

ИНФОРМАТИКА

УДК 681.326.7

СИММЕТРИЧНОЕ НЕРАЗРУШАЮЩЕЕ ТЕСТИРОВАНИЕ ОЗУ

А.П. ЗАНКОВИЧ, В.Н. ЯРМОЛИК

*Белорусский государственный университет информатики и радиоэлектроники
П. Бровка, 6, Минск, 220013, Беларусь**Поступила в редакцию 25 апреля 2005*

В статье представлены локально-симметричные методы периодического неразрушающего тестирования схем ОЗУ. Приведена оценка эффективности использования предлагаемых методов в сравнении с существующими методами Николаидиса и глобально-симметричными методами.

Ключевые слова: неразрушающее тестирование ОЗУ, маршевые тесты, условия скрытия и проявления ошибок, симметричное тестирование.

Введение

В настоящее время быстро прогрессирующие технологии производства полупроводниковых приборов позволяют создавать схемы ОЗУ с высокой степенью интеграции. Следствием использования новых технологий является повышение вероятности возникновения неисправностей. Для обеспечения надежного функционирования цифровых систем большое внимание должно быть уделено своевременному обнаружению неисправностей памяти, поскольку результаты исследований [1, 2] говорят о том, что отказы ОЗУ составляют до 70% общего числа отказов системы в целом.

Помимо традиционного заводского контроля на этапе производства, в настоящее время широко применяется встроенная аппаратура самотестирования (ВАСТ) [3, 4]. Такой подход позволяет периодически выполнять тестирование на внутренней частоте без применения внешнего оборудования. Большинство ВАСТ реализуют тесты сложности $O(N)$ (N – емкость запоминающего устройства), называемые маршевыми тестами [5]. При стандартном маршевом тестировании со всеми ячейками памяти в определенном порядке (\uparrow — от младших адресов к старшим, \downarrow — наоборот, \updownarrow — в любом направлении) выполняются операции, задаваемые несколькими маршевыми элементами. Используются операции: $w0, w1$ — запись в элемент памяти значений 0 или 1; $r0, r1$ — чтение текущего значения из элемента памяти и сравнение его со значением 0 или 1. Достоинствами маршевых тестов являются относительно высокая скорость выполнения и покрывающая способность, а также простота реализации ВАСТ. Главным их недостатком считается полная потеря хранящейся в памяти информации, что ограничивает применение классических маршевых тестов только моментами начального тестирования устройства. Для надежной работы постоянно включенных систем, помимо методов с использованием избыточных данных (коды Хэмминга, контроль на четность), применяются методы периодического неразрушающего тестирования [6–9], основанного на классических маршевых алгоритмах. В данной статье представлены новые локально-симметричные неразрушающие тесты ОЗУ.

Существующие неразрушающие тесты ОЗУ

Основным требованием, предъявляемым к неразрушающим тестам, является необходимость восстановления исходного состояния объекта тестирования после выполнения процедуры тестирования. Поэтому все тестовые воздействия, направленные на активизацию неисправностей и проявление их в виде ошибок на выходах схемы, должны носить обратимый характер. Простейшим способом реализации этого требования для случая неразрушающего тестирования памяти является сохранение всего ее содержимого в резервном буфере с последующим выполнением одного из классических разрушающих алгоритмов. Хотя этот способ позволяет использовать существующие ВАСТ, он потребует дублирования массива запоминающих элементов, а также дополнительного времени для сохранения и последующего восстановления содержимого памяти. Поэтому для большинства приложений он непригоден.

Вместо него в большинстве случаев используют методы, основанные на преобразовании классических маршевых тестов к неразрушающему виду. Алгоритм такого преобразования предложен М.Николаидисом в работе [4]. В результате все маршевые операции теста заменяются на неразрушающие аналоги. Операции чтения помещают значения из памяти в прямом (r_a) или обратном ($r_{\bar{a}}$) коде во временный буфер, размер которого равен размеру слова памяти. При реализации операций записи (w_a и $w_{\bar{a}}$) данные помещаются соответственно в прямом или обратном коде из буфера в текущую ячейку памяти. Таким образом, активизация неисправностей в этом случае производится путем обратимого инвертирования содержимого ячеек ОЗУ.

Примером теста, полученного по методу Николаидиса, является *TMarch C*-. Он состоит из двух частей: $\{\uparrow(r_a); \uparrow(r_a); \downarrow(r_a); \downarrow(r_a); \Downarrow(r_a)\}$ и $\{\uparrow(r_{\bar{a}}, w_{\bar{a}}); \uparrow(r_{\bar{a}}, w_{\bar{a}}); \downarrow(r_{\bar{a}}, w_{\bar{a}}); \downarrow(r_{\bar{a}}, w_{\bar{a}}); \Downarrow(r_{\bar{a}})\}$. Во время выполнения первой части неисправности не активизируются, что позволяет сформировать эталонную сигнатуру прочитанных данных. Вторая часть — базовый неразрушающий алгоритм — активизирует неисправности и формирует рабочую сигнатуру. Совпадение эталонной и рабочей сигнатур говорит об исправности ОЗУ.

Для устранения основного недостатка неразрушающего тестирования по методу Николаидиса, связанного с существенным (до 30%) увеличением сложности алгоритма, был предложен новый метод [7]. Его основная идея заключается в разбиении базового неразрушающего теста на две части, считывающие симметричные последовательности данных. Неисправности схемы памяти проявляют себя в виде нарушений симметрии этих последовательностей, что легко обнаружить, вычислив и сравнив в конце алгоритма их сигнатуры. Пусть $D = (D_0, \dots, D_{N-1})$ — некоторая последовательность данных. Обратной назовем последовательность $D^* = (D_{N-1}, \dots, D_0)$, инверсной — последовательность $\bar{D} = (\bar{D}_0, \dots, \bar{D}_{N-1})$, а обратной инверсной — последовательность $\bar{D}^* = (\bar{D}_{N-1}, \dots, \bar{D}_0)$. Всего возможны четыре типа комбинаций симметричных последовательностей данных (рис. 1).

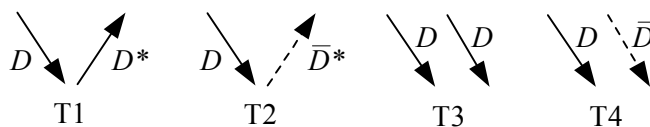


Рис. 1. Типы симметрии последовательностей данных

В качестве основы неразрушающего алгоритма в [7] берется базовый неразрушающий тест. Он разделяется на две половины, операции чтения которых должны генерировать взаимосимметричные последовательности данных. Этого можно добиться, вводя в маршевый алгоритм дополнительные операции чтения. Например, при добавлении в начало базового неразрушающего теста *TMarch C*- маршевого элемента $\downarrow(r_a)$ получается симметричный тест *UDMarch C*-. $\downarrow(r_a); \uparrow(r_a, w_{\bar{a}}); \uparrow(r_{\bar{a}}, w_a); \downarrow(r_a, w_{\bar{a}}); \downarrow(r_{\bar{a}}, w_a); \uparrow(r_a)$. Вертикальной линией помечена граница симметричных последовательностей, добавленная операция подчеркнута.

Локально-симметричные тесты ОЗУ

Главным недостатком описанных выше алгоритмов является то, что они рассматривают покрывающую способность построенных алгоритмов только после их получения. То есть при создании алгоритма в первую очередь рассматриваются вопросы придания ему определенной структуры (добавление начального алгоритма вычисления эталонной сигнатуры, приведение алгоритма к симметричному виду), что приводит, как правило, к его необоснованному усложнению. Авторами предложена методика анализа поведения неисправностей в процессе неразрушающего маршевого тестирования [10], которая позволяет построить алгоритмы с теоретически минимальной сложностью. Она основана на выделении условий проявления и скрытия неисправностей и выделении в тесте симметричных элементов. Симметричным элементом назовем два набора операций чтения маршевого неразрушающего теста, считывающих из ОЗУ взаимосимметричные последовательности данных.

Модификация значений памяти в процессе выполнения базового неразрушающего алгоритма выполняется путем инвертирования содержимого. Инвертирование исправной ячейки означает запись в нее противоположного значения. Неисправность может проявить себя сохранением прежнего состояния или изменением другой ячейки. Наведенное в результате ошибочное значение считывается и модифицируется последующими операциями до тех пор, пока повторная активация неисправности не установит в зависимой ячейке такое значение, какое бы она имела в случае отсутствия неисправности, т.е. произойдет скрытие ошибки.

Рассмотрим поведение ячейки с константной неисправностью SAF — логическое значение такой ячейки всегда равно 0 (SAF0) или 1 (SAF1), независимо от операций, производимых с этой или другими ячейками памяти. Условие обнаруживающего проявления константной неисправности SAF неразрушающим тестом — инверсия неисправной ячейки: следующая операция чтения вернет ошибочное значение. Прочитанное после повторного инвертирования значение совпадет со значением, которое было бы прочитано из исправной памяти. В общем случае, любое четное количество инвертирований наводит ошибку в неисправной ячейке, а нечетное скрывает ошибку.

Рассуждая аналогичным образом, авторы работы [9] построили условия проявляющего и скрывающего проявления для наиболее вероятных неисправностей ОЗУ — константных, переходных, адресных и неисправностей взаимного влияния. Это позволило определить, какие операции оперируют с ошибочными данными в процессе тестирования. На рис. 2 эти операции подчеркнуты темными линиями.

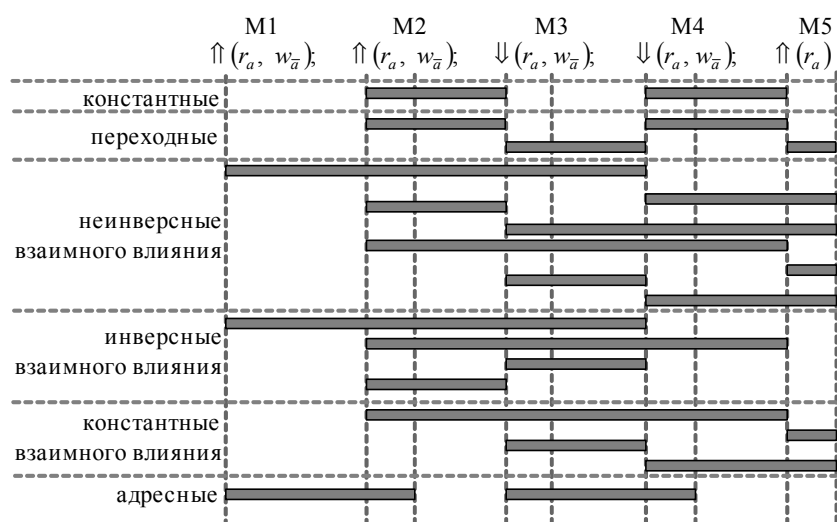


Рис. 2. Схема проявления неисправностей для неразрушающего теста $T_{March} C$

При рассмотрении неразрушающих тестов нельзя говорить о проявлении неисправности в виде ошибки с фиксированным значением. В этом случае ошибка определяется по результату сравнения содержимого ОЗУ до и после выполнения тестом активизирующих неисправность действий. Для симметричных неразрушающих тестов это означает нарушение симметрии

считываемых из ОЗУ последовательностей данных. Это нарушение можно определить на схеме проявления неисправности по отсутствию симметрии подчеркнутых операций. Для выделенного симметричного элемента, если подчеркнутая операция чтения из одной его симметричной стадии имеет неподчеркнутую пару во второй стадии, неисправность может быть обнаружена. В простейшем случае симметричный элемент могут составлять две одиночные операции чтения. Например, операции чтения из первого и второго маршевого элементов неразрушающего теста *TMarch C*- образуют симметричный элемент, позволяющий обнаруживать все константные неисправности SAF. Аналогичное утверждение справедливо и для симметричного элемента более сложной конфигурации, одну стадию которого образуют маршевые операции чтения из M_1 и M_2 , а вторую – из M_3 и M_4 .

Для определения обнаруживающей ошибки способности теста необходимо рассмотреть все случаи вызванных неисправностями нарушений симметрии считываемых последовательностей данных. Например, если выделить в *TMarch C*- три симметричных элемента на основе операций чтения маршевых элементов M_1 и M_2 (считываются потоки данных с симметрией типа T4), M_3 и M_4 (симметрия типа T4), M_4 и M_5 (симметрия типа T2), то это не позволит обнаружить все сгенерированные ошибки. Это произойдет из-за того, что ошибочные данные, вызванные частью неисправностей взаимного влияния, считываются маршевыми элементами M_3 , M_4 и M_5 и не считываются элементами M_1 и M_2 . Указанного недостатка можно избежать, выбрав другой набор симметричных элементов, например сформировав его из операций чтения маршевых элементов (M_1 , M_2) и (M_3 , M_4) (симметрия типа T2), M_4 и M_5 (симметрия типа T2).

Для записи локально-симметричных алгоритмов дополним описанную выше запись маршевых тестов следующим образом. Верхний индекс операций чтения будет содержать список номеров симметричных элементов, включающих данную операцию. Границу первого и второго симметричного потоков отметим вертикальной чертой. Нижний ее индекс задает тип симметрии образованного симметричного элемента, а верхний индекс – его порядковый номер. Тогда рассмотренный выше алгоритм запишется так:

$$\uparrow(r_a^{[1]}, w_a) \uparrow(r_a^{[1]}, w_a) \Big|_{T2}^{[1]} \Downarrow(r_a^{[1]}, w_a) \Downarrow(r_a^{[1,2]}, w_a) \Big|_{T2}^{[2]} \uparrow(r_a^{[2]}).$$

Оценка эффективности использования рассмотренных методов

На рис. 3 представлены наиболее популярные алгоритмы неразрушающего тестирования, построенные с помощью существующих и предлагаемого выше методов.

Одним из важнейших параметров создаваемых алгоритмов является их сложность, оцениваемая по суммарному количеству операций в алгоритме. На рис. 4,а видно, что локально-симметричные алгоритмы обладают меньшей сложностью, особенно для длинных тестов. Это связано с тем, что в данном случае необходимо добавление большого количества дополнительных операций, приводящих его к глобально-симметричному виду, чего можно избежать, рассматривая локальную симметрию.

Еще одним критерием оценки эффективности алгоритмов тестирования является их покрывающая способность. В случае неразрушающего тестирования ее можно рассматривать с двух сторон. С одной стороны, следует рассмотреть способность алгоритма активизировать некоторый набор неисправностей – его генерирующую способность, с другой — созданный алгоритм должен однозначно определять ошибочность прочитанного значения, т.е. обладать обнаруживающей способностью.

Поскольку при тестировании по методу Николаидиса активизация неисправностей производится только в процессе работы базового неразрушающего алгоритма, только он обладает генерирующей способностью. А так как он был получен из разрушающего маршевого алгоритма путем выполнения преобразований, не влияющих на покрывающую способность, то их покрывающие способности совпадают [6]. Это утверждение распространяется также на глобально- и локально-симметричные алгоритмы, поскольку в их основу положены базовые алгоритмы Николаидиса.

		Алгоритм
MATS	1	$\{\uparrow(r_a); \downarrow(r_a)\} \{\uparrow(r_a, w_a); \downarrow(r_a)\}$
	2	$\uparrow(r_a, w_a); \downarrow(r_a)$
	3	$\downarrow(r_a^{[1]}) \Big _{T1}^{[1]} \uparrow(r_a^{[1,2]}, w_a) \Big _{T1}^{[2]} \downarrow(r_a^{[2]})$
MATS++	1	$\{\uparrow(r_a); \downarrow(r_a, r_a)\} \{\uparrow(r_a, w_a); \downarrow(r_a, w_a, r_a)\}$
	2	$\uparrow(r_a, r_a, w_a); \downarrow(r_a, w_a, r_a)$
	3	$\downarrow(r_a^{[1]}) \Big _{T3}^{[1]} \uparrow(r_a^{[2]}, w_a) \Big _{T1}^{[2]} \downarrow(r_a^{[1]}, w_a, r_a^{[2]})$
March Y	1	$\{\uparrow(r_a, r_a); \downarrow(r_a, r_a); \uparrow(r_a)\} \{\uparrow(r_a, w_a, r_a); \downarrow(r_a, w_a, r_a); \uparrow(r_a)\}$
	2	$\downarrow(r_a); \uparrow(r_a, r_a, w_a, r_a) \Big \downarrow(r_a, r_a, w_a, r_a); \uparrow(r_a)$
	3	$\uparrow(r_a^{[1,2]}) \Big _{T1}^{[1]} \uparrow(r_a, w_a, r_a); \downarrow(r_a^{[1]}, w_a, r_a) \Big _{T1}^{[2]} \downarrow(r_a^{[2]})$
March C-	1	$\{\uparrow(r_a); \uparrow(r_a) \downarrow(r_a); \downarrow(r_a); \downarrow(r_a)\} \{\uparrow(r_a, w_a); \uparrow(r_a, w_a) \downarrow(r_a, w_a); \downarrow(r_a, w_a); \downarrow(r_a)\}$
	2	$\uparrow(r_a); \uparrow(r_a, w_a); \uparrow(r_a, w_a) \Big \downarrow(r_a, w_a); \downarrow(r_a, w_a); \downarrow(r_a)$
	3	$\uparrow(r_a^{[1]}, w_a); \uparrow(r_a^{[1]}, w_a) \Big _{T2}^{[1]} \downarrow(r_a^{[1]}, w_a); \downarrow(r_a^{[1,2]}, w_a) \Big _{T2}^{[2]} \uparrow(r_a^{[2]})$

Рис. 3. Алгоритмы неразрушающего тестирования: 1 — метод Николаидиса; 2 — глобально-симметричные; 3 — локально-симметричные

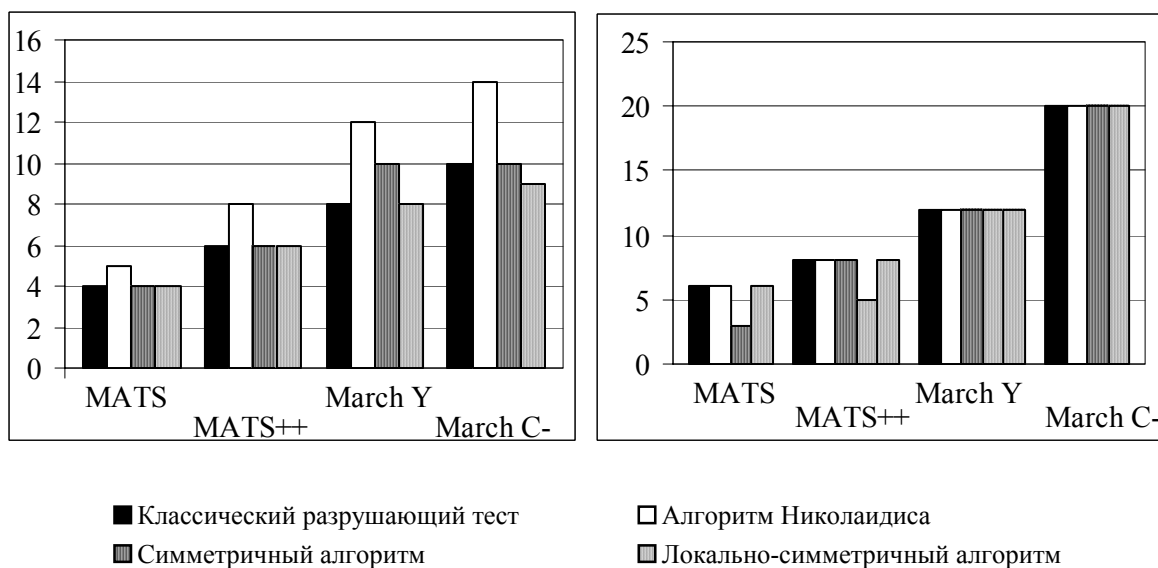


Рис. 4. Эффективность неразрушающих алгоритмов: а) сложность алгоритма; б) количество обнаруживаемых неисправностей

Обнаружение сгенерированных ошибок при тестировании по алгоритму Николаидиса производится при сравнении эталонной и рабочей сигнатур. Для рассматриваемых неисправностей ошибки гарантированно искажат только вторую из них, и, следовательно, все могут быть обнаружены. На обнаруживающую способность симметричных алгоритмов влияет выбор позиций, в которые добавляются корректирующие симметрию операции чтения. Дополнительная операция в самом начале алгоритма более предпочтительна, поскольку она выполняется перед активизацией большинства неисправностей. Однако в некоторых случаях (например, для алгоритма MATS) приведение к симметричному виду требует большого количества дополнитель-

ных операций. Этого можно избежать только за счет снижения обнаруживающей способности (рис. 4,б).

При периодическом выполнении неразрушающих тестов ОЗУ в работающей системе значения в запоминающих ячейках будут меняться в силу изменения набора запущенных программ, их состояния и обрабатываемых данных. Поскольку условия обнаружения многих неисправностей отличаются только начальными значениями ячеек, периодический запуск неразрушающего теста, удовлетворяющего только части условий, в системе с изменяющимся содержанием ОЗУ позволяет значительно повысить обнаруживающую способность теста.

Заключение

Сравнительный анализ предлагаемых локально-симметричных тестов ОЗУ показал высокую эффективность их применения. Обладая сравнимой с существующими неразрушающими тестами ОЗУ покрывающей способностью, они имеют меньшую алгоритмическую сложность, что сокращает общее время тестирования.

SYMMETRIC TRANSPARENT MEMORY TESTING

A.P. ZANKOVICH, V.N. YARMOLIK

Abstract

Novel method of periodical transparent memory testing is presented in the article. It is based on analysis of local symmetry in data read by March test. Comparison studie of proposed method and wildy accepted test of Nicolaidis and global symmetry methods is presented in the article.

Литература

1. *Prince B.* High-performance memories: new architecture DRAM's and SRAM's, evolution and function. Gr.Britain: John Wiley & Sons, 1996.
2. *Goor A.J. van de.* Testing Semiconductor Memories, Theory and Practice. Gr.Britain: John Wiley & Sons, 1991.
3. *McCluskey E.J.* // IEEE Design & Test of Computers. 1985. Vol. 2, N. 2. P. 21–28.
4. *Nicolaidis M.* // Proc. of Intern. Test Conf., USA. 1992. P. 598–606.
5. *Suk D., Reddy S.* // IEEE Trans. on Computers, 1981. Vol. C-30, N. 12. P. 982–985.
6. *Nicolaidis M.* // IEEE Trans. on Computers. 1996. Vol. 45, N. 10, P. 1141–1155.
7. *Yarmolik V.N., Hellebrand S., Wundelich H.-J.* // Proc. of the DATE'99 Conference, 1999. P. 702–707.
8. *Ярмолик В.Н., Калоша Е.П., Быков Ю.В. и др.* Проектирование самотестируемых СБИС // В 2 т. Мн.: БГУИР, 2001. Т. 2
9. *Занкович А.П., Ярмолик В.Н.* // Автоматика и телемеханика. М., 2003. № 9. С. 141–154.
10. *Yarmolik V.N., Zankovich A.P., Ivaniuk A.A.* // Proc. of Int. Conf. On MixDes, Poland. 2002. P. 545–548.