

Архитектура средств встроенного самотестирования микросхем памяти

В.А. Андриенко¹, В.Г. Рябцев², Т.Ю. Уткина¹

¹Черкасский государственный технологический университет

²Европейский университет, volodja18@ukr.net

Аннотация — Рассматривается архитектура встроенных средств самотестирования, обеспечивающая получение всех возможных сочетаний кодов данных, хранящихся в смежных ячейках, окружающих базовые ячейки, что приводит к выявлению отказов, вызванных взаимным влиянием запоминающих ячеек.

Ключевые слова — функции Уолша, микросхема памяти, средства самотестирования.

I. ВВЕДЕНИЕ

Постоянное увеличение емкости и быстродействия микросхем оперативной памяти приводит к значительному увеличению количества отказов и сбоев как на стадии их изготовления из-за несовершенства конструкции и технологии, так и при эксплуатации управляющих и информационных вычислительных комплексов. Для повышения отказоустойчивости и надежности оперативных запоминающих устройств (ОЗУ) применяется ряд мер, позволяющих обнаружить, локализовать и ликвидировать последствия неисправностей. К таким мерам относятся методы и средства встроенного самотестирования.

Реализация встроенного самотестирования (ВСТ) значительно снижает стоимость и повышает процент покрытия неисправностей, так как проверка может производиться на рабочих частотах и не требуется внешнего тестового оборудования, стоимость которого, как правило, во много раз выше, чем стоимость самой микросхемы памяти [1]. Существующие встроенные средства JTAG-сканирования, для доступа к ячейкам памяти при выполнении каждой операции записи или чтения, требуют ввода нескольких векторов интерфейса, что снижает частоту диагностирования, а, следовательно, и его эффективность [2].

При проектировании архитектуры средств ВСТ необходимо обеспечить сочетание высоких диагностических свойств формируемых тестов с небольшими аппаратными затратами. Наибольшее взаимное влияние оказывают друг на друга запоминающие смежные ячейки, поэтому при проектировании ВСТ необходимо обеспечить формирование всех возможных сочетаний, по крайней мере, в четырех смежных ячейках, окружающих базовую ячейку и расположенных относительно ее по "кресту". Формирование такого распределения данных программным способом может потребовать большого объема памяти микрокоманд, что приведет к увеличению объема дополнительных компонентов, увеличению рассеиваемой мощности и уменьшению информационной емкости микросхемы.

Цель данной работы – уменьшение аппаратных затрат встроенных средств самотестирования, обеспечивающих получение всех возможных сочетаний кодов данных, хранящихся в смежных ячейках, окружающих базовые ячейки, что приводит к выявлению отказов, вызванных взаимным влиянием запоминающих ячеек. При этом уменьшается емкость программы теста путем использования аппаратно-микропрограммного способа формирования теста, который объединяет высокое быстродействие с небольшими аппаратными затратами.

При проектировании архитектуры средств, встроенных в микросхемы памяти, обычно дополнительно в структуру добавляют группу мультиплексоров, которые коммутируют сигналы кодов операций, данных и адреса либо от внешних выводов, либо от внутри установленных формирователей. Устройство управления компонентами внутрисхемного тестирования обычно выполняют в виде автомата с "жесткой" логикой.

II. ПРИМЕНЕНИЕ ФУНКЦИЙ УОЛША ДЛЯ ФОРМИРОВАНИЯ ТЕСТОВ МИКРОСХЕМ ПАМЯТИ

С целью уменьшения количества состояний управляющего автомата, в работе [3] предложено использовать коды двумерных функций Уолша $Wal(k, l, x, y) = F\{H(k, x), H(l, y)\}$, которые формируются с помощью операции эквивалентности ($\sim$) между элементами множества строк первой матрицы Адамара $H(k, x)$ и элементами множества столбцов второй матрицы Адамара $H(l, y)$.

Значение кодов двумерных функций Уолша при допустимых значениях переменных $k = 0, 3$ и $l = 0, 3$, координат адреса строк x и координат адреса столбцов y можно определить по следующим формулам:

$$\begin{aligned} Wal(0, 0, x, y) &= 1; \\ Wal(0, 1, x, y) &= 0 \sim y_0; \\ Wal(0, 2, x, y) &= 0 \sim y_1; \\ Wal(0, 3, x, y) &= 0 \sim (y_1 \oplus y_0); \\ Wal(1, 0, x, y) &= x_0 \sim 0; \\ Wal(1, 1, x, y) &= x_0 \sim y_0; \\ Wal(1, 2, x, y) &= x_0 \sim y_1; \\ Wal(1, 3, x, y) &= x_0 \sim (y_1 \oplus y_0); \\ Wal(2, 0, x, y) &= x_1 \sim 0; \\ Wal(2, 1, x, y) &= x_1 \sim y_0; \\ Wal(2, 2, x, y) &= x_1 \sim y_1; \\ Wal(2, 3, x, y) &= x_1 \sim (y_1 \oplus y_0); \\ Wal(3, 0, x, y) &= (x_1 \oplus x_0) \sim 0; \\ Wal(3, 1, x, y) &= (x_1 \oplus x_0) \sim y_0; \\ Wal(3, 2, x, y) &= (x_1 \oplus x_0) \sim y_1; \\ Wal(3, 3, x, y) &= (x_1 \oplus x_0) \sim (y_1 \oplus y_0). \end{aligned}$$

Таблица 1

Распределение информации для запоминающего устройства емкостью 16 бит

	l			
k	0	1	2	3
0				
1				
2				
3				

Общее число комбинаций кодов двумерных функций Уолша при этом равно 16, что соответствует числу всех возможных сочетаний кодов в четырех смежных ячейках памяти, которые окружают базовую ячейку. Все комбинации кодов данных, получаемых с помощью функций Уолша, разные, поскольку в матрицах Адамара отсутствуют строки с одинаковыми элементами. Распределение информации для запоминающего устройства емкостью 16 бит, полученное с помощью двумерных функций Уолша, приведено в таблице 1.

В таблице 1 заштрихованные поля соответствуют 1, незаштрихованные – 0. При количестве ячеек памяти более 16, комбинации кодов мультиплексируются по всему объему памяти. Например, для переменных $l = 1, k = 1$ в качестве тестовых данных используется "шахматный" код, который будет записываться во все ячейки памяти. Если дополнить полученные комбинации кодов данных, приведенные в таблице 1, инверсными значениями, то образуется тестовая последовательность, которая обеспечивает получение всех возможных комбинаций кодов в смежных ячейках, когда базовые ячейки сохраняют 1 или 0.

Граф состояний цифрового автомата, формирующего коды функций Уолша, приведен на рис. 1.

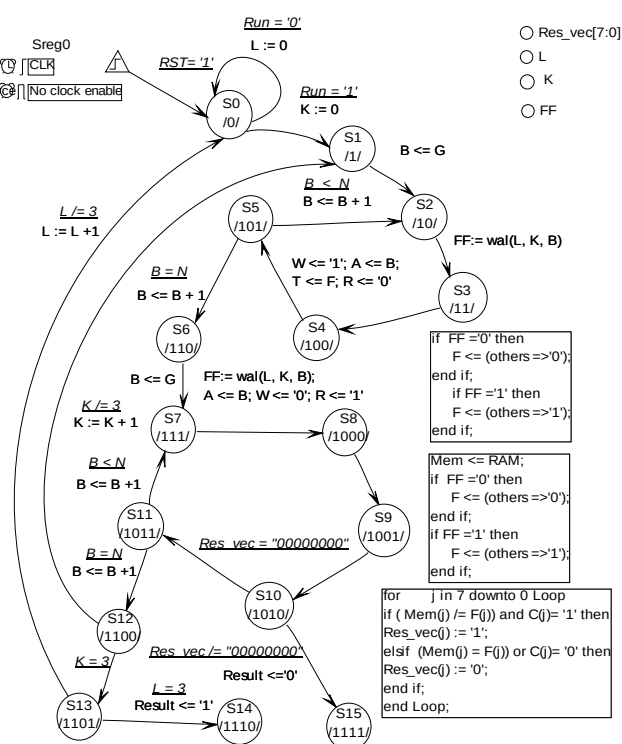


Рис. 1. Граф состояний цифрового автомата

Данный автомат применяется для формирования тестовых и эталонных воздействий при самотестировании полупроводниковых запоминающих устройств. Входные сигналы автомата образуют следующий кортеж: строб синхронизации – CLK , сигнал сброса – RST и сигнал "Пуск" – Run , считанные из ячеек памяти данные, которые поступают по шине Ram , коды разрешения сравнения данных C , определяющие размерность шины данных.

Под действием сигнала RST автомат переходит в состояние $S0$ и остается в нем, если отсутствует сигнал Run . По сигналу Run автомат начинает функционировать и в состояниях $S2-S5$ обеспечивает выполнение циклов записи кодов двумерных функций Уолша в ячейки тестируемой памяти. На выходе автомата формируются код адреса, коды данных, равных кодам двумерных функций Уолша, коды операций записи W и считывания R и сигнал результата тестирования $Result$.

Когда весь объем памяти заполнится данными, автомат переходит в состояние $S6$, а в состояниях $S7-S11$ осуществляется считывание данных из ячеек памяти и сравнение считанных данных с эталонными значениями, в качестве которых также используются коды двумерных функций Уолша.

Если на выходе автомата сформируется вектор результатов сравнения Res_vec , имеющий нулевое значение, то тест продолжается с новыми значениями переменных l и k .

Если содержимое вектора Res_vec равно нулю, то, после выполнения теста, на выходе автомата формируется сигнал $Result=1$ и автомат переходит в заключительное состояние $S14$, что соответствует работоспособному состоянию микросхемы.

Если, в процессе выполнения теста, в состоянии $S10$ будет зафиксировано неравенство считанных и эталонных данных, то автомат прекратит работу и перейдет в состояние $S15$, а на его выходе сформируется сигнал $Result=0$, что соответствует обнаружению неисправности в тестируемой микросхеме.

Длительность самотестирования составляет $64n$ циклов обращения к памяти, где n – емкость памяти.

Например, продолжительность тестирования микросхемы памяти емкостью 64 Мбит со временем цикла обращения равным 30 нс с помощью предлагаемого метода составляет 2 с.

Алгоритм работы микросхемы памяти в режиме самотестирования приведен на рис. 2.

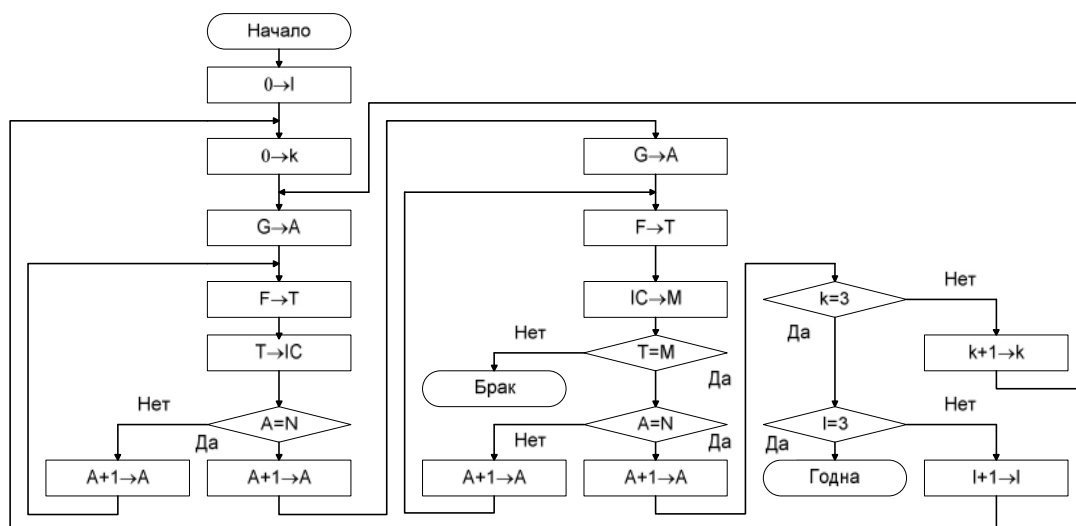


Рис. 2. Алгоритм самотестирования микросхемы памяти

Вначале счетчики K и L устанавливаются в нулевое состояние, затем выбирается начальный адрес массива ячеек памяти, в регистр контрольных данных заносится код двумерных функций Уолша и осуществляется запись информации в выбранную ячейку. Значение регистра текущего адреса увеличивается на единицу и аналогичным образом осуществляется запись кодов во все ячейки памяти.

Дальше опять устанавливается начальный адрес массива ячеек памяти, код двумерных функций Уолша запоминается в регистре эталонных данных, осуществляется считывание выбранной ячейки и сравнение данных. Операции повторяются для всех комбинаций счетчиков K и L . Для формирования инверсных данных тест повторяется дважды.

III. СТРУКТУРА КОМПОНЕНТОВ ВСТРОЕННЫХ СРЕДСТВ САМОТЕСТИРОВАНИЯ

Структурная схема формирователя кодов двумерных функций Уолша приведена на рис. 3 и состоит из двухразрядных счетчиков K и L , элементов ИСКЛЮЧАЮЩЕЕ ИЛИ $D1, D2, D4$, D -триггера $D3$ и инвертора $D5$. Под действием управляющих сигналов RST , CLK , Q и кодов микроопераций на выходе формирователя образуются коды двумерных функций Уолша.

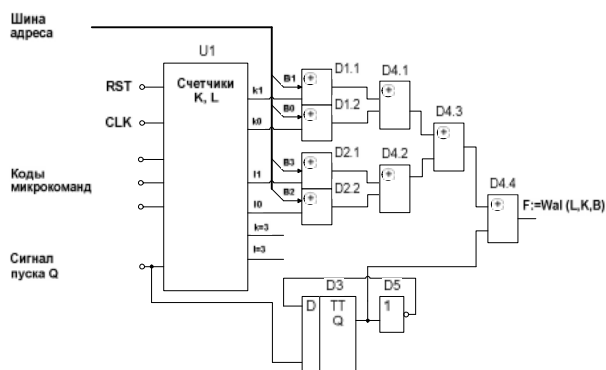


Рис. 3. Структурная схема формирователя кодов функций Уолша

Самотестирование микросхемы памяти следует выполнять два раза, т.к. после поступления каждого сигнала пуска Run код на выходе формирователя инвертируется при помощи включенных в схему D -триггера $D3$, инвертора $D5$ и элемента $D4.4$.

Для фиксирования состояний автомата достаточно четырехразрядного счетчика, структурная схема которого приведена на рис. 4 и содержит логические элементы $D1-D3$, $D6-D9$, счетчик $D4$, и RS -триггер $D5$.

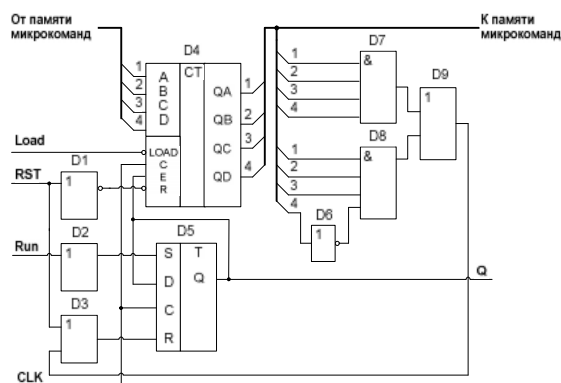


Рис. 4. Структурная схема счетчика состояний автомата

Коды с выходов счетчика используются в качестве кодов адреса памяти микрокоманд. На входы параллельного занесения счетчика поступают коды адреса перехода от памяти микрокоманд. Управление

работой счетчика состояний обеспечивается сигналами RST , Run , CLK и признаком параллельного занесения кода адреса перехода $Load$, который поступает от дешифратора микрокоманд. Если счетчик состояний автомата перейдет в состояние $S15=1111$ или $S14=1110$, то триггер $D5$ переключится в нулевое состояние и выполнение самотестирования микросхемы памяти прекращается.

Для проектирования моделей цифровых автоматов целесообразно использовать интегрированную среду проектирования цифровых устройств Active-HDL [4]. Основными функциями среды Active-HDL являются: синтез алгоритмов, представленных в виде графов состояний цифровых автоматов с конечным числом состояний, анализ ошибок вносимых пользователем, оптимизация алгоритмов, автоматическая компиляция алгоритмов в программы автоматизированного управления технологическим оборудованием на языке программирования VHDL и обработка файлов с проектами.

IV. ЗАКЛЮЧЕНИЕ

Технико-экономический эффект при применении кодов двумерных функций Уолша достигается за счет повышения качества микросхем полупроводниковой памяти на этапе их изготовления путем увеличения числа сочетаний кодов данных в смежных запоминающих ячейках, что обеспечивает выявление отказов и сбоев, вызванных взаимным влиянием ячеек памяти.

Если топология ячеек памяти не соответствует последовательности возрастания кодов адреса, то рекомендуется использовать дополнительные преобразователи кода адреса – скремблеры.

Если в микросхеме имеется несколько подматриц запоминающих ячеек, то для каждой из них необходимо применять отдельные формирователи кодов двумерных функций Уолша.

ЛИТЕРАТУРА

- [1] Ярмолик В. Н., Мурашко И. А. Методы и средства встроенного самотестирования с пониженным энергопотреблением // Доклады БГУИР. 2004. № 1. С. 102—114.
- [2] Городецкий А. Встроенные инструменты тестирования // Компоненты и технологии. 2009. № 3. С. 10—11.
- [3] Рябцев В. Г. Применение матриц Уолша для контроля БИС ОЗУ // Управляющие системы и машины. 1995. № 3. С. 28—30.
- [4] Семенец В. В., Хаханова И. В., Хаханов В. И. Проектирование цифровых систем с использованием языка VHDL. Харьков : ХНУРЭ, 2003. 492 с.