

# ПРОГРАММНОЕ СРЕДСТВО МОДЕЛИРОВАНИЯ И ПОИСКА НЕИСПРАВНОСТЕЙ ЗАПОМИНАЮЩИХ УСТРОЙСТВ

*Деменковец Д.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Ярмолик В.Н. – профессор, доктор технических наук*

В тексте доклада описывается программное средство моделирования и поиска неисправностей запоминающих устройств. Дается классификация неисправностей запоминающих устройств. Описываются функциональные возможности этого ПС при использовании маршевых тестов. Описываются получаемые программным средством оценки поиска неисправностей при маршевом тестировании.

Быстрое развитие полупроводниковых технологий привело к значительному увеличению объема и плотности оперативной памяти (ОЗУ, RAM) на одном чипе. Для тестирования памяти требуется больше времени. В связи с этим наблюдается повышенный интерес к проблемам диагностирования оперативной памяти (ОЗУ, RAM). Этот интерес связан с тем, что ОЗУ является главным компонентом современных вычислительных систем, таких как встроенных систем (system-on-a-chip, SoC) и сетей на кристалле (nets-on-a-chip). Для тестирования ОЗУ используются, как правило, тесты линейной длины семейства «Марш» («March») [1,2]. Актуальным является проверка эффективности маршевых тестов. Проверки полноты покрытия данных тестов. Оценка времени работы маршевых тестов. Проверка новых методов тестирования запоминающих устройств. Основной сложностью проверки является симуляция работы реальной памяти и поведение ячеек памяти при различных неисправностях. В докладе рассматриваются основные типы неисправностей запоминающих элементов, предложенных Ван де Гором оперативного запоминающего устройства (ОЗУ, RAM) [3].

Входными данными для программного средства моделирования являются маршевая последовательность или последовательности и последовательность адресов ячеек памяти (адресная последовательность). Выходными данными является: время, затраченное на тестирование, полнота

тестового покрытия, затраченное количество оперативной памяти компьютера, средняя загрузка процессора, оценка производительности, косвенная оценка энергоэффективности тестов.

Далее рассмотрим основные типы неисправностей запоминающих устройств, поддерживаемых проектируемым ПО. К неисправностям матрицы запоминающих устройств в первую очередь относят неисправности, в которых участвуют: одна ячейка; две ячейки; несколько ячеек, в общем случае более чем две, без ограничений на их количество. Общепринятыми являются следующие модели неисправностей ячеек памяти.

К неисправностям, затрагивающим одну ячейку ЗУ, относят [3]:

- 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. (уточнить, это с обратной связью, т.е. мы записываем, что-то в ответ получаем? Спросить многократные и однократные константные неисправности).

- Переходные неисправности (*transition faults - TF*). Подобные неисправности 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. Данную неисправность не нужно путать с типом неисправности взаимного влияния, которая рассматривается ниже.

Среди неисправностей, в которых участвуют две ячейки ЗУ, выделяют следующие неисправности [3]:

- Неисправности взаимного влияния (*coupling fault - CF*). При описании данной неисправности выделяют влияющую ячейку (*aggressor cell*), определяемую ее адресом *i*, изменение логического состояния которой воздействует на состояние зависимой ячейки (*victim cell*) с адресом *j*.

- Инверсные неисправности взаимного влияния (*inverse coupling faults - CF<sub>in</sub>*). При наличии данной неисправности изменение значения *b<sub>i</sub>* влияющей ячейки вызывает инвертирование значение *b<sub>j</sub>* зависимой ячейки. Возможны четыре вида данного типа неисправностей *CF<sub>in</sub>*.

- Неисправности взаимного влияния прямого действия (*idempotent coupling faults - CF<sub>id</sub>*). При изменении значения *b<sub>i</sub>* влияющего запоминающего элемента происходит принудительная установка определенного логического значения 0 или 1 в зависимом запоминающем элементе. Различают восемь неисправностей прямого действия.

- Статические неисправности взаимного влияния (*state coupling faults - CF<sub>st</sub>*). Переход зависимой ячейки в какое-либо состояние *b<sub>j</sub>* возможен при определенном значении *b<sub>j</sub>* влияющей ячейки. Возможно восемь неисправностей *CF<sub>st</sub>*.

Модели кодочувствительных неисправностей в данной статье не рассматриваются и данным программным средством не поддерживаются.

Среди множества тестов запоминающих устройств выделяют так называемые традиционные тесты, которые позволяют обеспечивать некоторую степень покрытия возможных неисправностей при соответствующих временных затратах. Для данного программного средства реализованы следующие широко известные тесты типа шахматная доска (*checkerboard*), тест бегущая 1/0 (*walking 1/0*), тест бегущая диагональ (*sliding diagonal*), тест пинг-понг (*ping-pong*), галопирующая 1/0 (*galloping 1/0*).

Во всех тестах запоминающих устройств операция чтения содержимого ячейки сопряжена с проверкой прочитанного значения 0/1 с заранее записанным значением 0/1 в данную ячейку.

Разрабатываемое программное средство поддерживает выполнение маршевых тестов (*march tests*). Маршевые тесты имеют линейную зависимость сложности  $O(N)$  от емкости *N* памяти. Данные тесты обладают рядом достоинств, среди которых приемлемая покрывающая способность неисправностей запоминающих устройств, описанных выше, а также простота реализации тестовой процедуры.

Данное программное средство моделирует только неисправности массива запоминающих элементов памяти, которые описаны выше, и на данном этапе не поддерживаются неисправности схем обрэмления, которые также относятся к неисправностям запоминающих устройств.

В данном программном средстве для удобства модель памяти представляется и рассматривается как двумерное пространство в виде матрицы с одинаковым количеством строк и столбцов. Входными тестовыми данными являются адреса запоминающих ячеек, состоящие из адресов по горизонтальной и вертикальной оси матрицы ячеек памяти

Моделирование константного типа (*stuck-at faults - SAF*) неисправности программным способом реализовано путем установки значения соответствующего свойства экземпляра объекта «ячейки памяти» с проверкой типа неисправности (соответствующего поля экземпляра объекта).

Симуляция поиска переходных неисправностей (*transition faults - TF*) реализованы схожим способом, с выполнением дополнительной проверка значения состояния ячейки (соответствующего свойства экземпляра объекта «ячейка памяти»). В зависимости от типа неисправности запись соответствующего состояния выполняется или остается неизменным.

Неисправности взаимного влияния реализованы с использованием паттерна (шаблона проектирования) «наблюдатель» с помощью ссылки на метод (делегата) [4].

Программная реализация позволяет в последующем легко расширять функциональные возможности и последующем позволит легко добавлять поддержку новых типов неисправностей, в том числе сложных и трудно обнаруживаемых. Данное программное средство позволяет симулировать значительные объемы оперативной памяти. Также ПС позволяет ручным способом организовать и проанализировать основные категории тестовые воздействия [3,5]: точечные тестовые наборы (point patterns), узкополосные тестовые наборы (strip patterns), блочные тестовые наборы (block patterns). В последующем на вход программы представить адресную последовательность (address set).

Программное средство специально разрабатывается на кафедре ПОИТ БГУИР для моделирования и верификации обнаруживающей способности проверяющих тестов на предмет диагностирования расширенного множества одиночных и кратных неисправностей.

Одним из основных критериев оценки эффективности тестов запоминающих устройств является время выполнения теста, которое зависит от количества операций обращения к тестируемому устройству и от времени отклика устройства. Обобщающая характеристика эффективности тестов называется сложность теста  $Q(\text{тест})$ . Эта характеристика оценивается как количество всех элементарных операций (чтения/запись), необходимых для реализации теста [6].

Под оценкой работы моделирования и поиска неисправностей запоминающих устройств понимается критерий качества маршевого теста, который определяется двумя параметрами:

- процент обнаруженных неисправностей в рамках принятой модели (процент покрытия);
- 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).

**Список использованных источников:**

1. Ярмолик, В. Н. Контроль и диагностика вычислительных систем: [монография] / В. Н. Ярмолик. – Минск : Бестпринт, 2019. – 387 с. : ил. 75.
2. Ярмолик, В. Н. Обзор методов неразрушающего тестирования ОЗУ / В. Н. Ярмолик, А. П. Занкович // Доклады БГУИР. - 2005. - № 4 (12). - С. 62 - 72.
3. Goor, A.J. Testing Semiconductor Memories, Theory and Practice / A.J. Googr. – UK, Chichester: John Wiley & Sons, 1991/ 536 p.
4. Тепляков С. Паттерны проектирования на платформе .NET : книга // С. Тепляков – СПб.: Изд-во: Питер, 2015. – 320 с.: ил.
5. Chan, F.T. Proportional sampling strategy guidelines for software testing practitioner / F.T. Chan, T.Y. Chen, I.K. Mak, T.T. Yu // Information Software Technology – 1996. – Vol.38. - №12. – P. 775-782.
6. Ярмолик, В. Н. Методы и средства встроенного самотестирования с пониженным энергопотреблением / В. Н. Ярмолик, И. А. Мурашко // Доклады БГУИР. - 2004. - № 1 (5). - С. 102 - 114.