

## ПРОГРАММНО-АППАРАТНОЕ СРЕДСТВО МИКРОПРОГРАММНОГО УПРАВЛЕНИЯ ТЕСТИРОВАНИЕМ ПАМЯТИ

*Головин Е.С., студент*

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

*Иванюк А.А. – д-р техн. наук, профессор*

Тестирование – это ключевой этап в любой разработке Систем на Кристалле (СнК). Особенно важным делает этот этап то, что в аппаратных средствах отсутствует возможность динамического реконфигурирования, потому каждая ошибка, совершенная проектировщиками, может привести к непригодности целой партии чипов. Однако на сегодняшний день тестирование СнК производится и после того, как устройство передаётся конечному потребителю. Достигается это за счёт различных средств встроенного самотестирования, которые проверяют ключевые для работы блоки на исправность. В данной работе представлено устройство встроенного самотестирования памяти с микропрограммным управлением, которое поддерживает режимы разрушающего и неразрушающего тестирования памяти.

В области тестирования памяти маршевые тесты стали стандартом индустрии в силу их максимального быстродействия и возможности обнаруживать все неисправности класса Stuck-at Fault, а также частичное обнаружение неисправностей взаимного влияния современными алгоритмами [1]. Маршевые тесты можно представить как набор маршевых элементов. Маршевым элементом называется набор операций чтения и записи данных, которые последовательно выполняются над одной ячейкой памяти. Операции, относящиеся к одному маршевому элементу, имеют одинаковое направление изменения адресов ОЗУ, за счёт чего и возможно выполнение всех операций на одной ячейке последовательно.

Маршевые тесты стали часто применяться в Memory Built-In Self-Test (MBIST) схемах, которые использовались для проверки ячеек ОЗУ на наличие неисправностей уже после того, как устройство попадало к конечным пользователям. Эти схемы выполняли один заданный алгоритм проверки памяти и строились на жёсткой логике. Однако у таких схем имеется один недостаток – полное отсутствие гибкости. Из-за этого качество проверок памяти напрямую зависело от обнаруживающих способностей выбранного за основу маршевого теста и никак не могло быть улучшено новыми алгоритмами, способными обнаруживать различные классы неисправностей.

Для решения проблемы с гибкостью стали использоваться Programmable MBIST схемы. Их основным отличием является программируемая логика, с помощью которой задаётся по некоторому коду произвольный маршевый тест. Это позволяет использовать разные алгоритмы тестирования для обнаружения большего количества потенциальных неисправностей ячеек ОЗУ, а также спасает устройство самотестирования от морального устаревания, которое может происходить с классическими MBIST в силу непригодности по современным стандартам выбранного за основу построения схемы маршевого теста.

При создании кода, по которому маршевые тесты будут записываться в проектируемый Programmable MBIST, учитывались следующие два критерия:

1. Компактность кода. Это необходимо для минимизации итогового размера схемы тестирования и используемой для хранения маршевого теста памяти.

2. Простота кода. Необходимо для минимизации вероятности ошибки в кодировании маршевого теста и упрощения отладки.

Было принято решение использовать код операций, состоящий из трёх бит. Во втором бите кодируется признак последней операции в маршевом тесте. В первом бите кодируется направление изменения адресов: если поле равно 1, то адрес изменяется от младшего к старшему, иначе – от старшего к младшему. Нулевой бит отвечает за тип операции: если поле равно 0, то выполняется операция записи, в обратном случае выполняется операция чтения. В предложенном коде отсутствует отдельный бит для кодирования данных. Их значение вычисляется в зависимости от следующей операции в маршевом тесте: если это запись, текущее значение данных инвертируется, если чтение, то значение данных остаётся неизменным с предыдущей операции [2].

В устройстве реализован алгоритм валидации хранящегося во внутренней памяти схемы самотестирования маршевого теста. Необходимость его обусловлена тем, что маршевые тесты имеют произвольную длину и признаком окончания теста в предложенном коде служит выставленный Last Operation бит в последней операции. Перед началом проведения тестирования памяти происходит проверка записанного алгоритма. В случае отсутствия операции с Last Operation битом тест считается не валидным и устройство не выполняет проверку памяти по заданному алгоритму.

Тесты памяти по их воздействию на хранящиеся данные принято разделять на разрушающие и неразрушающие. Разрушающим называется тест, который не гарантирует сохранность данных, хранящихся перед проверкой в тестируемой области памяти. Неразрушающим называется такой тест

