6. Floudas, C. A., Pardalos, P. M. (1990). A collection of Test Problems for Constrained Global Optimization Algorithms. Berlin Helldelberg: Springer-Verlag, 193.

Дата надходження рукопису 26.11.2014

Косолап Анатолий Иванович, доктор физико-математических наук, профессор, кафедра специализированных компьютерных систем, Украинский государственный химико-технологический университет, пр. Гагарина, 8, г. Днепропетровск, Украина, 49005
E-mail: anivkos@ua.fm

УДК 681.327
DOI: 10.15587/2313-8416.2014.34596

ОЦЕНКА ДИАГНОСТИЧЕСКИХ СВОЙСТВ ТЕСТА ПРОВЕРКИ ОПЕРАТИВНОЙ ПАМЯТИ March_ DSS

© Т. Ю. Потопахина, А. И. Сивальнева, А. Ю. Попова, И. Г. Либерг

Предлагаются исследования диагностической способности тестовой последовательности March_DSS для обнаружения неисправностей оперативной памяти. Представлены результаты оценивания диагностического качества теста March_DSS по сравнению с другими алгоритмами проверки на предмет обнаружения кратных неисправностей. Предлагаются рекомендации к применению исследованных тестовых последовательностей
Ключевые слова: *неисправности оперативной памяти, тестовые последовательности «March», оценка диагностической способности*

Researches of diagnostic ability of test sequence March DSS for detection of failure of a random access memory are offered. Analysis results of the March DSS test in comparison with other algorithms of check regarding detection of the multiple failures are provided. The recommendations are offered for use of investigated test sequences
Keywords: *RAM faults, March test sequences, evaluation of diagnostic opportunities*

1. Введение

В настоящее время наблюдается повышенный интерес к проблемам диагностирования оперативной памяти (ОЗУ, RAM). Этот интерес связан с тем, что ОЗУ является доминирующим компонентом как в составе современных компьютерных систем – внешняя оперативная память, так и в составе встроенных систем System on Chip (SoC) – встроенная оперативная память. Для диагностирования ОЗУ используются, как правило, тесты линейной длины семейства «Марш» («March»). В настоящей статье производится анализ теста March_ DSS с целью оценки его диагностических свойств.

2. Постановка задачи

До настоящего времени были предложены ряд моделей неисправностей оперативной памяти и тестовые последовательности для их обнаружения [1, 2]. Как правило, эти тесты предназначены для обнаружения неисправностей в рамках принятой модели. Одним из последних достижений в области диагностирования запоминающих устройств является тест March_ DSS, на который получен патент США [3, 4]. При этом остается открытым вопрос о сравнении данного теста с другими тестовыми последовательностями на предмет оценки диагностических свойств обнаружения неисправностей оперативной памяти.

2. Анализ литературных данных

Классической работой в области диагностирования ОЗУ является книга Ван де Гора [5]. Применение маршевых тестов исследовано в работе [1]. Актуальность использования тестов семейства «Марш» для современных систем System-on-Chip подтверждено в работе [3]. В патенте США [4] предлагается универсальная тестовая последовательность March_DSS. Попытка качественного анализа диагностических свойств этого теста сделана в работе [5]. В данной работе исследуются диагностические свойства теста March_ DSS с точки зрения процента обнаруженных неисправностей.

3. Цель статьи

В данной работе проводится оценка диагностических свойств теста March_DSS с помощью программного пакета RAMST, специально разработанного на кафедре АУТС НТУ «ХПИ» для моделирования и верификации обнаруживающей способности проверяющих тестов на предмет диагностирования расширенного множества одиночных и кратных неисправностей.

Под оценкой диагностических свойств мы будем понимать критерий качества теста определяемый по двум параметрам:

– во-первых, это процент обнаруженных неисправностей в рамках принятой модели;

– во-вторых, это временные затраты на тестирование памяти, которое масштабируется количеством обращений к ячейкам памяти в процессе генерации теста в зависимости от ее емкости N (параметр N – количество ячеек памяти).

4. Модели неисправностей

При описании моделей несправностей общепринятыми являются следующие обозначения [5]:

– операция ↑ обозначает переход ячейки памяти в состояние логической «1»;

– операция ↓ обозначает переход ячейки памяти в состояние логического «0»;

– операция ↕ обозначает любое изменение состояния ячейки памяти;

– операция $\forall$ обозначает выполнение любых действий по отношении к ячейке памяти;

– выражение <S/F> описывает значение или операцию S, которая приводит к возникновению в ячейки памяти неправильного значения F, где S $\in$ {0, 1, ↑, ↓ ,↕}, a F $\in$ {0,1};

– выражение <S_i, S_j/F> описывает влияние ячейки памяти по адресу i на ячейку памяти по адресу j таким образом, что ячейка по адресу j принимает неправильное значение F вместо значения S_j.

Общепринятыми являются следующие модели неисправностей ячеек памяти:

1) константные неисправности ячеек памяти (stuck-at fault, SAF) – ячейка памяти находится в одном и том же логическом состоянии «0» или «1», т. е. неисправность константа ноль для данной ячейки может быть обозначена как <$\forall$/0>, а константа единица – <$\forall$/1>;

2) неисправности переключений ячеек памяти (transient faults, TF), которые могут быть обозначены как <↑/0> или <↓/1>, т. е. ячейка памяти не может совершить переход из состояния логического «0» в логическую «1» или наоборот;

3) неисправности взаимного влияния одной ячейки памяти на другую (coupling faults, CF), которые в свою очередь делятся на

– статические (**CF_{st}**) – логическое состояние некоторой ячейки памяти по адресу i приводит к переходу состояния ячейки памяти по адресу j в определенное логическое значение, т. е. <0,0/1>, <1,0/1>, <0,1/0>, <1,1/0>;

– идемпотентные (**CF_{id}**) – изменение логического состояния некоторой ячейки памяти по адресу i приводит к переходу состояния ячейки памяти по адресу j в определенное логическое значение, т. е. <↑, 1/0> , <↑, 0/1>, <↓, 1/0>, <↓, 0/1>;

– инверсные (**CF_{in}**) – изменение логического состояния некоторой ячейки памяти по адресу i приводит к изменению текущего состояния ячейки памяти по адресу j на инверсное логическое значение, т. е. <↑/↕>, <↓/↕>.

В данной работе, модель неисправностей расширена на неисправности взаимного влияния ячеек памяти, вызванных проведением операций считывания:

– статические (**CF_{st}R**) – считывание состояния некоторой ячейки памяти по адресу i приводит к переходу состояния ячейки памяти по адресу j в определенное логическое значение, т. е. <R, 0/1>, <R, 1/0>;

– инверсные (**CF_{in}R**) – считывание логического состояния некоторой ячейки памяти по адресу i приводит к изменению текущего состояния ячейки памяти по адресу j на инверсное логическое значение, т. е. <R, ↕>.

Следует отметить, что почти все маршевые тесты обнаруживают практически 100 % одиночных неисправностей [6]. Таким образом, мы предлагаем производить оценку диагностических свойств тестов в рамках модели кратных неисправностей.

Всего допускаются следующие комбинации кратных неисправностей , приведенных ниже в табл. 1.

Таблица 1

Комбинации моделей кратных неисправности

№	Неисправности
1	$CF_{id} - CF_{id}$
2	$CF_{id} - CF_{in}$
3	$CF_{id} - CF_{stR}$
4	$CF_{id} - CF_{inR}$
5	$CF_{in} - CF_{in}$
6	$CF_{in} - CF_{stR}$
7	$CF_{in} - CF_{inR}$
8	$CF_{stR} - CF_{stR}$
9	$CF_{stR} - CF_{inR}$
10	$CF_{inR} - CF_{inR}$

Отличительной особенностью предложенной модели неисправностей является возможность задания кратных неисправностей между двумя парами ячеек памяти. Например, ячейка по адресу i влияет на ячейку по адресу k и в тоже время ячейка по адресу j также влияет на ячейку по адресу k.

5. Описание алгоритма March_DSS

Описание алгоритма будем производить в виде последовательности операторов (ШАГОВ), указывающих на порядок адресных переходов и на порядок выполнения операций записи-считывания. Например $\prod_{i:=1}^{N}(Vi)$ описывает последовательность операций по записи логического «0» в ячейки памяти начиная с первой и заканчивая последней N в порядке возрастания адресов, а оператор $\prod_{i:=N}^{1}(RiWi)$ описывает последовательность операций по считыванию ячеек памяти и записи в них логической «1» начиная с адреса последней ячейки N и заканчивая первой в порядке убывания адресов.

Алгоритм March_ DSS состоит из 16 шагов, каждый из которых записывается следующим образом:

ШАГ 1. Запись «0» $\prod_{i:=1}^{N}(Vi)$

ШАГ 2. Считать «0», записать «0», считать «0», записать «1» $\prod_{i:=1}^{N}(RiViRiWi)$

ШАГ 3. Считать «1» , записать «1», записать «0» $\prod_{i:=1}^{N}(RiWiVi)$

ШАГ 4. В обратном порядке считать «0» ,записать «0», записать «1» $\prod_{i:=N}^{1}(RiViWi)$

ШАГ 5. В обратном порядке считать «1», записать «1», считать «1», записать «0», записать «0», считать «0» $\prod_{i:=N}^{1}(RiWiRiViViRi)$

ШАГ 6. В обратном порядке считать «0», записать «0», считать «0» $\prod_{i:=N}^{1}(RiViRi)$

ШАГ 7. В обратном порядке считать «0», записать «1», считать «1» $\prod_{i:=N}^{1}(RiWiRi)$

ШАГ 8. В обратном порядке считать «1», записать «1» $\prod_{i:=N}^{1}(RiWi)$

ШАГ 9. В обратном порядке считать «1», записать «0», считать «0» $\prod_{i:=N}^{1}(RiViRi)$

ШАГ 10. В обратном порядке считать «0», записать «1», считать «1» $\prod_{i:=N}^{1}(RiWiRi)$

ШАГ 11. Считать «1» , записать «0» $\prod_{i:=1}^{N}(RiVi)$

ШАГ 12. В обратном порядке считать «0», записать «0» $\prod_{i:=N}^{1}(RiVi)$

ШАГ 13. Считать «0», записать «1» $\prod_{i:=1}^{N}(RiWi)$

ШАГ 14. В обратном порядке считать «1», записать «1», считать «1» $\prod_{i:=N}^{1}(RiWiRi)$

ШАГ 15. В обратном порядке считать «1», записать «0» $\prod_{i:=N}^{1}(RiVi)$

ШАГ 16. Считать «0» $\prod_{i:=1}^{N}(Ri)$

Результаты обнаружения неисправностей в процессе тестирования представлены в табл. 2.

Таблица 2

Процент обнаружения неисправностей тестом March_DSS

№	Наименование теста	Длина теста	Модель неисправности	% обнаружен. неисправностей
1	«March_ DSS»	46*N	$CF_{id} - CF_{id}$	98.33 %
			$CF_{id} - CF_{in}$	96.67 %
			$CF_{id} - CF_{stR}$	100.00 %
			$CF_{id} - CF_{inR}$	100.00 %
			$CF_{in} - CF_{in}$	93.33 %
			$CF_{in} - CF_{stR}$	100.00 %
			$CF_{in} - CF_{inR}$	100.00 %
			$CF_{stR} - CF_{stR}$	100.00 %
			$CF_{stR} - CF_{inR}$	100.00 %
			$CF_{inR} - CF_{inR}$	94.44 %

6. Описание алгоритмов March_C и March_L

Алгоритм теста March_C – состоит из 6 шагов, а его длина равна 10*N и он имеет следующий вид:

ШАГ 1. Запись нулей $\prod_{i:=1}^{N}(Vi)$

ШАГ 2. Считать «0», записать «1» $\prod_{i:=1}^{N}(RiWi)$

ШАГ 3. Считать «1» , записать «0» $\prod_{i:=1}^{N}(RiVi)$

ШАГ 4. В обратном порядке считать «0»,записать «1» $\prod_{i:=N}^{1}(RiWi)$

ШАГ 5. В обратном порядке считать «1»,записать «0» $\prod_{i:=N}^{1}(RiVi)$

ШАГ 6. В обратном порядке считать «0» $\prod_{i:=N}^{1}(Ri)$

Алгоритм теста March_L состоит из 9 шагов, а его длина равна 13*N:

ШАГ 1. Запись нулей $\prod_{i:=1}^{N}(Vi)$

ШАГ 2. Считать «0», записать «1» $\prod_{i:=1}^{N}(RiWi)$

ШАГ 3. В обратном порядке считать «0» $\prod_{i:=N}^{1}(Ri)$

ШАГ 4. Считать «1», записать «0» $\prod_{i:=1}^{N}(RiVi)$

ШАГ 5. В обратном порядке считать «0» $\prod_{i:=N}^{1}(Ri)$

ШАГ 6. В обратном порядке считать «0»,записать «1» $\prod_{i:=N}^{1}(RiWi)$

ШАГ 7. В обратном порядке считать «0» $\prod_{i:=1}^{N}(Ri)$

ШАГ 8. В обратном порядке считать «1»,записать «0» $\prod_{i:=N}^{1}(RiVi)$

ШАГ 9. В обратном порядке считать «0» $\prod_{i:=1}^{N}(Ri)$

Результаты обнаружения неисправностей в процессе тестирования последовательностями «March_C » и «March_L» представлены в табл. 3.

Таблица 3

Процент обнаружения неисправностей тестом March_C и March_L

№	Наименование теста	Длина теста	Модель неисправности	% обнаружен. неисправностей
1	«March_C - »	10*N	$CF_{id} – CF_{id}$	98.33 %
			$CF_{id} – CF_{in}$	96.67 %
			$CF_{id} – CF_{stR}$	99.77 %
			$CF_{id} – CF_{inR}$	100.00 %
			$CF_{in} – CF_{in}$	93.33 %
			$CF_{in} – CF_{stR}$	99.54 %
			$CF_{in} – CF_{inR}$	100.00 %
			$CF_{stR} – CF_{stR}$	100.00 %
			$CF_{stR} – CF_{inR}$	100.00 %
			$CF_{inR} – CF_{inR}$	94.44 %
2	«March_L»	13*N	$CF_{id} – CF_{id}$	98.33 %
			$CF_{id} – CF_{in}$	96.67 %
			$CF_{id} – CF_{stR}$	100.00 %
			$CF_{id} – CF_{inR}$	100.00 %
			$CF_{in} – CF_{in}$	93.33 %
			$CF_{in} – CF_{stR}$	100.00 %
			$CF_{in} – CF_{inR}$	100.00 %
			$CF_{stR} – CF_{stR}$	100.00 %
			$CF_{stR} – CF_{inR}$	100.00 %
			$CF_{inR} – CF_{inR}$	94.44 %

7. Выводы

Тестовая последовательность «March_Dss», которая проектировалась как наиболее мощный тест последнего времени для обнаружения неисправностей памяти, имеет такую же оценку диагностических свойств по обнаружению кратных неисправностей, в рамках представленной модели, как и тест «March_L». При этом длина теста «March_Dss» в 3 раза больше, чем у теста «March_L». В принципе это совпадает с качественной оценкой диагностических свойств, представленной в работе [4].

Для организации эффективного диагностирования оперативной памяти, на наш взляд, следует использовать тестовую последовательность «March_ C», которая обнаруживает около 82 % кратных неисправностей. Применение тестовой последовательности «March_L» требует на 3N больше временных затрат, однако позволяет обнаружить около 88 % всех кратных неисправностей. Применение теста «March_Dss» требует значительных временных затрат и с практической точки зрения, на наш взгляд, является нецелесообразным.

Следует отдельно рассмотреть вопрос об аппаратных затратах на реализацию генератора тестов и схемы анализа выходных реакций в системах System-on-Chip для исследованных выше тестов.

Литература

1. van de Goor, A. J. The Automatic Generation of March Tests [Text] / A. J. van de Goor // Rec. IEEE Int. Workshop on Memory Technology, Design and Testing, 1994. – P. 86–91. doi: 10.1109/mtdt.1994.397192

2. Lakshmi, H. R. Implementation of FSM-MBIST and Design of Hybrid MBIST for Memory cluster in Asynchronous SoC [Text] / H. R. Lakshmi, R. Varchaswini, Y. J. M Shirur // International Journal of Computer Applications Technology and Research. – 2014. – Vol. 3, Issue 4. – P. 216–220.

3. Patent N20090199057A1 USA, Int.Cl.G11 C29/00 [Text] / March DSS: Memory Diagnostic Test. – Publ. 06.08.2009.

4. Моамар, Д. Н. Метод оценки диагностических свойств тестов микросхем памяти [Текст] / Д. Н. Моамар, В.Г. Рябцев, Т.Ю. Уткина // Вісник Хмельницького національного університету. – 2011. – № 4. – С. 226–232.

5. Van de Goor, A. J. Testing Semiconductor Memories. Theory and Practice [Text] / A. J. van de Goor. – ComTex Publushing, Gouda,The Netherlands, 1998. – 512 p.

6. Либергб, И. Г. Исследование обнаруживающей способности алгоритмов проверки оперативной памяти [Текст] / И. Г. Либерг, Д. Н. Бовкун // Вестник НТУ ”ХПИ”. – 2004. – Вып. 17. – С. 127–137.

References

1. van de Goor, A. J. (1998). The Automatic Generation of March Tests. Rec. IEEE Int. Workshop on Memory Technology, Design and Testing, 86–91. doi: 10.1109/mtdt.1994.397192

2. Lakshmi, H. R., Varchaswini, R., M Shirur, Y. J. (2014). Implementation of FSM-MBIST and Design of Hybrid MBIST for Memory cluster in Asynchronous SoC. International Journal of Computer Applications Technology and Research, 3 (4), 216–220.

3 Patent N20090199057A1 USA, Int.Cl.G11 C29/00 (2009). March DSS: Memory Diagnostic Test. Publ. 06.08.2009.

4. Moamar, D. N., Rjabcev, V. G., Utkina, T. Ju. (2011). Metod ocenki diagnosticheskih svojstv testov mikroshem pamjati. Visnik Hmel'nic'kogo nacional'nogo universitetu, 4, 226–232.

5. van de Goor, A. J. (1998). Testing Semiconductor Memories. Theory and Practice. ComTex Publushing, Gouda, The Netherlands, 512.

6. Libergb, I. G., Bovkun, D. N. (2004). Issledovanie obnaruzhivajushhej sposobnosti algoritmov proverki operativnoj pamjati. Vestnik NTU ”KhPI”, 17, 127–137.

Рекомендовано до публікації д-р техн. наук , проф. Кривуля Г. Ф.
Дата надходження рукопису 25.11.2014

Потопахина Татьяна Юрьевна, кафедра автоматики и управления в технических системах, Национальный техничский университет «Харьковский политехнический институт», ул. Фрунзе, 21, г. Харьков, Украина, 61002
E-mail: potopahina21@mail.ru

Сивальнева Алина Игоревна, кафедра автоматики и управления в технических системах, Национальный техничский университет «Харьковский политехнический институт», ул. Фрунзе, 21, г. Харьков, Украина, 61002
E-mail: alinkasivalneva@rambler.ru

Попова Алина Юрьевна, кафедра автоматики и управления в технических системах, Национальный техничский университет «Харьковский политехнический институт», ул. Фрунзе, 21, г. Харьков, Украина, 61002
E-mail: alishka91@ukr.net

Либерг Игорь Геннадиевич, кандидат технических наук, доцент, кафедра автоматики и управления в технических системах, Национальный техничский университет «Харьковский политехнический институт», ул. Фрунзе, 21, г. Харьков, Украина, 61002
E-mail: i_liberg@ukr.net