

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ
COMPUTER-AIDED DESIGN

УДК 004.33.054
<https://doi.org/10.37661/1816-0301-2020-17-2-54-70>

Поступила в редакцию 19.01.2020
Received 19.01.2020

Принята к публикации 25.02.2020
Accepted 25.02.2020

Псевдоисчерпывающее тестирование запоминающих устройств на базе маршевых тестов типа March A

В. Н. Ярмолик^{1✉}, И. Мрозек², С. В. Ярмолик¹

¹Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь
✉E-mail: yarmolik10ru@yahoo.com

²Белостоцкий технический университет, Белосток, Польша

Аннотация. Показывается актуальность тестирования запоминающих устройств современных вычислительных систем. Анализируются методы и алгоритмы реализации тестовых процедур на базе классических маршевых тестов. Выделяются многократные маршевые тесты, позволяющие обнаруживать сложные кодочувствительные неисправности памяти. Для их обнаружения обосновывается необходимое условие, которому должны удовлетворять тестовые процедуры для покрытия сложных неисправностей. Это условие заключается в формировании псевдоисчерпывающего теста для заданного количества произвольных ячеек памяти. Исследуется эффективность однократного и двукратного применения тестов типа MATS++, March C– и March A, а также приводятся ее аналитические оценки для различного количества $k \leq 10$ ячеек памяти, участвующих в неисправности. Обосновывается применимость математической модели комбинаторной задачи собирателя купонов для описания многократного тестирования памяти. Приводятся значения средней, минимальной и максимальной кратности многократных тестов для обеспечения исчерпывающего множества двоичных комбинаций для заданного числа произвольных ячеек памяти. Экспериментально показывается справедливость аналитических оценок и подтверждается высокая эффективность формирования псевдоисчерпывающего покрытия тестами типа March A.

Ключевые слова: тестирование вычислительных систем, встроенное тестирование, многократное тестирование, маршевые тесты памяти, псевдоисчерпывающие тесты

Для цитирования. Ярмолик, В. Н. Псевдоисчерпывающее тестирование запоминающих устройств на базе маршевых тестов типа March A / В. Н. Ярмолик, И. Мрозек, С. В. Ярмолик // Информатика. – 2020. – Т. 17, № 2. – С. 54–70. <https://doi.org/10.37661/1816-0301-2020-17-2-54-70>

Pseudoexhaustive memory testing based on March A type march tests

Vyacheslav N. Yarmolik^{1✉}, Ireneusz Mrozek², Svetlana V. Yarmolik¹

¹Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus
✉Email: yarmolik10ru@yahoo.com

²Bialystok University of Technology, Bialystok, Poland

Abstract. The relevance of testing of memory devices of modern computing systems is shown. The methods and algorithms for implementing test procedures based on classical March tests are analyzed. Multiple March tests are highlighted to detect complex pattern-sensitive memory faults. To detect them, the necessary condition that test procedures must satisfy to deal complex faults, is substantiated. This condition is in the formation of a pseudo-exhaustive test for a given number of arbitrary memory cells. We study the effectiveness of single and

double application of tests like MATS ++, March C– and March A, and also give its analytical estimates for a different number of $k \leq 10$ memory cells participating in a malfunction. The applicability of the mathematical model of the combinatorial problem of the coupon collector for describing multiple memory testing is substantiated. The values of the average, minimum, and maximum multiplicity of multiple tests are presented to provide an exhaustive set of binary combinations for a given number of arbitrary memory cells. The validity of analytical estimates is experimentally shown and the high efficiency of the formation of a pseudo-exhaustive coverage by tests of the March A type is confirmed.

Keywords: testing of computing systems, embedded testing, multi-run testing, march memory tests, pseudo-exhaustive tests

For citation. Yarmolik V. N., Mrozek I., Yarmolik S. V. Pseudoexhaustive memory testing based on March A type march tests. *Informatics*, 2020, vol. 17, no. 2, pp. 54–70 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-54-70>

Введение. Удельный вес запоминающих устройств в структуре встроенных вычислительных систем достигает 94 % занимаемой системой площади кристалла [1–3]. Поэтому тестирование современных запоминающих устройств представляет собой востребованную и актуальную задачу, для решения которой применяются различные принципы и подходы. Среди них выделяется вероятностное тестирование и его многочисленные разновидности, которые являются эффективной аппроксимацией исчерпывающего и псевдоисчерпывающего тестирования [2–5].

Весьма востребованным методом тестирования запоминающих устройств служит многократное тестирование с изменяемыми начальными условиями на базе классических маршевых тестов [6–9]. Реализация многократного тестирования предполагает выполнение определенных процедур, основанных на изменении либо начального состояния запоминающего устройства, либо его адресной последовательности [6, 10–13]. Применение вероятностных принципов при формировании как адресных последовательностей, так и начальных состояний памяти позволило построить математическую модель тестирования памяти и оценить ее эффективность [2, 5].

Целью настоящей статьи является анализ эффективности многократных маршевых тестов запоминающих устройств с изменяемым начальным состоянием памяти. Объектом исследования служат три специфических множества маршевых тестов, таких как тесты типа MATS++, March C– и March A. Приводятся основные аналитические соотношения эффективности тестов, а также описываются их сложности для формирования псевдоисчерпывающих тестовых наборов, обеспечивающих всевозможные 2^k двоичные комбинации в произвольных k из N запоминающих ячеек памяти.

Однократные маршевые тесты. Маршевые тесты запоминающих устройств формируют в k произвольных ячейках памяти так называемую орбиту, которая представляет собой набор двоичных k -разрядных векторов [4, 11]. Конкретный набор векторов в орбите зависит от трех основных факторов [4]. Прежде всего однозначно определяет структуру орбиты используемый маршевый тест, так как он формулирует правила, по которым генерируются двоичные векторы в ячейках запоминающего устройства. Начальное состояние запоминающего устройства – второй фактор, от которого зависит вид конкретной орбиты. Последовательность формируемых маршевым тестом адресов, представляющая собой порядок обращения к запоминающим ячейкам памяти, является третьим фактором.

Для классических разрушающих маршевых тестов начальным состоянием памяти будет нулевое состояние, формирующееся в результате реализации фазы инициализации $\uparrow\downarrow(w0)$ теста [2, 4, 9]. Символ \uparrow обозначает возрастающую, а символ \downarrow убывающую последовательность адресов, которые отличаются взаимнообратным порядком генерирования, а совместное применение символов $\uparrow\downarrow$ обозначает безразличный порядок адресов [9]. Под фазой маршевого теста понимают набор операций чтения r и записи w нулевого ($r0, w0$) либо единичного ($r1, w1$) значения для каждой ячейки запоминающего устройства. Все операции одной фазы теста выделяются круглыми скобками, перед которыми указывается порядок адресов для данной фазы, а сами фазы в тесте разделяются символом «;» [9]. Последующие фазы теста за фазой инициализации, как правило, используют возрастающий либо убывающий порядок адресов, как это, например, определено для теста MATS ++ $\{\uparrow\downarrow(w0); \uparrow(r0, w1); \downarrow(r1, w0, r0)\}$ и теста March C– $\{\uparrow\downarrow(w0); \uparrow(r0, w1); \uparrow(r1, w0); \downarrow(r0, w1); \downarrow(r1, w0); \uparrow\downarrow(r0)\}$.

В работе [4] исследованы маршевые тесты типа MATS ++ и March C⁻, для которых данные из ячеек памяти считываются и в них записываются инверсные значения. Перед выполнением очередной фазы теста состояние памяти может быть нулевым либо единичным. Для $k > 1$ из N , где N – емкость памяти в битах произвольных ячеек памяти, во время выполнения фаз формируется возрастающая или обратная убывающая последовательность адресов. Поэтому в маршевых тестах типа MATS ++ и March C⁻ существует четыре различные орбиты [4]. Количество орбит и их порядок во время выполнения теста зависят от применяемого маршевого теста. Возможные орбиты для классических маршевых тестов типа MATS ++ и March C⁻, формируемые в произвольных k ячейках памяти, приведены в табл. 1 [4].

Таблица 1

Орбиты, формируемые маршевыми тестами типа MATS ++ и March C ⁻				
Орбита	O_0	O_1	O_2	O_3
Предыдущая фаза теста	(..., w0, ...)	(..., w0, ...)	(..., w1, ...)	(..., w1, ...)
Фаза теста	$\uparrow(\dots, w1, \dots)$	$\downarrow(\dots, w1, \dots)$	$\uparrow(\dots, w0, \dots)$	$\downarrow(\dots, w0, \dots)$
P_0	0 0 0 ... 0 0	0 0 0 ... 0 0	1 1 1 ... 1 1	1 1 1 ... 1 1
P_1	0 0 0 ... 0 1	1 0 0 ... 0 0	1 1 1 ... 1 0	0 1 1 ... 1 1
P_2	0 0 0 ... 1 1	1 1 0 ... 0 0	1 1 1 ... 0 0	0 0 1 ... 1 1
...
P_{k-2}	0 0 1 ... 1 1	1 1 1 ... 0 0	1 1 0 ... 0 0	0 0 0 ... 1 1
P_{k-1}	0 1 1 ... 1 1	1 1 1 ... 1 0	1 0 0 ... 0 0	0 0 0 ... 0 1
P_k	1 1 1 ... 1 1	1 1 1 ... 1 1	0 0 0 ... 0 0	0 0 0 ... 0 0

При описании предыдущей и текущей фаз теста, приведенных в табл. 1, показана последняя операция записи в фазе, после которой могут быть только операции чтения. Отметим, что орбиты O_0 , O_1 , O_2 и O_3 формируются фазами, которые инвертируют содержимое запоминающего устройства. Их применение многообразно и широко представлено в классических маршевых тестах. Например, фаза $\uparrow(r0, w1, r1, w0, r0, w1)$ теста March A и фаза $\uparrow(r0, w1, w0, w1, r1)$ теста March LA, так же как и фаза $\uparrow(r0, w1)$ теста MATS ++, формируют орбиту O_0 . В зависимости от количества фаз и их сочетаний в тесте формируется некоторое подмножество орбит [4]. Маршевый тест MATS ++ генерирует две орбиты, а именно O_0 во время реализации его фазы $\uparrow(r0, w1)$ и O_3 в результате применения фазы $\downarrow(r1, w0, r0)$, которые состоят из одинаковых двоичных наборов, сформированных в обратном порядке. В то же время March C⁻ позволяет генерировать все четыре возможные орбиты $O_i \in \{O_0, O_1, O_2, O_3\}$, что доказывает его более высокую эффективность [4].

Для тестов типа MATS++ и March C⁻ приведем следующее определение [4].

Определение 1. Орбита O_i , $i \in \{0, 1, 2, 3\}$, сформированная в произвольных k ячейках памяти как результат применения одной из фаз маршевого теста типа MATS++ и March C⁻, состоит из k -разрядных двоичных векторов P_0, P_1, \dots, P_k .

Данное определение обобщается для произвольной фазы любого маршевого теста и оценивает сложность орбиты, формируемой фазой маршевого теста. В дальнейшем будем придерживаться обозначения орбит O_i , формируемых тестами типа MATS++ и March C⁻, для которых справедливы следующие два определения [4].

Определение 2. Две орбиты O_j и O_r считаются эквивалентными или равными ($O_j = O_r$), если обе включают одинаковые двоичные векторы независимо от их порядка.

В случае маршевых тестов типа MATS ++ и March C⁻ справедливы равенства $O_0 = O_3$ и $O_1 = O_2$ (см. табл. 1). Для целей сравнения орбит введем меру расстояния $D(O_j, O_r)$ между орбитами O_j и O_r .

Определение 3. Расстоянием $D(O_j, O_r) = D(O_r, O_j)$ между орбитами O_j и O_r является количество неповторяющихся k -разрядных двоичных векторов P_0, P_1, \dots, P_k , входящих в их состав.

Тогда для эквивалентных орбит $O_j = O_r$ расстояние $D(O_j, O_r) = 0$, а для орбит O_j и O_r , которые состоят из неповторяющихся двоичных векторов P_0, P_1, \dots, P_k , входящих в их состав, $D(O_j, O_r) = k + 1$. Анализируя орбиты, представленные в табл. 1, можно показать, что $D(O_0, O_1) = D(O_2, O_3) = D(O_0, O_2) = D(O_1, O_3) = k - 1$, а $D(O_0, O_3) = D(O_1, O_2) = 0$.

Множество разновидностей орбит не ограничивается орбитами O_i , формируемыми тестами типа MATS ++ и March C-. Вторым подмножеством орбит Q_i являются орбиты, формируемые такими маршевыми тестами, как March A, March B, March LA, March LR, March M, March PS и др. [2, 7–9]. В дальнейшем подобные тесты будем называть маршевыми тестами типа March A, которые позволяют формировать орбиты Q_i , $i \in \{0, 1, 2, 3\}$, приведенные в табл. 2.

Таблица 2

Орбиты, формируемые маршевыми тестами типа March A

Орбита	Q_0	Q_1	Q_2	Q_3
Предыдущая фаза теста	(..., w0, ...)	(..., w0, ...)	(..., w1, ...)	(..., w1, ...)
Фаза теста	$\uparrow(\dots, w1, \dots, w0, \dots)$	$\downarrow(\dots, w1, \dots, w0, \dots)$	$\uparrow(\dots, w0, \dots, w1, \dots)$	$\downarrow(\dots, w0, \dots, w1, \dots)$
P_0	0 0 0 ... 0 0	0 0 0 ... 0 0	1 1 1 ... 1 1	1 1 1 ... 1 1
P_1	0 0 0 ... 0 1	1 0 0 ... 0 0	1 1 1 ... 1 0	0 1 1 ... 1 1
P_2	0 0 0 ... 1 0	0 1 0 ... 0 0	1 1 1 ... 0 1	1 0 1 ... 1 1
...
P_{k-2}	0 0 1 ... 0 0	0 0 1 ... 0 0	1 1 0 ... 1 1	1 1 0 ... 1 1
P_{k-1}	0 1 0 ... 0 0	0 0 0 ... 1 0	1 0 1 ... 1 1	1 1 1 ... 0 1
P_k	1 0 0 ... 0 0	0 0 0 ... 0 1	0 1 1 ... 1 1	1 1 1 ... 1 0

Для орбит Q_i , так же как и для орбит O_i , применимы определения 1–3 и следующее утверждение.

Утверждение 1. Расстояние Хэмминга между двумя векторами орбиты Q_i для k произвольных ячеек запоминающего устройства, расположенных по возрастанию их адресов, равняется $HD(P_0, P_r) = 1$ для $r \in \{1, 2, \dots, k\}$ и $HD(P_l, P_r) = 2$ для $l \neq r \in \{1, 2, \dots, k\}$.

Следствием утверждения 1 является отсутствие в пределах орбиты Q_i двух одинаковых двоичных векторов P_j и P_l для $j \neq l \in \{0, 1, 2, \dots, k\}$. Так же как и для O_i , множество орбит Q_i содержит эквивалентные орбиты, которые состоят из одинаковых двоичных векторов. Действительно, из табл. 2 видно, что $Q_0 = Q_1$ и $Q_2 = Q_3$. Соответственно, можно показать, что $D(Q_0, Q_1) = D(Q_2, Q_3) = 0$. Кроме того, $D(Q_0, Q_2) = D(Q_0, Q_3) = D(Q_1, Q_2) = D(Q_1, Q_3) = k + 1$. Отметим, что по сравнению с множеством орбит O_i в случае орбит Q_i достигается большее расстояние между ними, равное $k + 1$. Максимальное количество k -разрядных двоичных наборов для $k > 2$, равное $2k + 2$, может быть получено в результате применения фаз маршевого теста, формирующего, например, орбиты Q_0 и Q_3 .

Дальнейшее увеличение количества двоичных наборов возможно только путем формирования орбит из множества O_i . Каждая из орбит O_i содержит наборы, состоящие из всех единичных 1 1 1 ... 1 1 и всех нулевых 0 0 0 ... 0 0 значений, которые формируются и орбитами Q_0 и Q_3 . Кроме того, орбиты O_0 и O_2 содержат наборы 0 0 0 ... 0 1, 0 1 1 ... 1 1 и 1 0 0 ... 0 0, 1 1 1 ... 1 0, входящие в состав орбит Q_0 и Q_3 соответственно. Поэтому любая орбита O_i формирует только $k + 1 - 4 = k - 3$, $k > 3$, дополнительных k -разрядных двоичных наборов по отношению к наборам, сгенерированным ранее орбитами Q_0 и Q_3 . Таким образом, максимально возможное количество двоичных наборов в k ячейках памяти, формируемых однократным применением маршевого теста, будет определяться как $2k + 2 + 2(k - 3) = 4(k - 1)$. При этом необходимым условием получения такого количества наборов является присутствие в тесте фаз, формирующих соответствующие орбиты. Например, орбиты Q_0 и Q_3 совместно с орбитами O_0 и O_2 позволяют сформировать $4(k - 1)$ двоичных наборов. Одним из маршевых тестов, который формирует максимальное количество двоичных наборов, может быть тест

$$\{\uparrow\downarrow(w0); \uparrow(r0, w1); \downarrow(r1, w0, w1); \uparrow(r1, w0); \downarrow(r0, w1, w0) \uparrow(r0)\}. \quad (1)$$

В дальнейшем данный тест будем называть March O. Сложность данного теста равняется $12N$, и он позволяет формировать $4(k - 1)$ k -разрядных двоичных векторов (табл. 3).

Таблица 3

Орбиты, формируемые маршевым тестом *March O* для $k = 5$

Фаза теста	$\uparrow(r0, w1)$	$\downarrow(r1, w0, w1)$	$\uparrow(r1, w0)$	$\downarrow(r0, w1, w0)$
Орбита	O_0	Q_3	O_2	Q_1
P_0	0 0 0 0 0	1 1 1 1 1	1 1 1 1 1	0 0 0 0 0
P_1	0 0 0 0 1	0 1 1 1 1	1 1 1 1 0	1 0 0 0 0
P_2	0 0 0 1 1	1 0 1 1 1	1 1 1 0 0	0 1 0 0 0
P_3	0 0 1 1 1	1 1 0 1 1	1 1 0 0 0	0 0 1 0 0
P_4	0 1 1 1 1	1 1 1 0 1	1 0 0 0 0	0 0 0 1 0
P_5	1 1 1 1 1	1 1 1 1 0	0 0 0 0 0	0 0 0 0 1

Тест *March O* формирует в произвольных $k = 5$ ячейках запоминающего устройства $4(k - 1) = 4(5 - 1) = 16$ двоичных наборов. Отметим, что тесты типа *MATS++* и *March C-* для того же значения $k = 5$ соответственно формируют $k + 1 = 6$ и $2k = 10$ двоичных наборов. Для обеспечения эффективного обнаружения сложных неисправностей запоминающих устройств аналогичные сочетания орбит двух видов O_i и Q_i используются в классических маршевых тестах (рис. 1).

Для всех приведенных на рис. 1 маршевых тестов показаны сложность их реализации, оцениваемая количеством циклов обращения к запоминающему устройству емкостью N бит, а также фазы теста, которые формируют орбиты вида O_i и Q_i .

March A (15N)							
$\uparrow\downarrow(w0)$	$\uparrow(r0, w1, w0, w1)$	$\uparrow(r1, w0, w1)$	$\downarrow(r1, w0, w1, w0)$	$\downarrow(r0, w1, w0)$			
	O_0	Q_2	O_3	Q_1			
Algorithm B (17N)							
$\uparrow\downarrow(w0)$	$\uparrow(r0, w1, w0, w1)$	$\uparrow(r1, w0, r0, w1)$	$\downarrow(r1, w0, w1, w0)$	$\downarrow(r0, w1, r1, w0)$			
	O_0	Q_2	O_3	Q_1			
March U (13N)							
$\uparrow\downarrow(w0)$	$\uparrow(r0, w1, r1, w0)$	$\uparrow(r0, w1)$	$\downarrow(r1, w0, r0, w1)$	$\downarrow(r1, w0)$			
	Q_0	O_0	Q_3	O_3			
March LR (14N)							
$\uparrow\downarrow(w0)$	$\downarrow(r0, w1)$	$\uparrow(r1, w0, r0, w1)$	$\uparrow(r1, w0)$	$\uparrow(r0, w1, r1, w0)$	$\downarrow(r0)$		
	O_1	Q_2	O_0	Q_0			
March M (16N)							
$\uparrow\downarrow(w0)$	$\uparrow(r0, w1, r1, w0)$	$\uparrow\downarrow(r0)$	$\uparrow(r0, w1)$	$\uparrow\downarrow(r1)$	$\downarrow(r1, w0, r0, w1)$	$\uparrow\downarrow(r1)$	$\downarrow(r1, w0)$
	Q_0		O_0		Q_3		O_3
March PS (23N)							
$\uparrow\downarrow(w0)$	$\uparrow(r0, w1, r1, w0, r0, w1)$	$\uparrow(r1, w0, r0, w1, r1)$	$\uparrow(r1, w0, r0, w1, r1, w0)$	$\uparrow(r0, w1, r1, w0, r0)$			
	O_0	Q_2		O_2			Q_0

Рис. 1. Примеры маршевых тестов

Ранее указывалось, что максимально возможное количество двоичных наборов в k ячейках памяти, формируемых однократным применением маршевого теста, будет определяться как $4(k - 1)$, однако это достигается существенно большей сложностью теста, формирующего нужное сочетание орбит (см. рис. 1).

Двукратные маршевые тесты. Приведенные в предыдущем разделе результаты анализа позволяют сделать следующие выводы. Однократное применение маршевого теста дает возможность получить только одну, две или четыре неэквивалентные орбиты, которые зависят от начального состояния памяти и используемой адресной последовательности. В случае тестов типа *MATS++* применение маршевого теста позволяет генерировать одну орбиту, включающую $k + 1$ различных k -разрядных двоичных векторов. Две орбиты, включающие $2k$ неповторяющихся двоичных векторов, формируются тестами типа *March C-*. Максимальное количе-

ство k -разрядных двоичных векторов, равное $4(k - 1)$, обеспечивается четырьмя орбитами, генерируемыми тестами типа March A.

Для формирования всевозможных 2^k двоичных векторов в $k > 1$ произвольных ячейках памяти маршевый тест типа MATS ++ должен выполняться многократно при различных начальных условиях, обеспечивающих генерирование неэквивалентных орбит в k ячейках. Для $k > 2$ обеспечение 2^k двоичных векторов тестами типа MATS ++ и March C– достигается их применением более одного раза. Неоднократное применение любого маршевого теста, включая тесты типа March A, необходимо для достижения той же цели для $k > 3$. Во всех случаях повторное применение теста осуществляется при использовании различных начальных состояний памяти и (или) различных адресных последовательностей [2, 7, 8]. Подход, основанный на изменении адресных последовательностей, был ранее всесторонне исследован в работах [2, 7, 13], а первоначальные результаты по изменяемым начальным состояниям памяти представлены в [2, 4, 14]. В большей своей части они относились к тестам типа MATS ++. Для этих тестов был доказан и исследован ряд положений, касающихся орбит O_i , которые формируются тестами типа MATS++.

Двукратные маршевые тесты типа MATS++. Для тестов типа MATS++ справедливо следующее утверждение [4].

Утверждение 2. Для произвольных $1 < k \leq N$ ячеек памяти с адресами $\beta, \gamma, \dots, \delta, \varepsilon, \eta$, где $N \geq \beta > \gamma > \dots > \delta > \varepsilon > \eta \geq 0$, запоминающего устройства и различных 2^k их начальных состояний $P_0 = a_\beta a_\gamma \dots a_\delta a_\varepsilon a_\eta$ маршевый тест типа MATS++ при использовании фиксированной адресной последовательности, возрастающей либо убывающей, генерирует 2^k уникальных неэквивалентных орбит O_i .

Данное утверждение гарантирует формирование тестами типа MATS++ множества из $k + 1$ двоичного вектора, причем изменение начального состояния P_0 в указанных ячейках приводит к формированию очередных $k + 1$ векторов, включающих векторы, не совпадающие с ранее сгенерированными двоичными векторами. Для оценки эффективности двукратного применения тестов типа MATS++ в работе [4] приведено следующее утверждение.

Утверждение 3. Для заданного k , $1 < k \leq N$, и постоянного порядка адресов существует $2^k - (k^2 + k)/2 - 1$ орбит O_i , формируемых повторным применением теста типа MATS++ с отличающимися начальными состояниями P_0 и состоящих из двоичных векторов, которые не входят в исходную орбиту O_0 , и $(k^2 + k)/2$ орбит, включающих по два вектора, которые входят в орбиту O_0 .

Таким образом, изменение начального состояния P_0 ячеек памяти приводит к формированию только двух видов орбит. Их точное число, равное $M_1 = (k^2 + k)/2$ и $M_2 = 2^k - (k^2 + k)/2 - 1$, позволяет оценить среднее число M_{ave} двоичных векторов, формируемых в результате двукратного маршевого теста типа MATS ++. Основываясь на общей идее многократного тестирования запоминающих устройств [4], исходные состояния ячеек памяти перед последующим применением маршевого теста формируются как случайные равномерно распределенные значения 0 и 1. Это означает, что для тех же k ячеек памяти новая орбита формируется с равной вероятностью из всех возможных 2^k орбит. Тогда среднее количество новых двоичных векторов, формируемых во время повторного применения теста MATS ++, будет равняться $[M_1 \times (k - 1) + M_2 \times (k + 1)]/2^k$, а значение среднего числа $M_{ave}(\text{MATS ++})$ двоичных векторов, формируемых в результате двукратного маршевого теста MATS ++, определяется соотношением

$$M_{ave}(\text{MATS ++}) = (k + 1) + \frac{M_1 \times (k - 1) + M_2 \times (k + 1)}{2^k} = 2(k + 1) - \frac{(k + 1)^2}{2^k}. \quad (2)$$

Очевидно, что значение M_{ave} существенно увеличится при применении тестов, формирующих более сложные орбиты и большее их число, например тестов типа March C–.

Двукратные маршевые тесты типа March C–. Для оценки эффективности двукратного применения тестов типа March C– отметим, что подобные тесты формируют две неэквивалент-

ные орбиты вида O_i , которыми в зависимости от применяемого теста могут быть O_0 и O_1 , или O_2 и O_3 , либо обе пары орбит вместе. В дальнейшем будем применять обозначения O_{01} и O_{23} для идентификации указанных пар орбит. Данные пары орбит имеют только два повторяющихся двоичных набора P_0 и P_k (см. табл. 1), поэтому их совместное формирование при однократном применении теста обеспечивает в произвольных k ячейках памяти $(k+1) + (k+1) - 2 = 2k$ двоичных наборов. Рассмотрим случай пары орбит O_0 и O_1 (O_{01}), для которых начальное состояние $P_0(O_0)$ первой орбиты O_0 равняется начальному состоянию $P_0(O_1)$ второй орбиты O_1 . Для финальных состояний обеих орбит также выполняется равенство $P_k(O_0) = P_k(O_1)$. Остальные k -разрядные двоичные векторы P_1, P_2, \dots, P_{k-1} , входящие в обе орбиты, связаны соотношением $P_i(O_0) = \overline{P_{k-i}(O_1)}$, $i = \overline{1, k-1}$. Например, для $i = 1$ $P_1(O_0) = 0\ 0\ 0 \dots 0\ 1 = P_{k-1}(O_1) = 1\ 1\ 1 \dots 1\ 0$.

При произвольных k , $2 < k \leq N$, ячейках памяти с адресами $\beta, \gamma, \dots, \delta, \varepsilon, \eta$, где $N \geq \beta > \gamma > \dots > \delta > \varepsilon > \eta \geq 0$, запоминающего устройства и различных 2^k их начальных состояний $P_0 = a_\beta a_\gamma \dots a_\delta a_\varepsilon a_\eta$, где $a_i \in \{0, 1\}$, $i \in \{\beta, \gamma, \dots, \delta, \varepsilon, \eta\}$, а $\overline{a_i}$ принимает инверсное значение относительно a_i , структура всего множества неповторяющихся $2k$ двоичных векторов для случая O_{01} определена в табл. 4.

Таблица 4

Пара орбит O_{01} , состоящая из $2k$ двоичных векторов

Вектор	Индексы разрядов вектора					
	k	$k-1$...	3	2	1
P_0	a_β	a_γ	...	a_δ	a_ε	a_η
P_1	a_β	a_γ	...	a_δ	a_ε	$\overline{a_\eta}$
P_2	a_β	a_γ	...	a_δ	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_3	a_β	a_γ	...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
...
P_{k-1}	a_β	a_γ	...	a_δ	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_k	$\overline{a_\beta}$	$\overline{a_\gamma}$...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_{k+1}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	a_δ	a_ε	a_η
P_{k+2}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	a_δ	a_ε	a_η
...
P_{2k-2}	a_β	a_γ	...	a_δ	a_ε	a_η
P_{2k-1}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$

Для всех возможных $2^3 = 8$ ($k = 3$) начальных состояний $P_0(O_{01})$ соответствующие восемь орбит, полученные согласно приведенной в табл. 4 процедуре, представлены в табл. 5. Для обозначения конкретной орбиты в скобках указано начальное состояние трех ячеек, входящих в орбиту. Например, обозначение $O_{01}(011)$ определяет пару орбит с начальным состоянием $P_0(O_{01}) = 0\ 1\ 1$.

Таблица 5

Пары орбит для $k = 3$

Вектор	$O_{01}(000)$	$O_{01}(001)$	$O_{01}(010)$	$O_{01}(011)$	$O_{01}(100)$	$O_{01}(101)$	$O_{01}(110)$	$O_{01}(111)$
P_0	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
P_1	0 0 1	0 0 0	0 1 1	0 1 0	1 0 1	1 0 0	1 1 1	1 1 0
P_2	0 1 1	0 1 0	0 0 1	0 0 0	1 1 1	1 1 0	1 0 1	1 0 0
P_3	1 1 1	1 1 0	1 0 1	1 0 0	0 1 1	0 1 0	0 0 1	0 0 0
P_4	1 0 0	1 0 1	1 1 0	1 1 1	0 0 0	0 0 1	0 1 0	0 1 1
P_5	1 1 0	1 1 1	1 0 0	1 0 1	0 1 0	0 1 1	0 0 0	0 0 1

Следует отметить, что произвольные пары орбит O_{01} для $P_0(O_{01}) \neq 0\ 0\ 0 \dots 0$ формируются путем инвертирования столбцов исходной пары орбит $O_{01}(000\dots 0)$, соответствующих ненулевым значениям $P_0(O_{01})$. Например, пара орбит $O_{01}(011)$ формируется путем инвертирования первого и второго столбцов пары орбит $O_{01}(000)$, приведенной в табл. 5.

Из равенства $P_i(O_0) = \overline{P_{k-i}(O_1)}$, $i = 0, k$, справедливого для орбит O_0 и O_1 , следует, что использование инверсного начального состояния при формировании пары орбит O_{01} приводит к генерированию эквивалентного множества k -разрядных двоичных векторов, т. е. $O_{01}(P_0) = O_{01}(\overline{P_0})$. Например, из табл. 5 видно, что $O_{01}(000) = O_{01}(111)$. Таким образом, в результате случайного выбора начального состояния $P_0(O_{01})$ для пары орбит O_{01} при повторном применении теста типа March C– в двух случаях из 2^k возможных будут сгенерированы множества идентичных k -разрядных двоичных векторов. Множества повторяющихся векторов формируются при выборе того же самого начального состояния $P_0(O_{01})$, что и для исходной пары орбит, или его инверсного значения $\overline{P_0(O_{01})}$.

Повторное применение теста типа March C– может увеличить количество двоичных комбинаций, формируемых в произвольных k ячейках только при условии изменения их начального состояния P_0 . Для общего случая пары орбит O_{01} с начальным состоянием $P_0 = a_\beta a_\gamma \dots a_\delta a_\epsilon a_\eta$ новая пара орбит O_{01}^n будет получена в результате инвертирования r ($k > r \geq 1$) разрядов для всех двоичных k -разрядных векторов исходной пары орбит O_{01} . В качестве примера две новые пары орбит O_{01}^{n1} и O_{01}^{n2} как результат инвертирования определенных разрядов в векторах орбиты O_{01} для $k = 5$ показаны в табл. 6. В данном случае использовалось инвертирование соседних разрядов векторов, поэтому отрицания представляют собой единый непрерывный блок.

Пара орбит O_{01}^{n1} является результатом отрицания всего лишь одного разряда с индексом 2 для всех векторов O_{01} , а пара O_{01}^{n2} получена в результате инвертирования блока разрядов векторов с индексами 2, 3 и 4. В результате обе пары орбит O_{01}^{n1} и O_{01}^{n2} имеют шесть новых векторов и включают в себя по четыре вектора из оригинальной пары орбит O_{01} . Действительно, в O_{01}^{n1} используются векторы P_1, P_2 , входящие в O_0 , и инверсные векторы P_8, P_9 из орбиты O_1 , а в O_{01}^{n2} используются соответственно P_1, P_4 и P_6, P_9 (табл. 6).

Таблица 6

Примеры применения одного блока инверсий при формировании O_{01}^n для $k = 5$

Вектор	O_{01}					O_{01}^{n1}					O_{01}^{n2}				
	5	4	3	2	1	5	4	3	2	1	5	4	3	2	1
P_0	a_β	a_γ	a_δ	a_ϵ	a_η	a_β	a_γ	a_δ	a_ϵ	a_η	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_1	a_β	a_γ	a_δ	a_ϵ	$\overline{a_\eta}$	a_β	a_γ	a_δ	a_ϵ	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_2	a_β	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	a_γ	a_δ	a_ϵ	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	$\overline{a_\eta}$
P_3	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	a_δ	a_ϵ	$\overline{a_\eta}$
P_4	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	$\overline{a_\eta}$	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	$\overline{a_\eta}$
P_5	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η
P_6	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	a_δ	$\overline{a_\epsilon}$	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	a_η
P_7	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	a_η
P_8	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η
P_9	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η

Обобщая результаты приведенного анализа, можно сделать следующий вывод. Для случая одного блока последовательных отрицаний, примененных для O_{01} с целью получения новой пары орбит O_{01}^n (см. табл. 4 и 5), результирующая новая пара орбит включает только четыре двоичных вектора исходной пары орбит O_{01} . Остальные векторы отличаются от векторов, вхо-

дящих в O_{01} . Отметим, что для любого k существует k блоков с одним отрицанием, $k - 1$ блок с двумя последовательными отрицаниями, $k - 2$ блока с тремя последовательными отрицаниями и т. д.

Количество пар орбит, полученных в результате инвертирования последовательного блока бит для начального состояния P_0 исходной пары орбит O_{01} , может быть рассчитано следующим образом:

$$M = k + (k - 1) + (k - 2) + \dots + 2 + 1 = (k^2 + k) / 2. \quad (3)$$

Указанные пары орбит O_{01}^n включают в себя четыре вектора из O_{01} .

Применение других наборов отрицаний для получения новой пары орбит, которые включают в себя более одного блока последовательных отрицаний, приводит к получению абсолютно новой пары орбит за исключением одного случая при использовании двух блоков отрицаний. В табл. 7 показаны примеры новых пар орбит O_{01}^{n3} и O_{01}^{n4} как результат применения двух блоков отрицаний.

Таблица 7

Примеры применения более одного блока инверсий при получении O_{01} для $k = 5$

Вектор	O_{01}					O_{01}^{n3}					O_{01}^{n4}				
	5	4	3	2	1	5	4	3	2	1	5	4	3	2	1
P_0	a_β	a_γ	a_δ	a_ϵ	a_η	a_β	$\overline{a_\gamma}$	a_δ	$\overline{a_\epsilon}$	$\overline{a_\eta}$	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_1	a_β	a_γ	a_δ	a_ϵ	a_η	a_β	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η
P_2	a_β	a_γ	a_δ	a_ϵ	a_η	a_β	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η
P_3	a_β	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η
P_4	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	a_γ	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ϵ	a_η
P_5	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ϵ	a_η	a_β	a_γ	$\overline{a_\delta}$	a_ϵ	a_η
P_6	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	a_δ	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_7	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_8	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ϵ	$\overline{a_\eta}$	a_β	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_9	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ϵ	a_η	a_β	a_γ	$\overline{a_\delta}$	a_ϵ	$\overline{a_\eta}$

Новая пара орбит O_{01}^{n3} содержит k -разрядные векторы, непересекающиеся с векторами множества O_{01} , а пара орбит O_{01}^{n4} содержит повторяющиеся векторы P_2, P_3, P_7, P_8 , которые соответствуют векторам P_7, P_8, P_2, P_3 исходной пары орбит O_{01} . Четыре повторяющихся вектора в новой паре орбит обусловлены симметрией исходных орбит O_0 и O_1 , образующих пару O_{01} . Этот эффект возникает в случае двух блоков отрицания, в которые входят первый и k -й разряды векторов орбит. Число таких пар орбит будет вычисляться аналогично соотношению (3) для значения k , равного $k - 2$.

Общее количество новых пар орбит O_{01}^n , включающих в себя четыре вектора из O_{01} , определяется выражением

$$M_1 = M + (k - 2) + (k - 3) + \dots + 2 + 1 = \frac{(k^2 + k)}{2} + \frac{(k - 2)^2 + k - 2}{2} = k^2 - k. \quad (4)$$

Число M_2 новых пар орбит, состоящее из двоичных векторов, которые не участвуют в исходной паре орбит O_{01} , определяется как

$$M_2 = 2^k - (k^2 - k) - 2. \quad (5)$$

Обобщая приведенный анализ, можно сформулировать следующее утверждение, которое справедливо как для пар орбит O_{01} , так и для O_{23} , генерируемых тестами типа March C–.

Утверждение 4. Для заданного k , $2 < k \leq N$, и постоянного порядка адресов существует $2^k - k^2 + k - 2$ пар орбит O_{01} (O_{23}), формируемых тестами типа March C– с другими начальными состояниями P_0 и состоящих из двоичных векторов, которые не входят в исходную пару орбит O_{01} , (O_{23}), а также $k^2 - k$ пар орбит, включающих по четыре вектора, которые входят в две исходные орбиты O_{01} (O_{23}).

Среднее количество новых двоичных векторов, формируемых во время повторного применения теста типа March C–, будет равняться $[M_1 \times (2k - 4) + M_2 \times (2k)] / 2^k$, а значение среднего числа $M_{ave}(\text{March C–})$ двоичных векторов, формируемых в результате применения двукратного маршевого теста March C–, определяется соотношением

$$M_{ave}(\text{March C–}) = 2k + \frac{M_1 \times (2k - 4) + M_2 \times (2k)}{2^k} = 4k - \frac{k^2}{2^{k-2}}. \quad (6)$$

Очевидно, что значение M_{ave} существенно увеличится при применении тестов, формирующих большее число разнообразных орбит.

Двукратные маршевые тесты типа March A. Первоначально отметим, что тесты типа March A формируют орбиты вида Q_i , приведенные в табл. 2. Маршевые тесты, такие как March A, March B, March LA, March LR, March M, March PS и др., как правило, формируют пары орбит $Q_1, Q_2; Q_0, Q_2$ и Q_0, Q_3 (см. рис. 1). Для идентификации подобных пар орбит в дальнейшем будем применять обозначения Q_{12}, Q_{02} и Q_{03} . Как отмечалось ранее, указанные пары орбит формируют максимально возможное количество неповторяющихся k -разрядных двоичных векторов, равное $2k + 2$. Поэтому даже однократное применение теста, генерирующего, например, пару орбит Q_{03} , обеспечивает всевозможные двоичные комбинации в любых $k < 4$ ячейках запоминающего устройства. Действительно, при $k = 3$ орбиты Q_{03} обеспечивают $2 \times k + 2 = 2 \times 3 + 2 = 8$ всевозможных 2^3 трехразрядных двоичных векторов. Не нарушая общности, детально исследуем пару орбит Q_{03} для произвольных k , $3 < k \leq N$, ячеек памяти с адресами $\beta, \gamma, \dots, \delta, \varepsilon, \eta$, где $N \geq \beta > \gamma > \dots > \delta > \varepsilon > \eta \geq 0$, запоминающего устройства (табл. 8).

Таблица 8

Пара орбит Q_{03} , состоящая из $2k+2$ двоичных векторов

Вектор	Индексы разрядов вектора					
	k	$k-1$...	3	2	1
P_0	a_β	a_γ	...	a_δ	a_ε	a_η
P_1	a_β	a_γ	...	a_δ	a_ε	a_η
P_2	a_β	a_γ	...	a_δ	$\overline{a_\varepsilon}$	a_η
P_3	a_β	a_γ	...	$\overline{a_\delta}$	a_ε	a_η
...
P_{k-1}	a_β	a_γ	...	a_δ	a_ε	a_η
P_k	$\overline{a_\beta}$	a_γ	...	a_δ	a_ε	a_η
P_{k+1}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_{k+2}	$\overline{a_\beta}$	a_γ	...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_{k+3}	$\overline{a_\beta}$	a_γ	...	a_δ	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
...
P_{2k-1}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_{2k}	$\overline{a_\beta}$	a_γ	...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_{2k+1}	$\overline{a_\beta}$	a_γ	...	a_δ	$\overline{a_\varepsilon}$	$\overline{a_\eta}$

Первоначально отметим особенность структуры Q_{03} , которая заключается в том, что вторая часть Q_{03} (Q_3), состоящая из двоичных наборов $P_{k+1}, P_{k+2}, P_{k+3}, \dots, P_{2k}, P_{2k+1}$, равняется инверсии ее первой части (Q_0), включающей $P_0, P_1, P_2, \dots, P_{k-1}, P_k$. Действительно, $P_{k+1} = \overline{P_0}$ и $P_{k+1+j} = \overline{P_{k+1-j}}$, $j = 1, 2, \dots, k$. Вторая особенность Q_{03} обусловлена инверсным соотношением двух частей Q_0 и Q_3 пары орбит Q_{03} . Она состоит в том, что вес $w(\overline{P_i})$ инверсного k -разрядного вектора $\overline{P_i}$ определяется весом $w(P_i)$ произвольного исходного двоичного вектора P_i как $w(\overline{P_i}) = k - w(P_i)$ и, наоборот, как $w(P_i) = k - w(\overline{P_i})$, где вес $w(P_i)$ двоичного вектора P_i определяется количеством его единичных компонент. Из табл. 8 видно, что $w(P_{k+1}) = w(\overline{P_0}) = k - w(P_0)$, а $w(P_{k+1+j}) = w(\overline{P_{k+1-j}}) = k - w(P_{k+1-j})$.

Для нулевого начального состояния $P_0 = 0\ 0\ 0 \dots 0$ двоичные k -разрядные векторы пары орбит Q_{03} имеют веса $w(P_0) = 0$, $w(P_1) = w(P_2) = \dots = w(P_k) = 1$, $w(P_{k+1}) = k$ и $w(P_{k+2}) = w(P_{k+3}) = \dots = w(P_{2k+1}) = k - 1$. Примеры пар орбит Q_{03} для $k = 5$ и различных начальных состояний P_0 приведены в табл. 9.

Таблица 9

Восемь пар орбит для $k = 5$

Вектор	$Q_{03}(00000)$	$Q_{03}(00010)$	$Q_{03}(01010)$	$Q_{03}(11010)$	$Q_{03}(11110)$
P_0	00000	00010	01010	11010	11110
P_1	00001	00011	01011	11011	11111
P_2	00010	00000	01000	11000	11100
P_3	00100	00110	01110	11110	11010
P_4	01000	01010	00010	10010	10110
P_5	10000	10010	11010	01010	01110
P_6	11111	11101	10101	00101	00001
P_7	01111	01101	00101	10101	10001
P_8	10111	10101	11101	01101	01001
P_9	11011	11001	10001	00001	00101
P_{10}	11101	11111	10111	00111	00011
P_{11}	11110	11100	10100	00100	00000

Анализ пар орбит Q_{03} , представленных в табл. 8 и 9, показывает, что независимо от начального состояния P_0 для любого $k < 4$ в произвольных k ячейках запоминающего устройства однократные тесты типа March A, генерирующие Q_{03} , обеспечивают всевозможные k -разрядные двоичные векторы. Для $k \geq 4$ необходимо повторное применение указанных тестов. Первоначально рассмотрим двукратное применение подобных тестов для $k > 4$.

Учитывая то, что новая пара орбит Q_{03} формируется с равной вероятностью для всех возможных 2^k начальных состояний орбит P_0 , докажем следующее утверждение.

Утверждение 5. Для заданного k , $4 < k < N$, и постоянного порядка адресов существует $2^k - k^2 - k - 2$ пар орбит Q_{03} , формируемых тестами типа March A с другими начальными состояниями P_0 и состоящих из двоичных векторов, которые не входят в исходную пару орбит Q_{03} , а также $k^2 + k$ пар орбит Q_{03} , включающих по четыре вектора, которые входят в две исходные орбиты Q_0 и Q_3 .

Доказательство. Для заданного значения $k > 4$ существует 2^k пар орбит Q_{03} , которые представлены в табл. 8 и 9 и определяются начальным состоянием P_0 . Не нарушая общности рассуждений, предположим, что $P_0 = 0\ 0\ 0 \dots 0$ и, соответственно, векторы, входящие в Q_{03} , имеют значения весов $w(P_i) \in \{0, 1, k - 1, k\}$. Необходимо отметить, что Q_{03} включает всевозможные k -разрядные векторы с указанными весами 0, 1, $k - 1$ и k , т. е. по одному вектору с весами 0 и k и по k векторов с весами 1 и $k - 1$.

Новые начальные состояния $P_0 \neq 0\ 0\ 0 \dots 0$ в зависимости от количества единичных значений в векторе P_0 будут определять инверсию соответствующих разрядов пары орбит Q_{03} (см. табл. 9). Рассмотрим случай нового начального состояния P_0 , полученного в результате применения операции инверсии только одного бита первоначального состояния P_0 исходной

пары орбит Q_{03} . В этом случае новая пара орбит будет содержать четыре k -разрядных вектора, ранее использованных в исходной паре орбит Q_{03} , а именно векторы $P_0 = 0\ 0\ 0\ \dots\ 0$ и $P_{k+1} = 1\ 1\ 1\ \dots\ 1$, полученные в результате инвертирования одного бита в соответствующем векторе с весом 1 и его инверсии с весом $k - 1$. Кроме того, результат инвертирования одного бита в векторах $P_0 = 0\ 0\ 0\ \dots\ 0$ и $P_{k+1} = 1\ 1\ 1\ \dots\ 1$ приведет к повторению в новой орбите одного из векторов с весом 1 и весом $k - 1$ исходной пары орбит. Пример получения новой пары орбит $Q_{03}(00010)$ для $k = 5$ как результат отрицания второго бита начального состояния приведен в табл. 9. В этой же таблице дан пример получения $Q_{03}(01010)$ как результат применения двух отрицаний для формирования нового начального состояния P_0 . Отметим, что использование двух отрицаний также приводит к формированию новой орбиты, в которой всегда будут повторяться четыре вектора, используемых в исходной орбите. Только два вектора из полного множества векторов веса 1 в результате применения двух отрицаний для двух произвольных разрядов будут преобразованы в векторы с весом 1, а остальные векторы будут иметь вес 3. Аналогичное утверждение справедливо и для векторов с весом $k - 1$. Только два из них в результате инвертирования двух бит будут иметь вес $k - 1$, а остальные преобразуются в векторы с весом $k - 3$.

Учитывая инверсную структуру ($Q_0 = \overline{Q_3}$) пары орбит Q_{03} , применение $k - 1$ отрицания для получения новой пары орбит эквивалентно применению одного отрицания, а использование $k - 2$ отрицаний повторяет процедуру применения двух отрицаний. Во всех четырех случаях новые начальные состояния, полученные в результате применения 1, 2, $k - 2$ и $k - 1$ отрицаний, приводят к получению новой орбиты, содержащей четыре k -разрядных вектора, ранее использованных в исходной паре орбит Q_{03} . Тогда общее количество новых пар орбит Q_{03}^n , включающих в себя четыре вектора из Q_{03} , для $k > 4$ определяется выражением

$$M_1 = 2 \binom{k}{1} + 2 \binom{k}{2} = 2k + 2 \frac{k(k-1)}{2} = k^2 + k. \quad (7)$$

Для случая, когда количество отрицаний равняется трем либо больше трех, но меньше $k - 2$, все двоичные векторы пары орбит Q_{03} преобразуются в векторы, веса которых будут отличаться от $w(P_i) \in \{0, 1, k - 1, k\}$. Действительно, применение трех отрицаний для $k > 4$ приводит к тому, что вектор с весом 0 преобразуется в вектор с весом $w = 3$, а вектор с весом 1 преобразуется в вектор, для которого $w = 2$ либо $w = 4$. Кроме того, вектор, имеющий вес k , преобразуется в вектор с весом $k - 3$, а векторы, вес которых равен $k - 1$, преобразуются в векторы с весами $k - 2$ либо $k - 4$. Таким образом, для $k \geq 5$ использование трех отрицаний при формировании нового начального состояния приводит к формированию пары орбит Q_{03} , состоящей из двоичных k -разрядных векторов, которые не входят в исходную пару орбит Q_{03} . Аналогичным образом можно показать, что применение большего числа отрицаний, не превышающего $k - 3$ для заданного k , также приводит к формированию пары орбит Q_{03} , двоичные векторы которых не пересекаются с векторами исходной орбиты. Число M_2 новых пар орбит, состоящих из двоичных векторов, которые не участвуют в исходной паре орбит O_{01} , определяется равенством

$$M_2 = 2^k - (k^2 + k) - 2, \quad (8)$$

что и требовалось доказать.

Утверждение 5 справедливо для произвольного значения $k > 4$. В случае $k = 4$ новые начальные состояния P_0 могут быть получены путем применения одного, двух и трех отрицаний. Для одного и трех отрицаний новые орбиты так же, как и для $k > 4$, будут содержать по четыре k -разрядных вектора, а для двух отрицаний – по восемь векторов, которые использовались в исходной паре орбит. Это объясняется тем, что только для $k = 4$ два отрицания соответствуют $k - 2$ отрицаниям.

Среднее количество новых k -разрядных двоичных векторов, формируемых парой орбит Q_{03} во время повторного применения теста типа March A, будет равняться

$[M_1 \times (2k-2) + M_2 \times (2k+2)] / 2^k$ для $k > 4$. При $k = 4$ это количество равняется $[8 \times 4 + 6 \times 8] / 2^4 = 5$. Среднее число $M_{ave}(Q_{03})$ двоичных векторов, формируемых в результате применения двукратного маршевого теста типа March A за счет пары орбит Q_{03} , определяется соотношением

$$M_{ave}(Q_{03}) = 2 \cdot (4+1) + 5 = 15 \quad \text{для } k = 4,$$

$$M_{ave}(Q_{03}) = 2 \cdot (k+1) + \frac{M_1 \cdot (2k-2) + M_2 \cdot (2k+2)}{2^k} = 4(k+1) - \frac{2k^2 + 3k + 2}{2^k} \quad \text{для } k > 4. \quad (9)$$

Отметим, что соотношения (9) позволяют получить лишь нижнюю оценку $M_{ave}(\text{March A})$, так как тесты типа March A формируют не только пары орбит Q_j , но и пары орбит O_i (см. рис. 1).

С целью оценки эффективности двукратных маршевых тестов типа March A были получены экспериментальные результаты, показывающие среднее количество двоичных векторов, формируемых в k произвольных ячейках памяти. Для каждого значения k в табл. 10 приведены усредненные значения, полученные в результате 100 000 экспериментов, заключающихся в случайном формировании начального состояния памяти. Для каждого эксперимента определялось количество k -разрядных векторов, формируемых в k ячейках памяти, адреса которых выбирались случайным образом.

Таблица 10

Экспериментальные значения $M_{ave}(Test)$

k	2	3	4	5	6	7	8	9	10
2^k	4	8	16	32	64	128	256	512	1024
$M_{ave}(\text{MATS}++)$	3,75	6,00	8,43	10,87	13,21	15,51	17,69	19,81	21,91
$M_{ave}(\text{March } C-)$	4	7,55	12,10	17,03	21,93	26,63	31,13	35,48	39,53
$M_{ave}(\text{March A})$	4	8	14,48	21,98	29,64	37,05	44,11	50,86	57,34

В табл. 10 представлены также аналогичные экспериментальные результаты для тестов типа MATS++ и March C–, которые целиком соответствуют теоретическим результатам, полученным согласно соотношениям (2) и (6). Из таблицы видно, что, например, двукратный тест MATS++ для случая $k = 4$ в среднем формирует 8,43 четырехразрядного вектора в произвольных четырех ячейках памяти. В то же время тест типа March C– формирует уже 12,1 вектора, а тесты из множества March A формируют 14,48 двоичных наборов. Очевидное преимущество тестов типа March A достигается за счет их большей сложности в сравнении со сложностью тестов других типов, которая в случае теста March A равняется $15N$.

Эффективность маршевых тестов типа MATS ++, March C– и March A. Эффективность однократного применения маршевых тестов типа MATS ++ ($6N$), March C– ($10N$), March A ($15N$) и March O ($12N$), для которых в скобках показана их сложность, оценим средним количеством $C_1(Test)$ двоичных векторов, формируемых в произвольных k из N ячеек памяти за N циклов обращения к памяти. Соответствующие характеристики как результат деления числа двоичных векторов на коэффициент сложности теста для указанных разновидностей тестов примут вид

$$C_1(\text{MATS } ++) = (k+1)/6, \quad C_1(\text{March } C-) = 2k/10,$$

$$C_1(\text{March A}) = 4(k-1)/15, \quad C_1(\text{March O}) = 4(k-1)/12. \quad (10)$$

Отметим, что все указанные разновидности тестов типа MATS ++, March C– и March A при однократном их применении гарантируют формирование в каждой ячейке памяти ($k = 1$) нулевого и единичного значения. Тесты типа March C– и March A, кроме того, обеспечивают всевозможные комбинации для двух произвольных ячеек памяти ($k = 2$), а тесты типа March A – и для трех ячеек ($k = 3$). Значения численных характеристик эффективности однократных тестов, представленных в формуле (10) для различных значений k , приведены в табл. 11.

Таблица 11

Эффективность $C_1(Test)$ однократного применения тестов

k	2	3	4	5	6	7	8	9	10
$C_1(MATS ++)$	0,50	0,67	0,83	1,00	1,17	1,33	1,50	1,67	1,83
$C_1(March C-)$	–	0,60	0,80	1,00	1,20	1,40	1,60	1,80	2,00
$C_1(March A)$	–	–	0,80	1,07	1,33	1,60	1,87	2,13	2,40
$C_1(March O)$	–	–	1,00	1,33	1,66	2,00	2,33	2,66	3,00

Анализ характеристики $C_1(Test)$ показывает наибольшую эффективность тестов типа March A и наименьшую эффективность тестов MATS ++, особенно с ростом значения величины k . В то же время для $k = 4$ наибольшую эффективность показывают тесты March O и MATS ++, а тесты March C– и March A имеют одинаковую эффективность. Для $k = 5$ эффективность однократных тестов MATS ++ и March C– одинакова, однако уступает эффективности тестов March A и March O.

При оценке эффективности двукратных тестов $C_2(Test)$ используем полученные аналитические выражения (2) и (6) для среднего количества $M_{ave}(Test)$ в случае тестов MATS ++ и March C–, а также экспериментальные оценки, приведенные в табл. 10, для тестов March A.

Таблица 12

Эффективность $C_2(Test)$ двукратного применения тестов

k	2	3	4	5	6	7	8	9	10
$C_2(MATS ++)$	0,31	0,50	0,70	0,91	1,10	1,29	1,47	1,65	1,82
$C_2(March C-)$	–	0,33	0,60	0,84	1,09	1,32	1,55	1,76	1,98
$C_2(March A)$	–	–	0,48	0,73	0,99	1,24	1,47	1,69	1,91
$C_2(March O)$	–	–	0,60	0,92	1,24	1,54	1,84	2,12	2,39

Характеристика $C_2(Test)$, так же как и $C_1(Test)$, является сравнительной оценкой эффективности маршевых тестов, которая показывает среднее количество k -разрядных двоичных векторов в произвольных k ячейках памяти, формируемых конкретным маршевым тестом за N циклов обращения к памяти. Значения данной характеристики показывают несомненную эффективность тестов типа March A, особенно для небольших значений k . Высокая эффективность теста March O объясняется его ориентацией на формирование всевозможных k -разрядных двоичных комбинаций в произвольных k ячейках запоминающего устройства, что является необходимым условием обнаружения сложных неисправностей памяти [4].

Как было показано ранее в работе [4], формирование всевозможных двоичных комбинаций в произвольных k ячейках памяти достигается многократным применением маршевых тестов, среди которых обоснованно выделяются тесты типа March C–. Однако результаты, представленные в предыдущих разделах, показывают большую эффективность тестов типа March A, которые позволяют генерировать большее число орбит и, соответственно, двоичных k -разрядных векторов. Для подтверждения данного утверждения в табл. 13 представлены экспериментальные результаты, показывающие сравнительную эффективность многократных тестов March C– и March A для случая $k = 8$.

Таблица 13

Количество восьмиразрядных векторов (%) по отношению к их максимальному числу 2^8 , %

Количество итераций теста	$M_{ave}(March C-)$			$M_{ave}(March A)$		
	min	max	ave	min	max	ave
5	15,63	31,25	27,58	22,27	44,92	37,55
10	28,13	59,38	47,53	39,45	73,83	60,92
20	46,88	88,28	72,40	57,81	95,70	84,58
30	57,81	99,22	85,36	68,36	100,00	93,79
40	66,41	100,00	92,21	76,17	100,00	97,47
50	72,66	100,00	95,77	82,42	100,00	98,91
60	78,13	100,00	97,65	87,50	100,00	99,50
70	81,25	100,00	98,67	90,63	100,00	99,77
80	83,59	100,00	99,22	92,19	100,00	99,89
90	85,16	100,00	99,54	95,31	100,00	99,94
100	85,16	100,00	99,71	95,31	100,00	99,97

Графическая иллюстрация результатов экспериментов (рис. 2) показывает несомненное преимущество тестов типа March A по сравнению с другими классическими маршевыми тестами для обеспечения всевозможных двоичных комбинаций в произвольных восьми ячейках запоминающего устройства.

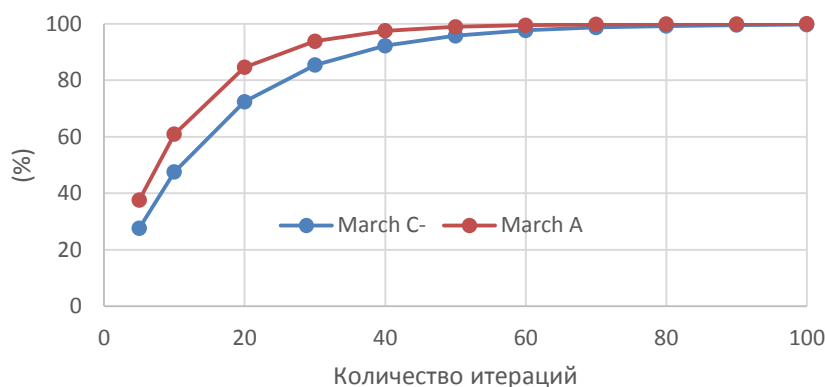


Рис. 2. Сравнительная эффективность тестов March C– и March A

На рис. 2 видно, что тест March A всегда превосходит тест March C–, так как для достижения той же полноты покрытия для теста March A требуется существенно меньше его итераций. Например, десятикратный тест March A обеспечивает в произвольных восьми ячейках памяти в среднем 60,92 % от 256 двоичных векторов, в то же время десятикратный тест March C– — только 47,53 % (см. табл. 13). Для произвольного количества k ячеек памяти высокая эффективность тестов типа March A подтверждается экспериментальными данными, представленными в табл. 14.

Таблица 14

Экспериментальные значения для $M_{ave}(\text{March A})$

k	3	4	5	6	7	8	9	10
2^k	8	16	32	64	128	256	512	1024
$M_{ave}(\text{March A})$	1	2,9	7,3	15,15	30,81	63,81	126,35	248
$M_{min}(\text{March A})$	1	2	4	6	12	28	60	131
$M_{max}(\text{March A})$	1	26	40	62	118	205	340	544
$M_{ave}(\text{March C-})$	2,33	4,42	9,34	19,15	40,55	82,62	169,32	339,48

Для сопоставительного анализа в последней строке табл. 14 представлены полученные ранее в работе [4] экспериментальные оценки $M_{ave}(\text{March C-})$, которые для всех k заметно уступают $M_{ave}(\text{March A})$. Например, для обеспечения всевозможных двоичных комбинаций путем многократного применения теста March A в произвольных 10 ячейках памяти в среднем необходимо применить данный тест 248 раз. Тот же результат может быть достигнут тестом March C–, средняя кратность которого принимает значение, равное 339,48.

Заключение. В работе проведен анализ маршевых тестов, применяемых для тестирования современных запоминающих устройств. Выделены три основных подмножества подобных тестов, а именно тесты типа MATS ++, March C– и March A. Исследована эффективность применения этих разновидностей тестов для реализации псевдоисчерпывающего тестирования запоминающих устройств. Получены аналитические соотношения, позволяющие оценить эффективность маршевых тестов для целей генерирования всевозможных k -разрядных двоичных комбинаций в k произвольных ячейках памяти. Полученные аналитические оценки эффективности реализации псевдоисчерпывающего тестирования памяти подтверждены экспериментальными результатами, показывающими преимущества тестов типа March A. При-

менение подобных тестов для целей реализации псевдоисчерпывающего тестирования требует существенно меньших временных затрат по сравнению с тестами типа MATS ++ и March C–.

Список использованных источников

1. Wang, L.-T. VLSI Test Principles and Architectures: Design for Testability / L.-T. Wang, C.-W. Wu, X. Wen. – Elsevier, 2006. – 808 p.
2. Ярмолик, В. Н. Контроль и диагностика вычислительных систем / В. Н. Ярмолик. – Минск : Бестпринт, 2019. – 387 с.
3. Иванюк, А. А. Проектирование встраиваемых цифровых устройств и систем / А. А. Иванюк. – Минск : Бестпринт, 2012. – 338 с.
4. Ярмолик, В. Н. Псевдоисчерпывающее тестирование запоминающих устройств на базе многократных маршевых тестов / В. Н. Ярмолик, И. Мрозек, В. А. Леванцевич // Информатика. – 2018. – № 1(15). – С. 110–121.
5. Sharma, A. K. Semiconductor Memories: Technology, Testing, and Reliability / A. K. Sharma. – London : John Wiley & Sons, 2002. – 480 p.
6. Niggemeyer, D. Integration of non-classical faults in standard march tests / D. Niggemeyer, M. Redeker, J. Otterstedt // Records of the IEEE Intern. Workshop on Memory Technology, Design and Testing. – San Jose, 1998. – P. 91–98.
7. Неразрушающее тестирование запоминающих устройств / В. Н. Ярмолик [и др.]. – Минск : Бестпринт, 2005. – 230 с.
8. Mrozek, I. Multi-run Memory Tests for Pattern Sensitive Faults / I. Mrozek. – Cham : Springer International Publishing AG, 2019. – 135 p.
9. Goor, A. J. Testing Semiconductor Memories, Theory and Practice / A. J. Goor. – Chichester : John Wiley & Sons, 1991. – 536 p.
10. Yarmolik, S. V. Address sequences and backgrounds with different hamming distance for multiple run march tests / S. V. Yarmolik // IEEE Intern. J. of Applied Mathematics and Computer Science. – 2008. – Vol. 18, no. 3. – P. 329–339.
11. Sokol, B. Address sequence for march tests to detect pattern sensitive faults / B. Sokol, S. V. Yarmolik // Proc. of 3rd IEEE Intern. Workshop on Electronic Design Test and Applications (DELTA'06). – Kuala Lumpur, Malaysia, 2006. – P. 354–357.
12. Sokol, B. Impact of the address changing on the detection of pattern sensitive faults / B. Sokol, I. Mrozek, V. N. Yarmolik // Information Processing and Security Systems. – London : Springer Science + Business Media, Inc., 2005. – P. 217–226.
13. Ярмолик, В. Н. Адресные последовательности для многократного тестирования ОЗУ / В. Н. Ярмолик, С. В. Ярмолик // Информатика. – 2014. – № 3(39). – С. 92–103.
14. Mrozek, I. Antirandom test vectors for BIST in Hardware / Software systems / I. Mrozek, V. N. Yarmolik // Fundamenta Informaticae. – 2012. – No. 119. – P. 1–23.
15. Mrozek, I. Iterative antirandom testing / I. Mrozek, V. N. Yarmolik // J. of Electronic Testing: Theory and Applications (JETTA). – 2012. – Vol. 9, no. 3. – P. 251–266.

References

1. Wang L.-T., Wu C.-W., Wen X. *VLSI Test Principles and Architectures: Design for Testability*. Elsevier, 2006, 808 p.
2. Yarmolik V. N. Kontrol' i diagnostika vuchislitel'nyh sistem. *Monitoring and Diagnostics of Computer Systems*. Minsk, Bestprint, 2019, 387 p. (in Russian).
3. Ivaniuk A. A. Proektirovanie vstraivaemuh cifrovuh ustroystv i sistem. *Designing Embedded Digital Devices and Systems*. Minsk, Bestprint, 2012, 338 p. (in Russian).
4. Yarmolik V. N., Mrozek I., Levancevich V. A. Pseudoisчерpuyayuschee testirovanie zapominayuschih ustroystv na baze mnogokratnych marshevyh testov [Pseudo-exhaustive storage testing based on multiple march tests]. *Informatika [Informatics]*, 2018, no. 1(15), pp. 110–121 (in Russian).
5. Sharma A. K. *Semiconductor Memories: Technology, Testing, and Reliability*. London, John Wiley & Sons, 2002, 480 p.
6. Niggemeyer D., Redeker M., Otterstedt J. Integration of non-classical faults in standard march tests. *Records of the IEEE International Workshop on Memory Technology, Design and Testing*, San Jose, 1998, pp. 91–98.

7. Yarmolik V. N., Murashko I. A., Kummert A., Ivaniuk A. A. Nerazrushayushee testirovanie zapominayuschih ustroystv. *Non-Destructive Storage Testing*. Minsk, Bestprint, 2005, 230 p. (in Russian).
8. Mrozek I. *Multi-run Memory Tests for Pattern Sensitive Faults*. Cham, Springer International Publishing AG, 2019, 135 p.
9. Goor A. J. *Testing Semiconductor Memories, Theory and Practice*. Chichester, John Wiley & Sons, 1991, 536 p.
10. Yarmolik S. V. Address sequences and backgrounds with different hamming distance for multiple run march tests. *IEEE International Journal of Applied Mathematics and Computer Science*, 2008, vol. 18, no. 3, pp. 329–339.
11. Sokol B., Yarmolik S. V. Address sequence for march tests to detect pattern sensitive faults. *Proceedings of 3rd IEEE International Workshop on Electronic Design Test and Applications (DELTA'06)*. Kuala Lumpur, Malaysia, 2006, pp. 354–357.
12. Sokol B., Mrozek I., Yarmolik V. N. Impact of the address changing on the detection of pattern sensitive faults. *Information Processing and Security Systems*. London, Springer Science + Business Media, Inc., 2005, pp. 217–226.
13. Yarmolik V. N., Yarmolik S. V. Adresnye posledovatel'nosti dlya mnogokratnogo testirovaniya OZU [Address sequences for repeated testing of RAM]. *Informatika [Informatics]*, 2014, no. 3(39), pp. 92–103 (in Russian).
14. Mrozek I., Yarmolik V. N. Antirandom test vectors for BIST in Hardware / Software systems. *Fundamenta Informaticae*, 2012, no. 119, pp. 1–23.
15. Mrozek I., Yarmolik V. N. Iterative antirandom testing. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 2012, vol. 9, no. 3, pp. 251–266.

Информация об авторах

Ярмолик Вячеслав Николаевич, доктор технических наук, профессор, Белорусский государственный университет радиоэлектроники и информатики, Минск, Беларусь.
E-mail: yarmolik10ru@yahoo.com

Мрозек Иренеуш, доктор, адъюнкт, Белостокский технический университет, Белосток, Польша.
E-mail: i.mrozek@pb.edu.pl

Ярмолик Светлана Вячеславовна, кандидат технических наук, Белорусский государственный университет радиоэлектроники и информатики, Минск, Беларусь.
E-mail: syarmolik@gmail.com

Information about the authors

Vyacheslav N. Yarmolik, Dr. Sci. (Eng.), Professor, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus.
E-mail: yarmolik10ru@yahoo.com

Ireneusz Mrozek, Dr., Lecture, Bialystok University of Technology, Bialystok, Poland.
E-mail: i.mrozek@pb.edu.pl

Svetlana V. Yarmolik, Cand. Sci. (Eng.), Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus.
E-mail: syarmolik@gmail.com