

ОЦЕНКА КАЧЕСТВА ПСЕВДО-КОЛЬЦЕВОГО ТЕСТИРОВАНИЯ УСТРОЙСТВ ОПЕРАТИВНОЙ ПАМЯТИ

Сергей ГРИЦКОВ, Герман СОРОКИН

Технический Университет Молдовы

Abstract: The paper deals with the determination of the quality of pseudo-ring testing estimated by a simulation based on FPGA of pseudo-ring tests as well as single and multiple stack-at faults of memory devices. Is considered the transition from simulation based on the software tools to modeling based on the hardware ones.

Ключевые слова: псевдо-кольцевое тестирование, качество тестирования, константные неисправности, система моделирования.

1. ВВЕДЕНИЕ

Одним из способов оценки методов тестирования является определение их алгоритмической сложности, которая определяет количество выполняемых итераций для обнаружения в определенной ячейке памяти того или иного типа неисправностей. Применение классических методов тестирования ограничено, так как их алгоритмическая сложность чаще всего носит нелинейный характер. Алгоритмическая сложность маршевых или March-тестов значительно возрастает при переходе от бит-ориентированной к word-ориентированной структуре устройств оперативной памяти [1]. Недостатки существующих методов тестирования приводят к необходимости разработки иных подходов к тестированию устройств оперативной памяти. Одним из таких методов является псевдо-кольцевое или π -тестирование [2].

Важным вопросом является установление качества тестирования, которое определяется обнаруживающей способностью тестов, то есть определением количества ячеек памяти, в которых тест сможет обнаружить определенную неисправность по отношению к общему числу ячеек памяти. Одним из решений данного вопроса является синтез моделей, позволяющих моделировать как проведение тестирования, так и моделирование самих неисправностей. В данной работе и будет рассмотрено моделирование псевдо-кольцевых тестов, а также – неисправностей в устройствах оперативной памяти.

2. Типовые неисправности устройств памяти и методы их обнаружения

При рассмотрении неисправностей на функциональном уровне представления структуры оперативной памяти можно выделить ~80 типов неисправностей. Важным параметром является частота появления тех или иных типов неисправностей. Необходимо отметить, что наиболее часто появляющимися типами неисправностей являются константные одиночные ($s@0$ – константа нуль и $s@1$ – константа единица) и кратные неисправности, частота появления которых составляет порядка 50% из всех возможных типов неисправностей [1,3]. На основе параметра частоты появлений можно ограничиться константными неисправностями. Для обнаружения данного типа неисправностей применим псевдо-кольцевое тестирование, которое подробно описано в [1,4].

Подчеркнем основные принципы π -тестирования. Псевдо-кольцевое тестирование основывается на эмулировании (создании виртуального) сдвигового регистра с обратными связями (LFSR). Структура виртуального LFSR определяется структурой неприводимого полинома $g(z) = \sum_{i=0}^k g_i z^i$, где $g_i \in \mathbf{GF}(2^m) / p(x)$, $p(x) = \sum_{i=0}^m p_i x^i$, $m=2,3,\dots$, $p_i \in \{0,1\}$. Идея π -тестирования основывается на перемещении виртуального регистра через все ячейки матрицы памяти. Процедура π -тестирования предусматривает выполнение так называемых π -итераций. Итерация π -теста заключается в следующем: инициализация виртуального LFSR, перемещение LFSR в пространстве памяти, считывание финального состояния **Fin** виртуального регистра LFSR и анализ полученного результата (сравнение ожидаемого результата со считанным). Для вычисления ожидаемого состояния применяем известное рекуррентное уравнение Колмогорова-Чепмена:

$$S_n = S_0 * A^n, \quad (1)$$

где S_0 – одномерный вектор состояния LFSR размерности k в начальный момент времени; A – сопровождающая двумерная матрица [4].

В [1] доказано, что при псевдо-кольцевом тестировании константных неисправностей можно применять приводимые полиномы, определяющие структуру LFSR (например, полином вида $g(z)=1+z+z^2$). Рассмотрим пример тестирования word-ориентированной памяти (к примеру, 4-х битной) с неисправностью $s@1$ в ячейке $i+2$ и структурой LFSR, определяемой полиномом вида $g(z)=1+z+z^2$ (рис. 1).

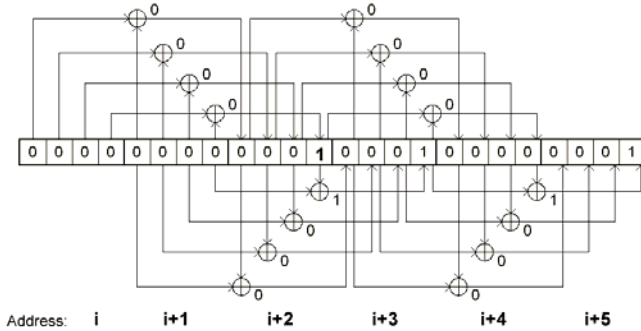


Рис. 1. Пример моделирования и обнаружения константной одиночной неисправности на основе регистров с внешними сумматорами.

Финальное значение LFSR без неисправностей в данном примере должно быть равно $\langle 0000\ 0000 \rangle$, но по рис. 2 видно, что это значение равно $\langle 0000\ 0001 \rangle$. Так как финальное значение LFSR не совпало с ожидаемым, можно утверждать, что тест с

начальным состоянием $\langle 0000\ 0000 \rangle$, данным направлением движения LFSR и данной структурой LFSR обнаружит неисправность $s@1$ в ячейке $i+2$. Также можно утверждать, что **главными параметрами** π -теста является *структура LFSR, направление его движения и его начальное состояние*, что приводится в [4].

Важным вопросом является, сможет ли данный тест обнаружить все константные неисправности, что и приводит к необходимости моделирования неисправностей.

3. МОДЕЛИРОВАНИЕ ПСЕВДО-КОЛЬЦЕВЫХ ТЕСТОВ И КОНСТАНТНЫХ НЕИСПРАВНОСТЕЙ

При разработке системы моделирования псевдо-кольцевых тестов можно применить принципы реализации моделей, описанные в [5]. Модель должна обладать настраиваемыми параметрами, которые позволили бы изменять параметры псевдо-кольцевых тестов, а также – типы имитируемых неисправностей. Принцип функционирования данной модели заключается в проведении псевдо-кольцевого тестирования памяти без и с неисправностями. Результаты тестирования (значения LFSR в памяти без неисправностей – ожидаемое значение, и LFSR в памяти с неисправностями – реальное значение) сравниваются и фиксируются счетчиком, который и выдает значение, определяющее обнаруживающую способность теста. Блок-диаграмму требуемой модели можно представить в следующем виде (рис. 2).

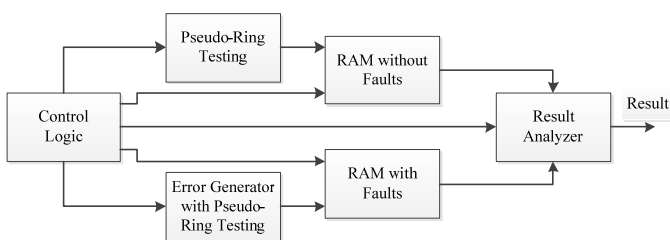


Рис. 2. Блок-диаграмма системы моделирования неисправностей оперативной памяти.

Блоки, указанные на рис. 2 обозначают следующее:

Pseudo-Ring Testing – блок, представляющий собой средства, необходимые для осуществления псевдо-

кольцевого тестирования, осуществляющие загрузку изначального значения теста, адресацию памяти, запись значений в память и чтение памяти, определяющие направление проведения итераций, а также – количество тестовых комбинаций.

RAM without Faults – блок статической оперативной памяти без неисправностей.

Error Generator with Pseudo-Ring Testing – блок, представляющий собой средства, необходимые как для осуществления псевдо-кольцевого тестирования, так и для генерирования неисправностей; блок позволяет моделировать неисправности последовательно в каждой ячейке памяти, выбирать тип моделируемой неисправности.

RAM with Faults – блок статической оперативной памяти, в которой будут имитироваться определенные неисправности.

Result Analyzer – блок, сравнивающий финальные значения, полученные после проведения каждой итерации теста над памятью с и без соответствующего типа неисправностей.

Control Logic – блок, осуществляющий синхронизацию работы всех вышеперечисленных блоков модели.

Принцип моделирования неисправностей заключается в том, что вся память разбивается на линейный массив, состоящий из бит. При проведении *i*-й итерации теста в *i*-й бит памяти записывается не вычисляемое значение при π -тестировании, а значение, определяемое текущей неисправностью. Таким образом, число тестовых итераций определяется количеством бит памяти.

Данная система моделирования реализована в системе Quartus II 9.1.

Полученная в результате языкового описания на VHDL система моделирования в виде RTL-диаграммы представлена на рис. 3.

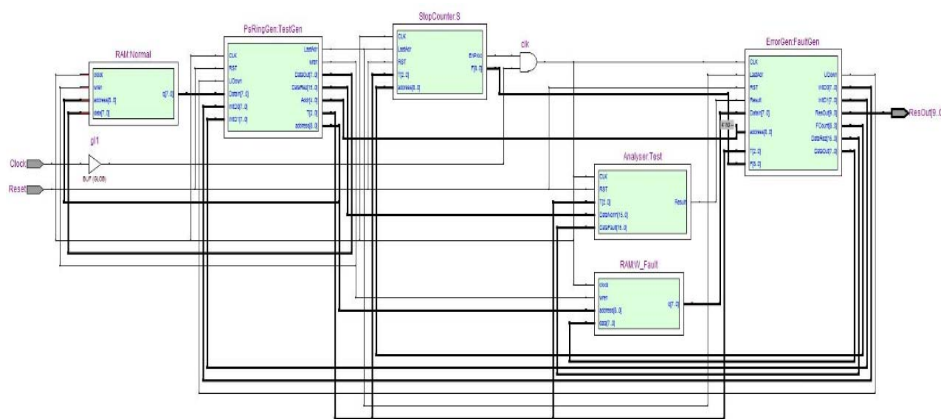


Рис. 3. RTL-диаграмма системы моделирования константных одиночных и кратных неисправностей.

На рис. 3 блоки имеют обозначают следующее:

RAM Normal – блок оперативной памяти без неисправностей.

PsRingGen_TestGen - блок осуществляющий псевдо-кольцевое тестирование.

Analyzer_Test – блок, который выдает импульс на выходе при обнаружении той или иной тестовой комбинации соответствующей неисправности.

RAM_W_Fault – блок оперативной памяти, в которой моделируются соответствующие неисправности.

ErrorGen_FaultGen – блок осуществляющий псевдо-кольцевое тестирование и моделирующий соответствующие неисправности в блоке памяти RAM_W_Fault.

ResOut – шина, на которой фиксируется количество обнаруженных неисправностей в процессе проведения одного тестового эксперимента после того, как блок **StopCounter_S** остановит работу модели.

Результатом синтеза данной системы является проведение временного моделирования в системе QSIM, что подтверждает корректность функционирования разработанной системы.

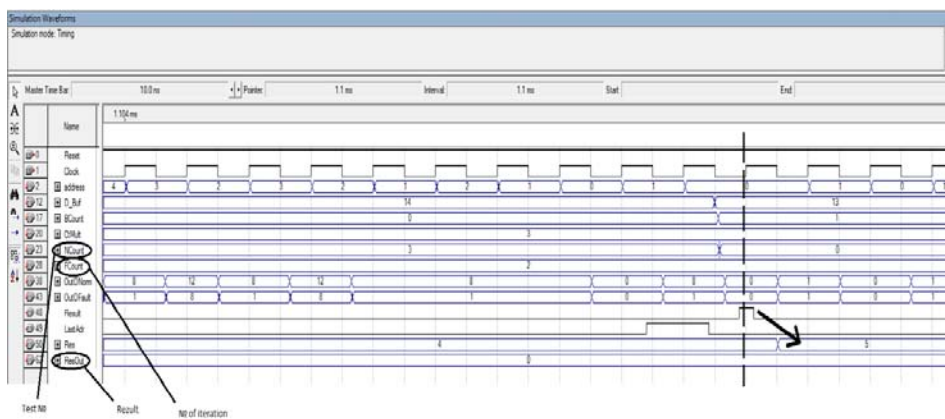


Рис. 4. Результат моделирования спроектированной системы моделирования константных одиночных и кратных неисправностей.

На рис. 4 *address* обозначает шину адреса оперативной памяти. *OutDNorm* – шина данных оперативной

памяти без неисправностей, *OutDFault* – шина данных оперативной памяти с моделируемыми неисправностями. *BCount* и *CTMult* – вспомогательные счетчики. По окончании одной тестовой итерации на линии *LastAdr* формируется импульс. Если на линии *Result* появится при этом импульс, значит, тестовая комбинация обнаружила соответствующую неисправность. В таком случае происходит увеличение счетчика *Res*, который подсчитывает число обнаруженных неисправностей в процессе проведения тестовых экспериментов, на единицу. После выполнения всех тестовых экспериментов значение из счетчика *Res* копируется в выходной буфер *ResOut*.

В результате тестовых экспериментов, проведенных в [1] можно утверждать, что тесты со структурой LFSR, определяемой полиномом $g(z)=1+z+z^2$, начальными состояниями <0000 0000> и <1111 1111>, и произвольным направлением движения LFSR обнаружат все константные неисправности, как одиночные, так и кратные.

Одним из важнейших параметров является время моделирования. При реализации VHDL-проекта на FPGA Cyclone II (kit DE-2) время моделирования на аппаратном уровне сократилось в 32000 раз по сравнению с моделированием в приложении QSIM системы Quartus II 9.1 [1].

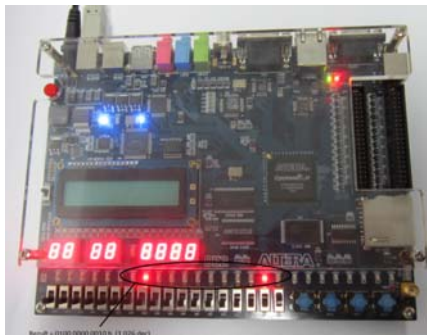


Рис. 5. Проведение моделирования на аппаратном уровне.

На рис. 5 выделены светодиоды, так как они отображают в двоичной системе количество обнаруженных неисправностей после проведения соответствующего π -теста.

Табл. 1. Зависимость времени проведения моделирования от характеристик тестирования.

Параметры моделируемой системы			Время моделирования	
Разрядность ячеек памяти	Кол-во итераций	Объем памяти, число ячеек	На ПК	На DE2
4 bit	8	512	~ 3 ч	0,34 с
8 bit	16	65 k	~ 197 910 ч (~ 23 года)	21 990 с (~ 6,1 ч)

Из табл. 1 видно, что моделирование устройств памяти со структурой ячеек 8 и более бит должно выполняться либо вычислительной системой, состоящей из сети компьютеров, либо – на аппаратном уровне, как было предложено в [1].

4. ЗАКЛЮЧЕНИЕ

Моделирование тестирования неисправностей позволяет определить качество (обнаруживающую способность) тестов, которая в данном случае составила 100% для псевдокольцевого тестирования константных неисправностей. Результаты моделирования π -тестирования показали, что при увеличении параметров тестирования памяти (а именно увеличении разрядности ячеек памяти) время моделирования растет экспоненциально (табл. 1). Для ускорения моделирования можно применить аппаратное ускорение (реализация системы на kit DE-2, для данного случая), что позволяет сократить в 32 000 раз временные затраты. Таким образом, время моделирования сократилось с 23 лет до 6 часов.

ЛИТЕРАТУРА

1. С. Грицков, *Само-тестирование встроенной оперативной памяти микроконтроллеров*, UTM, FRT, CPAE, мастерандская работа, 2012.
2. Gh. Bodean, D. Bodean, A. Labunetz, *New Schemes for Self-Testing RAM*, Technical University of Moldova, 2005.
3. Hamdioui, S., van de Goor, A.J., Rodgers, M., *March SS: a test for all static simple RAM faults*, Memory Technology, Design and Testing, 2002, pp.95-100.
4. S. Grițcov, A. Ghincul, Gh. Bodean, *Autotestarea pseudoinelară a microcontrolerelor nanosatelitului SATUM*, ICTEI, Chișinău, mai 2012, p. 260-267.
5. Alessandro Fin, Franco Fummi, *A VHDL Error Simulator for Functional Test Generation*, Proceedings of the conference on Design, automation and test in Europe, 2000, p. 390-395.