2. Ярмолик В.Н., Иванюк А.А., "Встроенное самотестирование памяти с использованием сигнатурного анализа", "Логическое проектирование", Минск, 1997.

УДК 681.3

АЛГОРИТМИЧЕСКИЕ И ПРОГРАММНЫЕ СРЕДСТВА СИНТЕЗА САМОТЕСТИРУЕМЫХ СБИС

Хомич О. В.

Белорусский государственный университет информатики и радиоэлектроники

Использование встроенных схем самотестирования является одним из важнейших методов обеспечения надежного функционирования СБИС. Эффективность самотестирования оценивается длительностью теста и аппаратными затратами на реализацию схемы тестирования.

Практически при всех методах диагностирования СБИС широкое применение находит использование линейного сдвигового регистра с сумматорами по модулю два в цепи обратной связи (LFSR) в качестве генератора псевдослучайных тестовых наборов в следствие относительно небольших аппаратных затрат на его реализацию [2].

Большинство комбинационных схем содержит неисправности, обнаружение которых с помощью только псевдослучайных тестовых наборов зачастую оказывается невозможно. В таких случаях для достижения максимальной полноты покрытия неисправностей широкое применение получило использование детерминированных тестовых наборов совместно с псевдослучайными тестовыми последовательностями. Чаще всего применяется одна из разновидностей метода сканирования пути, что не приводит к ухудшению временных характеристик работы тестируемых СБИС [1].

Основным недостатком всех используемых на данный момент методов является увеличение аппаратных затрат на реализацию генератора тестовых последовательностей, что в ряде случаев является критичным для построения самотестируемых СБИС.

Показано, что при использовании LFSR в качестве генератора псевдослучайных тестовых наборов большинство генерируемых последовательностей являются бесполезными с точки зрения обнаружения неисправностей в комбинационной схеме [1]. В то же время, обнаружение конкретных неисправных состояний СБИС часто возможно только с помощью детерминированных тестовых наборов, которые специфицируются в зависимости от конкретной тестируемой комбинационной схемы. Как правило, в псевдослучайных тестовых наборах число бит, требующих изменения для получения заданных тестовых наборов достаточно мало по сравнению с общей длиной тестовой последовательности [1].

Более привлекательным видится нахождение способа анализа порождающего полинома, позволяющего доказать наличие необходимых детерминированных тестовых наборов во всей совокупности генерируемых тестовых последовательностей, а также, при невозможности генерирования необходимых детерминированных тестовых наборов, нахождение примитивного полинома, обеспечивающего получение требуемых последовательностей. Данный анализ работы генератора возможен в следствие того, что количество примитивных полиномов заданной степепени для больших значений степени достаточно велико. Таким образом, доказательство наличия требуемых детерминированных наборов в выходной совокупности генерируемых последовательностей, а также, при необходимости, выбор удовлетворяющего требованиям примитивного полинома, является, по сути, решением задачи по обеспечению генерирования детерминированных тестовых последовательностей, применение которых обеспечивает необходимую полноту покрытия неисправностей СБИС.

Предлагается метод анализа порождающего полинома генератора псевдослучайных тестовых последовательностей, доказывающий наличие

необходимых детерминированных тестовых наборов во всей совокупности генерируемых тестовых последовательностей, а также, при невозможности совместного генерирования заданных детерминированных тестовых наборов, нахождение примитивного полинома, обеспечивающего получение требуемых последовательностей. Описываемый анализ генератора возможен на этапе проектирования схемы самотестирования СБИС. Преимущество предлагаемого подхода заключается в отсутствии аппаратных затрат для генерирования детерминированных тестовых наборов.

Исходными данными для поиска начального состояния генератора, обеспечивающего наличие необходимых детерминированных тестовых наборов во всей совокупности генерируемых псевдослучайных тестовых последовательностей является порождающий полином, описывающий работу генератора, длина цепи сканирования и множество детерминированных тестовых наборов со специфицированными значениями бит.

Алгоритм анализа тестового генератора содержит следующие этапы.

1. Необходимо проанализировать состояние цепи сканирования на предмет заполнения соответствующими битами тестовой последовательности. Тестовый генератор, реализованный на основе примитивного полинома, будет обеспечивать генерирование периодически повторяющихся тестовых наборов. В следствие этого для каждого бита цепи сканирования можно поставить в соответствие биты детерминированного тестового набора.
2. На основании анализа, проведенного на шаге 1, составляются системы уравнений для каждого тестового набора. Данные системы включают уравнения для каждого специфицированного бита. Количество систем определяется количеством детерминированных последовательностей, а количество уравнений в каждой системе - количеством специфицированных бит в соответствующем тестовом наборе.
3. Ищется полином, описывающий генератор тестовой последовательности, получаемой путем децимации исходной тестовой последователь-

ности. Причем коэффициент децимации необходимо взять равным длине цепи сканирования.

4. Каждому биту исходной последовательности ставится в соответствие бит децимированной последовательности.
5. С учетом соотношений, полученных на шаге 4, переходят от систем уравнений, полученных на шаге 2, к системам относительно бит децимированной последовательности.
6. Вычисляются коэффициенты для получения сдвинутых копий m-последовательности для величины сдвига, равной номерам бит децимированной тестовой последовательности, относительно которых составлены системы уравнений на шаге 5.
7. Системы уравнений, полученные на шаге 5, преобразовываются относительно начальных состояний генератора. Причем коэффициентами при неизвестных выступают соответствующие значения коэффициентов для получения копий сдвинутой m-последовательности.
8. Находятся решения систем, полученных на шаге 7. Решения систем уравнений являются состояниями генератора в разные такты его работы.
9. На основании значений решений систем уравнений, полученных на шаге 8, производится анализ возможности генерирования необходимых детерминированных тестовых наборов. При невозможности совместного генерирования заданных детерминированных тестовых наборов, производится выбор другого примитивного полинома соответствующей степени, и процедура анализа повторяется с шага 3.

Как видно из рассмотренного алгоритма, наиболее трудоемким и в большей степени определяющим эффективность предложенного анализа является этап, связанный с определением коэффициентов для получения значений m-последовательности, сдвинутой на произвольное количество тактов. На основании оценки быстродействия методов формирования сдвинутых копий m-последовательностей при реализации программного обеспечения за основу был выбран, так называемый, быстрый алгоритм

определения сдвинутых копий m-последовательности, предложенный в работе [3].

Предложенный метод анализа порождающего полинома генератора псевдослучайных тестовых последовательностей положен в основу программного комплекса, позволяющего определять возможность генерирования детерминированных тестовых наборов со специфицированными значениями бит при заданном порождающем полиноме и длине цепи сканирования. При невозможности генерирования необходимых детерминированных наборов для заданного тестового генератора производится поиск порождающего полинома эквивалентной степени, обеспечивающего наличие заданных тестовых последовательностей в выходной совокупности генерируемых тестовых наборов. В качестве исходных данных для эксперимента были использованы бенчмарки, предложенные в работе [1]. Полученные результаты позволяют рассматривать предложенный анализ тестового генератора, в качестве метода, обеспечивающего требуемое значение величины покрытия неисправностей без дополнительных аппаратных затрат на генерирование детерминированных тестовых наборов, что является очевидным преимуществом по сравнению с применяемыми в настоящее время методами самотестирования.

Литература

1. Gundolf Kiefer, Hans-Joachim Wunderlich. Using BIST Control for Pattern Generation. IEEE International Test Conference, pp. 347-355, 1997.
2. Sybille Hellebrand, Birgit Reeb, Stefen Tarnick, Hans-Joachim Wunderlich. Pattern Generation for Deterministic BIST Scheme. Proc. Int. Conf. On Computer-Aided Design, pp. 88-94, 1995.
3. Ярмолик В. Н., Демиденко С. Н. Генерирование и применение псевдослучайных сигналов в системах испытаний и контроля. - Мн.: Наука и техника, 1986.