ИНФОРМАТИКА ▪ INFORMATICS
ТОМ ▪ VOL. 18 3|2021 С. ▪ P. 18–35

УДК 004.33.054
https://doi.org/10.37661/1816-0301-2021-18-3-18-35

Оригинальная статья
Original Paper

Неразрушающие тесты с четным повторением адресов для запоминающих устройств

В. Н. Ярмолик[1]✉, И. Мрозек[2], В. А. Леванцевич[1], Д. В. Деменковец[1]

[1]*Белорусский государственный университет информатики и радиоэлектроники, ул. П. Бровки, 6, 220013, Минск, Беларусь*
✉*E-mail: yarmolik10ru@yahoo.com*

[2]*Белостокский технический университет, ул. Вейска, 45А, 15-351, Белосток, Польша*

Аннотация. Показывается актуальность задачи тестирования запоминающих устройств современных вычислительных систем. Исследуются математические модели неисправностей этих устройств и используемые методы тестирования наиболее сложных из них на базе классических неразрушающих маршевых тестов. Вводится понятие адресных последовательностей (*pA*) с четным повторением адресов, которые являются основой базового элемента, входящего в структуру новых неразрушающих маршевых тестов *March_pA*_1 и *March_pA*_2. Приводятся алгоритмы формирования подобных последовательностей и примеры их реализации. Показывается максимальная диагностическая способность новых тестов для случая простейших неисправностей, таких как константные (*SAF*) и переходные (*TF*), а также сложных кодочувствительных неисправностей (*PNPSFk*). Отмечается существенно меньшая временная сложность тестов *March_pA*_1 и *March_pA*_2 по сравнению с классическими неразрушающими тестами, которая достигается за счет меньших временных затрат на получение эталонной сигнатуры. Вводятся новые метрики расстояния для количественного сравнения эффективности применяемых *pA* при однократной реализации тестов *March_pA*_1 и *March_pA*_2. В основе новых метрик лежит расстояние $D(A(j), pA)$, определяемое разностью индексов повторяющихся адресов $A(j)$ в последовательности *pA*. Исследуются свойства новых характеристик последовательностей *pA* и оценивается их применимость для выбора оптимальных тестовых последовательностей *pA*, обеспечивающих высокую эффективность новых неразрушающих тестов. Приводятся примеры вычисления метрик расстояний и показывается зависимость эффективности новых тестов от численных значений метрик расстояния. Как и в случае классических неразрушающих тестов, рассматривается многократное применение тестов *March_pA*_1 и *March_pA*_2. Вводится характеристика $V(pA)$, которая численно равняется количеству отличающихся значений расстояния $D(A(j), pA)$ адресов $A(j)$ последовательности *pA*. Экспериментально показывается справедливость аналитических оценок и подтверждается высокая эффективность обнаружения неисправностей однократными и многократными тестами типа *March_pA*_1 и *March_pA*_2 на примере неисправностей взаимного влияния для $p = 2$.

Ключевые слова: тестирование вычислительных систем, запоминающие устройства, неразрушающие маршевые тесты, адресные последовательности с четным повторением адресов, многократное неразрушающее тестирование

Для цитирования. Неразрушающие тесты с четным повторением адресов для запоминающих устройств / В. Н. Ярмолик [и др.] // Информатика. – 2021. − Т. 18, № 3. – С. 18–35. https://doi.org/10.37661/1816-0301-2021-18-3-18-35

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 07.04.2021
Подписана в печать | Accepted 14.05.2021
Опубликована | Published 29.09.2021

Transparent memory tests with even repeating addresses for storage devices

Vyacheslav N. Yarmolik[1], Ireneusz Mrozek[2], Vladimir A. Levantsevich[1], Denis V. Demenkovets[1]

[1]Belarusian State University of Informatics and Radioelectronics,
st. P. Brovki, 6, 220013, Minsk, Belarus
✉E-mail: yarmolik10ru@yahoo.com

[2]Bialystok University of Technology,
st. Wiejska, 45A, 15-351, Białystok, Poland

Abstract. The urgency of the problem of memory testing of modern computing systems is shown. Mathematical models describing the faulty states of storage devices and the methods used for their detection are investigated. The concept of address sequences (*pA*) with an even repetition of addresses is introduced, which are the basis of the basic element included in the structure of the new transparent march tests *March_pA*_1 and *March_pA*_2. Algorithms for the formation of such sequences and examples of their implementations are given. The maximum diagnostic ability of new tests is shown for the case of the simplest faults, such as constant (*SAF*) and transition faults (*TF*), as well as for complex pattern sensitive faults (*PNPSFk*). There is a significantly lower time complexity of the *March_pA*_1 and *March_pA*_2 tests compared to classical transparent tests, which is achieved at the expense of less time spent on obtaining a reference signature. New distance metrics are introduced to quantitatively compare the effectiveness of the applied *pA* address sequences in a single implementation of the *March_pA*_1 and *March_pA*_2 tests. The basis of new metrics is the distance $D(A(j), pA)$ determined by the difference between the indices of repeated addresses $A(j)$ in the sequence *pA*. The properties of new characteristics of the *pA* sequences are investigated and their applicability is evaluated for choosing the optimal test *pA* sequences that ensure the high efficiency of new transparent tests. Examples of calculating distance metrics are given and the dependence of the effectiveness of new tests on the numerical values of the distance metrics is shown. As well as in the case of classical transparent tests, multiple applications of new *March_pA*_1 and *March_pA*_2 tests are considered. The characteristic $V(pA)$ is introduced, which is numerically equal to the number of different values of the distance $D(A(j), pA)$ of addresses $A(j)$ of the sequence *pA*. The validity of analytical estimates is experimentally shown and high efficiency of fault detection by the tests *March_pA*_1 and *March_pA*_2 is confirmed by the example of coupling faults for $p = 2$.

Keywords: testing of computer systems, memory, transparent march tests, address sequences with even repeating addresses, multiple transparent testing

For citation. Yarmolik V. N., Mrozek I., Levantsevich V. A., Demenkovets D. V. Transparent memory tests with even repeating addresses for storage devices. *Informatics*, 2021, vol. 18, no. 3, pp. 18–35 (In Russ.). https://doi.org/10.37661/1816-0301-2021-18-3-18-35

Conflict of interest. The authors declare of no conflict of interest.

Введение. Быстро прогрессирующие технологии производства изделий микроэлектроники позволяют создавать схемы запоминающих устройств (ЗУ) с высокой степенью интеграции, которые в современных вычислительных системах занимают до 94 % площади кристалла [1, 2]. Следствием использования новых технологий является увеличение вероятности возникновения различных неисправностей в ЗУ, которые могут составлять до 70 % от общего числа отказов системы в целом [3, 4]. Помимо традиционных методологий тестирования ЗУ в настоящее время широко применяется самотестирование памяти (*memory built-in self-test*) [5–7]. Такой подход позволяет периодически выполнять тестирование на рабочей частоте без применения внешнего оборудования. Из-за больших объемов современных ЗУ возможно применение только тестов сложности $O(N)$, где N – емкость в битах ЗУ, которые называются маршевыми тестами [3, 5, 6].

При реализации маршевого теста со всеми N ячейками ЗУ в заданном порядке ($\Uparrow$ – от младших адресов к старшим, $\Downarrow$ – в обратной последовательности, $\Uparrow\Downarrow$ – в любом направлении) выполняются операции, определяемые несколькими элементами, которые представляют собой маршевый тест. Для этого используются: $w0$, $w1$ − запись в запоминающую ячейку памяти значения 0 или 1; $r0$, $r1$ − чтение текущего значения из элемента памяти и сравнение его со значением 0 или 1. Достоинствами маршевых тестов являются относительно высокие скорость выполнения и их обнаруживающая способность, а также простота реализации как средства тестирования. Наиболее значимый недостаток классических маршевых тестов заключается в потере хранящейся в памяти информации после выполнения теста.

Для надежной работы ЗУ современных вычислительных систем помимо методов, использующих избыточные данные (коды Хэмминга, контроль на четность и др.), применяются методы неразрушающего тестирования (*transparent testing*), основанные на классических маршевых тестах [8–10].

При анализе эффективности маршевых тестов рассматриваются математические модели, описывающие проявление физических дефектов ЗУ в процессе их функционирования [3, 4, 9, 10]. Наиболее распространенные модели неисправностей ЗУ приведены ниже [4].

Константные неисправности (*stuck-at fault, SAF*) характеризуются тем, что при их возникновении логическое значение ячейки памяти всегда равно 0 (*SA*0) или 1 (*SA*1) независимо от операции, производимой с этой или другими ячейками ЗУ.

Переходные неисправности (*transition fault, TF*) отличаются тем, что ячейка неспособна осуществлять переход из состояния логического 0 в состояние логической 1 ($TF\uparrow$) либо наоборот ($TF\downarrow$).

При возникновении неисправности взаимного влияния (*coupling fault, CF*) изменение логического значения одной влияющей ячейки (агрессора) отражается на значении второй зависимой ячейки (жертвы). Ячейка с меньшим адресом может влиять на ячейку со старшим адресом ($\wedge$), и наоборот ($\vee$). Существует три типа неисправностей *CF*. Во-первых, инверсная неисправность *CF* (*inversion CF, CFin*), для которой изменение значения влияющей ячейки вызывает инвертирование значения зависимой. Возможны следующие виды неисправностей *CFin*: $\wedge(\uparrow,\bar{b})$, $\wedge(\downarrow,\bar{b})$, $\vee(\uparrow,\bar{b})$, $\vee(\downarrow,\bar{b})$, где $b\in\{0, 1\}$ – текущее состояние ячейки жертвы. Во-вторых, неисправность *CF* прямого действия (*idempotent CF, CFid*). В случае *CFid* изменение значения влияющей ячейки переводит зависимую ячейку в определенное состояние. Возможны восемь видов неисправностей *CFid*: $\wedge(\uparrow,0)$, $\wedge(\uparrow,1)$, $\wedge(\downarrow,0)$, $\wedge(\downarrow,1)$, $\vee(\uparrow,0)$, $\vee(\uparrow,1)$, $\vee(\downarrow,0)$, $\vee(\downarrow,1)$. В-третьих, константная неисправность *CFst* (*state CF, CFst*), состоящая в том, что переход зависимой ячейки в какое-либо состояние возможен только при определенном значении влияющей ячейки. Различают следующие неисправности *CFst*: $\wedge(0,0)$, $\wedge(0,1)$, $\wedge(1,0)$, $\wedge(1,1)$, $\vee(0,0)$, $\vee(0,1)$, $\vee(1,0)$, $\vee(1,1)$.

Кодочувствительные неисправности (*pattern sensitive faults, PSF*) описывают поведение нескольких ячеек памяти вплоть до N ячеек, где N – емкость памяти в битах [4, 5]. Для подобных неисправностей логическое состояние одной ячейки памяти, называемой базовой (*base cell*), может зависеть от содержимого (0 или 1) или от логических переходов из 1 в 0 или из 0 в 1 в соседних ячейках (*neighborhood cells*) ЗУ. Среди множества разновидностей кодочувствительных неисправностей выделяют ограниченные (*restricted*) и граничные (*neighborhood, NPSF*) неисправности [4, 5]. При тестировании современных ЗУ обычно придерживаются модели кодочувствительных неисправностей, для которой рассматривается небольшое число $k \leq 9$ ячеек памяти, входящих в неисправность *NPSFk*, а их местоположение может быть произвольным [4, 5, 9]. В качестве объекта исследования чаще всего рассматриваются пассивные кодочувствительные неисправности (*PNPSFk*), где k обозначает количество произвольных ячеек памяти емкостью N бит, участвующих в конкретной неисправности. Для *PNPSFk* содержимое базовой ячейки не может быть изменено из-за определенного набора данных в соседних $k-1$ ячейках [4, 5]. Отметим, что результаты, полученные для *PNPSFk*, легко обобщаются и для других классов кодочувствительных неисправностей в силу того, что *PNPSFk* является моделью наиболее трудно обнаруживаемых неисправностей памяти, покрывающей другие виды неисправностей [4, 5].

В статье предлагается подход к построению неразрушающих тестов, основанный на применении адресных последовательностей с четным повторением адресов, позволяющий уменьшить временную сложность тестов и увеличить их диагностическую способность при сохранении обнаруживающей способности. Показывается зависимость эффективности новых неразрушающих тестов от применяемых адресных последовательностей и предоставляется оптимальный выбор для их многократной реализации.

1. Неразрушающие маршевые тесты с четным повторением адресов. Основным требованием, предъявляемым к неразрушающим тестам, является необходимость восстановления исходного состояния объекта после выполнения процедуры его тестирования. Поэтому все тестовые воздействия, направленные на активизацию неисправностей и проявление их в виде ошибок на выходе ЗУ, должны носить обратимый характер [10].

В настоящей работе при построении нового класса эффективных неразрушающих тестов ЗУ предлагается использовать модифицированные адресные последовательности. Первоначально рассмотрим общие свойства адресных последовательностей и их модификацию для реализации новых неразрушающих тестов.

Определение 1. Адресной последовательностью называется последовательность $A(j) = a_{m-1}a_{m-2}\ldots a_2a_1a_0$, $j\in\{0, 1, 2, \ldots, 2^m - 1\}$, где $a_i\in\{0, 1\}$ для $i\in\{0, 1, 2, \ldots, m - 1\}$, состоящая из всех возможных 2^m m-разрядных двоичных комбинаций $a_{m-1}a_{m-2}\ldots a_2a_1a_0$, генерируемых в произвольном порядке, причем каждый адрес формируется только один раз [5].

Для любой последовательности бит a_i, $i\in\{0, 1, 2, \ldots, m - 1\}$, адресной последовательности A существует 2^{m-1} различных двоичных комбинаций $a_{m-1}a_{m-2}\ldots a_{i+1}a_{i-1}\ldots a_2a_1a_0$ для $a_i = 0$ и такое же количество комбинаций $a_{m-1}a_{m-2}\ldots a_{i+1}a_{i-1}\ldots a_2a_1a_0$ для $a_i = 1$. Для общего случая это свойство можно сформулировать следующим образом: для любого числа $1 \le r < m$ последовательностей бит a_α, a_β, …, a_χ адресов $A = a_{m-1}\,a_{m-2}\ldots a_2a_1a_0$ для каждой из 2^r двоичных комбинаций 00…0, 00…1, … и 11…1 значений $a_\alpha\, a_\beta\ldots\, a_\chi$ существует ровно 2^{m-r} различных двоичных комбинаций в остальных $m - r$ последовательностях бит. Основываясь на данном свойстве, можно сформулировать формальный метод построения генераторов адресных последовательностей, в которых каждый адрес имеет кратность повторения адресов, равную $q = 2^r$. Приведем указанный метод в виде утверждения.

Утверждение. *Удаление произвольных $1 \le r < m$ бит a_α, a_β,…, a_χ из исходной адресной последовательности $A = a_{m-1}a_{m-2}\ldots a_2a_1a_0$, удовлетворяющей определению 1, приводит к тому, что оставшиеся биты последовательности будут формировать последовательность адресов, каждый из которых повторяется ровно $q = 2^r$ раз.*

Подобные последовательности далее будем обозначать qA. При этом отметим, что алгоритм формирования qA такой же, как и для A. Отличием является использование в qA не всех бит исходной адресной последовательности A. В случае формирования каждого адреса дважды ($q = 2$) утверждение для получения последовательности $2A$ формулируется следующим образом: произвольная совокупность любых $m - 1$ разрядов $a_{m-1}a_{m-2}\ldots a_{i+1}a_{i-1}\ldots a_2a_1a_0$ из m разрядов $a_{m-1}a_{m-2}\ldots a_2a_1a_0$ исходной адресной последовательности A формирует адресную последовательность $2A$, в которой каждый $m - 1$ разрядный адрес генерируется дважды.

Обобщая предыдущие рассуждения, сформулируем определение адресной последовательности pA для произвольного четного p.

Определение 2. Адресной последовательностью pA, состоящей из $p2^m$ адресов, называется упорядоченная последовательность адресов $A(j) = a_{m-1}\,a_{m-2}\ldots a_2a_1a_0$, $j\in\{0, 1, 2, \ldots, 2^m - 1\}$, где $a_i\in\{0, 1\}$ для $i\in\{0, 1, 2, \ldots, m - 1\}$, состоящая из всех возможных 2^m m-разрядных двоичных комбинаций $a_{m-1}a_{m-2}\ldots a_2a_1a_0$, каждая из которых формируется ровно p раз.

В качестве примера рассмотрим счетчиковую (пересчетную) адресную последовательность $A_C = c_{m-1}c_{m-2}\ldots c_2c_1c_0$, где $c_i \in\{0, 1\}$, $i\in\{0, 1, 2, \ldots, m - 1\}$, формируемую в соответствии с алгоритмом двоичного m-разрядного суммирующего счетчика [5]. Для исходной счетчиковой последовательности $A_C = c_3c_2c_1c_0$ в табл. 1 даны примеры последовательностей $2A_C$ и $4A_C$, полученных согласно утверждению.

Таблица 1. Адресная последовательность A_C и ее модификации $2A_C$ и $4A_C$

Table 1. Address sequences A_C and its $2A_C$ and $4A_C$ modifications

$A_C = c_3c_2c_1c_0$	$2A_C = c_3c_2c_1$	$2A_C = c_3c_2c_0$	$2A_C = c_3c_1c_0$	$2A_C = c_1c_0c_3$	$4A_C = c_3c_2$	$4A_C = c_3c_1$	$4A_C = c_3c_0$	$4A_C = c_0c_3$
0000 (0)	000 (0)	000 (0)	000 (0)	000 (0)	00 (0)	00 (0)	00 (0)	00 (0)
0001 (1)	000 (0)	001 (1)	001 (1)	010 (2)	00 (0)	00 (0)	01 (1)	10 (2)
0010 (2)	001 (1)	000 (0)	010 (2)	100 (4)	00 (0)	01 (1)	00 (0)	00 (0)
0011 (3)	001 (1)	001 (1)	011 (3)	110 (6)	00 (0)	01 (1)	01 (1)	10 (2)
0100 (4)	010 (2)	010 (2)	000 (0)	000 (0)	01 (1)	00 (0)	00 (0)	00 (0)
0101 (5)	010 (2)	011 (3)	001 (1)	010 (2)	01 (1)	00 (0)	01 (1)	10 (2)
0110 (6)	011 (3)	010 (2)	010 (2)	100 (4)	01 (1)	01 (1)	00 (0)	00 (0)
0111 (7)	011 (3)	011 (3)	011 (3)	110 (6)	01 (1)	01 (1)	01 (1)	10 (2)
1000 (8)	100 (4)	100 (4)	100 (4)	001 (1)	10 (2)	10 (2)	10 (2)	01 (1)
1001 (9)	100 (4)	101 (5)	101 (5)	011 (3)	10 (2)	10 (2)	11 (3)	11 (3)
1010 (10)	101 (5)	100 (4)	110 (6)	101 (5)	10 (2)	11 (3)	10 (2)	01 (1)
1011 (11)	101 (5)	101 (5)	111 (7)	111 (7)	10 (2)	11 (3)	11 (3)	11 (3)
1100 (12)	110 (6)	110 (6)	100 (4)	001 (1)	11 (3)	10 (2)	10 (2)	01 (1)
1101 (13)	110 (6)	111 (7)	101 (5)	011 (3)	11 (3)	10 (2)	11 (3)	11 (3)
1110 (14)	111 (7)	110 (6)	110 (6)	101 (5)	11 (3)	11 (3)	10 (2)	01 (1)
1111 (15)	111 (7)	111 (7)	111 (7)	111 (7)	11 (3)	11 (3)	11 (3)	11 (3)

Для каждой адресной последовательности в табл. 1 приведены двоичные значения адресов и в скобках – их десятичные эквиваленты. В дальнейшем модифицированную последовательность qA в соответствии с утверждением будем обозначать как $qA_{M,ai,aj,\ldots,aq}$, где индекс M указывает на метод формирования исходной адресной последовательности A_M, а остальные значения a_i, a_j, …, a_q – на биты, удаленные из A_M. Например, $2A_C = c_3c_2c_1$ будем обозначать как $2A_{Cc0}$. Из табл. 1 видно, что двойные $2A_C$ и учетверенные $4A_C$ адресные последовательности зависят от выбранных разрядов исходной адресной последовательности A_C: $2A_{Cc0} = c_3c_2c_1$, $2A_{Cc1} = c_3c_2c_0$, $2A_{Cc2} = c_3c_1c_0$, $4A_{Cc1c0} = c_3c_2$, $4A_{Cc2c0} = c_3c_1$, $4A_{Cc2c1} = c_3c_0$ и от перестановок разрядов в qA_C: $2A_C = c_1c_0c_3$, $4A_C = c_0c_3$. Соответственно, общее количество последовательностей qA ($q = 2^r$, $1 \le r < m$), полученных из исходной последовательности A, определяется выражением

$$\binom{m}{m-\log_2 q}\cdot(m-\log_2 q)! = \binom{m}{m-r}\cdot(m-r)!,\ 1\le r<m. \tag{1}$$

В случае двойной адресной последовательности $2A_C$ ($r = 1$) количество последовательностей, полученных только по одному из методов и только для одной исходной адресной последовательности A_C, согласно (1) равняется $m!$. Для $m = 4$ количество подобных последовательностей равняется 24, а для реальных значений m их число чрезвычайно велико. Следует подчеркнуть, что в качестве исходной последовательности A может быть использована любая адресная последовательность. Хорошо апробированы и применимы на практике такие последовательности, как пересчетные (счетчиковые), последовательности Грея, анти-Грея, последовательности с максимальной переключательной активностью, последовательности с заданным расстоянием Хэмминга, ЛП$_\tau$-последовательности, М-последовательности и ряд других [11−15]. Очевидно, что число подобных последовательностей, а также разнообразие алгоритмов для их формирования и свойств получаемых последовательностей велико.

Основная идея неразрушающих маршевых тестов заключается в применении адресных последовательностей pA, где p, в том числе и $p = q = 2^r$, – четное целое число, согласно которому при четном инвертировании содержимого ячейки ЗУ его значение останется прежним. В соответствии с этим простейшим свойством операции инвертирования построим базовый элемент неразрушающего маршевого теста с использованием четной адресной последовательности pA. Как и в классических неразрушающих тестах, маршевый элемент должен начинаться с операции чтения rb содержимого $b\in\{0, 1\}$ текущей ячейки ЗУ. Это необходимо для однозначных последующих действий с текущей ячейкой ЗУ, которые основываются на прочитанном значении ее содержимого. Следующей должна быть операция записи инверсного значения $\bar{b}$ по отношению к только что полученному значению b, так как подобная операция является необходимым условием активизации подавляющего большинства неисправностей ЗУ [4, 5]. За операцией записи следует операция чтения этой же текущей ячейки ЗУ для проверки правильности выполнения операции инвертирования ее содержимого. Использование четных адресных последовательностей pA обеспечивает повторное инвертирование каждой ячейки ЗУ, в итоге сохраняя его исходное состояние. Далее переходим к следующему запоминающему элементу ЗУ в соответствии с адресной последовательностью pA. Возрастающую четную (p) адресную последовательность pA обозначим как $p\Uparrow$, а убывающую – как $p\Downarrow$. Таким образом, базовый элемент будет иметь вид

$$p\Uparrow(rb,\, w\bar{b}\,,\, rb). \tag{2}$$

Применение в базовом элементе четной адресной последовательности ($p\Uparrow$) приведет к тому, что каждая ячейка ЗУ последовательно выполнит переход из b в $\bar{b}$ и, наоборот, из $\bar{b}$ в b, сохранив таким образом свое начальное значение. Количество таких пар переходов равняется $p/2$. Правильность выполнения обоих переходов (↑ и ↓), а также операций чтения нулевых и единичных значений обеспечивает вторая операция чтения rb базового элемента (2). Для иллюстрации реализации базового элемента (2) рассмотрим процедуру тестирования ЗУ, содержащего $N = 8$ запоминающих элементов с исходным содержимым 0 1 0 1 0 1 1 0. В качестве адресной последовательности используем последовательности двойных адресов отраженного кода Грея $2A_{Gg3} = g_2g_1g_0$ [16] и последовательности Соболя $2A_{Ss0} = s_3s_2s_1$ [17, 18] для исходного значения $m = 4$. Пошаговое изменение содержимого ЗУ при реализации базового элемента (2) для обоих случаев двойной адресации представлено в табл. 2.

Таблица 2. Процедура реализации базового элемента (2) неразрушающего теста для двух видов адресации

Table 2. The procedure for the implementation of the basic element (2) of transparent test for two types of addressing

Адреса ЗУ *RAM addresses*		7	6	5	4	3	2	1	0	Адреса ЗУ *RAM addresses*		7	6	5	4	3	2	1	0
Содержимое *Contents*		0	1	1	0	1	0	1	0	Содержимое *Contents*		0	1	1	0	1	0	1	0
$2A_{Gg3}$	000 (0)	0	1	1	0	1	0	1	↑	$2A_{Ss0}$	100 (4)	0	1	1	↑	1	0	0	0
	001 (1)	0	1	1	0	1	0	↓	1		010 (2)	0	1	1	1	1	↑	0	0
	011 (3)	0	1	1	0	↓	0	0	1		110 (6)	0	↓	1	1	1	1	0	0
	010 (2)	0	1	1	0	0	↑	0	1		011 (3)	0	0	1	1	↓	1	0	0
	110 (6)	0	↓	1	0	0	1	0	1		111 (7)	↑	0	1	1	0	1	0	0
	111 (7)	↑	0	1	0	0	1	0	1		001 (1)	1	0	1	1	0	1	↑	0
	101 (5)	1	0	↓	0	0	1	0	1		101 (5)	1	0	↓	1	0	1	1	0
	100 (4)	1	0	0	↑	0	1	0	1		000 (0)	1	0	0	1	0	1	1	↑
	100 (4)	1	0	0	↓	0	1	0	1		100 (4)	1	0	0	↓	0	1	1	1
	101 (5)	1	0	↑	0	0	1	0	1		010 (2)	1	0	0	0	0	↓	1	1
	111 (7)	↓	0	1	0	0	1	0	1		110 (6)	1	↑	0	0	0	0	1	1
	110 (6)	0	↑	1	0	0	1	0	1		011 (3)	1	1	0	0	↑	0	1	1
	010 (2)	0	1	1	0	0	↓	0	1		111 (7)	↓	1	0	0	1	0	1	1
	011 (3)	0	1	1	0	↑	0	0	1		001 (1)	0	1	0	0	1	0	↓	1
	001 (1)	0	1	1	0	1	0	↑	1		101 (5)	0	1	↑	0	1	0	0	1
	000 (0)	0	1	1	0	1	0	1	↓		000 (0)	0	1	1	0	1	0	0	↓
Содержимое *Contents*		0	1	1	0	1	0	1	0	Содержимое *Contents*		0	1	1	0	1	0	1	0

На каждом шаге реализации базового элемента только одна запоминающая ячейка, выделенная символом ↑ или ↓, меняет свое состояние на противоположное. Из табл. 2 видно, что после выполнения базового элемента (2) начальное состояние ЗУ осталось неизменным.

Базовый элемент на основе четных адресных последовательностей позволяет синтезировать два неразрушающих маршевых теста:

$$\begin{gathered} March\_\,pA\_1\text{: } \{\Uparrow(rb);\, p\Uparrow(rb,\, w\bar{b}\,,\, rb);\, \Uparrow(rb)\},\ (2N{+}3pN); \\ March\_\,pA\_2\text{: } \{\Downarrow(rb);\, p\Uparrow(rb,\, w\bar{b}\,,\, rb);\, p\Downarrow(rb,\, w\bar{b}\,,\, rb);\, \Downarrow(rb)\},\ (2N{+}6pN). \end{gathered} \tag{3}$$

В обоих тестах (3) произвольный порядок адресов для первой и последней фаз, включающих только операции чтения, должен быть одинаков: возрастающий ⇑, как показано для *March_pA*_1, либо убывающий ⇓, как показано для *March_pA*_2. Это связано с тем, что первая фаза тестов *March_pA*_1 и *March_pA*_2 используется для сжатия исходного состояния ЗУ и получения эталонной сигнатуры S_F, а их последняя фаза – для получения реального значения сигнатуры S_R после выполнения предыдущего либо предыдущих базовых элементов. В случае проявления неисправностей в ходе выполнения базовых элементов их наличие будет определяться выполнением неравенства $S_F \neq S_R$. Выражения в круглых скобках для каждого из представленных тестов (3) показывают их временную сложность в зависимости от кратности p повторения адресов.

Базовый элемент (2) обеспечивает активизацию и обнаружение всех простейших неисправностей типа *SAF* и *TF*. Операция записи $w\bar{b}$ и четная адресация pA реализуют как чтение нуля и единицы из текущей ячейки, так и выполнение двух переходов ее состояния, что является условием активизации данных неисправностей. Их обнаружение обеспечивает вторая операция чтения rb, результат которой сравнивается со значением, полученным при выполнении первой операции чтения rb базового элемента (2). При отсутствии указанных неисправностей эти значения должны быть противоположными, а их равенство свидетельствует о наличии неисправности и ее местоположении. Таким образом, для простейших неисправностей типа *SAF* и *TF* предложенный тест *March_pA*_1 в отличие от известных неразрушающих тестов [8, 9] имеет максимальную диагностическую способность.

Аналогично максимально возможная для маршевых тестов диагностическая способность тестов *March_pA*_1 и *March_pA*_2 достигается и для случая сложных кодочувствительных неисправностей *PNPSFk*. Выполнение базового элемента в случае *PNPSFk* позволяет идентифицировать адрес базовой ячейки, которая не может выполнить один из переходов в этой ячейке для конкретного содержимого в соседних ячейках. В данном случае обнаруживаются только два их вида: $\langle u,u,u,\ldots,u,\uparrow,u,u,u,\ldots,u\rangle$ и $\langle d,d,d,\ldots,d,\downarrow,d,d,d,\ldots,d\rangle$, где u, $d \in \{0, 1\}$. Значения содержимого u и d соседних ячеек произвольны и зависят как от их начального состояния, так и от вида последовательности адресов pA. Например, для случая ЗУ с восемью ячейками и неисправностей *PNPSF*3 в ячейках с адресами 2, 4 и 5 тест *March_pA*_1 (3), применив адресную последовательность $2A_{Ss0}$, позволяет обнаруживать неисправности $\langle 1,1,\uparrow\rangle$, $\langle 0,0,\downarrow\rangle$, $\langle 1,\uparrow,0\rangle$, $\langle 0,\downarrow,1\rangle$, $\langle\uparrow,0,1\rangle$ и $\langle\downarrow,1,0\rangle$ (см. табл. 2). Таким образом, однократное применение теста *March_2A*_1 позволяет достичь полноты покрытия для неисправностей *PNPSF*3, равной 25 %.

Для обнаружения неисправностей взаимного влияния необходимо выполнить анализ состояния ячейки жертвы после активизации неисправности, что невозможно в рамках базового элемента (2). Невыполнение одного из переходов в ячейке жертвы как результат наличия неисправности *CF* приведет к нечетному количеству изменений ее состояния на противоположное. Соответственно, конечное содержимое ЗУ перед выполнением последней фазы чтения в обоих тестах *March_2A*_1 и *March_2A*_2 ЗУ будет отличаться от его исходного состояния, что приведет к выполнению неравенства $S_F \neq S_R$. Количественно полнота покрытия тестом *March_2A*_1 таких неисправностей независимо от адресной последовательности всегда постоянна, как это видно, например, из экспериментальных данных для *CFid* (табл. 3).

В табл. 3 показано, что независимо от адресной последовательности $2A_C$ полнота покрытия *FC* (*faults coverage*) неисправностей *CFid* тестом *March_2A*_1 всегда равняется 50 %. Приведенные данные получены для ЗУ емкостью N = 256 бит с нулевым начальным состоянием. Адресные последовательности $2A_{c0}$, …, $2A_{c7}$, $2A_{c8}$ сформированы из счетчиковой последовательности $A_C = c_8c_7c_6c_5c_4c_3c_2c_1c_0$ путем удаления соответствующего бита, например $2A_{c8} = c_7c_6c_5c_4c_3c_2c_1c_0$.

Неразрушающий маршевый тест *March_pA*_2 отличается от *March_pA*_1 наличием второго базового элемента с обратным порядком (p⇓) адресов pA, что расширяет его возможности обнаруживать сложные неисправности, сохраняя эффективность *March_pA*_1 для более простых неисправностей. Действительно, для неисправностей *CFid* тест *March_pA*_2 обнаруживает заметно большее их количество, вплоть до 100%-го покрытия (табл. 3).

Очевидно, что на полноту покрытия *FC* неисправностей *CFid* тестом *March_pA_2* в значительной степени влияет выбранная адресная последовательность *pA*. Для примера, приведенного в табл. 3, 100%-я полнота покрытия неисправностей *CFid* тестом *March_2A_2* достигается только для одной адресной последовательности $2A_{c8}$. Тест *March_2A_2* с данной адресной последовательностью для произвольной ячейки-агрессора ЗУ, состоящего из 256 ячеек, в остальных ячейках формирует условия обнаружения всех возможных неисправностей *CF*.

Таблица 3. Полнота покрытия неисправностей *CFid* тестами *March_2A_1* и *March_2A_2*, %

Table 3. Faults coverage of CFid faults by March_2A_1 and March_2A_2 tests, %

CFid	*March_2A_1*							*March_2A_2*						
	$2A_{c0}$	$2A_{c1}$	$2A_{c3}$	$2A_{c4}$	$2A_{c6}$	$2A_{c7}$	$2A_{c8}$	$2A_{c0}$	$2A_{c1}$	$2A_{c3}$	$2A_{c4}$	$2A_{c6}$	$2A_{c7}$	$2A_{c8}$
$\wedge(\uparrow;0)$	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,39	2,75	5,88	24,7	49,8	100
$\wedge(\uparrow;1)$	100	100	100	100	100	100	100	100	100	100	100	100	100	100
$\wedge(\downarrow;0)$	0,00	0,39	2,75	5,88	24,7	49,8	100	0,00	0,39	2,75	5,88	24,7	49,8	100
$\wedge(\downarrow;1)$	100	99,6	97,3	94,1	75,3	50,2	0,00	100	100	100	100	100	100	100
$\vee(\uparrow;0)$	0,00	0,39	2,75	5,88	24,7	49,8	100	0,00	0,39	2,75	5,88	24,7	49,8	100
$\vee(\uparrow;1)$	100	99,6	97,3	94,1	75,3	50,2	0,00	100	100	100	100	100	100	100
$\vee(\downarrow;0)$	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,39	2,75	5,88	24,7	49,8	100
$\vee(\downarrow;1)$	100	100	100	100	100	100	100	100	100	100	100	100	100	100
FC	50,0	50,0	50,0	50,0	50,0	50,0	50,0	50,0	50,2	51,4	52,9	62,4	75,0	100

В общем случае для нулевого начального состояния ЗУ ($b_i = 0$, где $i \in \{0, 1, \ldots, N-1\}$) условие обнаружения состоит в формировании тестом *March_2A_2* четырех состояний в остальных ячейках ЗУ, каждая из которых потенциально может быть жертвой:

$$\begin{array}{l} 0, 0, 0, \ldots, 0, \uparrow, 1, 1, 1, \ldots, 1; \\ 1, 1, 1, \ldots, 1, \downarrow, 0, 0, 0, \ldots, 0; \\ 1, 1, 1, \ldots, 1, \uparrow, 0, 0, 0, \ldots, 0; \\ 0, 0, 0, \ldots, 0, \downarrow, 1, 1, 1, \ldots, 1. \end{array} \tag{4}$$

Как видно из соотношений (4), для любой ячейки ЗУ, выступающей в роли агрессора, выполняются оба перехода $\uparrow$ и $\downarrow$ для двух возможных состояний в остальных $N-1$ ячейках. Соотношения (4) являются необходимым и достаточным условием обнаружения несвязных неисправностей взаимного влияния (*CF*), сформулированным при разработке теста *March C−* [4]. Анализ условий (4) показывает, что первые два соотношения реализуются первым базовым элементом теста *March_2A_2*, а два вторых – вторым базовым элементом, для которого используется обратный порядок адресной последовательности 2*A* (3).

Ранее отмечалось, что тест *March_2A_1* либо первый базовый элемент теста *March_2A_2* обнаруживают две неисправности *PNPSFk*: $\langle u,u,u,\ldots,u,\uparrow,u,u,u,\ldots,u\rangle$ и $\langle d,d,d,\ldots,d,\downarrow,d,d,d,\ldots,d\rangle$. Соответственно, второй базовый элемент будет обнаруживать дополнительно еще две неисправности $\langle u,u,u,\ldots,u,\downarrow,u,u,u,\ldots,u\rangle$ и $\langle d,d,d,\ldots,d,\uparrow,d,d,d,\ldots,d\rangle$ только в случае выполнения неравенства $u, u, u, \ldots, u \neq d, d, d, \ldots, d$ для произвольного *k*. Для нулевого начального состояния ЗУ последнее неравенство выполняется для любого *k* в случае реализации первым базовым элементом теста *March_2A_2* двух первых соотношений (4).

Приведенный выше анализ позволяет сформулировать обобщенное требование к тесту *March_2A_2* и, соответственно, к адресной последовательности 2*A*, используемой при его реализации для ЗУ емкостью *N* бит с произвольным начальным состоянием.

Для достижения максимальной способности к обнаружению неисправностей *CF* и *PNPSFk* с помощью теста *March_2A_2* для любой произвольной ячейки ЗУ, действующей как агрессор (в случае *CFid*) или как базовая ячейка (в случае *PNPSFk*), должно выполняться следующее условие для всех остальных $N-1$ ячеек памяти:

$$\begin{gathered} b, b, b, \ldots, b, \uparrow, \bar{b}, \bar{b}, \bar{b}, \ldots, \bar{b}; \\ \bar{b}, \bar{b}, \bar{b}, \ldots, \bar{b}, \downarrow, b, b, b, \ldots, b. \end{gathered} \qquad (5)$$

Символы $\uparrow$ и $\downarrow$ обозначают произвольную ячейку, в то время как случайные состояния $b \in \{0,1\}$ остальных ячеек упорядоченно изменяются на противоположные значения в соответствии с базовым элементом (2).

2. Количественные характеристики адресных последовательностей *pA*. Задача оценки максимальной покрывающей способности при тестировании ЗУ, зависящая от используемых адресных последовательностей, рассматривалась в рамках многократных маршевых тестов ЗУ [5, 11, 19, 20]. Исследовалась эффективность манхэттенского расстояния (*Manhattan distance*) и его взвешенных модификаций, таких как *Canberra distance* и *Bray Curtis distance*, для оценки адресных последовательностей, отвечающих определению 1 [5, 11]. Указанные характеристики использовались как мера различия между адресными последовательностями при формировании максимально отличающихся последовательностей адресов для последующих итераций теста ЗУ [5, 21, 22]. Было показано, что эффективность однократных маршевых тестов не зависит от вида адресной последовательности *A*, отвечающей определению 1 [4, 5, 23]. В то же время применение последовательностей с повторяющимися адресами *pA* для однократной реализации тестов (3) в значительной степени влияет на их эффективность (см. разд. 1).

Ранее уже отмечалось, что для четного *p* последовательность адресов *pA* состоит из всех возможных 2^m адресов $A(0)$, $A(1)$, $A(2)$, …, $A(2^m–2)$, $A(2^m–1)$, где $A(j) \in \{0, 1, 2, \ldots, 2^m-1\}$, $j \in \{0, 1, 2, \ldots, 2^m-1\}$, представляющих собой двоичные комбинации $a_{m-1}a_{m-2}\ldots a_2a_1a_0$, каждая из которых генерируется *p* раз. Это означает, что в произвольной последовательности *pA*, длина которой равняется $p2^m$, *p* адресов $A(j)$ генерируются на определенном расстоянии друг от друга, определяемом алгоритмом формирования *pA*. Например, для адресной последовательности $2A_{Cc1}$ адреса $A(2) = 010$ генерируются на расстоянии 2 друг от друга, а те же два адреса $A(2)$ в последовательности $2A_{Cc2}$ генерируются на расстоянии 4 (см. табл. 1). Очевидно, что каждый адрес $A(j)$ произвольной последовательности *pA* имеет фиксированные расстояния между его повторными формированиями. Например,

Индекс адреса	0	1	…	α	…	β	…	$p2^m-1$
Адрес				$A(j)$		$A(j)$		

Индексы адресов $\alpha < \beta < \ldots < p2^m-1$ в последовательности длиной $p2^m$ указывают местоположение всех повторяющихся *p* раз адресов $A(j)$. Предположим, что впервые адрес $A(j)$ формируется с индексом α в произвольной последовательности *pA*. Тогда расстояние между первым и вторым появлениями адреса $A(j)$ будет определяться разностью значений индексов α и β их позиций в последовательности *pA*, т. е. $D(A(j), pA) = \beta - \alpha = r$. Учитывая то, что период последовательности адресов *pA* равен $p2^m$, а расстояние между первым и вторым адресами $A(j)$ равно *r*, расстояние между вторым и первым адресами $A(j)$ будет равно $p2^m - r$. Расстояние между вторым и третьим формированиями адреса $A(j)$, как и для любой пары адресов $A(j)$, в последовательности *pA* определяется аналогичным образом.

В общем случае произвольная пара из возможных пар повторения адреса $A(j)$ в последовательности *pA* имеет два значения метрики расстояния $D(A(j), pA)$, а именно *r* и $p2^m - r$, которые не зависят от начального адреса с индексом 0 последовательности *pA*. Это следует из свойств циклических последовательностей и объясняется следующим образом. Учитывая то, что период последовательности адресов *pA* равен $p2^m$, а расстояние между первым и вторым адресами пары $A(j)$ равно *r*, расстояние между вторым и первым адресами $A(j)$ пары будет равно $p2^m - r$.

В предыдущем примере начальным адресом для последовательностей $2A_{Cc1}$ и $2A_{Cc2}$ был нулевой адрес $A(0) = 000$. Изменение начального адреса приводит к тому, что расстояние $D(A(2), 2A_{Cc1})$ равняется $r = 2$ либо $2^{m+1} - r = 2^4 - 2 = 14$. Расстояние $D(A(2), 2A_{Cc2})$ также принимает одно из двух значений: 4 или 12. Общее количество значений расстояния $D(A(j), pA)$ для четного p равняется $p(p - 1)$.

Для оценки последовательности pA в дальнейшем для каждого адреса $A(j)$ будем использовать минимальное значение расстояния $MD(A(j), pA)$ из всех возможных расстояний $D(A(j), pA)$. В качестве характеристики последовательности pA применим метрику среднего значения $AD(pA)$ минимальных расстояний $MD(A(j), pA)$ между повторяющимися адресами $A(j)$ последовательности pA, которая инвариантна относительно начального адреса. Эта метрика рассчитывается по формуле

$$AD(pA) = \frac{1}{2^m} \sum_{j=0}^{2^m - 1} MD(A(j),\ pA). \tag{6}$$

Основываясь на том, что наиболее практичным случаем адресных последовательностей pA с повторением адресов, отвечающих определению 2, является тот, когда $p = 2$, введем для таких последовательностей характеристику, которая вычисляется следующим образом:

$$AD(2A) = \frac{1}{2^m} \sum_{j=0}^{2^m - 1} \min\left[D(A(j),\ 2A),\ (2^{m+1} - D(A(j),\ 2A)) \right]. \tag{7}$$

Тогда для примера $2A_{Cc1}$ среднее расстояние между одинаковыми адресами, рассчитанное согласно (7), будет равно двум и это значение будет повторяться для любых значений начальных адресов последовательности $2A_{Cc1}$. Более того, зачастую алгоритм формирования последовательностей pA позволяет аналитически вычислить как значение $MD(A(j), pA)$ для любого адреса $A(j)$, так и величину $AD(pA)$. Например, для адресной последовательности $2A_{Cc0}$ независимо от значения m величина $MD(A(j), 2A_{Cc0})$ для любого адреса $A(j)$ всегда равняется единице, как это видно из табл. 1 для $m = 3$. Аналогично можно заметить, что для произвольного m $MD(A(j), 2A_{Cc1}) = 2$, $MD(A(j), 2A_{Cc2}) = 4$ и $MD(A(j), 2A_{Ccm}) = 2^m$. Следовательно, при формировании адресных последовательностей $2A_{Cci}$, $i \in \{0, 1, 2, \ldots, m\}$, согласно утверждению из счетчиковой последовательности адресов A_C, каждый из которых состоит из $m+1$ разрядов, среднее значение $AD(2A_{Cci})$ минимальных расстояний $MD(A(j), 2A_{Cci})$ будет вычисляться в соответствии с выражением

$$AD(2A_{Cci}) = \frac{1}{2^m} \sum_{j=0}^{2^m - 1} MD(A(j),\ 2A_{Cci}) = 2^i,\ i \in \{0, 1, 2, \ldots,\ m\}. \tag{8}$$

Следует отметить, что соотношение (8) справедливо для конкретного случая исходной адресной последовательности A_C и однозначно определенного алгоритма (см. утверждение) получения из нее последовательностей $2A_C$ для $p = 2$.

Приведем основные свойства метрик $D(A(j), pA)$, $MD(A(j), pA)$ и $AD(pA)$:

1. Минимальное значение $MD(A(j), pA)$ метрики $D(A(j), pA)$ не зависит от начального адреса последовательности pA, так как для любого начального адреса и любой пары повторяющихся адресов $A(j)$ оно принимает одно из двух значений: r или $p2^m - r$. Соответственно, значение метрики $AD(pA)$ также не зависит от начального адреса последовательности pA.

2. Для любой последовательности pA с периодом, равным $p2^m$, которая удовлетворяет определению 2, характеристики $D(A(j), pA)$, $MD(A(j), pA)$ и $AD(pA)$ принимают значения

$$1 \le D(A(j),\ pA) \le p2^m - (p - 1);\ \ 1 \le MD(A(j),\ pA) \le 2^m;\ \ 1 \le AD(pA) \le 2^m. \tag{9}$$

3. Максимальное значение метрики $AD(pA) = 2^m$ достигается только для случая, когда $MD(A(j), pA) = 2^m$ для всех $j \in \{0, 1, 2, \ldots, 2^m - 1\}$, что следует из соотношения (7). Это свойство в первую очередь присуще последовательностям, сформированным согласно утверждению.

Максимальное значение минимума $MD(A(j), 2A)$ между двумя адресами $A(j)$ ($p = 2$) достигается для случая, когда выполняется равенство $r = 2^{m+1} - r$, из которого следует, что $MD(A(j), 2A) = 2^m$. Для произвольного четного p также $MD(A(j), pA) = 2^m$. При этом все p копии адреса $A(j)$ также расположены равноудаленно друг от друга, т. е. на расстоянии $(p2^m)/p = 2^m$ (см. рис. 1).

4. Для удовлетворяющих определению 2 последовательностей pA, адреса которых модифицированы путем инвертирования произвольного числа их бит $a_{m-1}a_{m-2}\ldots a_2a_1a_0$, среднее значение $AD(pA)$ минимальных расстояний $MD(A(j), pA)$ между повторяющимися адресами $A(j)$ остается неизменным.

Свойство (4) следует из того, что применение операции отрицания для разрядов адресов $A(j)$ последовательности pA приводит к перераспределению их индексов и, соответственно, характеристик $MD(A(j), pA)$. Это означает, что в выражениях (6) и (7) для вычисления $AD(pA)$ используются те же слагаемые, только в другой последовательности. Например, для последовательности адресов $2A_{Gg3} = g_2g_1g_0$, приведенной в табл. 2, характеристика $AD(2A_{Gg3})$ (7) будет вычисляться как (15 + 13 + 9 + 11 + 1 + 3 + 7 + 5)/8 = 8. Между тем эта же характеристика для последовательности адресов $2A_{G'g3} = \overline{g}_2\overline{g}_1\overline{g}_0$, полученной в результате применения операции отрицания для разрядов $g_2g_1g_0$ последовательности $2A_{Gg3}$, вычисляется как (5 + 7 + 3 + + 1 + 11 + + 9 + 13 + 15)/8 = 8.

Рассмотренные метрики $D(A(j), pA)$ и $AD(pA)$ для последовательностей pA позволяют сформулировать условие, аналогичное условию (5) максимальной эффективности теста *March_pA_2* в терминах этих характеристик для $p = 2$.

Для общего случая адресных последовательностей pA условие максимальной эффективности теста *March_pA_2*, аналогичное условию (5) для $2A$, заключается в обеспечении наибольшего временного интервала между последовательными инвертированиями содержимого ячейки ЗУ с произвольным адресом $A(j)$. Это требование состоит в выборе адресных последовательностей с максимальным значением $AD(pA)$. Приведенное условие хорошо иллюстрируется на примере обнаружения тестом *March_2A_2* неисправностей взаимного влияния *CFid*. Объединяя данные, приведенные в табл. 3, и численные значения метрики $AD(2A_{Cci})$ (8), можно заметить однозначную зависимость полноты покрытия неисправностей от величины данной интегральной метрики.

Таблица 4. Полнота покрытия *FC*(*CFid*) неисправностей *CFid* тестом *March_2A_2*, %

Table 4. Faults coverage of FC (CFid) of CFid failures by March_2A_2 test, %

$2A_{Cci}$	$2A_{c0}$	$2A_{c1}$	$2A_{c2}$	$2A_{c3}$	$2A_{c4}$	$2A_{c5}$	$2A_{c6}$	$2A_{c7}$	$2A_{c8}$
FC(*CFid*)	50,00	50,20	50,59	51,37	52,94	56,08	62,35	74,99	100,00
$AD(2A_{Cci})$	1	2	4	8	16	32	64	128	256

Действительно, для $p = 2$ только в том случае, если для любой ячейки памяти с адресом $A(j)$ расстояние $D(A(j), 2A)$ будет равно $2^m = 2^8 = 256$, в ячейках памяти сформируются обратные состояния, указанные в выражении (5), и будет достигнута максимальная полнота покрытия неисправностей *CFid*.

3. Многократные неразрушающие тесты *March_pA_1* и *March_pA_2*. Идея неразрушающих тестов ЗУ предполагает многократное их применение, которое за счет изменяющихся начальных условий, чаще всего состояний ячеек ЗУ, позволяет обнаруживать любые неисправные состояния ЗУ [8−10]. В случае новых неразрушающих тестов *March_pA_1* и *March_pA_2* изменяющимися являются адресные последовательности pA, которые на каждой очередной итерации теста должны обеспечивать максимальное количество вновь обнаруживаемых неисправностей ЗУ по отношению к предыдущим итерациям. Аналогичная задача формирования максимально различных тестовых наборов и их последовательностей решалась в рамках управляемых вероятностных тестов (*control random testing*) [24−29].

Сущность управляемых вероятностных тестов заключается в том, что очередная тестовая последовательность формируется в терминах предварительно определенных и обоснованных характеристик. В качестве этих характеристик применяются меры расстояний, причем чаще всего используются расстояния Хэмминга и Евклида [24−29]. В случае управляемых вероятностных тестов принимается гипотеза о том, что для двух тестовых последовательностей, имеющих минимальное различие, количество обнаруживаемых неисправностей будет минимальным и, наоборот, для максимально различных последовательностей обнаруживающая способность будет максимальной [24−29]. При этом и различие, и подобие тестовых последовательностей оценивались с использованием ряда интегральных метрик расстояния [24−27].

В работе [29] была показана эффективность применения расстояния Евклида для выбора следующих адресных последовательностей A_i, используемых для многократного тестирования ЗУ с помощью метода формирования адресных последовательностей, который основывается на инвертировании определенных бит исходной адресной последовательности A_k. Алгоритм формирования многократных тестов состоит в применении масок в виде двоичного вектора $\Lambda = \lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0 \neq 0\ 0\ \ldots\ 0\ 0$, единичные значения которого определяют наличие инверсий разрядов адресов $a_{m-1}a_{m-2}\ldots a_2a_1a_0$ исходной базовой последовательности A_k по отношению к формируемой последовательности A_l [5].

Предположив, что исходная адресная последовательность A_k представляется двоичным кодом $a_{m-1}a_{m-2}\ldots a_2a_1a_0$, выражение для адресов $A(j)_l \neq A(j)_k$ последовательности A_l будет иметь вид [5, 29]

$$A_l = a_{m-1}^{\lambda_{m-1}} a_{m-2}^{\lambda_{m-2}} \ldots a_1^{\lambda_1} a_0^{\lambda_0} = (a_{m-1} \oplus \lambda_{m-1})(a_{m-2} \oplus \lambda_{m-2})\ldots(a_1 \oplus \lambda_1)(a_0 \oplus \lambda_0). \quad (10)$$

Отметим, что метод формирования адресных последовательностей (10) справедлив и для случая последовательностей с повторяющимися адресами pA. Например, для $m = 3$ и исходной последовательности $2A_k = 2A_{Cc1} = c_3c_2c_0$ = {000, 001, 000, 001, 010, 011, 010, 011, 100, 101, 100, 101, 110, 111, 110, 111}, применив вектор масок $\Lambda = \lambda_2\lambda_1\lambda_0 \neq 1\ 1\ 1$, на основании соотношения (10) получим новую последовательность адресов $2A_l$ = {111, 110, 111, 110, 101, 100, 101, 100, 011, 010, 011, 010, 001, 000, 001, 000}.

Для адресных последовательностей pA_k и pA_l, соответствующих определению 2, справедлива следующая теорема, которая аналогична теореме относительно A_k и A_l, доказанной в работе [29].

Теорема. *Для последовательностей адресов* pA_k *и* pA_l*, где* pA_k *включает* $p2^m$ *m-разрядных, сгенерированных произвольным образом двоичных адресов* $a_{m-1}a_{m-2}\ldots a_2a_1a_0$*, каждый из которых повторяется p раз, а адреса последовательности* pA_l *получены согласно равенству (10) на основании вектора отрицаний* $\lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0$*, для которого g значений* $\lambda_\alpha, \lambda_\beta, \ldots, \lambda_\chi, \lambda_\delta$ $(\alpha > \beta > \ldots > \chi > \delta)$ *равняются единице, расстояние Евклида* $ED(pA_k, pA_l)$ *вычисляется следующим образом:*

$$ED(pA_k, pA_l) = \sqrt{p2^m(2^{2\alpha} + 2^{2\beta} + \ldots + 2^{2\chi} + 2^{2\delta})}\,. \quad (11)$$

Действительно, для двух последовательностей адресов $2A_k = 2A_{Cc1} = c_3c_2c_0$ и $2A_l$, рассмотренных в предыдущем примере, расстояние Евклида соответствует значению, полученному из равенства (11): $ED(2A_k, 2A_l) = \sqrt{2\times 2^3 \times (2^{2\times 2} + 2^{2\times 1} + 2^{2\times 0})} = \sqrt{336}$.

В табл. 5 приведены численные значения расстояний $ED(2A_k, 2A_l)$, которые так же, как и в случае последовательностей адресов A_i [29], возрастают с ростом численного значения кода применяемой маски $\Lambda = \lambda_2\lambda_1\lambda_0$. Аналогичные значения расстояния Евклида и их различимость позволяют применять данную метрику для построения многократных неразрушающих тестов *March_pA_*1 и *March_pA_*2.

Таблица 5. Значения расстояния Евклида для m = 3
Table 5. Values of the Euclidean distance for m = 3

$2A_k = a_2a_1a_0$	$2A_l$						
	$a_2a_1\overline{a_0}$	$a_2\overline{a_1}a_0$	$a_2\overline{a_1}\,\overline{a_0}$	$\overline{a_2}a_1a_0$	$\overline{a_2}a_1\overline{a_0}$	$\overline{a_2}\,\overline{a_1}a_0$	$\overline{a_2}\,\overline{a_1}\,\overline{a_0}$
$ED(2A_k,2A_l)$	$\sqrt{16}$	$\sqrt{64}$	$\sqrt{80}$	$\sqrt{256}$	$\sqrt{272}$	$\sqrt{320}$	$\sqrt{336}$

Используем методологию построения многократных тестов, основанную на применении масок отрицаний при формировании адресных последовательностей [5, 29], для последующих итераций тестов *March_pA*_1 и *March_pA*_2.

Для общего случая многократных неразрушающих тестов *March_pA*_1 и *March_pA*_2 применим оптимальное сочетание набора двоичных векторов масок $\Lambda = \lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0$ с целью получения адресных последовательностей pA_0, pA_1, pA_2, …, pA_{w-1} тестов *March_pA*_1 и *March_pA*_2. В случае кратности тестов w, равной восьми, оптимальное сочетание масок для произвольного значения $m \geq 3$ приведено в табл. 6 [29], где значение индекса определяет оптимальную маску для последующей итерации многократного теста. Нулевое значение индекса означает однократное применение теста, для которого маска как таковая отсутствует, что эквивалентно ее нулевому значению. Для повторного применения теста маска, соответствующая наибольшей эффективности неразрушающего теста, принимает единичное значение для всех разрядов. Последующие итерации многократного теста основаны на применении приведенных в табл. 6 масок, оптимальность которых доказана в работе [29].

Таблица 6. Оптимальное сочетание масок $\Lambda = \lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0$ для $m \geq 3$
Table 6. Optimal combination of masks $\Lambda = \lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0$ *for* $m \geq 3$

Индекс *Index*	Λ							
	λ_{m-1}	λ_{m-2}	λ_{m-3}	λ_{m-4}	λ_{m-5}	…	λ_1	λ_0
0	0	0	0	0	0		0	0
1	1	1	1	1	1	…	1	1
2	1	0	0	0	0	…	0	0
3	0	1	1	1	1	…	1	1
4	1	1	0	0	0	…	0	0
5	0	0	1	1	1	…	1	1
6	1	0	1	1	1	…	1	1
7	0	1	0	0	0	…	0	0

Для оценки эффективности многократного применения новых неразрушающих тестов *March_pA*_1 и *March_pA*_2 были получены экспериментальные результаты для адресных последовательностей $2A_{c0}$, $2A_{c8}$ и $2A_{c4}$. Указанные адресные последовательности ранее использовались для получения полноты покрытия однократного применения данных тестов (см. табл. 3). В табл. 7 представлены результаты только двух итераций новых неразрушающих тестов, так как последующие их итерации с использованием оптимальных значений масок (см. табл. 6) не приводили к увеличению полноты покрытия неисправностей *CFid*.

Таблица 7. Полнота покрытия неисправностей *CFid* двукратными тестами *March_2A*_1 и *March_2A*_2, %
Table 7. Faults coverage of CFid faults by double tests March_2A_1 and March_2A_2 in percentage, %

CFid	*March*_2A_{c0}_1		*March*_2A_{c0}_2		*March*_2A_{c8}_1		*March*_2A_{c8}_2		*March*_2A_{c4}_1		*March*_2A_{c4}_2	
	0	1	0	1	0	1	0	1	0	1	0	1
$\wedge(\uparrow;0)$	0,0	0,0	0,0	0,0	0,0	100	100	0,0	0,0	5,88	5,88	0,0
$\wedge(\uparrow;1)$	100	0,0	100	0,0	100	0,0	100	0,0	100	0,0	100	0,0
$\wedge(\downarrow;0)$	0,0	0,0	0,0	0,0	100	0,0	100	0,0	5,88	0,0	5,88	0,0
$\wedge(\downarrow;1)$	100	0,0	100	0,0	0,0	100	100	0,0	94,12	5,88	100	0,0
$\vee(\uparrow;0)$	0,0	0,0	0,0	0,0	100	0,0	100	0,0	5,88	0,0	5,88	0,0
$\vee(\uparrow;1)$	100	0,0	100	0,0	0,0	100	100	0,0	94,12	5,88	100	0,0
$\vee(\downarrow;0)$	0,0	0,0	0,0	0,0	0,0	100	100	0,0	0,0	5,88	5,88	0,0
$\vee(\downarrow;1)$	100	0,0	100	0,0	100	0,0	100	0,0	100	0,0	100	0,0
FC	50	0,0	50	0,0	50	50	100	0,0	50	2,94	52,94	0,0

Полученные результаты позволяют сделать следующие выводы:

1. Многократное тестирование на основании модификаций исходной адресной последовательности (10) с применением оптимальных сочетаний масок, приведенных в табл. 6, целесообразно только на базе теста *March_pA_2*. Данный факт объясняется тем, что второй базовый элемент теста *March_pA_2* использует обратный порядок адресов. Это эквивалентно применению единичной маски ($\Lambda = 1\ 1\ 1\ldots\ 1$) при повторном использовании теста *March_pA_1*. Последующие маски, соответствующие индексам 2 и 3, также являются инверсными по отношению друг к другу. В то же время сложность реализации двукратного теста *March_2A_1* равняется $16N$, что заметно больше сложности $14N$ однократного теста *March_2A_2* (3).

2. При многократном использовании теста *March_pA_2* максимизация характеристики $AD(pA)$ не приводит к увеличению покрывающей способности при повторных применениях теста с оптимальным сочетанием масок. Как видно из табл. 7, на всех последующих итерациях теста независимо от адресной последовательности $2A_{c0}$, $2A_{c8}$ и $2A_{c4}$ и, соответственно, значений ее характеристик $AD(2A)$ полнота покрытия не увеличивается. Однако, как было показано ранее для случая $p = 2$, максимальные значения характеристики $AD(2A)$ однозначно определяют максимальную покрывающую способность однократного теста *March_pA_2* для неисправностей *CFid* и других сложных неисправностей ЗУ.

Отметим, что для адресной последовательности $2A_{c0}$, для которой значение $AD(2A_{c0}) = 1$ минимально, последовательности $2A_{c8}$, имеющей максимальное значение $AD(2A_{c8}) = 256$, а также последовательности адресов $2A_{c4}$ с характеристикой $AD(2A_{c4}) = 16$ общим является равенство расстояния $MD(A(j), 2A)$ для всех значений j. Действительно, для любого $j \in \{0, 1, 2, \ldots, 2^m - 1\}$ $MD(A(j), 2A_{c0}) = 1$, $MD(A(j), 2A_{c8}) = 256$ и $MD(A(j), 2A_{c4}) = 16$, как это ранее отмечалось при выводе соотношения (8).

Очевидно, что обнаружение сложных неисправностей ЗУ требует новых разнообразных состояний ЗУ для каждой итерации многократного теста *March_pA_2*. Остановимся на свойстве 4 метрик расстояния, рассмотренных в разд. 2. Согласно этому свойству применение операции отрицания для разрядов адресов $A(j)$ последовательности pA приводит к перераспределению их индексов j и, соответственно, характеристик $MD(A(j), pA)$, т. е. результатом операций отрицания является преобразование конкретного адреса $A(s)$ в адрес $A(t)$. При этом адрес $A(t)$ будет иметь характеристику $MD(A(t), pA)$, которую до преобразования имел адрес $A(s)$.

Таким образом, каждый адрес $A(s)$ последовательности адресов $2A$ в результате применения различных сочетаний отрицаний (10) принимает значения расстояния $MD(A(t), pA)$ от другого адреса $A(t)$. В случае когда все адреса $A(j)$ последовательности pA имеют одинаковые значения расстояний $MD(A(j), pA)$, применение отрицаний для получения новых адресных последовательностей, очевидно, не приведет к новому более разнообразному состоянию ячеек ЗУ. И, наоборот, если все адреса будут иметь собственные уникальные характеристики $MD(A(j), pA)$, перераспределяющиеся между адресами в результате применения выражения (10), произойдет формирование новых сочетаний состояний ячеек ЗУ. В результате создаются условия активизации и обнаружения сложных неисправностей ЗУ при многократном применении теста *March_pA_2*.

Введем новую характеристику $V(pA)$, которая численно равняется количеству отличающихся значений расстояния $MD(A(j), pA)$ адресов $A(j)$ последовательности pA. Для $p = 2$ характеристика $V(2A)$ принимает значения в диапазоне от 1 до 2^m. Например, для адресных последовательностей $2A_{Gg3}$ и $2A_{Ss0}$, приведенных в табл. 2, данная характеристика принимает значения $V(2A_{Gg3}) = 4$ и $V(2A_{Ss0}) = 1$. Значение характеристики $V(2A)$ для последовательностей адресов $2A_{c0}$ и $2A_{c8}$ (см. табл. 7) равняется единице. Это позволяет объяснить невысокую обнаруживающую способность многократного неразрушающего теста *March_pA_2* с их применением.

Для адресных последовательностей pA, свойства которых близки к свойствам случайных последовательностей, значения характеристик $AD(pA)$ и $V(pA)$ численно будут близки к своим средним значениям. Это свидетельствует об эффективности использования адресных последовательностей как для однократного применения теста *March_pA_2*, так и для повторных применений с модифицированными адресами согласно выражению (10).

Для оценки эффективности многократного применения неразрушающего теста *March_2A_2* были получены экспериментальные результаты для адресной последовательности $2A_{l4}$. Она представляет собой псевдослучайную последовательность, полученную на основании порождающего полинома $\varphi(x) = 1 + x^4 + x^9$ путем удаления четвертого разряда, что обеспечивает повторное формирование каждого восьмиразрядного адреса, кроме адреса, состоящего из всех нулей. Свойства такой последовательности близки к свойствам случайных последовательностей и характеризуются средними значениями характеристик $AD(pA)$ и $V(pA)$. Как в экспериментах с последовательностями адресов $2A_{c0}$, $2A_{c8}$ и $2A_{c4}$, в данном случае использовались модификации адресов в соответствии с выражением (10) на основании масок, приведенных в табл. 6. Результаты эксперимента даны в табл. 8.

Таблица 8. Полнота покрытия неисправностей *CFid* многократным тестом *March*_$2A_{l4}$_2, %

*Table 8. Faults coverage of CFid faults by multiple test March_*$2A_{l4}$*_2, %*

CFid \ *Index*	0	1	2	3	4	5	6	7	*FC*
∧(↑;0)	47,97	27,46	12,29	6,78	2,59	1,39	0,77	0,40	99,65
∧(↑;1)	85,11	12,54	1,95	0,37	0,02	0,01	0,00	0,00	100
∧(↓;0)	47,97	27,46	12,29	6,78	2,59	1,39	0,77	0,40	99,65
∧(↓;1)	85,11	12,54	1,95	0,37	0,02	0,01	0,00	0,00	100
∨(↑;0)	52,17	23,26	13,17	5,90	2,66	1,33	0,81	0,35	99,65
∨(↑;1)	80,91	16,74	1,99	0,33	0,02	0,00	0,00	0,00	99,99
∨(↓;0)	52,17	23,26	13,17	5,90	2,66	1,33	0,81	0,35	99,65
∨(↓;1)	80,91	16,74	1,99	0,33	0,02	0,00	0,00	0,00	99,99
FC	66,54	20,00	7,35	3,34	1,33	0,68	0,40	0,19	99,83

Для каждой из неисправностей *CFid* в зависимости от индекса повторного применения теста *March*_$2A_{l4}$_2 приведены значения увеличения их полноты покрытия в процентах. Например, при однократном применении теста полнота покрытия неисправности ∧(↑;0) составляет 47,97 % (табл. 8). Повторное применение теста увеличивает полноту покрытия на 27,46 %, а восьмикратное его использование обеспечивает суммарную полноту покрытия неисправности ∧(↑;0) 99,65 %. В последней строке (*FC*) табл. 8 приведены значения полноты покрытия всех восьми неисправностей *CFid*.

Полученные данные подтверждают эффективность адресной последовательности $2A_{l4}$ как при однократном применении теста *March*_$2A_{l4}$_2, так и при многократном тестировании ЗУ. Данные табл. 7 и 8 свидетельствуют о зависимости результатов тестирования ЗУ от свойств применяемых адресных последовательностей, данные табл. 3 – об эффективности использования последовательностей $2A$ с максимальным значением расстояния $AD(2A)$ для однократных тестов *March_2A_1* и *March_2A_2*. В свою очередь, анализ результатов, представленных в табл. 7 и 8, подтверждает гипотезу о применении последовательностей $2A$ с максимальным значением характеристики $V(2A)$ для обеспечения эффективности многократного теста *March_2A_2*.

Заключение. Предложенный подход к построению неразрушающих тестов ЗУ, основанный на применении адресных последовательностей с четным повторением адресов, показал свою высокую эффективность для случая $p = 2$. Двойное повторение адресов позволило уменьшить временную сложность предлагаемых неразрушающих тестов *March_2A_1* и *March_2A_2* по сравнению с известными методами и увеличить их диагностическую способность при сохранении высокой обнаруживающей способности неисправностей ЗУ. Исследована зависимость эффективности новых неразрушающих тестов *March_2A_1* и *March_2A_2* от применяемых адресных последовательностей и обоснован их оптимальный выбор для однократной и многократной реализации новых тестов. Интересным представляется дальнейшее исследование разработанных тестов *March_pA_1* и *March_pA_2* для случаев большей кратности p повторения адресов.

В первую очередь это касается сложных кодочувствительных неисправностей, так как многократное повторение адресов позволит обеспечить большее количество уникальных состояний ЗУ. Однако увеличение кратности повторения адресов существенно увеличивает временную сложность новых тестов, что оправдано только для случая сложных неисправностей современных ЗУ. Достаточно глубоким является вопрос формирования разнообразных адресных последовательностей с четным повторением адресов. Направления исследований в части генерирования адресных последовательностей с повторяющимися адресами и способы получения формальных методик оценки их свойств, влияющих на обнаруживающую способность неисправностей ЗУ, остаются практически открытыми.

Вклад авторов. В. Н. Ярмолик предложил идею неразрушающих тестов на базе многократно повторяющихся адресных последовательностей, В. А. Леванцевич и Д. И. Деменковец приняли участие в обобщении и анализе полученных результатов, И. Мрозек провел экспериментальные исследования.

Список использованных источников

1. Sharma, A. K. Advanced Semiconductor Memories: Architectures, Designs, and Applications / A. K. Sharma. – London : John Wiley & Sons, 2003. – 652 p.

2. The new hardware development trend and the challenges in data management and analysis / W. Pan [et al] // Data Science and Engineering. – 2018. – Vol. 6, no. 3. – P. 263–276.

3. Prince, B. High-Performance Memories: New Architecture DRAM's and SRAM's, Evolution and Function / B. Prince. – London : John Wiley & Sons, 1999. – 354 p.

4. Goor, A. J. Testing Semiconductor Memories: Theory and Practice / A. J. Goor. – Chichester, UK : John Wiley & Sons, 1991. – 536 p.

5. Ярмолик, С. В. Маршевые тесты для самотестирования ОЗУ / С. В. Ярмолик, А. П. Занкович, А. А. Иванюк. – Минск : Бестпринт, 2009. – 270 с.

6. Du, X. New Memory BIST and Repair Methods / X. Du. – Iowa : University of Iowa, 2004. – 276 p.

7. Chakraborty, A. Fault-Tolerance and Reliability Techniques for High-Density Random-Access Memories / A. Chakraborty, P. Mazumder. – Prentice Hall, 2002. – 448 p.

8. Nicolaidis, M. Theory of transparent BIST for RAMs / M. Nicolaidis // IEEE Transactions on Computers. – 1996. – Vol. 45, no. 10. – P. 1141–1156.

9. Неразрушающее тестирование запоминающих устройств / В. Н. Ярмолик [и др.]. – Минск : Бестпринт, 2005. – 230 с.

10. Ярмолик, В. Н. Обзор методов неразрушающего тестирования ОЗУ / В. Н. Ярмолик, А. П. Занкович // Доклады БГУИР. – 2005. – № 4(12). – С. 62–72.

11. Ярмолик, С. В. Многократные неразрушающие маршевые тесты с изменяемыми адресными последовательностями / С. В. Ярмолик, В. Н. Ярмолик // Автоматика и телемеханика. – 2007. – № 4. – С. 126–137.

12. Ярмолик, В. Н. Адресные последовательности для многократного тестирования ОЗУ / В. Н. Ярмолик, С. В. Ярмолик // Информатика. – 2014. – № 2(39). – С. 124–136.

13. Yarmolik, V. N. Modified gray and counter sequences for memory test address generation / V. N. Yarmolik, S. V. Yarmolik // Proc. of the 13th Intern. Conf. MIXDES Design of Integrated Circuits and Systems, Gdynia, Poland, 22–24 June 2006. – Gdynia, 2006. – P. 572–576.

14. Goor, A. J. Optimizing memory BIST Address Generator implementations / A. J. Goor, H. Kukner, S. Hamdioui // Proc. of the 2011 6th Intern. Conf. on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), Athens, Greece, 6–8 Apr. 2011. – Athens, 2011. – P. 572–576.

15. Ярмолик, С. В. Квазислучайное тестирование вычислительных систем / С. В. Ярмолик, В. Н. Ярмолик // Информатика. – 2013. – № 3(39). – С. 92–103.

16. Savage, C. A survey of combinatorial Gray code / C. Savage // SIAM Review. – 1997. – Vol. 39, no. 4. – P. 605–629.

17. Chi, H. Computational investigations of quasi-random sequences in generating test cases for specification-based tests / H. Chi, E. I. Jones // Proc. of the Winter Simulation Conf. WSC 2006, Monterey, CA, USA, 3–6 Dec. 2006. – Monterey, 2006. – P. 975–980.

18. Chen, T. Y. Quasi-random testing / T. Y. Chen, R. Merkel // IEEE Transaction on Reliability. – 2007. – Vol. 56, no. 3. – P. 562–568.

19. 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.

20. Ярмолик, С. В. Анализ количественных характеристик различия при тестировании ОЗУ / С. В. Ярмолик, А. Н. Курбацкий, В. Н. Ярмолик // Информатика. – 2008. – № 3(19). – С. 90–98.

21. Thompson, A. C. Minkowski Geometry / A. C. Thompson. – Cambridge, N. Y., 1996. – 364 p.

22. Tubbs, J. D. Note on binary template matching / J. D. Tubbs // Pattern Recognition. – 1989. – Vol. 22, no. 4. – P. 359–365.

23. 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.

24. Zhou, Z. Q. Using coverage information to guide test case selection in adaptive random testing / Z. Q. Zhou // Proc. 34th IEEE Computer Soft and Applications Conf., Seoul, Korea, 19–23 July 2010. – Seoul, 2010. – P. 208−213.

25. Chan, K. P. Good random testing / K. P. Chan, T. Y. Chen, D. Towey // Proc. 9th Ada-Europe Intern. Conf. on Reliable Software Technologies (LNCS), Palma de Mallorca, Spain, 14–18 June 2004. Palma de Mallorca, 2004. − P. 200−212.

26. Kuo, F. C. An in-depth study of mirror adaptive random testing / F. C. Kuo // Proc. 14th European Conf. on Soft Quality, Los Alamitos, CA, USA. – Los Alamitos, 2009. − P. 51−58.

27. Shiyi, Xu. Orderly random testing for both hardware and software / Xu Shiyi // Proc. Pacific Rim Intern. Symp. on Dependable Computing, Taipei, Taiwan, 15–17 Dec. 2008. – Taipei, 2008. − P. 160−167.

28. Ярмолик, С. В. Управляемое случайное тестирование / С. В. Ярмолик, В. Н. Ярмолик // Информатика. – 2011. – № 1(29). – С. 79–88.

29. Ярмолик, В. Н. Многократные управляемые вероятностные тесты / В. Н. Ярмолик, В. А. Леванцевич, И. Мрозек // Информатика. – 2015. – № 2(12). – С. 63–76.

References

1. Sharma A. K. *Advanced Semiconductor Memories: Architectures, Designs, and Applications*. London, John Wiley & Sons, 2003, 652 p.

2. Pan W., Li Z., Zhang Y., Weng C. The new hardware development trend and the challenges in data management and analysis. *Data Science and Engineering*, 2018, vol. 6, no. 3, pp. 263–276.

3. Prince B. *High-Performance Memories: New Architecture DRAM's and SRAM's, Evolution and Function*. London, John Wiley & Sons, 1999, 354 p.

4. Goor A. J. *Testing Semiconductor Memories: Theory and Practice*. Chichester, UK, John Wiley & Sons, 1991, 536 p.

5. Yarmolik S. V., Zankovich A. P., Ivaniuk A. A. Marshevue testu dlya samotestirovaniya OZU. *RAM Self-Test March Tests*. Minsk, Bestprint, 2009, 270 p. (In Russ.).

6. Du X. *New Memory BIST and Repair Methods*. Iowa, University of Iowa, 2004, 276 p.

7. Chakraborty A. *Fault-Tolerance and Reliability Techniques for High-Density Random-Access Memories*. Prentice Hall, 2002, 448 p.

8. Nicolaidis M. Theory of transparent BIST for RAMs. *IEEE Transactions on Computers*, 1996, vol. 45, no. 10, pp. 1141–1156.

9. Yarmolik V. N., Murashko I. A., Kummert A., Ivaniuk A. A. Nerazrushayuschee testirovanie zapominayuschih ustroistv. *Transparent Memory Testing*. Minsk, Bestprint, 2005, 230 p. (In Russ.).

10. Yarmolik V. N., Zankovich A. P. *Overview of transparent testing methods for RAM*. Doklady Belorusskogo gosudarstvennogo universiteta informatiki i radioèlektroniki [*Reports of the Belarusian State University of Informatics and Radioelectronics*], 2005, no. 4(12), pp. 62–72 (In Russ.).

11. Yarmolik S. V., Yarmolik V. N. *Multiple non-destructive marching tests with variable address sequences*. Avtomatika i telemehanika [*Automation and Remote*], 2007, no. 4, pp. 126–137 (In Russ.).

12. Yarmolik V. N., Yarmolik S. V. *Address sequences for repeated testing of RAM*. Informatika [*Informatics*], 2014, no. 2(39), pp. 124–136 (In Russ.).

13. Yarmolik V. N., Yarmolik S. V. Modified gray and counter sequences for memory test address generation. *Proceedings of the 13th International Conference MIXDES Design of Integrated Circuits and Systems, Gdynia, Poland, 22–24 June 2006*. Gdynia, 2006, pp. 572–576.

14. Goor A. J., Kukner H., Hamdioui S. Optimizing memory BIST Address Generator implementations. *Proceedings of the 2011 6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), Athens, Greece, 6–8 April 2011*. Athens, 2011, pp. 572–576.

15. Yarmolik S. V., Yarmolik V. N. *Quasi-random testing of computing systems*. Informatika [*Informatics*], 2013, no. 3(39), pp. 92–103 (In Russ.).

16. Savage C. A survey of combinatorial Gray code. *SIAM Review*, 1997, vol. 39, no. 4, pp. 605–629.

17. Chi H., Jones E. I. Computational investigations of quasi-random sequences in generating test cases for specification-based tests. *Proceedings of the Winter Simulation Conference WSC 2006, Monterey, California, USA, 3–6 December 2006*. Monterey, 2006, pp. 975–980.

18. Chen T. Y., Merkel R. Quasi-random testing. *IEEE Transaction on Reliability*, 2007, vol. 56, no. 3, pp. 562–568.

19. 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.

20. Yarmolik S. V., Kurbatski A. N., Yarmolik V. N. *Dissimilarity measure analysis for random access memory testing*. Informatika [*Informatics*], 2008, no. 3(19), pp. 90–98 (In Russ.).

21. Thompson A. C. *Minkowski Geometry*. Cambridge, New York, 1996, 364 p.

22. Tubbs J. D. Note on binary template matching. *Pattern Recognition*, 1989, vol. 22, no 4, pp. 359−365.

23. 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.

24. Zhou Z. Q. Using coverage information to guide test case selection in adaptive random testing. *Proceedings 34th IEEE Computer Soft and Applications Conference*, *Seoul*, *Korea*, *19–23 July 2010*. Seoul, 2010, pp. 208−213.

25. Chan K. P., Chen T. Y., Towey D. Good random testing. *Proceedings 9th Ada-Europe International Conference on Reliable Software Technologies (LNCS), Palma de Mallorca, Spain, 14–18 June 2004.* Palma de Mallorca, 2004, pp. 200−212.

26. Kuo F. C. An in-depth study of mirror adaptive random testing. *Proceedings 14th European Conference on Soft Quality, Los Alamitos, CA, USA.* Los Alamitos, 2009, pp. 51−58.

27. Shiyi Xu. Orderly random testing for both hardware and software. *Proceedings Pacific Rim International Symposium on Dependable Computing, Taipei, Taiwan, 15-17 December 2008*. Taipei, 2008, pp. 160−167.

28. Yarmolik S. V., Yarmolik V. N. *Controlled random testing*. Informatika [*Informatics*], 2011, no. 1(29), pp. 79–88 (In Russ.).

29. Yarmolik V. N., Levantsevich V. A., Mrozek I. *Multiple controlled probability tests*. Informatika [*Informatics*], 2015, no. 2(12), pp. 63–76 (In Russ.).

Информация об авторах

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

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

Леванцевич Владимир Александрович, магистр технических наук, старший преподаватель, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: lvn@bsuir.by

Деменковец Денис Викторович, магистр технических наук, старший преподаватель, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: demenkovets@bsuir.by

Information about the authors

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

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

Vladimer A. Levantsevich, M. Sci. (Eng.), Senior Lecture, Belarusian State University of Informatics and Radioelectronics.
E-mail: lvn@bsuir.by

Denis V. Demenkovets, M. Sci. (Eng.), Senior Lecture, Belarusian State University of Informatics and Radioelectronics.
E-mail: demenkovets@bsuir.by