

ТЕХНИЧЕСКИЕ НАУКИ

УДК 631.4:574

АНАЛИЗ ЭФФЕКТИВНОСТИ ОБНАРУЖЕНИЯ КОДОЧУВСТВИТЕЛЬНЫХ НЕИСПРАВНОСТЕЙ МАРШЕВЫМИ ТЕСТАМИ

ПЕТРОВСКАЯ ВИТА ВЛАДЛЕНОВНА

*магистрант кафедры программного обеспечения информационных технологий БГУИР,
Минск, Республика Беларусь*

ДЕМЕНКОВЕЦ ДЕНИС ВИКТОРОВИЧ

старший преподаватель кафедры программного обеспечения информационных технологий БГУИР, Минск, Республика Беларусь

Аннотация. *Статья посвящена обнаружению кодочувствительных неисправностей запоминающих элементов. В тексте рассмотрены тенденции развития запоминающих устройств и современные проблемы тестирования, приведена обобщенная схема системы памяти, классификация моделей неисправностей и используемый подход к тестированию. Статья содержит результаты обнаружения кодочувствительных неисправностей некоторыми маршевыми тестами и их анализ.*

Ключевые слова: *тестирование вычислительных систем, неисправности памяти, кодочувствительные неисправности, маршевые тесты*

Роль тестирования в процессе создания продукта сложно переоценить. Каждый этап производства может содержать ошибки. Задача тестирования – определить возникшую проблему, до того, как она нанесет серьезный ущерб. Правильность и эффективность тестирования – залог качественной продукции и успеха компании-производителя.

Проблема тестирования запоминающих устройств современных вычислительных систем, таких как встроенные системы (embedded systems), системы на кристалле (systems-on-a-chip) и сети на кристалле (nets-on-a-chip) является весьма актуальной задачей [1-3]. Особенно важную роль играет тестирование в процессе производства запоминающих устройств, поскольку наибольший удельный вес по числу микросхем и суммарной их стоимости в современных вычислительных системах имеет система памяти. Функциональная модель системы памяти, которая изображается в технических паспортах производителя, состоит из множества блоков. Обобщенная модель памяти представлена на рисунке 1 [4].

С помощью блока А задается адрес ячейки, над которой будет производиться операция чтения или записи. Ячейки хранятся в матрице, поэтому для определения ее расположения адрес содержит данные о строке и столбце, которые дешифрируются в блоках В и С. Массив запоминающих элементов обозначен на рисунке блоком D, логика чтения/записи происходит в блоках E, F, G.

Причинами некорректного состояния памяти могут быть неисправности электронного обрамления неисправности матрицы запоминающих элементов (ЗЭ) [5]. К неисправностям электронного обрамления относится дешифратор адреса и логика чтения/записи. Несмотря на простую организацию матрицы памяти существует огромное разнообразие возможных неисправностей, обнаружение и классификация некоторых из них является сложной задачей.

В процессе эволюции размер запоминающих устройств экспоненциально уменьшается, а объем памяти увеличивается, это приводит к увеличению плотности ячеек памяти. Близкое расположение запоминающих элементов приводит к возникновению как простых, так и сложных неисправностей: ячейки памяти могут влиять друг на друга, изменяя ее содержимое. Процесс обнаружения таких неисправностей является трудоемким.

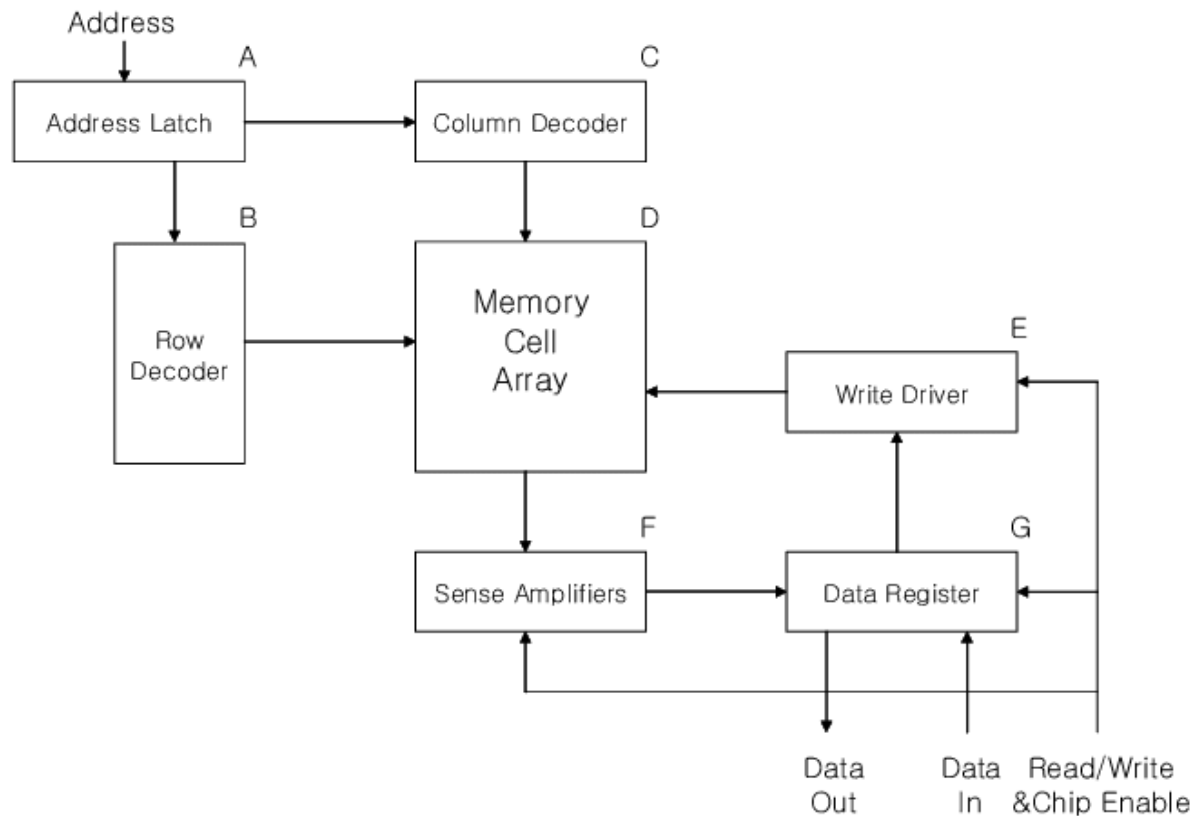


Рис.1. Функциональная модель памяти

Для того, чтобы сохранить уровень цен на запоминающие устройства, стоимость тестирования микросхем памяти, которая напрямую связана со временем тестирования, не может существенно увеличиваться. Объем современных запоминающих устройств не позволяет выполнять тесты сложности $O(n^2)$, где n – это количество бит памяти [6]. В настоящее время рассматриваются только методы тестирования, линейно зависящие от емкости памяти, эти тесты имеют общее название «маршевые тесты» (March Tests) [7].

Неисправные состояния запоминающих устройств описывают математическими моделями. Модель неисправности – это абстракция ошибки, вызванной конкретной физической неисправностью. Первые модели были предложены в 1980-ых годах, к ним относятся константные неисправности (stuck-at faults – SAF). Неисправный запоминающий элемент постоянно находится в состоянии логического нуля (s-a-0) или логической единицы (s-a-1), независимо от операций, выполняемых с неисправным элементом и другими элементами ЗУ.

Переходные неисправности (transition faults – TF) характеризуются невозможностью перехода состояния неисправного запоминающего элемента из 0 в 1 (TF \uparrow), или из 1 в 0 (TF \downarrow) при выполнении соответствующих операций записи. Если ячейка, имеет переходную неисправность, оказывается в состоянии, из которого она не может перейти в другое, то ее поведение повторяет поведение ячейки, содержащей константную неисправность.

Неисправности взаимного влияния (coupling fault – CF) характеризуются тем что, имеется влияющая ячейка (aggressor cell), изменение логического состояния которой воздействует на состояние зависимой ячейки (victim cell) [5]. Две последние модели описывают дефект памяти, в котором участвуют несколько ЗЭ.

Кодочувствительные неисправности (Pattern Sensitive Faults) характеризуются тем, что на состояние зависимой ячейки влияет содержимое всех других ячеек памяти. Неограниченная

PSF имеет сложность теста $(3n^2 + 2n) 2^n$ и непрактична для памяти большого размера [8]. На практике в тестировании используют модели, которые ограничивают число влияющих ячеек, граничные кодочувствительные неисправности (Neighborhood Pattern Sensitive Faults).

В модели неисправности NPSF_k выделяют одну базовую ячейку, на которую влияют соседние $k - 1$ ячеек. ЗЭ, участвующие в неисправности граничат друг с другом, соседние окружают базовую, образуя окрестность фон Неймана или окрестность Мура, представленные на рисунке 2.

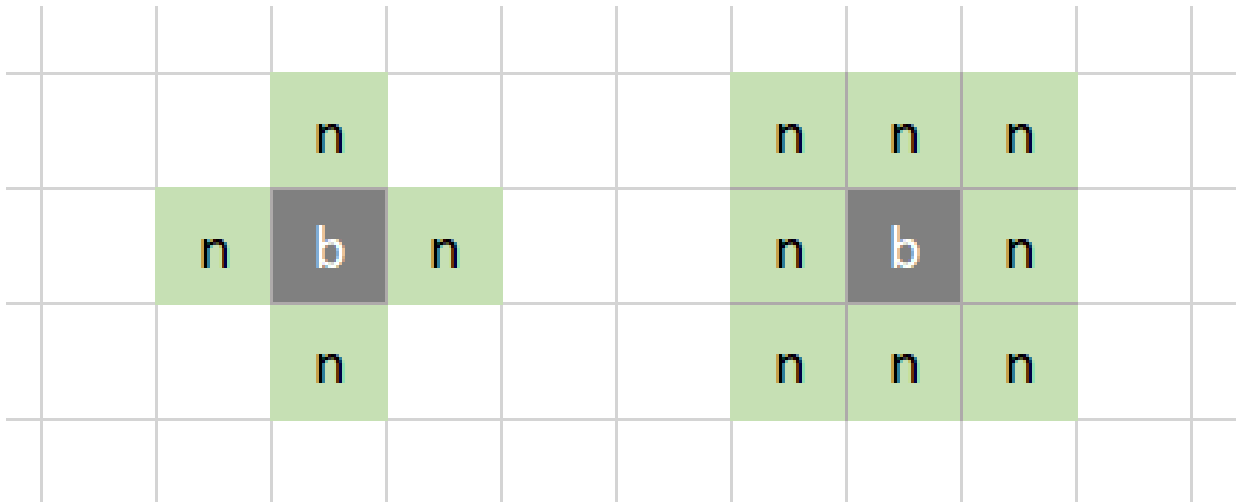


Рис.2. Окрестности кодочувствительных неисправностей (b – базовая ячейка, n – соседняя ячейка)

Выделяют три типа NPSF_k: пассивные (Passive) – базовая ячейка памяти не может изменить свое состояние, если в соседних ячейках установлен определенный код; активные (Active) – базовая ячейка изменяет свое состояния из-за изменения кода в соседних ячейках; статические (Static) – для определенной комбинации значений в соседних ячейках базовая принудительно устанавливается в ноль или единицу [5]. Всего возможных кодов для соседних ячеек, которые могут влиять на базовую ячейку 2^{k-1} . Количество возможных PNPSF_k составляет $k2^k$ [5].

Для активизации и обнаружения неисправных ячеек выполняется маршевый тест. Эффективность теста зависит от полноты покрытия неисправностей и временные затраты на тестирование [9].

Принято, что в определенный момент времени в массиве памяти находится только одна неисправность. Когда неисправностей несколько, они могут влиять друг на друга, в таком случае задача обнаружения дефектных ячеек становится нереально сложной. Для определения процента покрытия неисправностей генерируются все возможные модели, каждая из них проверяется отдельно маршевым тестом.

Маршевый тест состоит из конечной последовательности маршевых элементов. Маршевый элемент – это упорядоченный набор операций чтения и/или записи, который применяется к каждой ячейке памяти. Возможные операции: ‘r0’ – чтение из ячейки памяти ожидаемого нуля, ‘r1’ – чтение ожидаемой единицы, ‘w0’ – запись в ячейку памяти нуля, ‘w1’ – запись единицы. Маршевый элемент имеет свой порядок следования адресов: возрастающий (↑), убывающий (↓), возрастающий или убывающий (↑↓). Перед тем, как перейти к следующему адресу должны быть выполнены все операции маршевого элемента [5].

Для оценки эффективности обнаружения пассивных кодочувствительных неисправностей проведены следующие маршевые тесты: Scan, March Y, March PS, которые представлены в таблице 1.

Таблица 1 – Маршевые тесты: Scan, March Y, March PS

Маршевый тест	Маршевые элементы
Scan	{ $\uparrow\downarrow(w0)$; $\uparrow\downarrow(r0)$; $\uparrow\downarrow(w1)$; $\uparrow\downarrow(r1)$ }
March Y	{ $\uparrow\downarrow(w0)$; $\uparrow(r0,w1,r1)$; $\downarrow(r1,w0,r0)$; $\uparrow\downarrow(r0)$ }
March PS	{ $\uparrow\downarrow(w0)$; $\uparrow(r0, w1,r1,w0,r0,w1)$; $\uparrow(r1,w0,r0,w1,r1)$; $\uparrow(r1,w0,r0,w1,r1,w0)$; $\uparrow(r0,w1,r1,w0,r0)$; }

Проценты покрытия неисправностей тестами представлены в таблице 2. Процент покрытия равен отношению числа обнаруженных к общему количеству неисправностей, умноженный на 100%.

Таблица 2 – Проценты покрытия неисправностей маршевыми тестами

	PNPSF3	PNPSF4	PNPSF5	PNPSF9
Scan	12,5%	6,25%	3,125%	0,1953125%
March Y	25%	12,5%	6,25%	0,390625%
March PS	66,67%	37,50%	20%	1,388889%

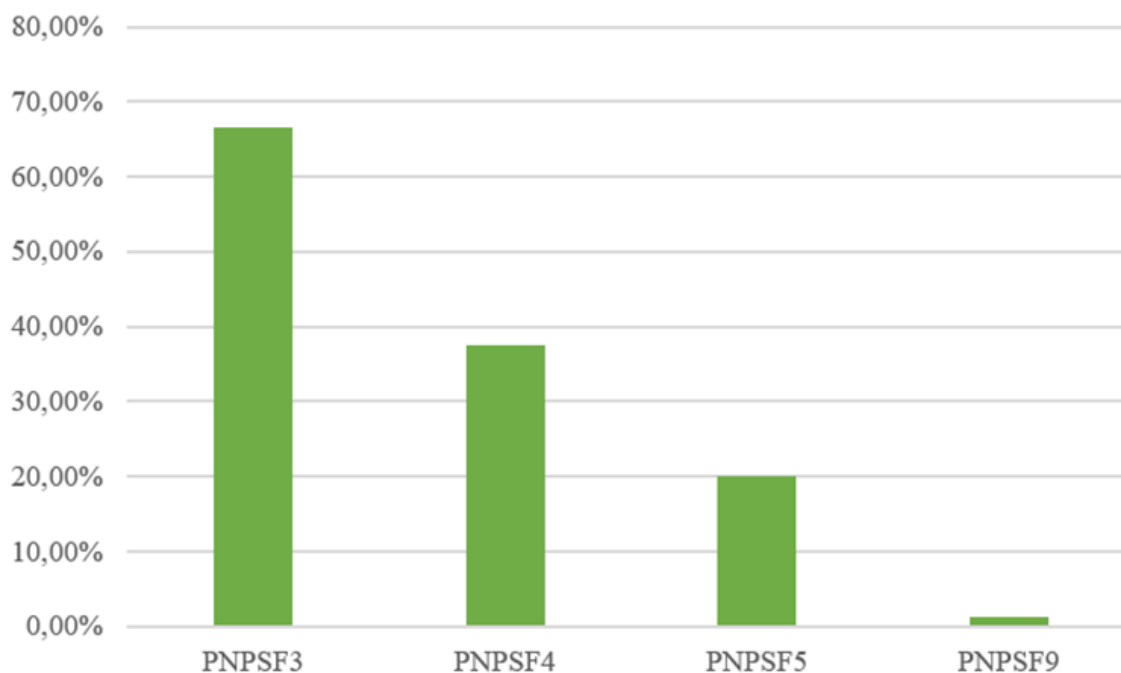
По данным из таблицы видно, что тест March PS обнаруживает наибольшее количество неисправностей. Обнаруженные тестами неисправности приведены в таблице 3.

Таблица 3 – Обнаруженные маршевыми тестами неисправности

Неисправность	Обнаруживающая последовательность операций	Scan	March Y	March PS
$\langle 1,1,1,\dots,1,\uparrow,0,0,0,\dots,0 \rangle$	$\downarrow(w0,r0,w1,r1)$	+		+
$\langle 1,1,1,\dots,1,\downarrow,0,0,0,\dots,0 \rangle$	$\uparrow(w1,r1,w0,r0)$			+
$\langle 0,0,0\dots,0,\uparrow,1,1,1,\dots,1 \rangle$	$\uparrow(w0,r0,w1,r1)$		+	+
$\langle 0,0,0\dots,0,\downarrow,1,1,1,\dots,1 \rangle$	$\downarrow(w1,r1,w0,r0)$		+	+
$\langle 0,0,0\dots,0,\uparrow,0,0,0,\dots,0 \rangle$	$\uparrow\downarrow(w0,r0,w1,r1,w0)$			+
$\langle 0,0,0\dots,0,\downarrow,0,0,0,\dots,0 \rangle$	$\uparrow\downarrow(w0,r0,w1,r1,w0,r0)$			+
$\langle 1,1,1\dots,1,\uparrow,1,1,1,\dots,1 \rangle$	$\uparrow\downarrow(w1,r1,w0,r0,w1,r1)$			+
$\langle 1,1,1\dots,1,\downarrow,1,1,1,\dots,1 \rangle$	$\uparrow\downarrow(w1,r1,w0,r0,w1)$			+

При увеличении окрестности растет количество возможных кодов в соседних ячейках. На рисунке 3 изображена диаграмма результатов теста March PNPSFk для различных k, которая показывает существенное уменьшение процента покрытия.

Рис. 3. Результаты теста March PS для различных k



ЛИТЕРАТУРА

1. Bushnell, M. L. Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits / M. L. Bushnell, V. D. Agrawal. – New York: Kluwer Academic Publishers, 2000. – 690 p.
2. Wang, L. T. VLSI Test Principles and Architectures: Design for Testability / L. T. Wang, C. W. Wu, X. Wen. – Amsterdam: Elsevier, 2006. – 808 p.
3. Ярмолик, В. Н. Контроль и диагностика вычислительных систем / В. Н. Ярмолик. – Минск: Бестпринт, 2019. – 387 с.
4. Keon-Jik, L. A simple diagnostic method for memory testing / L. Keon-Jik, L. Byeong-Jik, K. Seongwoon // International Conference on Electronics, Control and Signal Processing – 2003. – №16. – P. 1-6.
5. Ярмолик, С.В. Маршевые тесты для самотестирования ОЗУ: монография / С.В. Ярмолик, А.П. Занкович, А.А. Иванюк; под общ. ред. В.Н. Ярмолика. – Минск: Изд. центр БГУ, 2009. – 271 с.
6. Bushnell, M. Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits. / M. Bushnell, A. Vishwani. – NY: Kluwer Academic Publishers, 2001.
7. Ярмолик, С.В. Анализ количественных характеристик различия при тестировании ОЗУ / С.В. Ярмолик, А.Н. Курбацкий, В.Н. Ярмолик // Информатика. – 2008. - №3(19). – С. 90-98.
8. Mazumder, P. Parallel Testing for Pattern Sensitive Faults in Semiconductor Random Access Memory / P. Mazumder, J. Patel. – IEEE Transactions on Computers – 1989. – Vol.38. – №3. – С. 394-407.
9. Деменковец, Д. В. Программное средство моделирования и поиска неисправностей запоминающих устройств / Деменковец Д. В. // Радиотехника и электроника: сборник тезисов докладов 56-й научной конференции аспирантов, магистрантов и студентов, Минск, апрель-май 2020 года / Белорусский государственный университет информатики и радиоэлектроники. – Минск: БГУИР, 2020. – С. 58-60.