

ПРОГРАММНОЕ СРЕДСТВО СРАВНИТЕЛЬНОЙ ОЦЕНКИ МАРШЕВЫХ ТЕСТОВ ОЗУ

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

Селедец В.Н., Леванцевич В.А.

Ярмолик В.Н. – д.т.н., профессор

Рассматривается программная модель неисправностей оперативных запоминающих устройств. Описана архитектура программного средства, которое позволяет провести сравнительный анализ различных способов формирования тестов для обнаружения различных типов неисправностей.

Оперативные запоминающие устройства (ОЗУ) выполняют одну из важнейших функций в современных цифровых системах обработки и хранения информации. Результаты исследований показывают, что отказы ОЗУ составляют до 70% от общего числа отказов вычислительных систем. Причиной неисправных состояний ОЗУ является наличие физических или механических дефектов либо множества подобных дефектов, количество и многообразие которых практически неограниченно.

Для аналитического описания неисправных состояний ОЗУ используются математические модели неисправностей, так или иначе отражающие реальные физические дефекты ОЗУ.

Функциональные неисправности ОЗУ подразделяются на два подмножества (рисунок 1): неисправности матрицы запоминающих элементов и неисправности электронного обрамления. Второе подмножество включает неисправности дешифраторов адреса и неисправности логики чтения/записи. Доминирующее значение имеют неисправности матрицы запоминающих элементов ОЗУ.

К неисправностям матрицы запоминающих элементов ОЗУ относят неисправности, в которых участвуют: одна ячейка ОЗУ; две ячейки ОЗУ; несколько ячеек ОЗУ, в общем случае более чем две, без ограничений на их количество.

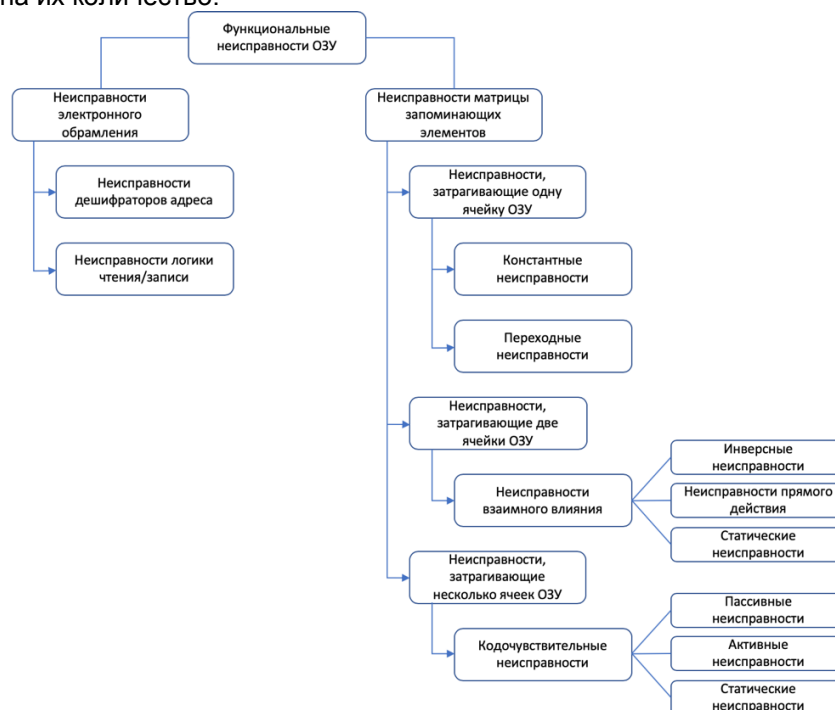


Рисунок 1 - Схема функциональных неисправностей ОЗУ

Тестирование является одним из наиболее эффективных методов повышения надёжности работы вычислительных устройств и систем. Основное назначение тестирования заключается в обнаружении максимально возможного числа неисправных состояний тестируемого объекта за минимально возможное время. В случае ОЗУ время тестирования зависит от его ёмкости N , которая подразделяет все тесты на три основных класса, а именно на тесты типа N^2 циклов, тесты типа $N^{3/2}$ и тесты типа N циклов, где циклом является время обращения (доступа) к ОЗУ.

Широкое применение находят тесты, имеющие линейную зависимость сложности от ёмкости ОЗУ, получившие название маршевые тесты.

Маршевый тест состоит из конечного числа маршевых элементов [1]. В свою очередь, каждый маршевый элемент содержит символ, определяющий порядок формирования адресной последовательности ОЗУ: ∇ - определяет последовательный перебор адресов ОЗУ по возрастанию, \odot - определяет последовательный перебор адресов по убыванию, $\odot\nabla$ - означает перебор по убыванию либо по возрастанию. Кроме этого, маршевый элемент содержит последовательность операций чтения и записи, заключённых в круглые скобки и разделяемых точкой с запятой. Каждая операция представляет собой элемент из следующего набора: 'r0' – операция чтения содержимого ЗЭ с ожидаемым значением 0, 'r1' – операция чтения ЗЭ с ожидаемым значением 1, 'w0' – операция записи 0 в ЗЭ, 'w1' – операция записи 1 в ЗЭ. Одна или несколько операций в маршевом элементе используется последовательно для адресуемой ячейки ОЗУ. Переход к следующей ячейке будет осуществлён только после выполнения всех операций в текущем маршевом элементе.

Количество маршевых элементов, их взаимное расположение, а также вид каждого из этих элементов определяет покрывающую способность маршевого теста, т. е. эффективность обнаружения всех видов возможных неисправностей ОЗУ.

Для того, чтобы определить эффективность маршевого теста, необходимо определить его основные характеристики, к которым относятся покрывающая способность, вычислительная сложность формирования, время обнаружения неисправности.

Для решения поставленной задачи разработано программное средство, позволяющее смоделировать запоминающее устройство, указать типы неисправностей запоминающих элементов и выполнить заданные маршевые тесты с целью определения их сравнительных характеристик.

Программное средство имеет объектно-ориентированную архитектуру, где каждый объект является реализацией определённой абстракции предметной области. Так, одной из главных абстракций является запоминающее устройство [2].

Запоминающее устройство в программном средстве представлено в виде класса, содержащего ёмкость N запоминающего устройства, массив N запоминающих элементов, и два метода: метод чтения значения запоминающего элемента и метод записи значения в запоминающий элемент. Оба метода принимают адрес запоминающего элемента в качестве параметра.

Запоминающий элемент характеризуется адресом, значением и типом неисправности.

В программном средстве используется ещё одна важная абстракция – алгоритм, представляющий собой способ формирования маршевого теста. Алгоритм, в свою очередь, состоит из трёх других абстракций: контроллера теста, который управляет ходом теста, генератора последовательности значений, которые записываются в запоминающие элементы и генератора адресов запоминающих элементов, по которым записываются сгенерированные значения. При использовании тестов с изменяемыми начальными состояниями, генератор последовательности значений позволяет предварительно записать последовательность значений в начале процедуры тестирования.

Ниже приведена упрощённая структурная схема программного средства (рисунок 2).

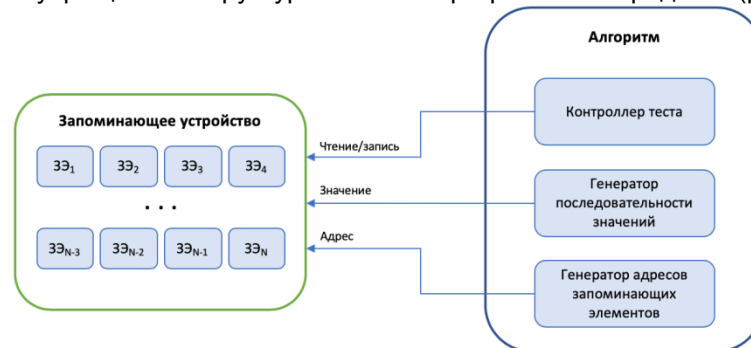


Рисунок 2 - Упрощённая структурная схема программного средства

Выводы. В результате решения поставленной задачи было разработано программное средство, позволяющее смоделировать запоминающее устройство, указать типы неисправностей запоминающих элементов, сформировать по определенным алгоритмам тесты и вывести результаты их работы, для сравнительного анализа.

Список использованных источников:

1. С.В. Ярмолик, А.П. Занкович, А.А. Иванюк Маршевые тесты для самотестирования ОЗУ.: Изд. Центр БГУ, 2009, с. 7-28.
2. А.В. Степанов, А.А. Иванюк, Программное средство моделирования функциональных неисправностей встроенных ОЗУ: Минск: А.Н. Вараксин, 2009, с. 1-2