ИНФОРМАТИКА

2017 апрель-июнь № 2

УДК 004.33.054

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

ПСЕВДОИСЧЕРПЫВАЮЩЕЕ ТЕСТИРОВАНИЕ ОЗУ

Анализируются методы тестирования современных запоминающих устройств, в том числе оперативных запоминающих устройств (ОЗУ), обосновывается применение псевдоисчерпывающих тестов для обнаружения сложных неисправностей памяти. Формулируется необходимое условие генерирования псевдоисчерпывающего теста для заданного количества запоминающих ячеек ОЗУ. Показывается, что задача генерирования псевдоисчерпывающего теста на базе многократных тестов ОЗУ с изменяемыми адресными последовательностями сводится к комбинаторной задаче собирателя купонов. Приводятся оценки минимальной, максимальной и средней кратности многократного теста для обеспечения исчерпывающего множества комбинаций для заданного числа ячеек ОЗУ, что подтверждает возможность формирования псевдоисчерпывающего теста для заданного числа ячеек ОЗУ.

Введение

Исчерпывающее тестирование (Exhaustive Testing) характеризуется максимальной эффективностью обнаружения неисправностей цифровых устройств и ошибок программного обеспечения [1−4]. Однако в силу большой временной сложности подобных тестов в настоящее время используются их различные аппроксимации и в первую очередь такие, как псевдоисчерпывающие тесты (Pseudoexhaustive Tests) [5−6]. Подобные тесты являются реальной альтернативой исчерпывающим тестам. Они основываются на формировании множества тестовых наборов (векторов), обеспечивающих всевозможные комбинации на заданном подмножестве элементов наборов теста, и называются локально исчерпывающими [1]. Чаще всего рассматриваются двоичные псевдослучайные тестовые наборы, которые формируют 2^k возможные двоичные комбинации для любых k из N разрядов тестовых наборов. Характерной особенностью псевдоисчерпывающих тестов является то, что их сложность, определяемая количеством тестовых наборов, существенно меньше сложности исчерпывающих тестов.

Вероятностное тестирование (Random Testing) и его многочисленные модификации, основанные на принципе черного ящика, также являются эффективной аппроксимацией исчерпывающего тестирования [1, 7]. Существующие модификации классического метода построения вероятностных тестов [1, 4−9] объединяются по принципу управления процедурой формирования очередного тестового набора [1]. Такие виды формирования модифицированных вероятностных тестов, как антивероятностные, быстрые антивероятностные, адаптивные, эффективные, зеркальные, упорядоченные, эволюционные, управляемые и др., основаны на вычислении некоторых характеристик для управляемого формирования очередного случайного тестового набора [1, 7, 9].

Существенным недостатком управляемых вероятностных тестов выступает необходимость перебора потенциальных кандидатов в тестовые наборы и вычисления для них характеристик, являющихся критериями для включения либо невключения их в вероятностный тест, что увеличивает вычислительную сложность формирования подобных тестов [1]. С целью уменьшения такой сложности широко обсуждаются и используются итеративные вероятностные тесты [1, 6], исчерпывающие и почти псевдоисчерпывающие вероятностные тесты [1, 10–11], вероятностные тесты с малым числом наборов [1, 4, 12], квазивероятностные [8, 13], а также многократные тесты для запоминающих устройств [14, 15].

Главным достоинством указанных разновидностей вероятностных тестов является использование некоторой обобщающей характеристики для теста в целом, а не для тестового набора в отдельности. Это позволяет значительно уменьшить вычислительную сложность построения подобных тестов. Кроме того, в ряде случаев при проектировании многократных тестов, в частности многократных маршевых тестов запоминающих устройств, не требуется вычисление каких-либо характеристик, построение таких тестов основывается на реализации предварительно определенной процедуры [8, 13–15].

Целью настоящей статьи является анализ эффективности псевдоисчерпывающего тестирования современных ОЗУ, основанного на многократном применении классических маршевых тестов с изменяемыми адресными последовательностями. В качестве основной характеристики маршевого теста ОЗУ для k произвольных запоминающих ячеек ОЗУ используется подмножество двоичных комбинаций, названное орбитой, которое формируется одной из фаз маршевого теста. Приводятся основные аналитические соотношения, а также оценки сложности теста для обеспечения всевозможных 2^k двоичных комбинаций в произвольных k из N запоминающих ячеек ОЗУ.

1. Многократные маршевые тесты ОЗУ

Ранее было показано [16], что при однократном применении маршевых тестов полнота покрытия сложных неисправностей ОЗУ, равно как и любых других неисправностей, остается неизменной. Отличием для разных версий конкретного маршевого теста являются только конфигурации неисправностей, в частности кодочувствительных, которые обнаруживаются либо не обнаруживаются при реализации данного теста для измененного содержимого ОЗУ и заданной последовательности адресов [4, 6, 14, 16, 17].

В качестве обобщающих (доминирующих) неисправностей, которые обнаруживаются маршевыми тестами, рассматривается модель кодочувствительных неисправностей PSFk (Pattern Sensitive Faults) или ее разновидности, определяемые как NPSFk (Neighborhood Pattern Sensitive Faults) и эффективно покрывающие более простые модели неисправностей ОЗУ [16]. Наиболее адекватной моделью, как показано в ряде источников [14, 16], являются неисправности PNPSFk, их количество для произвольных k из N ячеек ОЗУ и фиксированной базовой ячейки равняется $2\times 2^{k-1} = 2^k$, а общее число определяется выражением

$$Q(\text{PNPSF}k) = 2k2^{k-1}\binom{N}{k} = k2^k\binom{N}{k}. \tag{1}$$

Изменение содержимого ОЗУ, например, при его обновлении в результате вычислений позволяет существенно увеличить полноту покрытия при повторном применении маршевого теста [14, 17]. Для повышения эффективности повторного применения маршевых тестов радикальным подходом может быть изменение последовательности адресов, используемых в каждом последующем маршевом тесте [17].

Для оценки влияния изменения последовательности адресов на обнаруживающую способность многократного маршевого теста были проведены экспериментальные исследования ОЗУ емкостью N = 64 бит, состояние которого не изменялось для трех последовательно применяемых тестов. Тесты отличались только используемой адресной последовательностью. В трех последовательных маршевых тестах MATS+ и March C– применялись случайные последовательности адресов ячеек ОЗУ. В результате полнота покрытия при повторном и трехкратном применении маршевых тестов заметно возрастает, что подтверждают данные табл. 1 [14]. Здесь представлены результаты для трех кодочувствительных неисправностей PNPSFk, для каждой из которых получена суммарная полнота покрытия как результат последовательных процедур маршевого теста ОЗУ.

Таблица 1

Полнота покрытия для случайных последовательностей адресов, %

Тип теста	PPSF3			PPSF4			PPSF5		
MATS+	24,90	42,74	55,83	12,48	22,61	30,95	6,31	11,87	16,91
March C–	49,87	72,74	83,04	24,93	41,42	53,23	12,46	22,14	30,02

Из табл. 1 видно, что с ростом кратности применения маршевого теста растет полнота покрытия кодочувствительных неисправностей ОЗУ за счет увеличения двоичных комбинаций, формируемых маршевым тестом в произвольных k из N ячеек ОЗУ, что достигается за счет изменения последовательностей адресов. Отметим, что 100%-ная полнота покрытия рассмотренных кодочувствительных неисправностей для произвольных k из N ячеек ОЗУ будет обеспечена исчерпывающим тестом для этих ячеек. Генерирование исчерпывающих тестов для любых k из N ячеек ОЗУ достигается применением псевдоисчерпывающего теста ОЗУ для заданного k.

2. Адресные последовательности

Адресные последовательности находят широкое применение в различных приложениях вычислительной техники. В первую очередь это касается алгоритмов цифровой обработки сигналов, где встречаются довольно сложные типы адресации, примером которых может служить двоично-инверсная адресация данных [18]. Адресные последовательности интенсивно применяются и в современных методах тестирования как аппаратных, так и программных средств вычислительной техники [19], различного рода запоминающих устройств [14, 16], а также при реализации методов управляемого вероятностного [4–7] и квазислучайного [8] тестирования современных вычислительных систем.

Многократное применение маршевых тестов с изменяемыми адресными последовательностями является одним из радикальных методов повышения эффективности тестирования современных ОЗУ [14]. Для этой цели используются различные меры отличия адресных последовательностей (например, расстояние Хэмминга, арифметическое расстояние, манхэттенское расстояние, корреляционные зависимости), а также переключательная активность адресных последовательностей [4, 8, 12, 14].

Под адресной последовательностью понимают упорядоченную последовательность m-разрядных двоичных векторов $A(n) = a_{m-1}a_{m-2}a_{m-3}\ldots a_2a_1a_0$, где $a_i \in \{0,1\}$, $i \in \{0, 1, 2, \ldots, m-1\}$, принимающих всевозможные значения из множества 2^m двоичных векторов $\{000\ldots00, 000\ldots01, 000\ldots10, \ldots, 111\ldots11\}$ [14]. Значения адресов $A(n)$, так же как и их индексов n, принадлежат множеству $\{0, 1, 2, \ldots, 2^m-1\}$, т. е. $A(n), n \in \{0, 1, 2, \ldots, 2^m-1\}$. В качестве базовой адресной последовательности чаще всего используют счетную последовательность адресов, для которой выполняется свойство $A(n) = n$, т. е. $A(0) = 0$, $A(1) = 1$, $A(2) = 2$, …, $A(2^m-1) = 2^m-1$. Невыполнение данного свойства и приводит к многообразию различных адресных последовательностей [14]. Общее их количество, равное $2^m!$, для заданного значения m и адресного пространства, состоящего из 2^m адресов, принимает астрономические значения. Среди всевозможных адресных последовательностей выделяют четыре основных их подмножества: детерминированные, случайные, псевдослучайные и квазислучайные [8, 14].

Произвольная детерминированная, псевдослучайная или квазислучайная адресные последовательности удовлетворяют следующему основному свойству [14].

Свойство. *Адресная последовательность* $A(n) = a_{m-1}a_{m-2}\ldots a_2a_1a_0$, $n \in \{0, 1, 2, \ldots, 2^m-1\}$, *состоит из произвольной последовательности всевозможных* 2^m *адресов (двоичных комбинаций* $a_{m-1}\, a_{m-2} \ldots a_2a_1a_0$*), причем каждый адрес используется в такой последовательности только один раз.*

Для случайных адресных последовательностей различают два принципа их формирования, а именно генерирование случайных адресов (выборку с восстановлением) и генерирование неповторяющихся случайных адресов (выборку без восстановления). В первом случае адреса могут повторяться, во втором случае повторение адресов невозможно, что соответствует основному свойству.

3. Адресные последовательности многократных маршевых тестов ОЗУ

В качестве анализируемого ОЗУ будем рассматривать запоминающее устройство, состоящее из $N = 2^m$ однобитных запоминающих ячеек. Для тестирования подобных устройств, как правило, используются только маршевые тесты как единственно возможное решение для тестирования современных ОЗУ большой емкости [16]. Общая структура маршевых тестов состоит в последовательном обращении ко всем запоминающим ячейкам ОЗУ путем генерирования их адресов. Для каждой ячейки в соответствии с конкретным маршевым тестом выполняются операции чтения (r) логического нуля ($r0$) либо единицы ($r1$), а также операции записи (w) нуля либо единицы ($w0$), ($w1$). Количество операций чтения и записи, их порядок в каждой фазе теста, а также количество фаз, их структура и последовательность определяют вид маршевого теста [16]. Под фазой (элементом) маршевого теста, например $\Uparrow(r0, w1)$, понимают последовательное обращение ко всем ячейкам ОЗУ по возрастающим адресам ($\Uparrow$) с выполнением операций чтения и записи либо по убывающим адресам ($\Downarrow$). Возрастание адресов означает произвольную последовательность адресов, а убывание – их обратный порядок.

Отметим, что для современных ОЗУ, емкость которых существенно превышает 2^{30}, рассматриваются PNPSFk, как правило, для k, равных 3, 5 и 9, и других небольших значений $k \leq 10$ [10, 14, 16]. Необходимым условием обнаружение всего множества подобных неисправностей является обеспечение тестом ОЗУ всевозможных 2^k двоичных комбинаций в любых k из N ячеек ОЗУ. Очевидно, что данное условие может быть обеспечено в рамках исчерпывающих тестов, их сложность определяется величиной 2^N, где $N = 2^m$ – емкость ОЗУ, которая является большой величиной. Поэтому на практике применяются различные подходы, позволяющие аппроксимировать исчерпывающий тест. Другими словами, применяется тест либо тестовая процедура, сложность которых существенно меньше сложности исчерпывающего теста, а их обнаруживающая способность достаточно высокая [16].

Рассмотрим пример формирования двоичных комбинаций для $k = 6$ из N произвольных ячеек ОЗУ. Предположим, что фиксированными $k = 6$ ячейками ОЗУ являются ячейки b_α, b_β, b_χ, b_δ, b_ε, b_ϕ, где $b_\gamma \in \{0, 1\}$ для $\gamma \in \{\alpha, \beta, \chi, \delta, \varepsilon, \phi\}$ представляет текущее состояние ячеек, а их адреса α, β, χ, δ, ε, ϕ расположены в возрастающей последовательности ($\alpha < \beta < \chi < \delta < \varepsilon < \phi$). Для текущего состояния $b_\alpha b_\beta b_\chi b_\delta b_\varepsilon b_\phi = 000000$ в указанных ячейках при реализации первой фазы $\Uparrow(r0, w1)$ теста MATS+ получим состояния (табл. 2), которые называются орбитой [14].

Таблица 2

Состояния ячеек ОЗУ

Фаза теста MATS+	Содержимое ячеек					
	b_α	b_β	b_χ	b_δ	b_ε	b_ϕ
	0	0	0	0	0	0
$\Uparrow(r0,w1)$	**1**	0	0	0	0	0
	1	**1**	0	0	0	0
	1	1	**1**	0	0	0
	1	1	1	**1**	0	0
	1	1	1	1	**1**	0
	1	1	1	1	1	**1**

Приведенный пример реализации одной фазы маршевого теста с возрастающей последовательностью адресов показывает, что в рассматриваемых b_α, b_β, b_χ, b_δ, b_ε и b_ϕ ячейках ОЗУ были сформированы 7 из $2^6 = 64$ двоичных комбинаций. Здесь учитывалась исходная нулевая комбинация и шесть состояний, полученных в результате применения фазы $\Uparrow(r0, w1)$ маршевого теста. Используя фазу $\Downarrow(r0, w1)$ с обратной адресной последовательностью (убывающей от ϕ, ε, … до α) для тех же ячеек ОЗУ при нулевом начальном их состоянии, получим пять новых состояний: 000001, 000011, 000111, 001111, 011111. Выбрав адресную последовательность β, α, χ, δ, ε, ϕ для реализации фазы $\Uparrow(r0, w1)$ и применив те же нулевые начальные условия, получим следующее состояние запоминающих ячеек: 010000, 110000, 111000, 111100, 111110, 111111. Видно, что только одно состояние (010000) отличается от ранее полученных (табл. 2).

Рассмотренный пример позволяет сформулировать несколько общих положений для произвольных k из N ячеек ОЗУ. Приведем их для случая нулевого начального состояния ОЗУ и одного маршевого элемента, который последовательно инвертирует состояние ячеек ОЗУ. Представим указанные положения в виде утверждений.

Утверждение 1. *Однократное применение инвертирующего маршевого элемента позволяет получить* $k+1$ *двоичную комбинацию для произвольных* k *из* N *ячеек ОЗУ.*

Утверждение 2. *Применение последующего инвертирующего маршевого элемента с другой последовательностью адресов для* k *из* N *ячеек ОЗУ позволяет сформировать в них новые двоичные комбинации, минимальное количество которых равняется одной комбинации, а максимальное –* $k-1$.

3. Анализ сложности псевдоисчерпывающего теста ОЗУ

В качестве адресных последовательностей будем рассматривать только случайные адресные последовательности, каждая из которых формирует одну из неповторяющихся последовательностей адресов ОЗУ. В качестве маршевого теста используем простейший тест $\{\Uparrow(w0); \Uparrow(r0, w1)\}$, выполняющий инвертирование нулевого содержимого ОЗУ. Процедура тестирования ОЗУ будет состоять в многократном применении приведенного теста. При этом возможны два варианта генерирования случайным образом очередного адреса ячейки ОЗУ: с восстановлением либо без восстановления. Для адресов выбранных k ячеек ОЗУ так же, как и для адресной последовательности в целом, возможны выборка без восстановления и выборка с восстановлением. В первом случае повторение адресов невозможно, а во втором – адреса могут повторяться. В качестве примера рассмотрим множество на рис. 1.

1		2	
0	0	0	0
1	0	0	1
1	1	1	1

Рис. 1. Множество орбит для $k = 2$

Видно, что для $k = 2$ существуют только две орбиты. Исчерпывающий тест, т. е. генерирование всевозможных комбинаций (00, 01, 10, 11) из двух бит, обеспечивается двумя орбитами (1–2) в любой их последовательности. С увеличением значения k до трех получим шесть различных орбит (рис. 2).

1			2			3			4			5			6		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1
1	1	0	1	0	1	1	1	0	0	1	1	1	0	1	0	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Рис. 2. Множество орбит для $k = 3$

Исчерпывающий тест для $k = 3$ обеспечивается орбитами 1–4–5 и 2–3–6. Следует отметить, что число орбит три является минимальным для обеспечения исчерпывающего теста, состоящего из восьми двоичных комбинаций (000, 001, 010, …, 111).

В качестве второго примера можно привести множество, которое состоит из пяти орбит (1–2–3–5–6). Это множество орбит также обеспечивает исчерпывающий тест, причем множество указанных орбит является неизбыточным.

Общее количество орбит для произвольного значения k определяется как $Q_{\text{tou}} = k!$ [14]. Число Q_{tou}, например, для $k = 4$ будет $4! = 24$ (рис. 3).

1	2	3	4	5	6
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1 0 0 0	1 0 0 0	1 0 0 0	1 0 0 0	1 0 0 0	1 0 0 0
1 1 0 0	1 0 1 0	1 0 0 1	1 1 0 0	1 0 1 0	1 0 0 1
1 1 1 0	1 1 1 0	1 1 0 1	1 1 0 1	1 0 1 1	1 0 1 1
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
7	8	9	10	11	12
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 1 0 0	0 1 0 0	0 1 0 0	0 1 0 0	0 1 0 0	0 1 0 0
1 1 0 0	0 1 1 0	0 1 0 1	1 1 0 0	0 1 1 0	0 1 0 1
1 1 1 0	1 1 1 0	1 1 0 1	1 1 0 1	0 1 1 1	0 1 1 1
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
13	14	15	16	17	18
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0
1 0 1 0	0 1 1 0	0 0 1 1	1 0 1 0	0 1 1 0	0 0 1 1
1 1 1 0	1 1 1 0	1 0 1 1	1 0 1 1	0 1 1 1	0 1 1 1
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
19	20	21	22	23	24
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
1 0 0 1	0 1 0 1	0 0 1 1	1 0 0 1	0 1 0 1	0 0 1 1
1 1 0 1	1 1 0 1	1 0 1 1	1 0 1 1	0 1 1 1	0 1 1 1
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1

Рис. 3. Множество орбит для $k = 4$

Для данного случая, используя, например, орбиты 1–6–8–9–16–24, можно обеспечить исчерпывающий тест для четырех ячеек ОЗУ. Такое множество орбит является не единственным множеством, состоящим из шести орбит и обеспечивающим исчерпывающее множество четырехразрядных двоичных векторов. Однако применение меньшего чем шесть количества орбит для $k = 4$ не позволяет получить исчерпывающий тест. Как и для $k = 3$, исчерпывающий тест обеспечивается множеством орбит, состоящим не менее чем из трех орбит. В то же время для $k = 4$ множество орбит 1–2–5–6–8–9–13–14–15–16–17–23 является неизбыточным и обеспечивает все возможные двоичные комбинации из четырех бит.

Определим предельные значения количества орбит, необходимых для реализации исчерпывающего теста для произвольного значения k. Сразу рассмотрим случай выборки адресов ячеек ОЗУ без восстановления. Это означает, что формирование конкретной орбиты будет осуществляться путем случайного генерирования адресов ячеек, входящих в орбиту с повторением (выборка с восстановлением).

Вначале наложим ограничение на повторяемость орбит, т. е. рассмотрим случай выборки орбит без восстановления. Максимальная оценка $Q_{\max}$ количества орбит, необходимых для формирования исчерпывающего теста, содержит 2^k тестовых набора. Как следует из утверждения 2, применение одной орбиты независимо от ее вида обеспечивает $k + 1$ двоичных векторов, каждый из которых состоит из k бит. Напомним, что для каждой орбиты используется начальное значение, как правило, нулевое; оно записывается в выбранные k ячейки. В худшем случае каждая последующая орбита будет обеспечивать как минимум один новый двоичный вектор по отношению к множеству векторов, сгенерированных ранее использованными орбитами для случая выборки орбит без восстановления.

Тогда максимальная оценка $Q_{\max}$ будет вычисляться как

$$Q_{\max} = 1 + (2^k - (k+1)) = 2^k - k. \tag{2}$$

Полученная оценка $Q_{\max}$ является нижней ее границей для случая выборки орбит с восстановлением, т. е. когда орбиты могут повторяться.

Минимальное количество орбит $Q_{\min}$ для формирования исчерпывающего теста определяется соотношением

$$Q_{\min} = \binom{k}{\lceil k/2 \rceil}. \tag{3}$$

Здесь принимается во внимание тот факт, что каждая из орбит формирует только один набор (комбинацию), который состоит из определенного количества единиц и нулей. Так, например, для $k = 4$ вторая орбита (см. рис. 3)

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$$

формирует один набор (комбинацию), состоящий из одной единицы и трех нулей (1000), один набор из двух единиц и двух нулей (1010) и одну комбинацию их трех единиц и одного нуля (1110). Для случая $k = 4$ число наборов из одной единицы и трех нулей так же, как и число наборов из трех единиц и одного нуля, равняется четырем, а для случая двух единиц и двух нулей – шести.

Отметим, что максимальное количество двоичных комбинаций образуют двоичные векторы из k нулевых и единичных двоичных значений, для которых число единиц и нулей одинаково в случае четных величин k и отличается на единицу для нечетных значений k.

Фактически, чтобы обеспечить исчерпывающий тест для k ячеек ОЗУ, необходимо генерирование всех двоичных векторов из $\lceil k/2 \rceil$ единиц и $\lfloor k/2 \rfloor$ нулей, количество которых максимально по сравнению с другими комбинациями нулей и единиц. Это следует из свойства числа сочетаний из k по r для $k > 1$, $0 < r < k$:

$$\binom{k}{\lceil k/2 \rceil} > \binom{k}{r}.$$

Таким образом, $Q_{\min}$ и будет равняться количеству двоичных векторов из $\lceil k/2 \rceil$ единиц и $\lfloor k/2 \rfloor$ нулей (3). Такое же количество будет и для случая $\lfloor k/2 \rfloor$ единиц и $\lceil k/2 \rceil$ нулей, для которых $\lfloor k/2 \rfloor + \lceil k/2 \rceil = k$, что следует из свойства числа сочетаний

$$\binom{k}{\lceil k/2 \rceil} = \binom{k}{k - \lceil k/2 \rceil}.$$

Оценим среднее число Q_{ave} орбит, которое необходимо для формирования исчерпывающего теста на произвольных k ячейках ОЗУ. Оценку получим для случая выборки орбит с восстановлением. Для этого нужно проанализировать всевозможные сочетания орбит, которые обеспечат 2^k тестовых набора в k ячейках ОЗУ, что даже для малых значений k является затруднительным в силу большого числа орбит, равного $k!$. Поэтому, используя результаты, полученные при выводе оценки $Q_{\min}$, сформулируем необходимое условие генерирования исчерпывающего теста на произвольных k ячейках ОЗУ в виде следующего утверждения.

Утверждение 3. *Необходимым условием для формирования исчерпывающего теста в произвольных k из N ячеек ОЗУ является генерирование для этих ячеек множества орбит, в которых все двоичные коды, содержащие $\lfloor k/2 \rfloor$ единиц и $k - \lfloor k/2 \rfloor$ нулей, будут сгенерированы хотя бы по одному разу.*

Количество таких кодов в зависимости от k определяется величиной $Q_{\min}$. Для небольших величин k их значения приведены в табл. 3.

Таблица 3

Численные значения Q_{tou}, $Q_{\max}$ и $Q_{\min}$

k	2	3	4	5	6	7	8	9	10
Q_{tou}	2	6	24	120	720	5040	40 320	362 880	3628 800
$Q_{\min}$	2	3	6	10	20	35	70	126	252
$Q_{\max}$	3	5	12	27	58	121	248	503	1014

Например, как следует из табл. 3, для $k = 4$ необходимым является формирование такого подмножества орбит, в которых будут присутствовать все 6 двоичных кодов (0011, 0101, 1001, 0110, 1010, 1100), а для $k = 6$ требуется сгенерировать орбиты, содержащие все 20 двоичных кодов вида (000111, 001011, 010011, …, 111000). Отметим, что конкретная орбита для произвольного k содержит только один двоичный код, который включает $\lfloor k/2 \rfloor$ единиц и $k - \lfloor k/2 \rfloor$ нулей. Указанные коды присутствуют одинаковое число раз во всевозможных орбитах для заданного значения k (см. рис. 1–3). Таким образом, при равновероятной выборке одной из орбит с вероятностью $1/Q_{\min}$ будет выбран один из $Q_{\min}$ кодов, который включает $\lfloor k/2 \rfloor$ единиц и $k - \lfloor k/2 \rfloor$ нулей.

В подобной интерпретации рассматриваемая задача определения среднего числа количества орбит Q_{ave}, которые необходимы для формирования исчерпывающего теста на произвольных k ячейках ОЗУ, по мнению авторов, сводится к классической задаче собирателя купонов (Coupon Collector’s Problem) [20]. Для случая одной коллекции равновероятных купонов ($Q_{\min}$ орбит) среднее количество Q_{ave} случайной выборки купонов (орбит) для получения всех купонов как минимум по одному разу (обеспечения необходимого условия получения исчерпывающего теста, см. утверждение 3) определяется согласно соотношению [20]

$$Q_{\text{ave}} = 1 + \frac{Q_{\min}}{Q_{\min} - 1} + \frac{Q_{\min}}{Q_{\min} - 2} + \ldots + \frac{Q_{\min}}{2} + Q_{\min} = Q_{\min} \sum_{n=1}^{Q_{\min}} \frac{1}{n} \,. \tag{4}$$

Для больших значений $Q_{\min}$ можно использовать формулу аппроксимации Эйлера гармонической серии (4):

$$Q_{\text{ave}} = Q_{\min} (\log_e Q_{\min} + \gamma). \tag{5}$$

Значение $\gamma \approx 0{,}577\,22$ представляет собой константу Эйлера – Маскерони [20]. Численные значения Q_{ave} приведены в табл. 4.

Таблица 4

Численные значения Q_{ave}

k	2	3	4	5	6	7	8	9	10
$Q_{\min}$	2	3	6	10	20	35	70	126	252
Q_{ave}	3	5,50	14,70	29,29	71,95	145,14	338,06	683,52	1538,58

Анализ величин Q_{ave} показывает, что для обеспечения исчерпывающего множества двоичных комбинаций для k из N произвольных ячеек ОЗУ и, соответственно, реализации псевдоисчерпывающего теста ОЗУ для заданного k необходимо использовать многократный тест ОЗУ с изменяемыми адресными последовательностями, среднее значение кратности которого принимает приемлемые значения. Например, для обнаружения всего множества сложных неисправностей PPSF5 среднее значение кратности теста равняется 29,29, что гарантирует их 100%-ное обнаружение. Отметим, что данное утверждение справедливо для случая генерирования адресов без повторений и простейшего теста ОЗУ типа $\{\Uparrow(w0); \Uparrow(r0, w1)\}$.

4. Экспериментальный анализ псевдоисчерпывающих тестов ОЗУ

В качестве меры эффективности псевдоисчерпывающих тестов ОЗУ была выбрана ее средняя сложность Q_{ave} , определяемая средним количеством тестов в многократных тестах ОЗУ с изменяемыми адресными последовательностями. При оценке данной величины в разд. 3 приведен ряд ограничений и допущений, что позволило аналитически оценить ее значение, а также получить оценки значений Q_{min} и Q_{max} (см. табл. 3 и 4).

Для экспериментального подтверждения полученных аналитических результатов был проведен статистический анализ среднего значения Q_{ave} многократного применения теста $\{\Uparrow(w0);\ \Uparrow(r0,\ w1)\}$ с изменяемыми адресными последовательностями. В первом случае адреса ОЗУ генерировались без повторений, что несколько усложняло их формирование, а во втором – с повторениями. В обоих случаях средние оценочные значения Q_{ave} были получены на основании 10 000 экспериментов, а их конкретные величины приведены соответственно в табл. 5 и 6.

Таблица 5

Экспериментальные численные значения Q_{ave} для случая без повторения адресов

k	2	3	4	5	6	7	8	9	10
Q_{min}	2	3	6	14	31	79	197	499	998
Q_{ave}	2,99	6,69	15,53	35,02	77,21	168,06	362,96	762	1634,98
Q_{max}	14	32	66	122	242	499	898	1850	3037

Таблица 6

Экспериментальные численные значения Q_{ave} для случая с повторением адресов

k	2	3	4	5	6	7	8	9	10
Q_{min}	2	3	6	15	37	98	235	559	1267
Q_{ave}	3,96	9,14	20,50	45,69	98,89	211,60	446,86	950,15	1987,92
Q_{max}	21	44	78	183	300	5669	1364	2466	4135

Графически значения Q_{ave} для двух случаев формирования случайных адресов представлены на рис. 4.

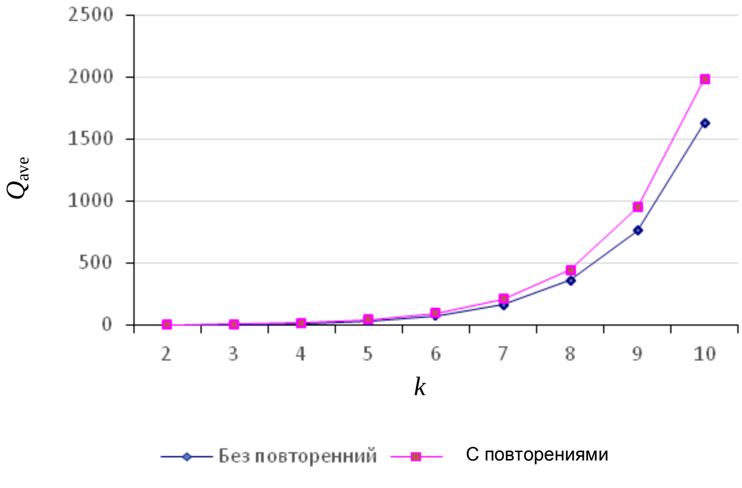

Рис. 4. Экспериментальные значения Q_{ave}

Анализ приведенных результатов показывает, что аналитическая оценка среднего количества маршевых тестов (4) подтверждается данными из табл. 5 и 6, а также рис. 4. Аналитические значения Q_{ave}, по сути, являются нижней оценкой средней сложности многократного применения простейшего маршевого теста вида $\{\Uparrow(w0);\ \Uparrow(r0,\ w1)\}$ для реализации псевдоисчерпывающего теста ОЗУ для заданного значения k. Подобный псевдоисчерпывающий тест ОЗУ – это результат многократного применения маршевого теста с изменяемыми адресными последовательностями. Аналогичный результат будет получен для любого MATS+ подобного теста, который характеризуется формированием орбит, приведенных для частных случаев значений k на рис. 1–4. В табл. 7 представлены экспериментальные результаты для теста MATS+ и адресных последовательностей без восстановления.

Очевидно, что использование классического маршевого теста, такого как March C–, позволит обеспечить псевдоисчерпывающее тестирование за меньшее число итераций. Это подтверждается экспериментальными результатами (табл. 8).

Таблица 7

Численные значения Q_{ave} для теста MATS+ без повторения адресов

k	2	3	4	5	6	7	8	9	10
Q_{min}	2	3	6	15	37	98	235	559	1267
Q_{ave}	3,96	9,14	20,50	45,69	98,89	211,60	446,86	950,15	1987,92
Q_{max}	21	44	78	183	300	5669	1364	2466	4135

Таблица 8

Численные значения Q_{ave} для теста March C– без повторения адресов

k	2	3	4	5	6	7	8	9	10
Q_{min}	1	2	3	6	12	33	82	185	469
Q_{ave}	1	2,5	5,88	13,79	31,93	72,3	157	345	730
Q_{max}	1	8	17	49	89	181	361	853	1619

Как отмечалось ранее, все множество маршевых тестов ОЗУ, применяемых для тестирования, основано на использовании последовательной выборки адресов ячеек ОЗУ, в том числе и случайной с повторяющимися адресами и без повторяющихся адресов. Во втором случае активно используются генераторы псевдослучайных последовательностей, которые могут гарантировать процедуру выборки без восстановления. В то же время неповторяющиеся адресные последовательности для всего ОЗУ могут формировать повторяющиеся адресные последовательности и, соответственно, одинаковые орбиты для произвольных k из N ячеек ОЗУ, где значение k существенно меньше емкости ОЗУ. Последнее утверждение и иллюстрируется близостью результатов эксперимента для двух способов формирования случайных адресных последовательностей (см. рис. 4).

Заключение

Проведен анализ способов тестирования современных ОЗУ. Обосновано применение многократных маршевых тестов с изменяемыми адресными последовательностями. Основным результатом данной статьи является оценка сложности реализации псевдоисчерпывающего теста для ОЗУ. Полученные аналитические оценки сложности псевдоисчерпывающих тестов ОЗУ подтверждены экспериментальными результатами. Приведенные численные характеристики позволяют сделать вывод о реальности применения псевдоисчерпывающего теста для современных ОЗУ. Как оказалось, применение не более 100 итераций простейшего маршевого теста $\{⇑(w0);\ ⇑(r0,\ w1)\}$ делает возможным исчерпывающее тестирование любого подмножества из $k \leq 6$ ячеек ОЗУ как с неповторяющимися, так и с повторяющимися адресными последовательностями. Использование любого другого маршевого теста с большим числом разнообразных фаз позволит достичь лучших результатов. Действительно, исчерпывающий тест ОЗУ для $k \leq 6$ обеспечивается тестом March C– в среднем за 31,93 итераций (см. табл. 8).

Список литературы

1. An Orchestrated Survey on Automated Software Test Case Generation / S. Anand [et al.] ; A. Bertolino, J. Jenny, L. Zhu, H. Zhu (ed.) // Journal of Systems and Software. – 2014. – Vol. C–39, no. 4. – P. 582–586.

2. 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.

3. Malaiya, Y.K. The coverage problem for random testing / Y.K. Malaiya, S. Yang // In Proc. of Intern. Test Conference (ITC 1984). – Philadelphia, PA, USA, 1984. – P. 237–242.

4. Ярмолик, С.В. Управляемые вероятностные тесты / С.В. Ярмолик, В.Н. Ярмолик // Автоматика и телемеханика. – 2012. – № 10. – С. 142–155.

5. Furuya, K. A probabilistic approach to locally exhaustive testing / K. Furuya // IEEE Transactions on IEICE. – 1989. – Vol. E72, no. 5. – P. 656–660.

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

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

9. Mrozek, I. Multiple Controlled Random Testing / I. Mrozek, V. Yarmolik // Fundamenta Informaticae. – 2016. – Vol. 144, no. 1. – P. 23–43.

10. 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.

11. Segall, I. Using binary decision diagrams for combinatorial test design / I. Segall, R. Tzoref-Brill, E. Farchi // Proc. of the Intern. Symp. on Software Testing and Analysis (ISSTA'11). – Toronto, Canada, 2011. − P. 254–264.

12. 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.

13. Chen, T.Y. Quasi-Random Testing / T.Y. Chen, R. Merkel // IEEE Trans. on Reliability. − 2007. – Vol. 56, no. 3. – P. 562–568.

14. 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.

15. Mrozek, I. Method for Generation Multiple Controlled Random Tests / I. Mrozek, V. Yarmolik // Proc. of the Computer Information Systems and Industrial Management (CISIM 2016). – Vilnius : Springer International Publisher, 2016. – P. 429–440.

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

17. Niggemeyer, D. Integration of Non-classical Faults in Standard March Tests / D. Niggemeyer, M. Redeker, J. Otterstedt // Proc. of the IEEE Intern. Workshop on Memory Technology, Design and Testing (MTDT'98). – Washington, USA, 1998. – P. 91–98.

18. Choinski, T.C. Generation of Digit Reversed Address Sequences for Fast FourierTransforms / T.C. Choinski, T.T. Tylaska // IEEE Transactions on Computers. – 1991. – Vol. 40, no. 6. – P. 780–784.

19. Wang, W.L. A Complete Memory Address Generator for Scan Based March Algorithms / W.L. Wang, K.J. Lee // Proc. of the IEEE Intern. Workshop on Memory Technology, Design, and Testing (MTDT'05). – Taipei, Taiwan, 2005. – P. 83–88.

20. 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.

Поступила 13.02.2017

[1]*Белорусский государственный университет*
информатики и радиоэлектроники,
Минск, П. Бровки, 6
e-mail: yarmolik10ru@yahoo.com,
lvn@bsuir.by

[2]*Белостокский технический университет,*
Белосток, ул. Вейска, 45А, 15-351
e-mail: i.mrozek@pb.edu.pl

V.N. Yarmolik, I. Mrozek, B.A. Levantsevich

PSEUDOEXHAUSTIVE RAM TESTING

Modern RAM tests and methods for their generation are analyzed and investigated. The wide application of pseudoexhaustive tests as the main test procedure for modern computer systems has been proved. The main estimates and metrics for so kind of tests are obtained. The values of analytical estimates have been validated by the experimental investigations.