ИНФОРМАТИКА

2006 январь-март № 1

УДК 004.33.054

С.В. Ярмолик, В.Н. Ярмолик

ОБНАРУЖЕНИЕ КОДОЧУВСТВИТЕЛЬНЫХ НЕИСПРАВНОСТЕЙ ЗАПОМИНАЮЩИХ УСТРОЙСТВ С МНОГОКРАТНЫМ ИСПОЛЬЗОВАНИЕМ МАРШЕВЫХ ТЕСТОВ

Рассматривается эффективность применения маршевых тестов для обнаружения кодочувствительных неисправностей запоминающих устройств (ЗУ) и дается оценка их покрывающей способности. Исследуется вопрос о многократном применении маршевых тестов и анализируется влияние изменения начальных адресов ЗУ при повторном применении маршевых тестов для достижения их максимальной эффективности. В заключение приводятся экспериментальные данные, показывающие эффективность многократного применения маршевых тестов.

Введение

Все множество неисправностей запоминающих ячеек ЗУ делится на несколько подмножеств. Такое деление весьма условно и может быть выполнено по различным критериям. Наиболее часто используется деление неисправностей ЗУ в зависимости от количества ячеек, участвующих в описании конкретной неисправности.

К неисправностям массива ячеек ЗУ в первую очередь относят неисправности, затрагивающие одну, две или несколько ячеек ЗУ (в общем случае более чем две без ограничений на их количество, т. е. так называемые кодочувствительные неисправности) [1].

Наиболее часто используются константные неисправности (*stuck-at faults – SAF*), которые характеризуются тем, что конкретная ячейка ЗУ (неисправная) постоянно находится в одном из возможных состояний, в состоянии нуля (0) или состоянии единица (1), независимо от выполненных ранее операций записи в данную ячейку противоположного значения [1]. Различают неисправность константного нуля (*stuck-at 0 – SAF0*) и неисправность константной единицы (*stuck-at 1 – SAF1*). Переходные неисправности (*transition faults – TF*) характеризуются невозможностью логического перехода состояния ячейки ЗУ (неисправной) из 0 в 1 (*transition up*, $\langle\uparrow\rangle$) или из 1 в 0 (*transition down*, $\langle\downarrow\rangle$) при выполнении соответствующих операций записи [1].

Среди неисправностей, в которых участвуют две ячейки ЗУ, выделяют неисправности взаимного влияния (*coupling faults – CF*), а среди них – инверсные неисправности взаимного влияния (*inversion coupling faults – CFin*). В неисправности *CFin* участвуют две ячейки ЗУ: a_i и a_j, $i \neq j$. При этом a_i называется агрессором (*aggressor*), а a_j – жертвой (*victim*). Расположение агрессора и жертвы в адресном пространстве ЗУ произвольно. При наличии данной неисправности логический переход из 1 в 0 или из 0 в 1 в агрессоре a_i приводит к инверсии логического значения в жертве a_j [1]. Таким образом, различают два вида инверсных неисправностей: $\langle\uparrow,a_j\rangle$ и $\langle\downarrow,a_j\rangle$.

Для неисправности прямого действия (*idempotent coupling faults – CFid*) во время логического перехода из 1 в 0 или из 0 в 1 во влияющей ячейке-агрессоре a_i происходит принудительная установка определенного логического значения 0 или 1 в ячейке-жертве a_j, на которую оказывает влияние агрессор [1]. Различают четыре неисправности прямого действия: $\langle\uparrow,0\rangle$, $\langle\uparrow,1\rangle$, $\langle\downarrow,0\rangle$ и $\langle\downarrow,1\rangle$.

Наиболее значимым классом неисправностей ЗУ являются кодочувствительные неисправности (*pattern sensitive faults – PSF*), затрагивающие несколько ячеек ЗУ. Для подобных неисправностей логическое состояние одной ячейки ЗУ может зависеть от ее содержимого (0 или 1) или от логических переходов из 1 в 0 или из 0 в 1 соседних ячеек ЗУ. Выделяют базовую ячейку (*base cell*), которая в данном случае является аналогом жертвы, и соседние ячейки (*neighborhood cells*), выступающие в роли агрессоров. Их количество и местоположение может быть произвольным. В зависимости от эффекта влияния на базовую ячейку различают следующие кодочувствительные неисправности [1–3]:

пассивные (*Passive PSF – PPSF*) – состояние базовой ячейки не может быть изменено для определенного кода в соседних ячейках ЗУ;

активные (*Active PSF – APSF*) – базовая ячейка изменяет свое состояние из-за изменения кода в соседних ячейках, которое происходит в результате изменения состояния на противоположное только в одной соседней ячейке, в то время как остальные ячейки сохраняют предыдущее состояние;

статические (*Static PSF – SPSF*) – для определенной комбинации значений в соседних ячейках состояние базовой ячейки принудительно устанавливается в состояние 0 или состояние 1. Главным отличием статических неисправностей от активных кодочувствительных неисправностей является длительность процесса установления неверного значения в базовой ячейке. Для статических неисправностей это время существенно больше [1].

В качестве объекта исследования в дальнейшем будем рассматривать пассивные кодочувствительные неисправности (*PPSFk*), где k обозначает количество произвольных ячеек ЗУ емкостью N бит, участвующих в конкретной неисправности, причем одна из них является базовой. Выделяют k различных классов *PPSFk* в зависимости от местоположения в адресном пространстве ЗУ базовой ячейки по отношению к соседним ячейкам. Например, в случае $k = 5$ существует пять следующих классов *PPSFk*: $b_{i0}n_{i1}n_{i2}n_{i3}n_{i4}$, $n_{i0}b_{i1}n_{i2}n_{i3}n_{i4}$, $n_{i0}n_{i1}b_{i2}n_{i3}n_{i4}$, $n_{i0}n_{i1}n_{i2}b_{i3}n_{i4}$, $n_{i0}n_{i1}n_{i2}n_{i3}b_{i4}$, где $i_0<i_1<i_2<\ldots<i_{k-1}$. В соседних ячейках возможны любые из 2^{k-1} двоичных наборов, каждый из которых определяет конкретную неисправность. Тогда общее количество возможных *PPSFk* для ЗУ емкостью N бит определяется согласно выражению [4]:

$$Q(PPSFk) = k2^{k-1} \times \binom{N}{k}. \tag{1}$$

Отметим, что результаты, полученные для *PPSFk*, легко обобщаются и на другие классы кодочувствительных неисправностей в силу того, что *PPSFk* является наиболее трудно обнаруживаемым типом неисправностей [1].

1. Оценка обнаруживающей способности маршевых тестов

Традиционно маршевый тест состоит из последовательности фаз, каждая из которых содержит операции чтения (r) и записи (w), причем все фазы всегда начинаются с операции чтения [1]. В качестве примера рассмотрим тест MATS+(*5N*): $\{\updownarrow (w0); \uparrow (r0, w1); \downarrow (r1, w0)\}$, неразрушающая версия которого имеет вид $\{\uparrow (ra, wa^*); \downarrow (ra^*, wa)\}$, где $a\in\{0,1\}$ есть значение содержимого ячейки ЗУ, а a^* – инверсное ему значение; символ $\uparrow$ обозначает возрастающую последовательность адресов, а символ $\downarrow$ – убывающую [5].

Применим указанный тест для тестирования ЗУ, содержащего N=8 ячеек. Исходное состояние всех ячеек ЗУ $A = a_0a_1a_2a_3a_4a_5a_6a_7 = 00000000$ и последующие их состояния при реализации теста MATS+ приведены в табл. 1, а состояния ЗУ для первых двух фаз неразрушающего теста March C–(*10N*): $\{\uparrow (ra, wa^*); \uparrow (ra^*, wa); \downarrow (ra, wa^*); \downarrow (ra^*, wa); \downarrow (ra)\}$ – в табл. 2.

Жирным шрифтом выделены ячейки, над которыми выполняются операции чтения и записи инверсного значения. Очевидно, что только при выполнении операции записи в базовую ячейку будет активизирована неисправность *PPSFk*, а ее обнаружение будет происходить при выполнении операции чтения в последующей фазе. Отметим, что в качестве генератора последовательности адресов использовалась счетчиковая последовательность, а в качестве начального адреса – $i_0 = 0$.

Таблица 1

Состояния ячеек ЗУ для i_0=0
при использовании маршевого теста MATS+

Фаза теста	Содержимое ЗУ							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
	0	0	0	0	0	0	0	0
↑ *(ra, wa*)*	***1***	0	0	0	0	0	0	0
	1	***1***	0	0	0	0	0	0
	1	1	***1***	0	0	0	0	0
	1	1	1	***1***	0	0	0	0
	1	1	1	1	***1***	0	0	0
	1	1	1	1	1	***1***	0	0
	1	1	1	1	1	1	***1***	0
	1	1	1	1	1	1	1	***1***
↓ *(ra*, wa)*	1	1	1	1	1	1	1	***0***
	1	1	1	1	1	1	***0***	0
	1	1	1	1	1	***0***	0	0
	1	1	1	1	***0***	0	0	0
	1	1	1	***0***	0	0	0	0
	1	1	***0***	0	0	0	0	0
	1	***0***	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Таблица 2

Состояния ячеек ЗУ для i_0=0
при использовании маршевого теста March C–

Фаза теста	Содержимое ЗУ							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
	0	0	0	0	0	0	0	0
↑ *(ra, wa*)*	***1***	0	0	0	0	0	0	0
	1	***1***	0	0	0	0	0	0
	1	1	***1***	0	0	0	0	0
	1	1	1	***1***	0	0	0	0
	1	1	1	1	***1***	0	0	0
	1	1	1	1	1	***1***	0	0
	1	1	1	1	1	1	***1***	0
	1	1	1	1	1	1	1	***1***
↑ *(ra*, wa)*	***0***	1	1	1	1	1	1	1
	0	***0***	1	1	1	1	1	1
	0	0	***0***	1	1	1	1	1
	0	0	0	***0***	1	1	1	1
	0	0	0	0	***0***	1	1	1
	0	0	0	0	0	***0***	1	1
	0	0	0	0	0	0	***0***	1
	0	0	0	0	0	0	0	***0***

Можно заметить, что для теста MATS+ и любой базовой ячейки при записи в нее нового значения в остальных ячейках ЗУ находится только один фиксированный набор. Отсюда следует, что из всего множества 2^{k-1} возможных комбинаций в произвольных k–1 соседних ячейках будет сформирован только один набор и, соответственно, обнаружена только одна из 2^{k-1} неисправностей *PPSFk*. Тогда общее количество кодочувствительных неисправностей $Q_1(PPSFk(i_0))$ [4, 6], обнаруживаемых тестом MATS+ при любом начальном значении адреса i_0, вычисляется по формуле

$$Q_1(PPSFk\,(i_0)) = k \times \binom{N}{k}, \tag{2}$$

а полнота покрытия $FC_{MATS+}(PPSFk(i_0))$ определяется как процентное отношение количества обнаруживаемых неисправностей к общему их числу:

$$FC_{MATS+}(PPSFk\,(i_0)) = \frac{Q_1(PPSFk\,(i_0))}{Q(PPSFk\,)} 100\,\% = \frac{1}{2^{k-1}} 100\,\%. \tag{3}$$

Например, для *PPSF5* имеем $FC_{MATS+}(PPSFk(i_0))=(1/2^4)100\,\%=6{,}25\,\%$.

Выражение для полноты покрытия (3) справедливо для любого маршевого теста, содержащего последовательность фаз {... ↑ (*ra*, ..., *wa**); ↓ (*ra**, ...); ...} подобно тесту MATS+, в котором отсутствует фрагмент {... ↑ (*ra*, ..., *wa**); ↑ (*ra**, ..., *wa*); ↕(*ra*, ...) ...}, присущий тесту March C–.

Если же тест содержит фрагмент {... ↑ (*ra*, ..., *wa**); ↑ (*ra**, ..., *wa*); ↕ (*ra*, ...) ...}, то в этом случае для произвольной фиксированной базовой ячейки и произвольных фиксированных соседних ячеек будут обнаруживаться уже две неисправности *PPSFk* (см. табл. 2), их общее число $Q_2(PPSFk(i_0))$ будет описываться выражением [4, 6]

$$Q_2(PPSFk\,(i_0)) = 2k \times \binom{N}{k}, \tag{4}$$

а полнота покрытия $FC_{March\ C-}(PPSFk(i_0))$ определяться как

$$FC_{MarchC-}(PPSFk\,(i_0)) = \frac{Q_2(PPSFk\,(i_0))}{Q(PPSFk\,)} 100\,\% = \frac{1}{2^{k-2}} 100\,\%. \qquad (5)$$

В случае *PPSF5* теста March C– $FC_{March\ C-}(PPSFk(i_0))=(1/2^3)100\,\%=12{,}5\,\%$.

Таким образом, однократное применение маршевого теста позволяет обнаружить не более чем $2^{-k+2}100$ % неисправностей *PPSFk*. Так, для $k = 2$, т. е. для случая наличия неисправностей взаимного влияния, достигается максимальная полнота покрытия. Для неисправностей *PPSFk* большей кратности полнота покрытия уменьшается экспоненциально.

2. Многократное применение маршевых тестов

С целью увеличения полноты покрытия *PPSFk* используют повторное многократное применение маршевого теста для различных начальных значений ЗУ [7], что позволяет существенно увеличить полноту покрытия кодочувствительных неисправностей. Однако эффективность данного подхода в значительной мере зависит от степени изменений содержимого ЗУ перед повторным применением маршевого теста [7].

В качестве альтернативного подхода рассмотрим эффективность многократного применения маршевых тестов с модифицированной последовательностью адресов, основываясь на свойствах маршевых тестов [8]. Для примера возьмем случай, когда маршевый тест используется дважды (тестовые процедуры 1 и 2) для различной последовательности адресов.

Тестовая процедура 1 реализуется при использовании произвольного генератора адресов ЗУ. Единственным требованием к нему является перебор всех адресов без повторений в произвольном порядке. Например, генератором адресов может быть счетчиковая последовательность, генерируемая двоичным счетчиком. Для тестовой процедуры 2 использовалась та же последовательность адресов ЗУ, что и для тестовой процедуры 1, но с различными модификациями в обоих экспериментах. В эксперименте 1 при выполнении тестовой процедуры 2 инвертировался старший бит последовательности адресов тестовой процедуры 1, а в эксперименте 2 – только младший бит адресной последовательности.

Результаты экспериментов по оценке эффективности обнаружения неисправностей *PPSF5* даны в табл. 3 и 4. В графе «Тестовая процедура 2» приведена суммарная полнота покрытия, полученная в результате последовательного выполнения тестовых процедур 1 и 2. Анализ результатов свидетельствует о зависимости полноты покрытия от метода модификации последовательности адресов.

Таблица 3

Результаты эксперимента 1

Маршевый тест	Тестовая процедура 1	Тестовая процедура 2
MATS+	6,25	12,5
March C–	12,5	24,6

Таблица 4

Результаты эксперимента 2

Маршевый тест	Тестовая процедура 1	Тестовая процедура 2
MATS+	6,25	9,82
March C–	12,5	19,6

Изменения последовательности адресов влечет за собой существенное усложнение устройства, формирующего адреса ЗУ. Отметим, что на современном этапе развития методов тестирования ЗУ практически всегда задача тестирования решается с использованием средств самотестирования. Это, в свою очередь, предполагает дополнительные аппаратные затраты, в том числе на устройство, генерирующее последовательность адресов.

Как уже отмечалось ранее, в качестве последовательности адресов будем применять счетчиковую последовательность, а в качестве метода модификации адресной последовательности, используемой для второй и последующих тестовых процедур, рассмотрим изменение только начальных адресов в последующих тестовых процедурах.

3. Изменение начальных адресов при тестировании ЗУ

Как показано в работе [8], количество обнаруживаемых неисправностей маршевыми тестами не зависит от последовательности адресов, используемых при реализации теста. Происходит только их перераспределение между множеством обнаруживаемых и необнаруживаемых неисправностей. В качестве примера, иллюстрирующего данное утверждение, рассмотрим тестовую процедуру для MATS+ и March C–, когда начальный адрес i_1 для второй процедуры тестирования равен 1. Отметим, что для первой процедуры тестирования $i_0 = 0$ (табл. 5 и 6).

Таблица 5

Состояния ячеек ЗУ для i_1=1
при использовании маршевого теста MATS+

Фаза теста	Содержимое ЗУ							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
	0	0	0	0	0	0	0	0
↑ (*ra*, *wa**)	0	***1***	0	0	0	0	0	0
	0	1	***1***	0	0	0	0	0
	0	1	1	***1***	0	0	0	0
	0	1	1	1	***1***	0	0	0
	0	1	1	1	1	***1***	0	0
	0	1	1	1	1	1	***1***	0
	0	1	1	1	1	1	1	***1***
	1	1	1	1	1	1	1	1
↓ (*ra**, *wa*)	***0***	1	1	1	1	1	1	1
	0	1	1	1	1	1	1	***0***
	0	1	1	1	1	1	***0***	0
	0	1	1	1	1	***0***	0	0
	0	1	1	1	***0***	0	0	0
	0	1	1	***0***	0	0	0	0
	0	1	***0***	0	0	0	0	0
	0	***0***	0	0	0	0	0	0

Таблица 6

Состояния ячеек ЗУ для i_1=1
при использовании маршевого теста March C–

Фаза теста	Содержимое ЗУ							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
	0	0	0	0	0	0	0	0
↑ (*ra*, *wa**)	0	***1***	0	0	0	0	0	0
	0	1	***1***	0	0	0	0	0
	0	1	1	***1***	0	0	0	0
	0	1	1	1	***1***	0	0	0
	0	1	1	1	1	***1***	0	0
	0	1	1	1	1	1	***1***	0
	0	1	1	1	1	1	1	***1***
	1	1	1	1	1	1	1	1
↑ (*ra**, *wa*)	1	***0***	1	1	1	1	1	1
	1	0	***0***	1	1	1	1	1
	1	0	0	***0***	1	1	1	1
	1	0	0	0	***0***	1	1	1
	1	0	0	0	0	***0***	1	1
	1	0	0	0	0	0	***0***	1
	1	0	0	0	0	0	0	***0***
	0	0	0	0	0	0	0	0

Как и для случаев, рассмотренных в табл. 1 и 2 ($i_0 = 0$), для теста MATS+ получено одно состояние, а для теста March C– – два состояния в N–1 = 7 ячейках ЗУ для всевозможных фиксированных положений базовой ячейки. Для каждой базовой ячейки b, исключая случай, когда $b = a_0$, получаем новое значение в одной из соседних ячеек (ячейке a_0), что меняет содержимое множеств обнаруживаемых и необнаруживаемых *PPSFk*.

Например, для теста MATS+, базовой ячейки $b = a_1$ и начального адреса $i_0 = 0$ обнаруживаются два типа *PPSF5*, а именно 15 неисправностей $b_{i0}n_{i1}n_{i2}n_{i3}n_{i4} = b0000$ и 20 неисправностей $n_{i0}b_{i1}n_{i2}n_{i3}n_{i4} = 1b000$ (см. табл. 1). Для нового начального адреса $i_1 = 1$ (табл. 5) получаем 15 неисправностей $b_{i0}n_{i1}n_{i2}n_{i3}n_{i4} = b0000$ и 20 новых неисправностей $n_{i0}b_{i1}n_{i2}n_{i3}n_{i4} = 0b000$, которые не были обнаружены в результате первой процедуры тестирования тестом MATS+.

Если базовая ячейка имеет другую фиксированную позицию, предположим $b = a_3$, тогда для $i_0 = 0$ получаем четыре класса *PPSF5*. Они включают 1 неисправность $b_{i0}n_{i1}n_{i2}n_{i3}n_{i4} = b0000$, 12 неисправностей $n_{i0}b_{i1}n_{i2}n_{i3}n_{i4} = 1b000$, 18 неисправностей $n_{i0}n_{i1}b_{i2}n_{i3}n_{i4} = 11b00$ и 4 неисправности $n_{i0}n_{i1}n_{i2}b_{i3}n_{i4} = 111b0$. Для нового начального адреса $i_1 = 1$ получим другое распределение неисправностей (см. табл. 5). В этом случае имеем 1 неисправность $b_{i0}n_{i1}n_{i2}n_{i3}n_{i4} = b0000$, 12 неисправностей $n_{i0}b_{i1}n_{i2}n_{i3}n_{i4}$ (8 неисправностей 1b000 и 4 0b000), а также 18 неисправностей $n_{i0}n_{i1}b_{i2}n_{i3}n_{i4}$ (6 неисправностей 11b00 и 12 неисправностей 01b00) и 4 неисправности $n_{i0}n_{i1}n_{i2}b_{i3}n_{i4} = 011b0$. Для нового начального адреса $i_1 = 1$ и базовой ячейки $b = a_3$ количество новых неисправностей *PPSF5*, обнаруживаемых только во время второй процедуры тестирования *PPSF5*, равняется 20, как и для случая базовой ячейки $b = a_1$, и состоит из 4 неисправностей 0b000, 12 неисправностей 01b00 и 4 неисправностей 011b0. Совершенно другой результат получается, когда базовая ячейка $b = a_0$ и начальный адрес $i_1 = 1$. В этом случае все неисправности, обнаруживаемые в течение второй процедуры тестирования, отличаются от неисправностей, обнаруженных в течение первой процедуры тестирования тестом MATS+ для $i_0 = 0$. Действительно, во время первой процедуры обнаруживаются только неисправности *b*0000, а во время второй – только *b*1111. Принимая во внимание всевозможные позиции для базовой ячейки, общее количество неисправностей *PPSF5*, обнаруживаемых только во время второй процедуры тестирования, равняется 175.

Легко показать, что количество дополнительных неисправностей *PPSFk*, обнаруженных во время второй процедуры применения теста MATS+ с начальным адресом $i_1 = 1$, равно

$$(N-1)\binom{N-2}{k-2}+\binom{N-1}{k-1}. \qquad (6)$$

Для i_1=2 это значение будет вычисляться следующим образом:

$$(N-2)\left[\binom{2}{1}\binom{N-3}{k-2}+\binom{2}{2}\binom{N-3}{k-3}\right]+2\left[\binom{N-2}{k-1}+\binom{N-2}{k-2}\right]. \qquad (7)$$

Для i_1=3 получим выражение

$$\begin{gathered}(N-3)\left[\binom{3}{1}\binom{N-4}{k-2}+\binom{3}{2}\binom{N-4}{k-3}+\binom{3}{3}\binom{N-4}{k-4}\right]+\\ +3\left[\binom{N-3}{k-1}+\binom{2}{1}\binom{N-3}{k-2}+\binom{2}{2}\binom{N-3}{k-3}\right].\end{gathered} \qquad (8)$$

Отметим, что новый адрес i_1 в случае теста MATS+ делит все множество из *N* возможных базовых ячеек на две группы. Это деление реализуется в зависимости от расстояния $s = i_1 – i_0$ ($i_1 > i_0$) между начальными адресами i_1 и i_0. В соответствии с приведенной метрикой первая группа включает *N*–*s* ячеек с s различными битами в *N*–1 ячейках ЗУ (исключая базовую ячейку), вторая группа включает s ячеек с *N*–*s* различными значениями в соседних ячейках. Тогда общее количество $Q_1(PPSFk(s))$ неисправностей *PPSFk*, обнаруживаемых только во время второй процедуры реализации теста MATS+ для $N>>k$, определяется как

$$Q_1(PPSFk\,(i_1))=(N-s)\times\sum_{i=1}^{\min(s,(k-1))}\binom{s}{i}\binom{N-s-1}{k-i-1}+s\times\sum_{i}^{\min(s,(k-1))}\binom{s-1}{i-1}\binom{N-s}{k-i}. \qquad (9)$$

Данное выражение справедливо для любых начальных адресов, удовлетворяющих равенству $s = i_1 – i_0\,(i_1 > i_0)$.

Анализ последнего соотношения показывает, что количество $Q_1(PPSFk(s))$ неисправностей, обнаруживаемых в течение второй тестовой процедуры, зависит только от значения *s* и достигает максимума при $s = N/2$. Такой же результат может быть получен при максимизации количества новых значений в соседних *N*–1 ячейках для всевозможных фиксированных положений базовой ячейки или при максимизации суммы Хэмминговых расстояний *S_HD* для всевозможных положений базовой ячейки в двух последовательных процедурах тестирования для начальных адресов i_0 и i_1. Значение *S_HD*(*s*) вычисляется как *S_HD*(*s*) = *s*(*N*–*s*) + (*N*–*s*)*s* = 2*s*(*N*–*s*). Тогда максимум значения S_HD(*s*) будет достигаться для величины *s*, значение которой вычисляется из соотношения

$$\frac{\partial\ (S\_HD(s))}{\partial\ s}=\frac{\partial\ (2s(N-s))}{\partial\ s}=(2N-4s)=0. \qquad (10)$$

В результате получим, что $s = N/2$, т. е. максимальная покрывающая способность при проведении только двух процедур тестирования будет достигнута путем использования начальных адресов, удовлетворяющих соотношению $i_1 – i_0 = N/2$.

Для общего случая, когда тестирование ЗУ проводится в результате выполнения q последовательных процедур тестирования, оптимальное соотношение начальных адресов $i_0, i_1, i_2, ..., i_q$, где $i_0<i_1<i_2<...<i_q$ для каждой из процедур, будет определяться максимальным значением $S\_HD(q)$, где $S\_HD(q)$ есть сумма Хэмминговых расстояний $S\_HD(s)$ для всевозможных положений базовой ячейки во всех парах последовательных процедур тестирования. Другими словами, $S\_HD(q) = S\_HD(s_1) + S\_HD(s_2) + ... + S\_HD(s_q) + S\_HD(s_0)$, где $s_i = i_i – i_{i–1}$, $i \in \{1, 2, …, q\}$, а $s_0 = s_1+s_2+…+s_q$. Окончательно для $S\_HD(q)$ получим

$$S\_HD\ (q) = 2\sum_{i=0}^{q} s_i (N - s_i). \qquad (11)$$

Для трехкратного применения маршевых тестов сумма Хэмминговых расстояний

$$S\_HD(q) = 4s_1N + 4s_2N - 4s_1^2 - 4s_2^2 - 4s_1s_2. \qquad (12)$$

Данное выражение достигает максимума при $s_1 = s_2 = \frac{1}{3}N$.

Таким образом, максимальная покрывающая способность трехпроходных маршевых тестов типа MATS+ достигается тогда, когда разница между начальными адресами двух следующих друг за другом тестов равна одной третьей от всего размера памяти.

4. Результаты практических и вычислительных экспериментов

Для подтверждения полученных выше выражений был проведен ряд вычислительных и практических экспериментов. Для них использовалось такое построение памяти, в котором моделировалась кодочувствительная неисправность с заданным числом влияющих ячеек (*PPSFk*). Во время эксперимента случайным образом выбиралась базовая ячейка и k–1 соседних (влияющих) ячеек со случайно выбранным кодом, после чего применялись однократно, двукратно, трехкратно и N-кратно тесты MATS+ и March C– и фиксировалось, обнаружил ли данный тест эту неисправность. После проведения миллиона таких экспериментов выявлялись процентные отношения обнаруженных тестом кодочувствительных неисправностей памяти ко всем смоделированным неисправностям.

Экспериментальные результаты представлены в табл. 7–11. В табл. 7 приведены результаты экспериментов при однократном применении маршевых тестов MATS+ и первых двух фаз с последующей фазой чтения маршевого теста March C– для разных значений k.

Таблица 7

Покрывающая способность однократного применения тестов, %

k	2	3	4	5	6
MATS+	50	25	12,5	6,2	3
March C–	100	50	25	12,5	6,2

В табл. 8–10 приведены экспериментальные данные для двукратного применения маршевого теста MATS+ и первых двух фаз с последующей фазой чтения маршевого теста March C– для $k = 5$ и разного размера памяти N. Символом s обозначена разница между начальными адресами первого и второго применений теста ($s = i_1–i_0$), причем в табл. 9 и 10 для $N>>k$ включен дополнительный столбец MATS+т, который содержит теоретические результаты для маршевого теста MATS+, полученные по формуле (9) для конкретного размера памяти N и типа кодочувствительной неисправности.

Таблица 8

Покрывающая способность двукратного применения тестов для N=8, %

s	0	1	2	3	4	5	6	7
MATS+	6,2	10,1	11,76	12,3	**12,4**	12,3	11,76	10,1
March C–	12,4	18,5	22,20	23,9	**24,5**	23,9	22,2	18,5

Таблица 9

Покрывающая способность двукратного применения тестов для N=16, %

s	MATS+	MATS+т	MarchC–	s	MATS+	MATS+т	MarchC–
0	6,12	6,25	12,50	8	**12,33**	**12,34**	**23,90**
1	8,10	8,2031	15,50	9	12,20	12,29	23,50
2	9,50	9,6354	18	10	12,00	12,131	23,15
3	10,60	10,658	20	11	11,70	11,838	22,50
4	11,30	11,367	21,40	12	11,30	11,367	21,40
5	11,70	11,838	22,50	13	10,60	10,663	20,00
6	12,00	12,131	23,15	14	9,50	9,6805	18,00
7	12,20	12,29	23,50	15	8,10	8,5546	15,50

Таблица 10

Покрывающая способность двукратного применения тестов для N=32, %

s	MATS+	MATS+т	March C–	s	MATS+	MATS+т	March C–	s	MATS+	MATS+т	March C–
0	6,20	6,25	12,50	11	11,70	11,854	22,60	22	11,60	11,675	22,20
1	7,20	7,2266	14	12	11,85	11,994	23	23	11,30	11,452	21,75
2	8,00	8,0771	15,40	13	11,95	12,099	23,20	24	11,00	11,179	21,10
3	8,70	8,8143	16,60	14	12	12,172	23,40	25	10,70	10,85	20,40
4	9,30	9,4497	17,80	15	12,05	12,215	23,48	26	10,40	10,458	19,60
5	10	9,9944	18,80	16	**12,07**	**12,229**	**23,53**	27	10	9,9944	18,80
6	10,40	10,458	19,60	17	12,05	12,215	23,48	28	9,30	9,4497	17,80
7	10,70	10,85	20,40	18	12	12,172	23,40	29	8,70	8,8145	16,60
8	11,00	11,179	21,10	19	11,95	12,099	23,20	30	8,00	8,0817	15,40
9	11,30	11,452	21,75	20	11,85	11,994	23	31	7,20	7,2988	14,00
10	11,60	11,675	22,20	21	11,70	11,854	22,60				

Из табл. 8–10 видно, что наибольшая эффективность двукратного применения маршевых тестов достигается в том случае, когда разница между первым и вторым начальными адресами равна $N/2$.

В табл. 11 приведены данные трехкратного применения тех же маршевых тестов для $N = 8$. Символами s_1 и s_2 обозначается разность между начальными адресами соответственно первого и второго, второго и третьего проходов маршевых тестов ($s_1 = i_1 – i_0$, $s_2 = i_2 – i_1$).

Таблица 11

Покрывающая способность трехкратного применения тестов для N=8, %

s_1	s_2	MATS+	March C–	s_1	s_2	MATS+	March C–	s_1	s_2	MATS+	March C–
1	1	14	24,87	2	2	17,33	31,90	3	4	16,20	30,10
1	2	15,67	28,37	2	3	**17,70**	**32,88**	4	1	16,20	29,80
1	3	16,13	29,87	2	4	17,33	31,90	4	2	17,30	31,90
1	4	16,13	29,90	2	5	15,60	28,66	4	3	16,20	30,15
1	5	15,60	28,40	3	1	16,20	30,10	5	1	15,60	28,66
1	6	14	24,90	3	2	**17,70**	**32,90**	5	2	15,70	28,64
2	1	15,60	28,30	3	3	17,70	32,91	6	1	14,00	27,87

В результате получили, что максимальная покрывающая способность достигается при $s_1 = 2$, $s_2 = 3$ или, наоборот, при $s_1 = 3$, $s_2 = 2$. Это подтверждает полученное выше выражение (12).

В табл. 12 приведены данные для N-кратного применения маршевого теста MATS+ и первых двух фаз маршевого теста March C– с последующей фазой чтения для $N = 16$.

Таблица 12

Покрывающая способность N-кратного применения тестов для N=16, %

k	2	3	4	5	6	7
MATS+	100	75	50	31,10	18,40	11
March C–	100	100	75	50	31	18,20

Из табл. 12 видно, что 100%-я покрывающая способность обнаружения кодочувствительных неисправностей при использовании маршевых тестов достигается только для *PPSF2* (MATS+ и March C–) и *PPSF3* (March C–).

Заключение

Покрывающая способность многопроходных маршевых тестов при обнаружении кодочувствительных неисправностей памяти зависит от выбора начального адреса повторного применения теста. Адрес должен выбираться с таким расчетом, чтобы для любой базовой ячейки код во всевозможных влияющих ячейках максимально отличался от того кода, который был при предыдущем применении теста. Максимальное отличие будет достигаться в том случае, когда во всевозможных k–1 влияющих ячейках (каковыми могут быть любые k–1 из N–1) будут сформированы двоичные комбинации с максимальным Хэмминговым расстоянием. Это условие выполняется для двухпроходных маршевых тестов, когда разница между начальными адресами первого и второго применения тестов равна одной второй от всего размера памяти. Аналогичные данные получаются и для трехпроходных маршевых тестов, когда максимальная покрывающая способность этих тестов достигается при такой разнице между начальными адресами маршевых тестов, которая равна одной третьей от всего размера памяти.

В общем случае получаем, что для достижения наилучших результатов во время тестирования памяти при многократном использовании маршевых тестов разница между начальными адресами этих тестов должна быть равна $\frac{1}{q}$ от всего объема памяти, где q – количество проходов маршевых тестов. Однако при многократном применении маршевых тестов нельзя достигнуть 100%-й покрывающей способности для кодочувствительных неисправностей памяти при k>2 для MATS+ и k>3 для March C–, применяя только счетчиковую последовательность адресов.

Список литературы

1. Goor A.J. van de. Testing Semiconductor Memories, Theory and Practice. – Chichester, UK: John Wiley & Sons, 1991.

2. Karpovsky M.G., Yarmolik V.N. Transparent Memory Testing for Pattern Sensitive Faults // IEEE International Test Conference. – Washington, 1994. – P. 860–869.

3. Karpovsky M.G., Yarmolik V.N., Goor A.J. van de. Pseudo-Exhaustive Word-Oriented DRAM Testing // European Test Conference. – Munich, 1995. – P. 126–132.

4. Ярмолик С.В., Ярмолик В.Н. Эффективность многократного применения маршевых тестов для выявления кодочувствительных неисправностей // Известия Белорусской инженерной академии. – 2005. – № 1(91)/1. – С. 79–83.

5. Nicolaidis M. Theory of Transparent BIST for RAMs // IEEE Transaction of Computers. – Vol. 45. – № 10. – 1996.

6. Yarmolik V.N., Sokol B., Yarmolik S.V. Counter sequences for memory test address generation // Proc. 12th International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES 2005). – Krakow, 2005. – P. 413–418.

7. Mrozek I., Yarmolik V.N. Detection of Pattern Sensitive Faults by Multiple Transparent March Tests // Proc. 10th International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES'03). – Lodz, 2003 – P. 542–545.

8. Niggemeyer D., Otterstedt J., Redeker M. Detection of Non-classical Memory Faults using Degrees of Freedom in March Testing // Rec. 11th Workshop «Testmethods and Reliability of Circuits and Systems». – Potsdam, 1999.

Поступила 22.09.05

Белорусский государственный университет
информатики и радиоэлектроники,
Минск, П. Бровки, 6
e-mail: yarmoli@bsuir.unibel.by

S.V. Yarmolik,V.N. Yarmolik

MEMORY PATTERN SENSITIVE FAULTS DETECTION USING MULPIPLE RUNS OF MARCH TESTS

This paper deals with memory pattern sensitive faults detection problem. It shows the efficiency of using march tests for detection memory passive sensitive faults. It shows also the efficiency of multiple runs of march tests for detection PPSF and analyzes the optimal address seeds. At the conclusion the experimental data, showing the fault coverage of multiple uses of march tests are given.