinational Logical Devices // Automatic Control and Computer Sciences. 2019. Vol. 53, is. 1. P. 1—11. DOI: 10.3103/S0146411619010061.
22. Freiman C. V. Optimal Error Detection Codes for Completely Asymmetric Binary Channels // Information and Control. 1962. Vol. 5, is. 1. P. 64—71. DOI: 10.1016/S0019-9958(62)90223-1.
23. Piestrak S. J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
24. Сапожников В. В., Сапожников Вл. В., Ургансков Д. И. Универсальные структуры двоичных счетчиков единиц по произвольному модулю счета // Электронное моделирование. 2002. Т. 24, № 4. С. 65—81.
25. Сапожников В. В., Сапожников Вл. В., Ургансков Д. И. Блочная структура двоичного счетчика единиц по произвольному модулю счета // Электронное моделирование. 2005. Т. 27, № 4. С. 65—81.
26. Nikolos D. Self-Testing Embedded Two-Rail Checkers // J. of Electronic Testing: Theory and Applications. 1998. Vol. 12, is. 1—2. P. 69—79. DOI: 10.1023/A:1008281822966.
27. Сапожников В. В., Сапожников Вл. В. Самопроверяемые дискретные устройства. СПб: Энергоатомиздат, 1992. 224 с.
28. Пархоменко П. П., Согомонян Е. С. Основы технической диагностики (оптимизация алгоритмов диагностирования, аппаратурные средства). М.: Энергоатомиздат, 1981. 320 с.
29. Nicolaidis M., Zorian Y. On-Line Testing for VLSI — A Compendium of Approaches // J. of Electronic Testing: Theory and Application. 1998. Vol. 12, is. 1—2. P. 7—20. DOI: 10.1023/A:1008244815697.
30. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Основы теории надежности и технической диагностики. СПб: Лань, 2019. 588 с.
31. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Обнаружение опасных ошибок на рабочих выходах комбинационных логических схем // Автоматика на транспорте. 2015. Т. 1, № 2. С. 195—211.
32. Hamming R. W. Error Detecting and Correcting Codes // Bell System Technical J. 1950. Vol. 29, is. 2. P. 147—160.
33. Efanov D., Sapozhnikov V., Sapozhnikov Vl. On One Method of Formation of Optimum Sum Code for Technical Diagnostics Systems // Proc. of 14th IEEE East-West Design & Test Symposium (EWDTS'2016), Yerevan, Armenia, 14—17 October 2016. P. 158—163. DOI: 10.1109/EWDTS.2016.7807633.
34. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Взвешенные коды с суммированием для организации контроля логических устройств // Электронное моделирование. 2014. Т. 36, № 1. С. 59—80.

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
01.07.19 г.

Ссылка для цитирования: Ефанов Д. В. Применение кодов Варшамова—Тененгольца при синтезе устройств автоматизации // Изв. вузов. Приборостроение. 2020. Т. 63, № 2. С. 99—112.

APPLICATION OF VARSHAMOV—TENENHOLTS CODES IN AUTOMATION DEVICES SYNTHESIS

D. V. Efanov

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

Specific of Varshamov—Tenenholts codes application to the problems of construction of automation systems with the account for possibilities of faults detecting as well as correcting their dangerous ma-

nifestations is considered. Principles of the Varshamov—Tenenholts codes construction are described, a brief analysis of characteristics of errors detected in code words is performed. A block diagram of the coding equipment for the codes under consideration is presented; the approach is based on the use of binary adders in the residue ring of arbitrary module known in advance and a comparator. The coding equipment allows to select from the complete set of binary code vectors of a given length those that form the corresponding Varshamov—Tenenholts code. Examples of the considered codes application in implementation of devices and automation systems with various properties of the architecture (fault detection, error correction) are given. Algorithms for the synthesis of fault-tolerant combinational and sequential automation devices based on the Varshamov—Tenenholts codes are proposed.

Keywords: automatic control systems, error control coding, codes with error correction, Varshamov—Tenenholts code, checkability architecture, diagnostic equipment, error detecting, error correction

REFERENCES

1. Pradhan D.K. *Fault-Tolerant Computer System Design*, NY, Prentice Hall, 1996, 560 p.
2. Drozd A.V., Kharchenko V.S., Antoshchuk S.G., Drozd Yu.V., Drozd M.A., Sulima Yu.Yu. *Rabocheye diagnostirovaniye bezopasnykh informatsionno-upravlyayushchikh sistem* (Working Diagnostics of Safe Information and Control Systems), Khar'kov, 2012, 614 p. (in Russ.)
3. Glebova E.S., Blinnikov A.A. *Journal of Instrument Engineering*, 2015, no. 9(58), pp. 765–769. (in Russ.)
4. Borecký J., Kohlík M., Kubátová H. *Microprocessors and Microsystems*, 2017, vol. 52, pp. 251–260, DOI: 10.1016/j.micpro.2017.06.015.
5. Hahanov V. *Cyber Physical Computing for IoT-driven Services*, NY, Springer International Publishing AG, 2018, 279 p.
6. Shamanov V.I. *Automation on transport*, 2018, no. 2(4), pp. 223–240. (in Russ.)
7. Sapozhnikov V.V., Sapozhnikov VI.V., Khristov Kh.A., Gavzov D.V. *Metody postroyeniya bezopasnykh mikroelektronnykh sistem zheleznodorozhnoy avtomatiki* (Methods of Construction of Safe Microelectronic Systems of Railway Automation), Moscow, 1995, 272 p. (in Russ.)
8. Volkovoy A.V., Lysenko I.V., Kharchenko V.S., Shurygin O.V. *Mnogoversionnyye sistemy i tekhnologii dlya kriticheskikh prilozheniy* (Multi-Version Systems and Technologies for Critical Applications), Khar'kov, 2008, 224 p. (in Russ.)
9. Kharchenko V., Kondratenko Yu., Kacprzyk J. *Springer Book series "Studies in Systems, Decision and Control"*, 2017, vol. 74, 305 p.
10. Fujiwara E. *Code Design for Dependable Systems: Theory and Practical Applications*, John Wiley & Sons, 2006, 720 p.
11. Sogomonyan E.S., Slabakov E.V. *Samoproveryaemye ustroystva i otkazoustoychivye sistemy* (The Self-Checked Devices and Failure-Safe Systems), Moscow, 1989, 208 p. (in Russ.)
12. Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D. *New Methods of Concurrent Checking*, Edition 1, Dordrecht, Springer Science+Business Media B.V., 2008, 184 p.
13. Tel'pukhov D.V., Demeneva A.I., Zhukova T.D., Khrushchev N.S. *Electronic Engineering. Series 3. Microelectronics*, 2018, no. 1, pp. 15–22. (in Russ.)
14. Efanov D.V., Sapozhnikov V.V. *Automation and Remote Control*, 2018, no. 9(79), pp. 1609–1620.
15. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V. *Kody Khemminga v sistemakh funktsional'nogo kontrolya logicheskikh ustroystv* (Hamming Codes in Functional Control Systems of Logical Devices), St. Petersburg, 2018, 151 p. (in Russ.)
16. Varshamov R.R., Tenengolz G.M. *Automation and Remote Control*, 1965, no. 2(XXVI), pp. 288–292.
17. Berger J.M. *Information and Control*, 1961, no. 2–3(4), pp. 297–299, DOI: 10.1016/S0019-9958(61)80024-7.
18. Das D., Toubia N.A. *Proceedings of the 17th IEEE VLSI Test Symposium*, USA, CA, Dana Point, April 25–29, 1999, pp. 370–376.
19. Das D., Toubia N.A., Seuring M., Gossel M. *Proceedings of IEEE 6th International On-Line Testing Workshop (IOLTW)*, Spain, Palma de Mallorca, July 3–5, 2000, pp. 171–176, DOI: 10.1109/OLT.2000.856633.
20. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V. *Electronic Modeling*, 2018, no. 2(40), pp. 43–70, DOI: 10.15407/emodel.40.02.043. (in Russ.)
21. Efanov D.V., Sapozhnikov V.V., Sapozhnikov VI.V. *Automatic Control and Computer Sciences*, 2019, no. 1(53), pp. 1–11, DOI: 10.3103/S0146411619010061.
22. Freiman C.V. *Information and Control*, 1962, no. 1(5), pp. 64–71, DOI: 10.1016/S0019-9958(62)90223-1.
23. Piestrak S.J. *Design of Self-Testing Checkers for Unidirectional Error Detecting Codes*, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 1995, 111 p.
24. Sapozhnikov V.V., Sapozhnikov VI.V., Urganskov D.I. *Electronic Modeling*, 2002, no. 4(24), pp. 65–81. (in Russ.)
25. Sapozhnikov V.V., Sapozhnikov VI.V., Urganskov D.I. *Electronic Modeling*, 2005, no. 4(27), pp. 65–81.
26. Nikolos D. *Journal of Electronic Testing: Theory and Applications*, 1998, no. 1–2(12), pp. 69–79, DOI: 10.1023/A:1008281822966.
27. Sapozhnikov V.V., Sapozhnikov VI.V. *Samoproveryaemye diskretnyye ustroystva* (Self-Checked

- Discrete Devices), St. Petersburg, 1992, 224 p. (in Russ.)
28. Parkhomenko P.P., Sogomonyan E.S. *Osnovy tekhnicheskoy diagnostiki (optimizatsiya algoritmov diagnostirovaniya, apparaturnyye sredstva)* (Fundamentals of Technical Diagnostics (Optimization of Diagnostic Algorithms, Hardware)), Moscow, 1981, 320 p. (in Russ.)
 29. Nicolaidis M., Zorian Y. *Journal of Electronic Testing: Theory and Application*, 1998, no. 1–2(12), pp. 7–20, DOI: 10.1023/A:1008244815697.
 30. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V. *Osnovy teorii nadezhnosti i tekhnicheskoy diagnostiki* (Fundamentals of the Theory of Reliability and Technical Diagnostics), St. Petersburg, 2019, 588 p. (in Russ.)
 31. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V. *Automation on transport*, 2015, no. 2(1), pp. 195–211. (in Russ.)
 32. Hamming R.W. *Bell System Technical Journal*, 1950, no. 2(29), pp. 147–160.
 33. Efanov D., Sapozhnikov V., Sapozhnikov VI. *Proceedings of 14th IEEE East-West Design & Test Symposium (EWDTS'2016)*, Yerevan, Armenia, October 14–17, 2016, pp. 158–163, DOI: 10.1109/EWDTS.2016.7807633.
 34. Sapozhnikov V.V., Sapozhnikov VI.V., Efanov D.V. *Electronic Modeling*, 2014, no. 1(36), pp. 59–80. (in Russ.)

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. Application of Varshamov—Tenenholts codes in automation devices synthesis. *Journal of Instrument Engineering*. 2020. Vol. 63, N 2. P. 99—112 (in Russian).

DOI: 10.17586/0021-3454-2020-63-2-99-112