

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ ГЕНЕРАТОРА ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ НА ОСНОВЕ ПРИНЦИПА ДЕЦИМАЦИИ

А. А. Ясонов

*Учреждение образования «Гомельский государственный технический
университет имени П. О. Сухого», Беларусь*

Научный руководитель И. А. Мурашко

В процессе своего создания современное микроэлектронное устройство проходит два вида тестирования – функциональное тестирование и производственное тестирование. Несмотря на схожесть названий, целей и, иногда, методов функционального и производственного тестирования, между ними существует большая разница. Цель функционального тестирования – обнаружить ошибки в функционировании устройства или доказать, что устройство выполняет заданные функции без ошибок. Задача производственного тестирования – обнаружить производственные дефекты в уже изготовленном устройстве или доказать, что они отсутствуют. Основные требования при производственном тестировании – обеспечить приемлемый процент покрытия неисправностей (faultcoverage – FC). В настоящее время считается приемлемым FC порядка 95 %.

Очевидным способом производственного тестирования является тестирование устройства с использованием уже существующих функциональных тестов, что довольно часто применяется. Практика показывает однако, что такой подход не слишком хорош. Во-первых, наборы функциональных тестов для сложных устройств, как правило, довольно объемны. Следовательно, дорогостоящее тестовое оборудование будет сильно загружено, пока будет проверена вся партия кристаллов. Во-вторых, функциональные тесты не предназначены для выявления производственных дефектов, поэтому нельзя быть уверенным в том, что они способны выявить максимальное количество дефектов. Поэтому практическим стандартом стал способ производственного тестирования с использованием специального порта – TAP-порта, или JTAG интерфейса, который перешел в системы на кристалле по наследству от печатных плат.

Основная идея использования JTAG интерфейса заключается в следующем. На стадии синтеза устройства по RTL описанию в современных компиляторах имеется возможность так устанавливать параметры синтезируемого устройства, что в него вносятся дополнительные цепи, соединяющие все триггеры схемы в одну цепочку (цепочка может быть и несколько). Одновременно компилятор генерирует систему тестовых воздействий для построенной цепочки так, чтобы процент покрытия неисправностей был приемлемым (считается приемлемым процент покрытия равный 97 %). В процессе тестирования тестовые комбинации проходят по цепочке до нужных элементов. Достоинствами использования JTAG интерфейса по сравнению с ранее описанным подходом является более эффективное использование тестового оборудования и больший процент обнаруживаемых дефектов. Недостатками использования JTAG интерфейса можно считать необходимость наличия тестового оборудования. Это не позволяет тестировать устройство конечному пользователю, в том числе в процессе эксплуатации, что бывает часто необходимо, например, в системах криптографии. Внесение дополнительных цепей, не существующих на RTL уровне, также не всегда бывает приемлемым, кроме того, они могут привести к неконтролируемой деградации производительности устройства, что в высокопроизводительных системах также неприемлемо.

Есть еще один подход к тестированию устройств – самотестирование (BIST – built-in-self-test), в котором отсутствуют указанные недостатки. Основными достоинствами самотестирования является дешевизна, а также возможность тестирования устройства конечным пользователем. Главным достоинством, конечно, является отсутствие необходимости использования какого-либо внешнего дополнительного оборудования.

Основная идея самотестирования – использование внутренних аппаратных ресурсов для целей тестирования. Разумеется, полностью обойтись без дополнительного оборудования в процессе самотестирования невозможно, но желательно, чтобы оно было минимальным. В настоящее время считается приемлемым 10–15 % дополнительной аппаратуры при условии достижения нужного процента FC.

В организации самотестирования можно выделить следующие основные проблемы:

- генерация тестовых воздействий;
- сбор информации;
- внесение дополнительных точек наблюдения;
- переконфигурация устройства или его отдельных частей.

Генерация тестовых воздействий и сбор информации – функции BIST-контроллера. BIST-контроллер – это специализированное устройство, предназначенное исключительно для организации самотестирования. На его долю обычно приходится

90–100 % дополнительного оборудования, следовательно, главное требование, которое к нему предъявляется, в дополнение к указанным функциям – компактность.

Использование случайных тестовых воздействий гораздо экономичнее как с точки зрения аппаратных затрат, так и с точки зрения количества тестовых векторов, необходимых для достижения нужного процента ФС. Для этого целесообразно использовать хорошо известное устройство – сдвигающий регистр с линейными обратными связями (linierfeedbackshiftregister – LFSR).

Чтобы выходная последовательность такого устройства была максимально длинной, необходимо, чтобы обратные связи соответствовали коэффициентам полинома, для которого коэффициент пересчета (возврат в исходное состояние) был 2^N , где N – число разрядов регистра. На рис. 1 изображена реализация LFSR, соответствующего полиному $X^8 + X^6 + X^5 + X + 1$.

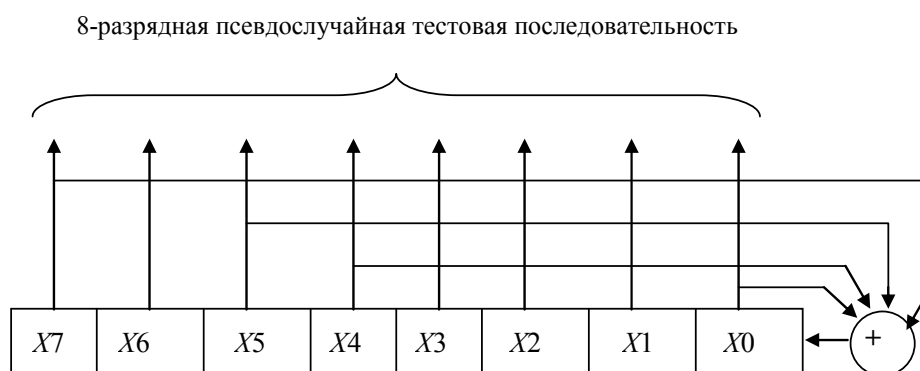


Рис. 1. Сдвигающий регистр с линейными обратными связями

Разрядность LFSR выбирают, исходя из практических соображений. Она не может быть меньше $\log_2(K)$, где K – число тестовых векторов. Как правило, 16-битового LFSR достаточно для тестирования большинства цифровых устройств. В случаях, когда LSFR используется для тестирования больших устройств, состоящих из нескольких СФ блоков, его разрядность может быть увеличена до 32.

При разработке встроенных средств самотестирования часто возникает задача подбора порождающего полинома определенной степени для повышения качества тестирования. Таблицы примитивных полиномов содержат, как правило, по одному полиному каждой степени. Кроме того, уже для 19-й степени существует уже 27594 примитивных полиномов. Хранить таблицы полиномов нецелесообразно из-за их объемов. Поэтому разработаны различные методики формирования множества полиномов заданной степени. Для решения этой задачи также может быть применено свойство децимации M -последовательности (последовательности максимальной длины). Алгоритм нахождения примитивных полиномов достаточно прост.

1. Из таблицы выбирается полином заданной степени m . Для него строится порождающая матрица V и находится период M -последовательности: $L = 2^m - 1$.

2. Выбирается коэффициент децимации q . При этом на q накладываются следующие ограничения:

– $(L, q) = 1$, т. е. L и q должны быть взаимно просты;

– $q \neq 2^i$, $i = 1, 2, 3, \dots$. Данное ограничение связано с тем, что при $q = 2^i$ новый полином будет совпадать с исходным;

476 Секция IX. Информационные технологии и моделирование

– $q \neq (L-1)2^i$, $i = 1, 2, 3, \dots$. Данное ограничение связано с тем, что при $q \neq (L-1)2^i$ формируется взаимно-обратный полином, который можно вычислить исходя из следующего выражения. Для полинома $\varphi(x)$ степени m существует взаимно-обратный полином $\varphi^{-1}(x)$, который связан с ним соотношением $\varphi^{-1}(x) = x^m \varphi\left(\frac{1}{x}\right)$. Если $\varphi(x)$ примитивный, то и взаимно-обратный полином также является примитивным.

3. Находится V^m и в соответствии с формулой $p(x) = \det(V \oplus Ix)$, где V – порождающая матрица, единичная диагональная матрица того же ранга, что и матрица V , находится новый полином.