

УДК 004.33.054

С.В. Ярмолик, В.Н. Ярмолик

ТЕСТИРОВАНИЕ ОЗУ НА ОСНОВЕ АДАПТИВНОГО СЖАТИЯ ВЫХОДНЫХ ДАННЫХ

Предлагается новая концепция неразрушающего тестирования оперативных запоминающих устройств (ОЗУ) на базе адаптивного сжатия выходных данных. Данная концепция основывается на использовании характеристики ОЗУ на базе адаптивного сжатия выходных данных, получаемой путем суммирования по модулю два всех адресов ячеек памяти, которые содержат единичные значения. Показывается, что эта характеристика может быть использована в качестве эталонной сигнатуры при тестировании ОЗУ. Рассматриваются основные свойства предлагаемых новых неразрушающих тестов, основанных на применении адаптивного сжатия выходных данных.

Введение

В современных вычислительных системах используются запоминающие устройства больших объемов и соответственно с большой плотностью элементов на кристалле. Для обеспечения надежного хранения данных в ОЗУ таких систем необходимы эффективные методы тестирования их исправности. Для достижения этих целей важную роль играет периодическое тестирование для проверки целостности данных, проверки их сохранности после включения питания или во время эксплуатации вычислительной системы по своему назначению. Так, частые операции с содержимым памяти требуют периодических проверок для выявления возможных неисправностей запоминающих устройств. Для случая встроенных запоминающих устройств в больших интегральных системах, таких, как микропроцессоры или цифровые устройства телекоммуникации, обычные технологии, применяемые для тестирования внешней памяти, не могут быть использованы в связи с ограниченностью доступа к встроенным запоминающим устройствам. Для решения данной проблемы была предложена идея встроенного самотестирования ОЗУ. Ряд теоретических и практических результатов по самотестированию ОЗУ был предложен в [1–7].

Известно, что для встроенных ОЗУ используются классические маршевые тесты из-за их высокой покрывающей способности и линейной сложности [8]. Более того, классические маршевые тесты легко преобразуются в неразрушающие тесты [9–11], а также в симметричные неразрушающие алгоритмы [12], которые сохраняют содержимое памяти и поэтому весьма эффективны для периодического тестирования и самотестирования ОЗУ [8]. Однако в связи с увеличением емкости современных ОЗУ и многократным применением маршевых тестов время тестирования становится серьезной проблемой наряду с проблемой уменьшения аппаратных затрат на реализацию средств самотестирования. В частности, применение стандартных неразрушающих средств встроенного самотестирования [8] может привести к неоправданному увеличению времени самотестирования, так как для их реализации необходимы временные затраты для вычисления эталонных сигнатур перед каждой процедурой реализации неразрушающего теста. Начальная фаза неразрушающего теста, применяемая для вычисления эталонной сигнатуры, в среднем имеет временную сложность, составляющую 50 % сложности используемого маршевого теста [9].

С целью уменьшения временной сложности неразрушающего теста с сохранением всех достоинств неразрушающего тестирования в настоящей статье рассматриваются алгоритмы адаптивного сжатия выходных данных (Self-Adjusting Output Data Compression – SAODC) для тестирования и диагностирования встроенных ОЗУ [13]. Данный подход позволяет полностью избежать временных издержек для вычисления эталонной сигнатуры. Будет показано, что алгоритмы адаптивного сжатия выходных данных могут быть использованы для эффективного обнаружения многократных константных неисправностей, переходных неисправностей и неисправностей взаимного влияния. При реализации встроенных средств самотестирования неразрушающих тестов, основанных на адаптивном сжатии выходных данных, аппаратные затраты сопоставимы с аппаратными затратами обычных неразрушающих тестов.

В данной статье используются следующие модели неисправностей [14]:

1. *Константные неисправности (Stuck-at fault – SAF)* – ячейка или линия находятся в одном из состояний логического ‘0’ или ‘1’.

2. *Неисправности типа обрыв (Stuck-open fault – SOF)* – открытая ячейка или обрыв линии.

3. *Переходные неисправности (Transition fault – TF)* – ячейка не может изменить свое состояние.

4. *Неисправность сохранения данных (Data retention fault – DRF)* – ячейка после некоторого времени теряет свое логическое значение из-за влияния паразитных факторов.

5. *Неисправности взаимного влияния (Coupling fault – CF)*. Различают три вида данных неисправностей:

инверсные неисправности взаимного влияния (Inversion coupling fault – CF_{in}) – изменение состояния в одной ячейке (агрессоре) ведет к изменению состояния в другой ячейке (жертве);

неисправности взаимного влияния прямого действия (Idempotent coupling fault – CF_{id}) – переход в одной ячейке в определенное состояние происходит под влиянием изменения состояния в другой ячейке [1, 3];

статические неисправности взаимного влияния (State coupling fault – CF_{st}) – ячейка-жертва устанавливается в фиксированное состояние, если ячейка-агрессор находится в определенном состоянии.

Таким образом, существуют следующие типы неисправностей взаимного влияния, распознаваемые маршевыми тестами:

1 – $\wedge < \uparrow, 0 >$; 2 – $\wedge < \downarrow, 0 >$; 3 – $\wedge < \uparrow, 1 >$; 4 – $\wedge < \downarrow, 1 >$; 5 – $\wedge < \uparrow, b >$; 6 – $\wedge < \downarrow, b >$; 7 – $\vee < \uparrow, 0 >$; 8 – $\vee < \downarrow, 0 >$; 9 – $\vee < \uparrow, 1 >$; 10 – $\vee < \downarrow, 1 >$; 11 – $\vee < \uparrow, b >$; 12 – $\vee < \downarrow, b >$.

С помощью знаков \uparrow и \downarrow описывается переход во влияющей ячейке-агрессоре, соответственно из ‘0’ в ‘1’ и из ‘1’ в ‘0’. Знак \wedge обозначает возрастающий порядок адресов ($j < i$), а \vee – убывающий ($j > i$), где j является адресом влияющей ячейки, i – ячейки-жертвы.

6. *Мостиковая неисправность (Bridging fault – BF)* описывает паразитные мостиковые соединения типа И либо типа ИЛИ между ячейками ОЗУ.

7. *Кодочувствительные неисправности (Neighborhood pattern sensitive fault – NPSF)*. Различают *активные (динамические) NPSF*, *пассивные NPSF* и *статические NPSF*.

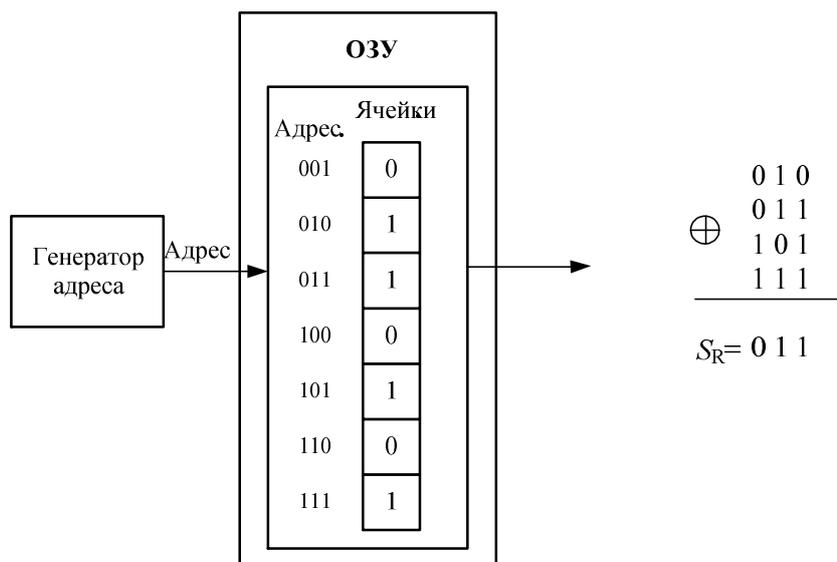
8. *Неисправность дешифратора адреса (Address decoder fault – AFs)*. Неисправности дешифратора адреса классифицируют следующим образом: *нет доступа ни к одной ячейке для определенного адреса ОЗУ; по определенному адресу активизируются несколько ячеек ОЗУ; нет доступа к определенной ячейке; определенная ячейка имеет несколько адресов, по которым она доступна.*

1. Адаптивное сжатие выходных данных ОЗУ

Концепция адаптивного сжатия выходных данных (АСВД) при реализации средств встроенного самотестирования ОЗУ была предложена в [13]. Согласно этой концепции эталонная характеристика (сигнатура) S_R начального содержимого бит-ориентированного ОЗУ вычисляется как сумма по модулю два всех адресов ячеек, содержащих значение ‘1’. Пример вычисления S_R для ОЗУ с $2^m - 1 = 2^3 - 1 = 7$ ячейками показан на рисунке.

Согласно теории адаптивного сжатия выходных данных ОЗУ тестируемое запоминающее устройство содержит все ячейки, кроме ячейки с нулевым адресом [13]. Пусть бит-ориентированное ОЗУ с $N = 2^m - 1$ ячейками будет представлено массивом $M[a]$, $a \in A$, с адресным пространством $A = \{1, 2, 3, \dots, 2^m - 1\}$. Тогда $A_1 = \{a \in A \mid M[a] = 1\}$ обозначает все адреса ячеек, содержащих ‘1’, а $A_0 = \{a \in A \mid M[a] = 0\}$ соответственно обозначает все адреса ячеек, содержащих ‘0’. Для A_1 и A_0 выполняется равенство $A_1 \cup A_0 = A$. Сумма по модулю два всех адресов A_1 и есть характеристика S_R [13]:

$$S_R = \bigoplus_{a \in A_1} a. \quad (1)$$

Эталонная сигнатура S_R для бит-ориентированного ОЗУ

Для данной характеристики справедливы следующие соотношения [13]:

$$S_R = \bigoplus_{a \in A_1} a = S_{R1} = \bigoplus_{a \in A_1} a = S_{R0} = \bigoplus_{a \in A_0} a; \quad (2)$$

$$S_R = \bigoplus_{a \in A} a = 0^m. \quad (3)$$

Здесь 0^m обозначает последовательность из m повторяющихся нулей. Используя подобное обозначение для общего случая, последовательность 000110001100011 может быть описана в компактном виде как $(0^3 1^2)^3$.

Основываясь на соотношениях (2) и (3), можно сделать следующее заключение. Для исправного бит-ориентированного ОЗУ с $N=2^m-1$ ячейками и произвольным содержимым $D = d_1 d_2 d_3 \dots d_N$, где $d_i \in \{1, 0\}$, $i \in \{1, 2, \dots, N\}$, эталонная сигнатура S_R равняется сумме по модулю два всех адресов ячеек, содержащих '1', или сумме по модулю два всех адресов ячеек, содержащих '0' ($S_R = S_{R1} = S_{R0}$). Для примера, показанного на рисунке, $S_R = S_{R1} = 010 \oplus 011 \oplus 101 \oplus 111 = 011$ и $S_{R0} = 001 \oplus 100 \oplus 110 = 011$. Сформулируем основное свойство для АСВД [13].

Свойство 1. Для вычисления эталонной сигнатуры S_R согласно адаптивному сжатию выходных данных могут быть использованы любые алгоритмы формирования адресной последовательности и произвольный порядок генерирования адресов, а именно возрастающий или убывающий.

Свойство 1 есть следствие равенства (1) и того факта, что операция поразрядного суммирования по модулю два является коммутативной и ассоциативной операцией. Согласно данному свойству эталонная характеристика (сигнатура) S_R содержимого памяти определяется как сумма по модулю два всех адресов ячеек, содержащих '1'. Порядок следования адресов не влияет на значение этой характеристики.

2. Неразрушающие тесты ОЗУ, основанные на АСВД

Маршевые тесты состоят из маршевых фаз, которые, в свою очередь, состоят из операции чтения или записи. Операции фазы применяются к ячейке памяти, после чего осуществляется переход к следующей ячейке. Порядок следования ячеек определяет адресная последовательность, которая может быть прямой и обратной. Так, символ wd обозначает операцию записи d ; wd^* – операцию записи инверсного значения d^* по отношению к предыдущему значению d ячейки ОЗУ; rd – операцию чтения, когда ожидаемое значение равняется d ; rd^* – операцию чтения, когда ожидаемое значение d^* есть инверсное по отношению к d ; \uparrow обозначает возрастание.

тающий порядок адресов, \Downarrow – убывающий порядок адресов, \Updownarrow – произвольный (возрастающий или убывающий) порядок адресов.

При реализации маршевых тестов выходные данные ОЗУ, формируемые операцией чтения, имеют два значения, а именно оригинальное значение, описываемое двоичной последовательностью $D = d_1 d_2 d_3 \dots d_N$, и инверсное значение $D^* = d_1^* d_2^* d_3^* \dots d_N^*$. Например, в случае неразрушающего маршевого теста March C–: $\{\uparrow\downarrow(rd, wd^*); \uparrow\downarrow(rd^*, wd); \downarrow\downarrow(rd, wd^*); \downarrow\downarrow(rd^*, wd); \uparrow\downarrow(rd)\}$ в течение первой, третьей и пятой фаз считываются оригинальные данные $D = d_1 d_2 d_3 \dots d_N$, в то время как во второй и четвертой фазах генерируются и считываются инверсные значения $D^* = d_1^* d_2^* d_3^* \dots d_N^*$.

АСВД может быть использовано при преобразовании оригинальных маршевых тестов в их неразрушающие версии. Предлагаемая процедура преобразования основывается на методике, предложенной М. Николаидсом [9]. Произвольный маршевый тест может быть преобразован в неразрушающий тест на базе АСВД согласно **процедуре 1**:

1. Фаза инициализации $\uparrow\downarrow(w0)$ заменяется первичной фазой $\uparrow\downarrow(rd)$ вычисления эталонной характеристики S_R . Во время данной фазы для исследуемого ОЗУ вычисляется сигнатура S_R согласно (1).

2. В остальных фазах маршевого теста все операции $r0$ ($r1$) заменяются операцией rd (rd^*).

3. Все операции $w0$ ($w1$) заменяются операцией wd (wd^*). В результате получается неразрушающий маршевый тест.

Например, для маршевых тестов MATS+ и March C– их неразрушающие версии будут иметь вид: $\{\uparrow\downarrow(rd); \uparrow\downarrow(rd, wd^*); \downarrow\downarrow(rd^*, wd)\}$ и $\{\uparrow\downarrow(rd); \uparrow\downarrow(rd, wd^*); \uparrow\downarrow(rd^*, wd); \downarrow\downarrow(rd, wd^*); \downarrow\downarrow(rd^*, wd); \uparrow\downarrow(rd)\}$.

Следует отметить, что первичная фаза $\uparrow\downarrow(rd)$ используется во всех неразрушающих тестах, построенных согласно процедуре 1. Это значит, что эталонная сигнатура S_R , которая получается во время данной фазы, не зависит от типа используемого маршевого теста.

В общем случае любой неразрушающий тест ОЗУ, полученный согласно процедуре 1, формирует регулярные выходные данные. Их вид зависит от трех аргументов, в первую очередь от размера блока памяти, для которого применяется неразрушающий тест. Параметры блока ОЗУ определяются начальным адресом i_a и конечным адресом f_a ($i_a < f_a$). Кроме того, регулярные выходные данные зависят от содержимого блока памяти D , а также от количества и вида операций чтения.

В случае маршевых тестов ОЗУ более чем с одной операцией чтения в одной фазе значения, полученные во время применения первой операции чтения, более значимы по сравнению со значениями, полученными в результате остальных операций чтения. Например, в случае стандартного теста March B: $\{\uparrow\downarrow(w0); \uparrow\downarrow(r0, w1, r1, w0, r0, w1); \uparrow\downarrow(r1, w0, w1); \downarrow\downarrow(r1, w0, w1, w0); \downarrow\downarrow(r0, w1, w0)\}$ во второй фазе используются три операции чтения. Это значит, что в неразрушающей версии эта фаза будет иметь вид $\uparrow\downarrow(rd, wd^*, rd^*, wd, rd, wd^*)$ или после удаления всех операций записи $\uparrow\downarrow(rd, rd^*, rd)$. Очевидно, что значения, полученные в результате первой операции чтения rd , более важны для выявления сложных неисправностей памяти типа CFs и должны быть использованы для принятия решения об исправности ОЗУ. Для последующих двух операций чтения rd^* и rd в последовательности операций чтения $\uparrow\downarrow(rd, rd^*, rd)$ значения, полученные в результате первой операции чтения, должны быть использованы как эталонные.

Итак, можно сделать вывод, что независимо от числа операций чтения в фазе маршевого теста только результат первой операции чтения должен быть использован для получения эталонной сигнатуры S_R и реального значения сигнатуры во всех фазах маршевого теста. Значение, полученное во время этой операции чтения (первой), может быть использовано как эталонная сигнатура для следующей операции чтения.

3. Адаптивное сжатие выходных данных ОЗУ

Для блока памяти, определенного двумя адресами i_a и f_a ($i_a < f_a$), и любого алгоритма генерирования адресов и их порядка следования в пределах данного блока, осуществляющего одно обращение (операцию чтения) ко всем ячейкам памяти, справедливы следующие утверждения:

У т в е р ж д е н и е 1. Эталонная характеристика $S_R = S_R(i_a, f_a, D)$ для блока памяти, описываемого начальным адресом i_a , конечным адресом f_a и содержимым D , определяется как сумма по модулю два последовательных адресов блока памяти для ячеек, содержащих '1', согласно выражению (1) [13].

У т в е р ж д е н и е 2. Эталонная характеристика $S_R = S_R(i_a, f_a, (D \oplus D^*))$ суммы по модулю два данных D и D^* тестируемого блока памяти равняется сумме по модулю два $S_R(i_a, f_a, D)$ и $S_R(i_a, f_a, D^*)$ или всех адресов блока памяти:

$$S_R(i_a, f_a, (D \oplus D^*)) = S_R(i_a, f_a, D) \oplus S_R(i_a, f_a, D^*) = S_R(i_a, f_a, (1^{f_a+1-i_a})). \quad (4)$$

Утверждения 2 следует из свойства линейности, описываемого соотношениями (2) и (3) [13].

У т в е р ж д е н и е 3. Эталонная характеристика S_R для тестируемого ОЗУ с адресным пространством $A = \{1, 2, 3, \dots, 2^m - 1\}$ ($i_a = 000 \dots 01 = 0^{m-1}1$, $f_a = 111 \dots 11 = 1^m$), содержащего во всех ячейках '1', имеет значение $S_R = 000 \dots 00 = 0^m$.

Например, в случае, представленном на рисунке, и при полном пространстве адресов памяти получим $001 \oplus 010 \oplus 011 \oplus 100 \oplus 101 \oplus 110 \oplus 111 = 000$, что удовлетворяет утверждению 3 и основному свойству адаптивного сжатия выходных данных [13]. В то же время в случае блока памяти, состоящего из подмножества последовательных адресов памяти, результат может принимать значения, отличные от нулевого. Данное утверждение показано на рисунке для блока памяти с начальным адресом $i_a = 010$ и конечным адресом $f_a = 100$. Действительно, для этого примера окончательное значение $S_R(i_a, f_a, (1^3)) = S_R(2, 4, (1^3)) = 010 \oplus 011 \oplus 100 = 101$.

Для блока ОЗУ с фиксированным начальным адресом $i_a = 000 \dots 01 = 0^{m-1}1$ и произвольным конечным адресом f_a , содержащим все '1' – $D = 111 \dots 11 = 1^{f_a+1-1} = 1^{f_a}$, значение сигнатуры $S_R = S_R(i_a, f_a, D) = S_R((0^{m-1}1, f_a, 1^{f_a}))$ зависит только от конечного адреса f_a . Таким образом, $S_R = S_R(f_a)$, где $S_R(f_a)$ представляет собой сумму по модулю два всех адресов блока памяти, начиная с начального адреса $i_a = i_a(m-1)i_a(m-2)i_a(m-3) \dots i_a(1)i_a(0) = 000 \dots 01 = 0^{m-1}1$ и до конечного адреса $f_a = f_a(m-1)f_a(m-2)f_a(m-3) \dots f_a(1)f_a(0)$ [16]. В таблице приведены значения суммы по модулю два для последовательных адресов ОЗУ, где, например, для $f_a = 2$ имеем $S_R(f_a = 2) = S_R(000 \dots 010) = 000 \dots 001 \oplus 000 \dots 010 = 000 \dots 011$.

Эталонная характеристика для тестируемого блока памяти

f_a	$f_a = f_a(m-1)f_a(m-2)f_a(m-3) \dots f_a(1)f_a(0)$	$S_R(f_a) = s_{m-1}s_{m-2}s_{m-3} \dots s_2s_1s_0$
1	000...0001	000...0001
2	000...0010	000...0011
3	000...0011	000...0000
4	000...0100	000...0100
5	000...0101	000...0001
6	000...0110	000...0111
7	000...0111	000...0000
8	000...1000	000...1000
9	000...1001	000...0001
10	000...1010	000...1011
11	000...1011	000...0000
12	000...1100	000...1100
13	000...1101	000...0001
14	000...1110	000...1111
15	000...1111	000...0000
...
2^m-1	111...1111	000...0000

Как было показано в [16], для значения $S_R(f_a)$ справедливо следующее утверждение:

У т в е р ж д е н и е 4. Для блока памяти, содержащего все '1', с начальным адресом $i_a = i_a(m-1)i_a(m-2)i_a(m-3) \dots i_a(1)i_a(0) = 000 \dots 01 = 0^{m-1}1$ и произвольным конечным адресом $f_a = f_a(m-1)f_a(m-2)f_a(m-3) \dots f_a(1)f_a(0)$, эталонная характеристика $S_R(f_a = f_a(m-1)f_a(m-2)$

$f_a(m-3) \dots f_a(1)f_a(0)$ принимает значение, которое зависит только от адреса $f_a = f_a(m-1)f_a(m-2)f_a(m-3) \dots f_a(1)f_a(0)$ и определяется согласно выражению

$$\begin{aligned} S_R(f_a) &= f_a = f_a(m-1)f_a(m-2)f_a(m-3) \dots f_a(3)f_a(2)f_a(1)f_a(0) \text{ для } f_a(0) = 0; \\ S_R(f_a) &= 000 \dots 00f_a(1)^* \text{ для } f_a(0) = 1, \end{aligned} \quad (5)$$

где $f_a(1)^*$ – инверсное значение $f_a(1)$.

Для вычисления $S_R(f_a) = s_{m-1}s_{m-2}s_{m-3} \dots s_3s_2s_1s_0$, соответствующего приведенному выше утверждению, можно предложить следующий алгоритм:

Алгоритм 1

Входные данные: $f_a = f_a(m-1)f_a(m-2)f_a(m-3) \dots f_a(1)f_a(0)$.

Start:

If $f_a \bmod 4 = 0$, then $S_R(f_a) = f_a$;
 Else if $f_a \bmod 4 = 1$, then $S_R(f_a) = 000 \dots 001$;
 Else if $f_a \bmod 4 = 2$, then $S_R(f_a) = f_a + 1$;
 Else if $f_a \bmod 4 = 3$, then $S_R(f_a) = 000 \dots 000$.

End

Основываясь на утверждении 4, можно сформулировать такое же утверждение и для произвольных начального и конечного адресов $i_a = i_a(m-1)i_a(m-2)i_a(m-3) \dots i_a(1)i_a(0)$ и $f_a = f_a(m-1)f_a(m-2)f_a(m-3) \dots f_a(1)f_a(0)$.

Утверждение 5. Эталонная характеристика $S_R(i_a, f_a, (I^{f_a+1-i_a}))$ для блока памяти с начальным адресом i_a и конечным адресом f_a , содержащего все '1' – $D = 111 \dots 11 = I^{f_a+1-i_a}$, равняется сумме по модулю два характеристик $S_R(i_a - 1)$ и $S_R(f_a)$, определенных согласно утверждению 4.

Например, $S_R(3, 7, (I^{7+1-3})) = S_R(3-1) \oplus S_R(7) = S_R(2) \oplus S_R(7) = 000 \dots 0011 \oplus 000 \dots 0000 = 000 \dots 0011$ (см. таблицу).

Последние два утверждения позволяют сформулировать вычислительную процедуру для определения $S_R(i_a, f_a, (I^{f_a+1-i_a}))$. Эта процедура может быть трансформирована в алгоритм, подобный алгоритму 1, принимая во внимание, что $(i_a - 1) \bmod 4 = (i_a + 3) \bmod 4$. В данном алгоритме будут использованы арифметические операции '+' и '-', а также сумма по модулю два \oplus .

Алгоритм 2

Входные данные: $i_a = i_a(m-1)i_a(m-2)i_a(m-3) \dots i_a(1)i_a(0)$, и $f_a = f_a(m-1)f_a(m-2)f_a(m-3) \dots f_a(1)f_a(0)$.

Start:

if $i_a \bmod 4 = 0$,
 if $f_a \bmod 4 = 0$, then $S_R = f_a$;
 Else if $f_a \bmod 4 = 1$, then $S_R = 000 \dots 001$;
 Else if $f_a \bmod 4 = 2$, then $S_R = f_a + 1$;
 Else if $f_a \bmod 4 = 3$, then $S_R = 000 \dots 000$.

End if

End if

if $i_a \bmod 4 = 1$,

if $f_a \bmod 4 = 0$, then $S_R = (i_a - 1) \oplus f_a$;
 Else if $f_a \bmod 4 = 1$, then $S_R = i_a$;
 Else if $f_a \bmod 4 = 2$, then $S_R = (i_a - 1) \oplus (f_a + 1)$;
 Else if $f_a \bmod 4 = 3$, then $S_R = i_a - 1$.

End if

End if

if $i_a \bmod 4 = 2$,

if $f_a \bmod 4 = 0$, then $S_R = f_a + 1$;
 Else if $f_a \bmod 4 = 1$, then $S_R = 000 \dots 000$;
 Else if $f_a \bmod 4 = 2$, then $S_R = f_a$;
 Else if $f_a \bmod 4 = 3$, then $S_R(f_a) = 000 \dots 001$.

End if

End if
 if $i_a \bmod 4 = 3$,
 if $f_a \bmod 4 = 0$, then $S_R = i_a \oplus f_a$;
 Else if $f_a \bmod 4 = 1$, then $S_R = i_a - 1$;
 Else if $f_a \bmod 4 = 2$, then $S_R = i_a \oplus (f_a + 1)$;
 Else if $f_a \bmod 4 = 3$, then $S_R = i_a$.
 End if
 End if
 End

Выходные данные: $S_R(i_a, f_a, (1^{f_a+1-i_a})) = S_R = s_{m-1}s_{m-2}s_{m-3} \dots s_3s_2s_1s_0$.

Для примера определим значение $S_R(i_a, f_a, (1^{f_a+1-i_a}))$ для $i_a = 3$ и $f_a = 13$. Согласно алгоритму 2 $i_a \bmod 4 = 3 \bmod 4 = 3$ и $f_a \bmod 4 = 13 \bmod 4 = 1$, тогда $S_R = S_R(i_a, f_a, (1^{f_a+1-i_a})) = S_R(3, 13, (1^{f_a+1-i_a})) = i_a - 1 = 0011 - 0001 = 0010$ или как результат суммы по модулю два адресов рассматриваемого блока получим $S_R = S_R(i_a, f_a, (1^{f_a+1-i_a})) = 0011 \oplus 0100 \oplus 0101 \oplus 0110 \oplus 0111 \oplus 1000 \oplus 1001 \oplus 1010 \oplus 1011 \oplus 1100 \oplus 1101 = 0010$.

Приведенные выше утверждения и алгоритмы для адаптивного сжатия выходных данных позволяют сформулировать следующие основные свойства для АСВД, которое используется для реализации неразрушающего тестирования ОЗУ, определенного процедурой 1:

Свойство 2. Эталонная характеристика $S_R = S_R(i_a, f_a, D^*)$, полученная согласно алгоритму АСВД для содержимого D^* блока тестируемого ОЗУ, который описывается начальным i_a и конечным f_a адресами, может быть определена как

$$S_R(i_a, f_a, D^*) = S_R(i_a, f_a, D) \oplus S_R(i_a, f_a, (1^{f_a+1-i_a})), \quad (6)$$

где $S_R(i_a, f_a, (1^{f_a+1-i_a}))$ вычисляется согласно алгоритму 2.

Свойство 2 может быть рассмотрено как следствие соотношения (4).

Свойство 3. Эталонная характеристика S_R так же, как и значение реальной характеристики S_{TEST} , для случая исправного состояния блока ОЗУ для двух последовательных значений данных D и D^* не зависит от содержимого D блока ОЗУ, а определяется только начальным i_a и конечным f_a адресами рассматриваемого блока, что следует из соотношения

$$S_R(i_a, f_a, (DD^*)) = S_R(i_a, f_a, (D \oplus D^*)) = S_R(i_a, f_a, (1^{f_a+1-i_a})). \quad (7)$$

Свойство 4. Неразрушающий тест на базе АСВД характеризуется аналогичной или большей обнаруживающей и диагностической способностью неисправностей ОЗУ по сравнению с классическим неразрушающим тестированием в силу многократного сравнения S_{TEST} с эталонным значением S_R .

Для неразрушающих тестов, основанных на АСВД, это сравнение производится после каждой фазы маршевого теста ОЗУ. В случае классических неразрушающих тестов S_{TEST} сравнивается с эталонным значением однократно после выполнения всех фаз теста.

Свойство 5. Средства неразрушающего самотестирования ОЗУ на базе АСВД имеют низкие аппаратную и временную сложности.

Низкая аппаратная сложность определяется минимальным набором стандартных устройств, используемых для встроенного самотестирования. Так же, как и в случае известных технологий неразрушающего самотестирования, аппаратные средства для реализации неразрушающего самотестирования ОЗУ на базе АСВД включают минимальный набор дополнительных устройств. В их число входят такие устройства, как генератор адресов ОЗУ, схема вычисления эталонной сигнатуры, схема вычисления реальной сигнатуры, схема хранения эталонной сигнатуры, схема сравнения реальной сигнатуры с эталонной и устройство управления самотестированием. По сравнению с известными методами неразрушающее самотестирование ОЗУ на базе АСВД характеризуется минимальными затратами на реализацию схем вычисления эталонной и реальной сигнатур. Эти устройства представляют собой набор из m двухвходовых сумматоров по модулю два, где m – разрядность адреса ОЗУ.

Минимальная временная сложность неразрушающего тестирования на базе АСВД обеспечивается минимальным временем для вычисления эталонной сигнатуры, которое определяется емкостью N ОЗУ и равняется N циклам обращения к ОЗУ. Для сравнения рассмотрим два стандартных неразрушающих теста MATS+ $\{\uparrow(rd, wd^*); \downarrow(rd^*, wd)\}$ и March C- $\{\uparrow(rd, wd^*); \uparrow(rd^*, wd); \downarrow(rd, wd^*); \downarrow(rd^*, wd); \uparrow(rd)\}$. Эталонная сигнатура S_R для обоих стандартных тестов определяется с помощью сигнатурного анализатора, описываемого полиномом $\varphi(x)$ [1]. Эталонная сигнатура формируется как результат сжатия на анализаторе данных в соответствии с операциями чтения $\{\uparrow(rd); \downarrow(rd^*)\}$ для MATS+ и $\{\uparrow(rd); \uparrow(rd^*); \downarrow(rd); \downarrow(rd^*); \uparrow(rd)\}$ для March C-. В первом случае необходимо $2N$ циклов обращения к ОЗУ, а во втором – $5N$ циклов. Суммарная сложность приведенных выше тестов составляет $6N$ для MATS+ и $14N$ для March C-. Использование неразрушающих тестов MATS+ и March C- на базе АСВД характеризуется временной сложностью $5N$ и $10N$ соответственно. В данном случае для вычисления эталонного значения сигнатуры S_R независимо от используемого маршевого теста необходимо только N дополнительных циклов обращения к ОЗУ.

Заключение

Предложены новые неразрушающие тесты на базе АСВД, которые характеризуются малыми аппаратными и временными затратами на реализацию в рамках самотестирования встроенных ОЗУ. Отличительной особенностью неразрушающих тестов на базе АСВД является многократное сравнение эталонной сигнатуры с реальной. Количество сравнений определяется количеством фаз используемого неразрушающего теста. Поэтому новые неразрушающие тесты имеют более высокую покрывающую способность всех видов неисправностей ОЗУ. Кроме того, неразрушающие тесты на базе АСВД могут быть эффективно использованы для обнаружения сложных неисправностей ОЗУ, а также их диагностирования.

Список литературы

1. Bardell, P.H. Built-In Self-Test for VLSI: pseudorandom techniques / P.H. Bardell, W. McAnney, J. Savir. – N.Y. : John Wiley&Sons, 1987. – 576 p.
2. Dekker, R. Realistic Built-In Self-Test for Static RAMs / R. Dekker, F. Beenker, L. Thijasen // IEEE Design and Test of Computers. – 1989. – Vol. 6, № 1. – P. 26–34.
3. Jain, S.K. Built-In Self-Testing of Embedded Memories / S.K. Jain, S.H. Stroud // IEEE Design and Test of Computers. – 1986. – Vol. 3, № 5. – P. 27–37.
4. Rajsuman, R. RAMBIST Builder: A Methodology for Automatic Built In Self-Test Design of Embedded RAMs / R. Rajsuman // IEEE International Workshop on Memory Technology, Design and Testing (MTDT'96) : Rec., Singapore, 13–14 Aug. 1996 / IEEE Computer Society. – Singapore, 1996. – P. 50–56.
5. Treuer, R. Built-In Self-Diagnosis for Repairable Embedded RAMS / R. Treuer, V.K. Agarwal // IEEE Design and Test of Computer. – June 1993. – Vol.10, № 2. – P. 24–33.
6. Chen, T. Design of a Self-Testing and Self-Repairing Structure for Highly Hierarchical Ultra-Large Capacity Memory Chips / T. Chen, G. Sunada // IEEE Transactions on VLSI Systems. – 1993. – Vol. 1, № 2. – P. 88–97.
7. Le, K.T. A Novel Approach for Testing Memories Using a Built-in Self Testing Technique / K.L. Le, K.K. Saluja // Test Conference: proc. Int. Conf., Washington D.C., USA, September 1986 / IEEE Computer Society. – Washington, USA, 1986. – P. 830–839.
8. Goor, A.J. Testing Semiconductor Memories, Theory and Practice / A.J. Goor. – UK, Chichester : John Wiley & Sons, 1991. – 536 p.
9. Nicolaidis, M. Transparent BIST for RAMS / M. Nicolaidis // Proc. Int. Test Conference, Baltimore, USA, September 1992 / IEEE Computer Society. – Washington, USA, 1992. – P. 598–607.
10. Неразрушающее тестирование запоминающих устройств / В.Н. Ярмолик [и др.]. – Минск : Бестпринт, 2005. – 230 с.

11. Cockburn, B. Synthesized Transparent BIST for Detecting Scrambled Pattern-Sensitive Faults in RAMs / B. Cockburn, Y.-F. Sat // Test Conference : proc. Int. Conf., Washington D.C., USA, October 1995 / IEEE Computer Society. – Washington, USA, 1995. – P. 23–32.
12. Hellebrand, S. Symmetric Transparent BIST for RAMs / S. Hellebrand, Y.-J. Wundelich, V.N. Yarmolik // IEEE Design, Automation and Test in Europe Conference (DATE'99) : proc. Int. Conf., Munich, Germany, 9–12 March 1999 / ACM Special Interest Group on Design Automation. – N.Y., USA, 1999. – P. 702–707.
13. Yarmolik, V.N. Self-Adjusting Output Data Compression: An Efficient BIST Technique for RAMs / S. Hellebrand, Y.-J. Wundelich, V.N. Yarmolik // IEEE Design, Automation and Test in Europe Conference (DATE'99) : proc. Int. Conf., Paris, France, 23–26 February 1998 / ACM Special Interest Group on Design Automation. – N.Y., USA, 1998. – P. 173–179.
14. Bushnell, M. Essential of electronic testing for digital, memory and mixed-signal VLSI circuits / M. Bushnell, A. Vishwani. – N.Y. : Kluwer Academic Publishers, 2001.
15. Efficient Online and Offline Testing of Embedded DRAMs / S. Hellebrand [et al.] // IEEE Transactions on COMPUTERS. – 2002. – Vol. 51, № 7. – P. 801–809.
16. Yarmolik, V. Built-In Self-Test and Diagnosis for Ram Based on Self-Adjusting Output Data Compression / V. Yarmolik, A.A. Ivaniuk, M. Krips // Design and Diagnostics of Electronic Circuits and Systems DDECS 2002 : proc. of Fifth Int. Workshop, Brno, Czech Republic, 17–19 April 2002. – Czech Republic, 2002. – P. 360–363.

Поступила 12.03.09

*Белорусский государственный университет
информатики и радиоэлектроники,
Минск П. Бровка, 6
e-mail: yarmolik@cosmostv.by,
yarmolik10ru@yahoo.com*

S.V. Yarmolik, V.N. Yarmolik

TRANSPARENT RAM TESTING BASED ON SELF-ADJUSTING OUTPUT DATA COMPRESSION

In this paper a new concept of transparent RAM Testing based on Self-Adjusting Output Data Compression (SAODC) has been proposed and developed. The presented technique is based on a memory characteristic derived as the modulo-2 sum of all addresses pointing to non-zero memory cells. This characteristic can be used as the RAM under test fault-free signature. The main properties of new transparent tests are investigated and validated.