ISSN 1816-0301 (print)

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ

COMPUTER-AIDED DESIGN

УДК 004.33.054

Поступила в редакцию 01.11.2017
Received 01.11.2017

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

[1]Белорусский государственный университет информатики и радиоэлектроники, Минск, Республика Беларусь
[2]Белостокский технический университет, Белосток, Польша

ПСЕВДОИСЧЕРПЫВАЮЩЕЕ ТЕСТИРОВАНИЕ ЗАПОМИНАЮЩИХ УСТРОЙСТВ НА БАЗЕ МНОГОКРАТНЫХ МАРШЕВЫХ ТЕСТОВ

Аннотация. Анализируются методы тестирования современных запоминающих устройств. Показывается обоснованность применения псевдоисчерпывающих тестов для обнаружения сложных неисправностей памяти. Формулируется необходимое условие генерирования псевдоисчерпывающего теста для заданного количества ячеек запоминающего устройства. Показывается, что задача генерирования псевдоисчерпывающего теста на базе многократных тестов запоминающих устройств с изменяемым начальным состоянием сводится к комбинаторной задаче собирателя купонов. Приводятся оценки средней, минимальной и максимальной кратности многократного теста для обеспечения исчерпывающего множества комбинаций для заданного числа ячеек запоминающего устройства. Экспериментально показывается справедливость аналитических оценок и подтверждается возможность псевдоисчерпывающего тестирования запоминающих устройств.

Ключевые слова: маршевые тесты, псевдоисчерпывющие тесты, задача собирателя купонов

Для цитирования. Ярмолик, В. Н. Псевдоисчерпывающее тестирование запоминающих устройств на базе многократных маршевых тестов / В. Н. Ярмолик, И. Мрозек, В. А. Леванцевич // Информатика. – 2018. – Т. 15, № 1. – С. 110–121.

V. N. Yarmolik[1], I. Mrozek[2], B. A. Levantsevich[1]

[1]Belarusian State University of Computer Science and Electronic Engineering, Minsk, Republic of Belarus
[2]Politechnika Białostocka, Białystok, Poland

PSEUDO-EXHAUSTIVE MEMORY DEVICES TESTING BASED ON MULTIPLE MARCH TESTS

Abstract. Methods for modern memory devices are analyzed. The validity of using pseudo-exhaustive tests to detect complex memory faults is shown. A necessary condition for generating a pseudo-exhaustive test for a given number of memory cells is formulated. It is shown that the problem of generating a pseudo-exhaustive test based on multiple memory tests with a variable background is reduced to the combinatorial task of the coupon collector. Estimates of the mean, minimum, and maximum multiplicity of a multiple test are given to provide an exhaustive set of combinations for a given number of cells of a memory device. The validity of analytical estimates is shown experimentally and the possibility of pseudo-exhaustive memory testing is confirmed.

Keywords: march tests, pseudo-exhaustive tests, coupon collector problem

For citation. Yarmolik V. N., Mrozek I., Levantsevich B. A. Pseudo-Exhaustive Memory Testing Based on Multiple March Tests. *Informatics,* 2018, vol. 15, no. 1, pp. 110–121 (in Russian).

Введение. Высокой эффективностью обнаружения неисправностей современных цифровых устройств, и в особенности запоминающих устройств, характеризуются исчерпывающие тесты и их различные аппроксимации в виде псевдоисчерпывающих тестов [1−4]. Чаще всего в качестве псевдоисчерпывающих тестов рассматриваются двоичные тестовые наборы, которые обеспечивают 2^k возможных двоичных комбинаций для любых k из N разрядов тестовых наборов, где $k << N$ [3]. Вероятностное тестирование и его многочисленные разновидности, основанные на принципе черного ящика, также являются эффективной аппроксимацией исчерпывающего тестирования [5]. Существующие модификации классического метода построения вероятностных тестов объединяются принципом управления процедурой формирования очередного тестового набора [6]. Существенным недостатком подобных тестов является вычислительная сложность формирования тестовых наборов [3, 5].

С целью уменьшения вычислительной сложности формирования управляемых вероятностных тестов широко обсуждаются и используются итеративные вероятностные тесты [6], вероятностные тесты с малым числом наборов [7], а также многократные тесты для запоминающих устройств [8–10]. При проектировании многократных тестов, в частности многократных маршевых тестов запоминающих устройств, не требуется вычисление каких-либо характеристик, а построение таких тестов основывается на реализации определенной процедуры [3, 4, 8–10].

Целью настоящей статьи является исследование эффективности применения многократных маршевых тестов с изменяемым начальным состоянием памяти для псевдоисчерпывающего тестирования современных запоминающих устройств. Приводятся основные аналитические соотношения, а также оценки сложности процедуры формирования псевдоисчерпывающих тестов, обеспечивающих всевозможные 2^k двоичные комбинации в произвольных k из N запоминающих ячеек памяти.

1. Многократные маршевые тесты запоминающих устройств. Ранее [11] было показано, что при однократном применении маршевых тестов запоминающих устройств полнота покрытия сложных неисправностей запоминающих устройств, а также любых других неисправностей остается неизменной. В качестве обобщающих (доминирующих) неисправностей, обнаруживаемых маршевыми тестами, рассматривается модель кодочувствительных неисправностей PSFk или ее разновидности, определяемые как NPSFk, которые эффективно покрывают более простые модели неисправностей запоминающих устройств [11]. Как показано в ряде источников [10, 11], наиболее адекватной моделью являются неисправности PNPSFk, количество которых для произвольных k из N ячеек памяти и фиксированной базовой ячейки $2 \times 2^{k-1} = 2^k$.

Для повышения эффективности повторного применения маршевых тестов радикальным подходом может быть как изменение последовательности адресов, применяемых в каждом последующем маршевом тесте [3, 12], так и изменение начального состояния памяти [7, 13]. Изменение содержимого памяти, например, при ее обновлении в результате вычислений позволяет существенно увеличить полноту покрытия при повторном применении маршевого теста [7, 10, 13].

В статье [3] было обосновано применение многократных маршевых тестов с изменяемыми адресными последовательностями для реализации псевдоисчерпывающего тестирования. Численные характеристики, представленные в данной работе, позволяют сделать вывод о реальности формирования псевдоисчерпывающего теста для современных запоминающих устройств. Применение не более 100 итераций простейшего маршевого теста позволяет реализовать исчерпывающее тестирование любого подмножества из $k \leq 6$ ячеек памяти. Определяющим недостатком метода, основанного на изменении адресных последовательностей, является необходимость хранения таких последовательностей с целью их воспроизведения.

Применение изменяемых начальных состояний современных запоминающих устройств рассматривалось в контексте их оптимальности для многократных маршевых тестов, кратность которых ограничивалась небольшими значениями [7, 10, 13]. В частности, были получены оптимальные значения для двукратных и трехкратных маршевых тестов, а также сделан оптимальный выбор начального состояния на основании расстояния Хэмминга и евклидова расстояния [7, 10, 13].

2. Анализ маршевых тестов запоминающих устройств. Маршевые тесты запоминающих устройств состоят из последовательных фаз с конкретными комбинациями операций записи (w) и чтения (r). Каждая фаза реализует последовательное обращение ко всем ячейкам запоминающего устройства согласно их адресам. Символ ↑ обозначает прямую (возрастающую) последовательность адресов памяти, символ ↓ – обратную (убывающую) последовательность, а символ ↕ – возрастающую или убывающую последовательность адресов [11]. Обозначения $w0$ и $w1$ используются для определения операции записи 0 и 1 в ячейку памяти, а $r0$ и $r1$ – операции чтения 0 и 1. В качестве примера рассмотрим два наиболее часто применяемых на практике маршевых теста, а именно MATS ++ {↕ ($w0$); ↑ ($r0$, $w1$); ↓ ($r1$, $w0$, $r0$)} и *March C−* {↕ ($w0$); ↑ ($r0$, $w1$); ↑ ($r1$, $w0$); ↓ ($r0$, $w1$); ↓ ($r1$, $w0$); ↕ ($r0$)} [11]. Сложность теста MATS ++ равна $6N$, а сложность *March C−* составляет $10N$, где N представляет собой емкость запоминающего устройства в битах [11].

Как уже отмечалось в разд. 1, маршевые тесты запоминающих устройств при их реализации генерируют в k произвольных ячейках памяти так называемую орбиту, которая представляет собой набор двоичных k-разрядных векторов [3, 12]. Конкретный набор векторов, составляющих орбиту, зависит от трех основных факторов [3]. Прежде всего используемый маршевый тест однозначно определяет структуру орбиты, так как он формулирует правила, по которым генерируются двоичные векторы в ячейках запоминающего устройства. Эти правила представляются набором фаз маршевого теста, их структурой и используемой последовательностью адресов. Начальное состояние запоминающего устройства является вторым аргументом, который принимается во внимание при формировании конкретной орбиты. Адресная последовательность, представляющая собой порядок адресов запоминающих ячеек памяти, которые формируются маршевым тестом, является третьим фактором. Порядок адресов в последовательности может быть произвольным с единственным ограничением, что каждый адрес формируется ровно один раз, а их последовательность обратима [3].

Анализ классических маршевых тестов показывает, что существует только одна или две отличающиеся орбиты для k произвольных ячеек памяти, которые генерируются во время выполнения теста [14]. Действительно, для фаз классических маршевых тестов применяются два возможных начальных состояния в произвольных k ячейках памяти, а именно все нули и все единицы. Как правило, во время выполнения теста используются фазы, для которых данные из ячеек памяти считываются, и в них записываются инверсные значения. Наряду с двумя возможными вариантами начальных состояний (единичное и нулевое) для k из N произвольных ячеек памяти во время выполнения фаз применяется прямая последовательность адресов (↑) или обратная последовательность (↓). Поэтому в маршевых тестах существуют четыре разные орбиты. Количество орбит и их порядок во время выполнения теста зависят от применяемого маршевого теста. Все возможные комбинации начальных состояний и адресных последовательностей для классических маршевых тестов и соответствующие им орбиты, формируемые в произвольных k ячейках памяти, приведены в табл. 1.

Таблица 1

Орбиты, формируемые классическими маршевыми тестами

Орбиты	O_0	O_1	O_2	O_3
Начальное состояние P_0	000…00	000…00	111…11	111…11
Последовательность адресов	Прямая (↑)	Обратная (↓)	Прямая (↑)	Обратная (↓)
P_0	000…00	000…00	111…11	111…11
P_1	000…01	100…00	111…10	011…11
P_2	000…11	110…00	111…00	001…11
…	…	…	…	…
P_{k-2}	001…11	111…00	110…00	000…11
P_{k-1}	011…11	111…10	100…00	000…01
P_k	111…11	111…11	000…00	000…00

Например, маршевый тест MATS ++ генерирует две орбиты, а именно O_0 во время реализации его фазы ↑ ($r0$, $w1$) и O_3 в результате применения фазы ↓ ($r1$, $w0$, $r0$). В то же время *March C−* позволяет генерировать все четыре возможные орбиты, что доказывает его более высокую эффективность для тестирования запоминающих устройств.

Для орбиты O, порожденной маршевым тестом, верны следующие общие утверждения [3].

Утверждение 1. *Орбита O, сформированная в произвольных k ячейках памяти как результат применения маршевого теста, состоит из* $k+1$ *k-разрядного двоичного вектора* $P_0, P_1, ..., P_k$.

Данное утверждение следует из структуры орбиты, показанной в табл. 1. Следует отметить, что для маршевого теста из-за фазы инициализации ↕ ($w0$) или любых фаз теста, таких как ↕ (... $w0$ ...), P_0 принимает нулевое значение $P_0 = 000 ... 0$, т. е. все k ячеек памяти устанавливаются в нулевое состояние. Значение начального состояния ячеек $P_0 = 111 ... 1$ может быть сгенерировано тестом, который имеет фазу инициализации ↕ ($w1$) или включает фазу ↕ (... $w1$ ...).

Утверждение 2. *Расстояние Хэмминга* $HD(P_i, P_{i+j})$ *между двумя векторами состояний* P_i *и* P_{i+j} *орбиты O для k произвольных ячеек запоминающего устройства, где* $i \in \{0, 1, 2, ..., k–1\}$ *и* $j \in \{0, 1, 2, ..., k–i\}$, *равно j.*

Это утверждение верно для различных 2^k начальных состояний в k ячейках памяти и произвольной адресной последовательности.

Следует отметить важное следствие утверждения 2 о том, что в пределах орбиты O нет двух одинаковых двоичных векторов P_i и P_l для $i \neq l$. Для любого начального состояния P_0 расстояние Хэмминга между любыми двумя векторами P_i и P_{i+j} всегда одинаково и определяется соотношением $HD(P_i, P_{i+j}) = j$. Например, если $j = 1$, то $HD(P_i, P_{i+1}) = 1$.

Для случая псевдоисчерпывающих тестов основная цель состоит в генерировании всех возможных 2^k двоичных векторов для произвольных k ячеек памяти. При этом последовательность векторов не имеет значения. Оптимальным решением является создание другой орбиты, отличной от сформированных ранее орбит, для получения максимального количества дополнительных векторов. Все орбиты должны быть максимально разными с точки зрения генерируемых двоичных векторов.

Утверждение 3. *Две орбиты* O_i *и* O_l *считаются эквивалентными или равными* ($O_i = O_l$), *если обе орбиты включают одинаковые двоичные векторы независимо от их порядка.*

В случае классических маршевых тестов $O_0 = O_3$ и $O_1 = O_2$ (см. табл. 1).

Все множество классических маршевых тестов, как отмечалось ранее [3, 10, 15], делится на два основных множества. Первое множество включает так называемые тесты типа MATS ++. Эти тесты характеризуются тем, что генерируют одну орбиту из четырех возможных или две, но эквивалентные (O_0 и O_3 или O_1 и O_2). Второе множество, называемое маршевыми *March C−* подобными тестами, характеризуется формированием при их реализации по меньшей мере двух различных орбит, а именно O_0, O_1; O_0, O_2; O_1, O_3; O_2, O_3.

Проведенный анализ позволяет сделать следующий вывод. Однократное применение маршевого теста дает возможность получить только одну или две орбиты, которые зависят от начального состояния памяти и используемой адресной последовательности. Для тестов типа MATS ++ применение маршевого теста позволяет генерировать $k + 1$ различных k-разрядных двоичных векторов или $2k$ для тестов типа *March C−*. Очевидно, что для получения всех возможных 2^k двоичных векторов для k произвольных ячеек памяти маршевый тест должен применяться более одного раза в случае различных начальных состояний или различных адресных последовательностей. Подход, основанный на изменении адресных последовательностей, был всесторонне исследован ранее [3, 10, 12].

3. Изменение начального состояния запоминающих устройств. Рассмотрим бит-ориентированное запоминающее устройство, содержащее $N = 2^m$ ячеек, каждая из которых имеет уникальный m-разрядный адрес. Для случая произвольного количества $k < 2^m$ ячеек памяти с адресами β, γ, ..., δ, ε, η, где $2^m > \beta > \gamma > ... > \delta > \varepsilon > \eta \geq 0$, оригинальная орбита O_o (табл. 2) формируется согласно процедуре, показанной в табл. 1 [3].

Таблица 2

Орбита O_o, состоящая из k+1 двоичного вектора, для k ячеек памяти

Орбита O_o						
Вектор	Индексы разрядов вектора					
	$k-1$	$k-2$	…	2	1	0
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	$\overline{a_\gamma}$	…	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_k	$\overline{a_\beta}$	$\overline{a_\gamma}$	…	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$

Начиная с исходного состояния памяти $P_0 = a_\beta a_\gamma \ldots a_\delta a_\varepsilon a_\eta$, где $a_i \in \{0, 1\}$, $i \in \{\beta, \gamma, \ldots, \delta, \varepsilon, \eta\}$ и $\overline{a_i}$ принимает инверсное значение относительно a_i, в орбите формируются k + 1 различных двоичных векторов, которые удовлетворяют утверждениям 1 и 2. В качестве примера рассмотрим случай, когда k = 3. Для всех возможных 2^3 = 8 начальных состояний восемь орбит, полученных согласно табл. 2, приведены в табл. 3.

Таблица 3

Восемь возможных орбит для k = 3

O_0			O_1			O_2			O_3			O_4			O_5			O_6			O_7		
0	0	0	0	0	1	0	1	0	0	1	1	1	0	0	1	0	1	1	1	0	1	1	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
0	1	1	0	1	0	0	0	1	0	0	0	1	1	1	1	1	0	1	0	1	1	0	0
1	1	1	1	1	0	1	0	1	1	0	0	0	1	1	0	1	0	0	0	1	0	0	0

Для случая изменяемого начального состояния в k ячейках запоминающего устройства справедливы следующие утверждения.

Утверждение 4. *Для произвольных k > 1 ячеек запоминающего устройства и различных 2^k начальных состояний $P_0 = a_\beta a_\gamma \ldots a_\delta a_\varepsilon a_\eta$ произвольный маршевый тест при использовании фиксированной адресной последовательности генерирует 2^k уникальных неэквивалентных орбит.*

Доказательство. Так как $a_i \in \{0, 1\}$, $i \in \{\beta, \gamma, \ldots, \delta, \varepsilon, \eta\}$, для k ячеек запоминающего устройства существует 2^k различных начальных состояний. Для случая фиксированного порядка адресов каждое начальное состояние генерирует только одну орбиту. Таким образом, для случая неизменяемого порядка адресов количество различных орбит не может быть больше 2^k. Число орбит менее 2^k может быть только в случае, когда для двух различных начальных состояний P_0 две орбиты эквивалентны (утверждение 3). Это означает, что для орбиты O_o, показанной в табл. 2, с начальным состоянием $P_0 = a_\beta a_\gamma \ldots a_\delta a_\varepsilon a_\eta$ существует эквивалентная орбита O_n с $P_0 \neq a_\beta a_\gamma \ldots a_\delta a_\varepsilon a_\eta$. Согласно утверждению 3 для случая, когда $O_n = O_o$, орбита O_n должна быть построена в результате перестановки двоичных векторов, используемых в орбите O_o. Это означает, что двоичные векторы $P_0 = a_\beta a_\gamma \ldots a_\delta a_\varepsilon a_\eta$ и $P_k = \overline{a_\beta}\ \overline{a_\gamma} \ldots \overline{a_\delta}\ \overline{a_\varepsilon}\ \overline{a_\eta}$ орбиты O_o с расстоянием Хэмминга HD(P_0, P_k) = k также должны входить в состав орбиты O_n. Единственная новая позиция для вектора $P_k = \overline{a_\beta}\ \overline{a_\gamma} \ldots \overline{a_\delta}\ \overline{a_\varepsilon}\ \overline{a_\eta}$ – это позиция P_0 в орбите O_n, а для вектора $P_0 = a_\beta a_\gamma \ldots a_\delta a_\varepsilon a_\eta$ – позиция P_k, что следует из утверждения 2. Если в орбите O_n начальное состояние $P_k = \overline{a_\beta}\ \overline{a_\gamma} \ldots \overline{a_\delta}\ \overline{a_\varepsilon}\ \overline{a_\eta}$, то для той же последовательности адресов P_1 при k > 1 не является двоичным вектором, принадлежащим O_o. Это означает неэквивалентность орбит O_o и O_n, т. е. $O_o \neq O_n$.

Утверждение 5. *Для заданного k и постоянного порядка адресов существуют $2^k - (k^2 + k)/2 - 1$ орбит O_n с иными начальными состояниями P_0, состоящих из двоичных векторов, которые не используются в орбите O_o, и $(k^2 + k)/2$ орбит, включающих по два вектора, которые входят в орбиту O_o.*

Доказательство. Для общего случая орбиты O_o с начальным состоянием $P_0 = a_\beta a_\gamma \ldots a_\delta a_\varepsilon a_\eta$ новая орбита O_n будет получена в результате инвертирования r ($k \geq r \geq 1$) разрядов для всех двоичных векторов исходной орбиты O_o. В качестве примера две новые орбиты O_w и O_v как результат инвертирования определенных разрядов в векторах орбиты O_o для $k = 5$ показаны на рис. 1. В данном случае использовалось инвертирование соседних разрядов векторов, поэтому отрицания представляют собой единый непрерывный блок.

	O_o					O_w					O_v				
	4	3	2	1	0	4	3	2	1	0	4	3	2	1	0
P_0	a_β	a_γ	a_δ	a_ε	a_η	a_β	a_γ	a_δ	$\overline{a_\varepsilon}$	a_η	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	a_η
P_1	a_β	a_γ	a_δ	a_ε	$\overline{a_\eta}$	a_β	a_γ	a_δ	$\overline{a_\varepsilon}$	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_2	a_β	a_γ	a_δ	$\overline{a_\varepsilon}$	$\overline{a_\eta}$	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	a_γ	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$	a_β	a_γ	$\overline{a_\delta}$	a_ε	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	a_δ	a_ε	$\overline{a_\eta}$
P_4	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ε	$\overline{a_\eta}$	a_β	a_γ	a_δ	a_ε	$\overline{a_\eta}$
P_5	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ε	$\overline{a_\eta}$	$\overline{a_\beta}$	a_γ	a_δ	a_ε	$\overline{a_\eta}$

Рис. 1. Результат инвертирования соседних разрядов векторов орбиты

Орбита O_w является результатом отрицания всего лишь одного разряда с индексом 1 для всех векторов орбиты O_o, а орбита O_v получена в результате инвертирования блока разрядов векторов с индексами 1, 2 и 3. В результате обе орбиты O_w и O_v имеют три новых вектора и включают в себя два вектора из оригинальной орбиты O_o. Так, в орбите O_w используются векторы P_1 и P_2 из орбиты O_o, а в орбите O_v – соответственно P_1 и P_4. В обоих случаях эти пары векторов меняют свои позиции в орбите. Для новой орбиты O_w вектор P_1 занимает положение P_2 и, наоборот, P_2 находится на позиции P_1. То же самое можно заключить и для векторов P_1 и P_4 в случае орбиты O_v. Остальные $k - 1$ векторы из-за примененных отрицаний являются новыми по сравнению с множеством векторов орбиты O_o.

Обобщая приведенный анализ, можно сделать следующий вывод. Для случая одного блока последовательных отрицаний, примененных для O_o, с целью получения новой орбиты O_n, как это было проиллюстрировано примерами построения O_w и O_v, результирующая новая орбита включает только два двоичных вектора исходной орбиты O_o. Остальные векторы отличаются от векторов орбиты O_o. Учитывая утверждения 3 и 4, в новой орбите, очевидно, будут повторяться двоичные векторы P_i и P_j из множества векторов $P_0, P_1, P_2, \ldots, P_k$. При этом вектор P_i займет позицию вектора P_j и, наоборот, P_j – позицию P_i. 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. Тогда число M_1 новых орбит O_n, которые включают в себя два вектора из оригинальной орбиты O_o, может быть рассчитано как

$$M_1 = k + (k - 1) + (k - 2) + \cdots + 2 + 1 = (k^2 + k)/2. \quad (1)$$

Применение других наборов отрицаний для получения новой орбиты, которые включают в себя более одного блока последовательных отрицаний, приводит к получению абсолютно новой орбиты. Например, орбиты O_x и O_z включают другие наборы двоичных векторов по сравнению с векторами орбиты O_o, как показано для случая $k = 5$ на рис. 2.

	O_o					O_x					O_z				
	4	3	2	1	0	4	3	2	1	0	4	3	2	1	0
P_0	a_β	a_γ	a_δ	a_ε	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	$\overline{a_\varepsilon}$	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	a_δ	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_1	a_β	a_γ	a_δ	a_ε	$\overline{a_\eta}$	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	$\overline{a_\varepsilon}$	a_η	a_β	$\overline{a_\gamma}$	a_δ	$\overline{a_\varepsilon}$	a_η
P_2	a_β	a_γ	a_δ	$\overline{a_\varepsilon}$	$\overline{a_\eta}$	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	a_ε	a_η	a_β	$\overline{a_\gamma}$	a_δ	a_ε	a_η
P_3	a_β	a_γ	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ε	a_η	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ε	a_η
P_4	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ε	a_η	a_β	a_γ	$\overline{a_\delta}$	a_ε	a_η
P_5	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$	a_β	a_γ	$\overline{a_\delta}$	a_ε	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ε	a_η

Рис. 2. Результат применения более одного блока инверсий

Таким образом, можно сделать вывод, что применение более чем одного блока последовательных отрицаний при получении новой орбиты O_n разрушает два вектора P_i и P_j, которые сохраняются в случае одного блока отрицаний. Число M_2 новых орбит, состоящее из двоичных векторов, которые не участвуют в исходной орбите O_o, можно оценить как

$$M_2 = 2^k - (k^2 + k)/2 - 1. \tag{2}$$

Для случая $k = 3$ согласно (1) и (2) получим $M_1 = 6$ и $M_2 = 1$, что подтверждается результатами табл. 3. Действительно, только одна орбита O_5 имеет совершенно разные двоичные векторы в сравнении с O_0. Остальные орбиты O_1, O_2, O_3, O_4, O_6 и O_7 включают ровно два двоичных вектора из орбиты O_0. Численные значения M_1 и M_2 для разных значений k представлены в табл. 4.

Таблица 4

Численные значения M_1 и M_2

k	2	3	4	5	6	7	8	9	10
M_1	3	6	10	15	21	28	36	45	55
M_2	0	1	5	16	42	99	219	466	968

Ограничение количества разновидностей новых орбит только двумя видами и их точное число M_1 и M_2 позволяют оценить среднее число M_{ave} двоичных векторов, формируемых в результате двукратного маршевого теста. В случае двукратного теста MATS ++ это значение можно получить следующим образом. Для произвольного значения ячеек k однократное применение теста MATS ++ формирует в них $k + 1$ двоичный вектор (см. утверждение 1). Основываясь на общей идее многократного тестирования запоминающих устройств [10], исходные состояния памяти перед последующим применением маршевого теста формируются как случайные равномерно распределенные данные. Это означает, что для тех же k ячеек памяти новая орбита из всех возможных 2^k орбит генерируется с равной вероятностью. Тогда среднее количество новых двоичных векторов, формируемых во время повторного применения теста MATS ++, будет определяться как $[M_1 \times (k-1) + M_2 \times (k+1)]/2^k$, а значение среднего числа M_{ave} двоичных векторов, формируемых в результате двукратного маршевого теста MATS ++, определяется соотношением

$$M_{ave} = (k+1) + \frac{M_1(k-1) + M_2(k+1)}{2^k} = 2(k+1) - \frac{(k+1)^2}{2^k}. \tag{3}$$

Численные значения $M_{ave}/2^k$ для различных значений k приведены в табл. 5.

Таблица 5

Численные значения $M_{ave}/2^k$

k	2	3	4	5	6	7	8	9	10
$M_{ave}/2^k$	0,937	0,750	0,527	0,339	0,206	0,121	0,069	0,038	0,021

Представленное в табл. 5 любое значение, умноженное на 100 %, может быть интерпретировано как число генерируемых двоичных векторов в процентном исчислении в результате двукратного применения маршевого теста MATS ++. Для экспериментальной проверки приведенных в табл. 5 аналитических результатов были получены реальные экспериментальные значения для разных k. Эти значения являются средней величиной $M_{ave}/2^k$ в процентах, полученной на основе 100 000 ее экспериментальных значений. Результаты экспериментов приведены в табл. 6.

Таблица 6

Экспериментальные значения

k	2	3	4	5	6	7	8	9	10
$(M_{ave}/2^k)\times100$ %	93,76	75,04	52,74	33,98	20,64	12,12	6,91	3,87	2,14

Из табл. 5 и 6 видно, что экспериментальные и теоретические значения $M_{ave}/2^k$ очень близки и практически одинаковы. Это объясняется точностью аналитических соотношений (1)–(3). Очевидно, что аналогичные оценки могут быть получены для трехкратного, четырехкратного и т. д. применений маршевых тестов. Однако для этого нужно построить достаточно громоздкие и сложные комбинаторные соотношения, сложность которых увеличивается с ростом k. Поэтому в разд. 4 рассмотрим более общую модель для любого значения кратности многократных маршевых тестов.

Утверждение 6. *Минимальное количество Q_{min} орбит, полученных в результате многократного применения маршевого теста с изменяемыми начальными состояниями и необходимых для получения всех 2^k возможных двоичных векторов в произвольных k ячейках, должно удовлетворять неравенству*

$$Q_{min} \geq \left\lceil \frac{2^k}{k+1} \right\rceil, \tag{4}$$

где $k = 2, 3, \ldots$

Значение Q_{min} может рассматриваться как нижняя граница для минимального числа маршевых тестов типа MATS ++ при достижении исчерпывающего покрытия 2^k двоичными векторами всех состояний в k произвольных ячейках памяти. Эта нижняя граница была получена для наилучшего случая, когда все последующие орбиты включают по $k + 1$ новому двоичному вектору, ранее не использованному в предыдущих орбитах, генерируемых MATS ++ подобными маршевыми тестами. В случае маршевых тестов типа *March C−*, учитывая, что каждая последующая итерация их применения формирует не более чем $2k$ новых двоичных векторов, Q_{min} принимает следующий вид:

$$Q_{\min} \geq \left\lceil \frac{2^{k-1}}{k} \right\rceil, \tag{5}$$

где $k = 2, 3, \ldots$

Согласно (4) для $k = 3$ минимальное число орбит $Q_{min} = 2$. Это означает, что минимальное число кратности, например, теста MATS ++ для достижения исчерпывающего покрытия в трех ячейках памяти равно двум. Действительно, пары орбит O_0, O_5; O_1, O_4; O_2, O_7 и O_3, O_6 генерируют все $2^3 = 8$ двоичных векторов для $k = 3$ (см. табл. 3).

4. Математическая модель многократного тестирования запоминающих устройств. Как было показано в предыдущих разделах, для генерирования всех 2^k возможных двоичных комбинаций в произвольных k ячейках памяти возможно применение многократного маршевого теста. Все последующие итерации применения выбранного теста в рамках многократного тестирования должны реализовываться с разными начальными состояниями и (или) различными адресными последовательностями, что является необходимым условием для обеспечения формирования новых орбит.

Предположим, что последовательность начальных состояний и адресные последовательности формируются случайным образом, а начальные состояния и адреса представляют собой случайные равномерно распределенные и независимые переменные. Тогда рассматриваемая задача определения среднего числа количества орбит Q_{ave}, которые необходимы для формирования исчерпывающего теста на произвольных k ячейках памяти [3], сводится к классической задаче теории комбинаторных вероятностей, а именно задаче собирателя купонов (Coupon Collector's Problem), когда в качестве купонов рассматривались равномерно распределенные орбиты, формируемые маршевыми тестами [3]. Предположив, что в качестве купона будет рассматриваться один двоичный вектор, а не орбита, постановка классической задачи собирателя купонов в терминах многократного тестирования запоминающих устройств может быть сформулирована следующим образом.

Определение. Применение одной итерации многократного маршевого теста генерирует один из 2^k возможных двоичных наборов для k произвольных ячеек памяти, который является равномерно распределенной случайной величиной с вероятностью $1/2^k$.

Отметим, что определение использует термин «двоичный вектор», а не термин «орбита», что, по сути, означает применение только одной фазы – инициализации произвольного маршевого теста. Эти ограничения позволяют использовать задачу собирателя купонов для описания случая изменяемых начальных состояний памяти при формировании псевдоисчерпывающих тестов на базе многократных тестов памяти. В качестве меры сложности псевдоисчерпывающего теста ранее была показана эффективность применения среднего числа кратности Q_{ave} используемого многократного теста. Эта величина Q_{ave} в случае формирования одного двоичного вектора для каждой итерации теста в соответствии с определением показывает среднее количество случайной выборки купонов (двоичных векторов) для получения всех купонов (2^k векторов) как минимум по одному разу. Таким образом, обеспечивается необходимое условие получения исчерпывающего теста [3]. Это значение вычисляется в соответствии с выражением [3, 16]

$$Q_{ave} = 1 + \frac{2^k}{2^k - 1} + \frac{2^k}{2^k - 2} + \ldots + \frac{2^k}{2} + 2^k = 2^k \sum_{n=1}^{2^k} \frac{1}{n}\ . \tag{6}$$

При больших значениях 2^k можно использовать аппроксимацию Эйлера гармонического ряда (6) в виде формулы

$$Q_{ave} = 2^k \left(\log_e 2^k + \gamma \right).$$

Величина $\gamma \approx 0{,}577\ 22$ является постоянной величиной Эйлера – Маскерони [16].

Численные значения Q_{ave} для ряда значений k даны в табл. 7.

Таблица 7

Численные значения Q_{ave}

k	2	3	4	5	6	7	8	9	10
2^k	4	8	16	32	64	128	256	512	1024
Q_{ave}	8,33	21,74	54,09	129,87	303,61	695,44	1567,83	3490,05	7689,39

Приведенные численные значения Q_{ave} показывают, что многократное применение маршевого теста с ограничениями, принятыми в определении, позволяет воспроизводить псевдоисчерпывающий тест для произвольных k ячеек памяти с умеренной временной сложностью. Среднее число Q_{ave} кратности теста принимает приемлемые значения. Например, для формирования $2^4 = 16$ двоичных векторов, необходимых для обнаружения сложных неисправностей запоминающих устройств, таких как PPSF4 [3], среднее значение кратности Q_{ave} многократного маршевого теста равно 54,09.

Следует отметить, что в рамках многократного тестирования памяти с изменяемыми начальными состояниями и (или) изменяемыми адресными последовательностями во время одной итерации теста генерируется не один двоичный вектор, а так называемая орбита, состоящая из $k + 1$ двоичных векторов. Поэтому обобщенная математическая модель, полученная для случая одного двоичного вектора на одну итерацию теста, должна быть адаптирована для реальной процедуры тестирования памяти.

5. Многократное тестирование с изменяемыми начальными состояниями. Как было показано в разд. 3, классический маршевый тест позволяет генерировать $k + 1$ различных двоичных векторов в случае тестов типа MATS ++ или $2k$ векторов в случае тестов типа *March C−*. Все двоичные векторы в обоих случаях различны и зависят от исходного начального состояния памяти, определяемого для k ячеек памяти начальным вектором P_0. Применяемый маршевый тест генерирует орбиту исходя из структуры теста и начального вектора P_0, который является случайной равномерно распределенной величиной. Для упрощения дальнейших рассуждений предположим, что в одной орбите двоичные векторы являются независимыми и равномерно распределенными случайными величинами, а вероятность формирования каждого из них равняется $1/2^k$. Основываясь на этом, предположим, что оценка Q_{ave} (6), полученная для общей математической модели, может быть адаптирована для многократных классических маршевых тестов с изменяемым начальным состоянием тестируемых запоминающих устройств. Тогда среднее значение Q_{ave}(MATS ++) кратности для многократных тестов типа MATS ++, необходимое для генерирования всех 2^k двоичных векторов, можно вычислить как Q_{ave}(MATS ++) = $= Q_{ave}/(k + 1)$. В случае тестов типа *March C−* может использоваться соотношение Q_{ave}(*March C−*) = $= Q_{ave}/(2k)$. Отметим, что независимо от применяемого теста двоичные векторы, входящие в орбиту, формируются без восстановления, т. е. не повторяются, а математическая модель задачи собирателя купонов рассматривается для случая с восстановлением. Тогда теоретические оценки кратности многократных маршевых тестов Q_{ave}(MATS ++) = $Q_{ave}/(k + 1)$ и Q_{ave}(*March C−*) = $= Q_{ave}/(2k)$ могут рассматриваться как верхние оценки: соответственно Q_{ave}(MATS ++) и Q_{ave}(*March C−*).

В табл. 8 и 9 представлены теоретические и экспериментальные оценки кратности для двух множеств многократных маршевых тестов памяти, обеспечивающих формирование всевозможных 2^k двоичных наборов для k произвольных ячеек памяти. Наряду с экспериментальной оценкой Q_{ave} были получены значения максимальной Q_{max} и минимальной кратности используемых тестов. Величина Q_{ave} представляет собой среднее значение количества итераций тестов в процентах, полученное на основании 100 000 экспериментальных значений.

Таблица 8

Теоретические и экспериментальные значения Q_{ave}(MATS++)

k	2	3	4	5	6	7	8	9	10
2^k	4	8	16	32	64	128	256	512	1024
Теор. Q_{ave}	2,77	5,44	10,82	21,64	43,37	86,93	174,20	349,00	699,03
Эксп. Q_{ave}	–	4,44	9,42	19,54	40,90	83,57	170,36	341,10	680,86
Эксп. Q_{min}	–	2	4	7	17	41	96	201	413
Эксп. Q_{max}	–	16	31	60	110	201	366	706	1405

Таблица 9

Теоретические и экспериментальные значения Q_{ave}(*March* C–)

k	2	3	4	5	6	7	8	9	10
2^k	4	8	16	32	64	128	256	512	1024
Теор. Q_{ave}	2,77	3,62	6,76	12,98	25,30	49,67	97,98	193,89	384,47
Эксп. Q_{ave}	–	2,33	4,42	9,34	19,55	40,55	82,62	169,32	339,48
Эксп. Q_{min}	–	2	2	4	7	18	37	86	213
Эксп. Q_{max}	–	7	17	28	65	113	207	391	769

Анализ приведенных выше данных позволяет сделать два основных вывода. Прежде всего численные значения показывают, что псевдоисчерпывающее тестирование, реализуемое многократными маршевыми тестами на основе изменяемого начального состояния памяти, можно описать в терминах вероятностных математических моделей, использующих геометрическое распределение. Например, применяя результаты, полученные для задачи собирателя купонов, можно оценить среднее число Q_{ave} многократных тестов, которое достаточно близко к реальным экспериментальным значениям. В случае тестов типа MATS ++ максимальная относительная погрешность 18,3 % получается для $k = 3$. С ростом k эта погрешность уменьшается, принимая значения 13,7, 9,7 и 2,5 % для $k = 4$, $k = 5$ и $k = 10$. Значения Q_{ave} среднего числа многократных тестов имеют вполне приемлемые величины, что позволяет использовать многократные тесты памяти с изменяемым начальным состоянием для реализации псевдоисчерпывающего тестирования памяти даже для случаев, когда $k = 10$. Следует подчеркнуть, что среднее значение Q_{ave} для тестов MATS ++ примерно в два раза выше по сравнению с тестами типа *March C–*. Это объясняется тем фактом, что второе множество тестов генерирует приблизительно в два раза больше двоичных векторов в произвольных k ячейках памяти.

Сравнивая полученные результаты с результатами для случая псевдоисчерпывающих тестов на базе многократных маршевых тестов с изменяемыми адресными последовательностями [3], можно отметить, что при изменении начальных состояний кратность маршевых тестов в среднем уменьшается в два раза. Например, экспериментальное среднее значение Q_{ave} для многократных тестов MATS ++ при изменении адресов без их повторения для $k = 6$ равняется 76,77, а $Q_{min} = 35$ и $Q_{max} = 214$ [3]. В случае изменяемых начальных состояний запоминающих устройств при применении того же теста для $k = 6$ имеем $Q_{ave} = 43{,}37$, $Q_{min} = 17$ и $Q_{max} = 110$ (см. табл. 8).

Заключение. В работе проведен анализ методов тестирования современных запоминающих устройств, обосновано применение многократных маршевых тестов с изменяемым начальным состоянием. Полученные аналитические оценки сложности псевдоисчерпывающих тестов запоминающих устройств подтверждены экспериментальными данными. Основным результатом данного исследования является оценка сложности реализации псевдоисчерпывающего теста для запоминающих устройств. Численные характеристики, представленные в статье, позволяют сделать вывод о реальности использования псевдоисчерпывающего теста для современных запоминающих устройств. Применение не более 50 итераций простейшего маршевого теста $\{\uparrow(w0);\ \uparrow(r0,\ w1,\ r1)\}$ даст возможность осуществить исчерпывающее тестирование любого подмножества из $k < 6$ ячеек памяти. Использование любого другого маршевого теста с большим числом разнообразных фаз позволит достичь лучших результатов. Действительно, исчерпывающий тест памяти для $k < 6$ обеспечивается тестом *March C–* в среднем за 25,3 итераций (см. табл. 9).

Список использованных источников

1. Barzilai, Z. Exhaustive Generation of Bit Pattern with Application to VLSI Self-Testing / Z. Barzilai, D. Coppersmith, A. Rozenberg // IEEE Transactions on Computers. – 1983. – Vol. C–31, no. 2. – P. 190–194.
2. Das, D. Exhaustive and Near-Exhaustive Memory Testing Techniques and their BIST Implementations / D. Das, M. G. Karpovsky // Journal of Electronic Testing. − 1997. – Vol. 10. − P. 215−229.
3. Ярмолик, В. Н. Псевдоисчерпывающее тестирование ОЗУ / В. Н. Ярмолик, И. Мрозек, В. А. Леванцевич // Информатика. – 2017. – № 2(54). – С. 58–69.
4. Ярмолик, С. В. Итеративные почти псевдоисчерпывающие вероятностные тесты / С. В. Ярмолик, В. Н. Ярмолик // Информатика. – 2010. – № 2(26). − С. 66−75.
5. Mrozek, I. Method for Generation Multiple Controlled Random Tests / I. Mrozek, V. Yarmolik // Proc. of the Computer Information Systems and Industrial Management (CISIM 2016), 14–16 September 2016. – Vilnius, Lithuania, 2016. – P. 429–440.
6. Mrozek, I. Iterative Antirandom Testing / I. Mrozek, V. Yarmolik // Journal of Electronic Testing: Theory and Applications (JETTA). – 2012. – Vol. 9, no. 3. – P. 251–266.
7. Yarmolik, S. V. The Syntheses of Probability Tests with a Small Number of Kits / S. V. Yarmolik, V. N. Yarmolik // Automatic Control and Computer Science. – 2011. − Vol. 45, no. 3. − P. 133−141.
8. Yarmolik, V. N. Address Sequences for Multiple Run March Tests / V. N. Yarmolik, S. V. Yarmolik // Automatic Control and Computer Sciences. – 2006. – No. 5. – C. 59–68.
9. Mrozek, I. Antirandom Test Vectors for Bist in Hardware/Software Systems / I. Mrozek, V. N. Yarmolik // Fundamenta Informaticae. – 2012. – No. 119. – P. 1–23.

10. 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.

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

12. 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, 24–25 August 1998. – Washington, DC, USA, 1998. – P. 91–98.

13. Mrozek, I. Optimal Backgrounds Selection for Multi Run Memory Testing / I. Mrozek, V. N. Yarmolik // Proc. of the 11th IEEE Workshop on Design and Diagnostic Circuits and Systems (DDECS 2008). – Bratislava, Slovakia, 2008. – P. 1–7.

14. Karpovsky, M. G. Transparent Random Access Memory Testing for Pattern Sensitive Faults / M. G. Karpovsky, V. N. Yarmolik // J. Electron. Testing: Theory and Applications (JETTA). – 1994. – Vol. 5, no. 1. – P. 91–113.

15. Karpovsky, M. G. Transparent Memory Testing for Pattern Sensitive Faults / M. G. Karpovsky, V. N. Yarmolik // Proc. Intern. Test Conf. IEEE Publisher. – Washington, DC, USA, 1994. – P. 860–869.

16. Flajolet, P. Birthday Paradox, Coupon Collectors, Caching Algorithms and Self-Organizing Search / P. Flajolet, D. Gardy, L. Thimonier // Discrete Appl. Math. – 1992. – No. 39. – P. 207–229.

Информация об авторах

Ярмолик Вячелав Николаевич – доктор технических наук, профессор, Белорусский государственный университет информатики и радиоэлектроники (ул. П. Бровки, 6, Минск, Республика Беларусь). E-mail: yarmolik10ru@yahoo.com

Мрозек Иренеуш – доктор, адъюнкт, Белостокский технический университет (ул. Вейска, 45А, 15-351, Белосток, Польша). E-mail: i.mrozek@pb.edu.pl

Леванцевич Владимир Александрович – магистр технических наук, ассистент, Белорусский государственный университет информатики и радиоэлектроники (ул. П. Бровки, 6, Минск, Республика Беларусь). E-mail: lvn@bsuir.by

Information about the authors

Vyacheslav N. Yarmolik – D. Sc. (Engineering), Professor, Belarusian State University of Informatics and Radioelectronics (6, P. Brovka Str., 220013, Minsk, Republic of Belarus). E-mail: yarmolik10ru@yahoo.com

Ireneusz Mrozek – Ph. D., Lecture, Bialystok University of Technology (45A 15-351, Wiejska Str., Białystok, Poland). E-mail: i.mrozek@pb.edu.pl

Vladimir A. Levantsevich – Master of Engineering, Assistant, Belarusian State University of Informatics and Radioelectronics (6, P. Brovka Str., 220013, Minsk, Republic of Belarus). E-mail: lvn@bsuir.by