

ISSN 1816-0301

ИНФОРМАТИКА

1(25)

ЯНВАРЬ-МАРТ

2010

УДК 519.873:519.718.7

Л.А. Золоторевич, А.В. Ильникова

ПОСТРОЕНИЕ ТЕСТОВ КОНТРОЛЯ ЦИФРОВЫХ СИСТЕМ НА УРОВНЕ МЕЖРЕГИСТРОВЫХ ПЕРЕДАЧ

Дается анализ состояния проблемы контроля сложнофункциональных больших интегральных схем. Рассматриваются задачи направленного построения тестов контроля на уровне межрегистровых передач (RTL) на языке VHDL. Класс функциональных неисправностей, рассматриваемых при направленном построении теста, соответствует неисправностям константного типа реализаций СБИС.

Введение

Требования к надежности цифровых систем постоянно возрастают не только в областях, в которых отказ может привести к катастрофическим событиям, но и во всех других применениях. Одной из самых важных особенностей систем обеспечения высокой надежности объектов электроники является их способность определять наличие или отсутствие ошибки функционирования объекта [1]. Построение тестов на всем интервале развития интегральной схемотехники является наукоемкой проблемой, которая до настоящего времени не получила эффективного теоретического и практического решения. Задача построения тестов принадлежит к классу NP-трудных [2]. В связи с высокой сложностью построения тестов контроля и диагностирования функционально сложных цифровых систем применяются методы контролепригодного проектирования, которые направлены на снижение сложности задачи. В то же время проблема построения тестов остается актуальной, так как все используемые подходы к проектированию не решают по разным причинам в полной мере задачу контроля цифровых систем.

Актуальность задачи генерации тестов обусловлена потребностью в тестах на всех этапах проектирования для анализа корректности проектов, так как методы формальной верификации не достигли уровня практического применения. На этапе производства тесты необходимы для контроля исправности готовых изделий, на этапе эксплуатации – для подтверждения исправности, работоспособности или правильного функционирования устройства [1]. На каждом из этапов жизненного цикла изделия необходимо контролировать объект с учетом моделей неисправностей, соответствующих реальным физическим дефектам, которые возникают в связи с рядом конструктивно-технологических факторов и дестабилизирующим воздействием внешней среды.

Стремление анализировать контролепригодность проекта на самых начальных этапах проектирования цифровой системы, желательно еще до проведения декомпозиции проекта с целью определения программно-реализуемой части проекта, положило начало исследований по иерархическому построению тестов – HTG (hierarchical test generation). В настоящее время известны две стратегии построения теста – нисходящее и восходящее проектирование. При восходящем проектировании [3] тесты, сгенерированные на более низком уровне, используются при построении тестов на более высоком уровне. При нисходящем проектировании информация, полученная на более высоком уровне, используется при построении тестов на более низком уровне представления [4]. Большое число публикаций по построению тестов свидетельствует о том, что наиболее широко исследуется возможность построения тестов на уровне межрегистровых передач RTL (register-transfer level) [5].

Ниже дается анализ состояния проблемы контроля сверхбольших интегральных схем. Предлагается метод построения иерархических тестов контроля. Метод основан на представлении объекта на системном уровне или RTL-уровне в виде диаграмм принятия решений. В отличие от известных классов неисправностей, заимствованных из области тестирования программных средств, рассматриваемые функциональные неисправности соответствуют неисправностям константного типа физического объекта. Решая задачу построения теста на начальных этапах проектирования цифровой системы, метод использует тесты функциональных блоков технологических библиотек проектирования, применяемых при синтезе объекта.

1. Состояние проблемы контроля цифровых устройств и систем

Самый простой подход к контролю комбинационных схем – это исчерпывающее тестирование, при котором на схему подается 2^n входных наборов, где n – количество входов схемы. Однако на практике этот подход не применим, так как при 100 внешних входах схемы необходимо подать примерно 10^{30} входных наборов. Альтернативным подходом является псевдослучайное тестирование, которое не гарантирует полноту контроля устройства.

Существующие в настоящее время системы контроля базируются на следующих подходах:

- функциональном контроле, который основан на исследовании устройства на рабочей частоте и проводится в течение рабочего функционирования устройства в режиме on-line;
- тестовом диагностировании объекта, которое основано на разработанных тестах контроля в различных классах неисправностей (константного типа, обрыва, замыкания и др.) в режиме приостановки эксплуатации объекта DUT (device under test).

Метод функционального контроля в режиме on-line требует большой тестовой последовательности для получения высокого качества контроля, что ограничивает его применение для схем большого объема.

Контроль в режиме off-line осуществляется с помощью специально разработанных тестов и основан в основном на применении модели одиночной неисправности константного типа SSA (single stuck-at) и внешнего диагностического оборудования для тестирования систем АТЕ (Automatic Test Equipment). На определенном этапе развития цифровой схемотехники считалось, что модель константной неисправности покрывает более 80 % всех дефектов, которые необходимо исследовать. Кроме того, предполагалось, что тест, покрывающий все одиночные неисправности константного типа, покрывает большинство множественных неисправностей. Однако уменьшение размеров элемента, переход в субмикронный диапазон, применение МОП-технологии требуют рассмотрения неисправностей расширенного класса, а также множественных неисправностей. Известны результаты в области моделирования [6] и построения тестов контроля МОП-структур в классах неисправностей типа «постоянно закрытый транзистор» (stuck-OPEN), замыкания.

Особого внимания требует проблема контроля неисправностей задержки. Неисправности задержки обусловлены ошибками проектирования, внешними факторами, влияющими на величину задержек компонентов структуры, конструктивно-технологическими особенностями, определяющими физическую длину межсоединений, и др. Неисправности задержки могут приводить к нарушению алгоритма функционирования объекта или уменьшению скорости работы при правильно реализуемой логической функции. Применяемые методы контроля неисправностей задержки базируются как на методах временного моделирования, так и на статическом временном анализе, основанном на детерминированном или статистическом подходе. Особенностью статического временного анализа является то, что он предполагает вычисление реализуемых функций [7]. Методы параметрического контроля, известные как IDDQ-тестирование, основаны на измерении величины тока в линиях схемы при статическом режиме работы устройства и позволяют определять неисправности задержки [8]. Они дают возможность обнаружить также и некоторые другие неисправности, которые нельзя проверить логическими методами (например, неисправности типа «постоянно открытый транзистор»). Недостатками данного подхода являются большие стоимость и время контроля.

Появившаяся в начале века и быстро развивающаяся тенденция проектирования систем на кристалле SoC (system-on-chip), последовательный переход в субмикронный диапазон все больше сокращают область применения внешнего диагностического оборудования для тестирования систем АТЕ в связи с применением методов контролепригодного проектирования DFT (design for testability). В то же время проблема построения тестов остается актуальной наряду с применением разных методов обеспечения контролепригодности и контроля объектов.

2. Особенности контроля систем на кристалле

Контролепригодность объекта определяется основными характеристиками проекта – показателями управляемости и наблюдаемости. Управляемость объекта – это его способность устанавливать требуемые логические состояния в узлах схемы, задавая некоторые значения вход-

ных сигналов, т. е. возможность управлять состояниями внутренних узлов со стороны внешних входов объекта. Наблюдаемость – это способность наблюдать хотя бы на одном из выходов схемы (в одной из контрольных точек) за состоянием внутреннего узла, управляя входными состояниями сигналов. Методы улучшения показателей управляемости и наблюдаемости подразделяются на две категории. Первая – это способность решить проблему применительно к конкретному изделию, улучшая его характеристики (например, путем декомпозиции). Другие подходы разрабатываются для формального применения во время проектирования объекта. Один из таких подходов основан на DFT-технике LSSD (level-sensitive scan design). Идея LSSD заключается во внутреннем сканировании цепей устройства, осуществляемом с помощью программного обрыва линий обратной связи контролируемого объекта, и последовательном анализе состояния каждой внутренней переменной объекта (каждого элемента памяти) на специальном выходе. Реализация данного подхода требует применения дополнительного оборудования, а также существенных временных затрат на проведение контроля [9].

Высокая размерность и сложность современных цифровых систем (порядка миллиарда транзисторов или сотни миллионов простых логических элементов), необходимость контроля неисправностей задержки, высокая стоимость построения теста, большой объем тестовой информации делают все менее эффективными системы тестового диагностирования, требующие внешнего диагностического оборудования и его применения в режиме off-line. Можно сказать, что в случае систем на кристалле контроль off-line с привлечением внешнего диагностического оборудования практически малоэффективен. Поэтому все более широко применяются средства встроенного самотестирования BIST (built-in self-test). Такой подход к обеспечению контроля не требует внешнего оборудования, которое интегрируется в состав активной инфраструктуры непосредственно на кристалле. Одним из методов реализации идеи BIST является применение методов компактного тестирования. Наиболее точным из методов компактного тестирования является метод сигнатурного анализа, основанный на применении сдвигового регистра с линейной обратной связью LFSR (linear feedback shift register), который может быть и генератором псевдослучайной последовательности. Анализ результатов контроля осуществляется на основе сравнения полученной сигнатуры с эталонной [9].

Применение методов встроенного самотестирования оказывается незаменимым при трудном доступе к некоторым блокам системы со стороны внешних входов (проблема контроля глубоких схем), при решении задачи сохранения интеллектуальной собственности в случае повторного применения IP-блоков и др.

Рассматриваемый метод контроля BIST привлекателен по той причине, что не требует внешнего диагностического оборудования и может применяться на всех этапах жизненного цикла изделия. Существенным недостатком метода является то, что он не гарантирует полноты проверки объекта. Поэтому данный метод хорошо совмещается с применением заранее разработанной тестовой последовательности, которая включается в последовательность, применяемую при сигнатурном анализе.

3. Представление объекта на поведенческом уровне и уровне RTL

В последнее время при решении задач проектирования и построения тестов объектов цифровой электроники символьными методами используются диаграммы принятия решений. Рассмотрим возможность их применения для создания математической платформы при решении поставленной задачи. Известны работы по представлению систем булевых функций в виде двоичных диаграмм принятия решений BDD (Binary Decision Diagrams) и по их редуцированию [10]. На уровне BDD эффективно решаются задачи построения тестов в классе неисправностей константного типа входных переменных [11]. В работе [12] предложено представление логической схемы в виде BDD, которая вначале была названа альтернативным графом, а впоследствии получила название структурно синтезируемой бинарной диаграммы (или схемы) принятия решений SSBDD (structurally synthesized binary decision diagram). В SSBDD, в отличие от BDD, заложена информация о структуре схемы, реализующей соответствующую функцию. Данная особенность BDD является существенной, так как позволяет расширить класс рассматриваемых неисправностей при построении теста на ее основе.

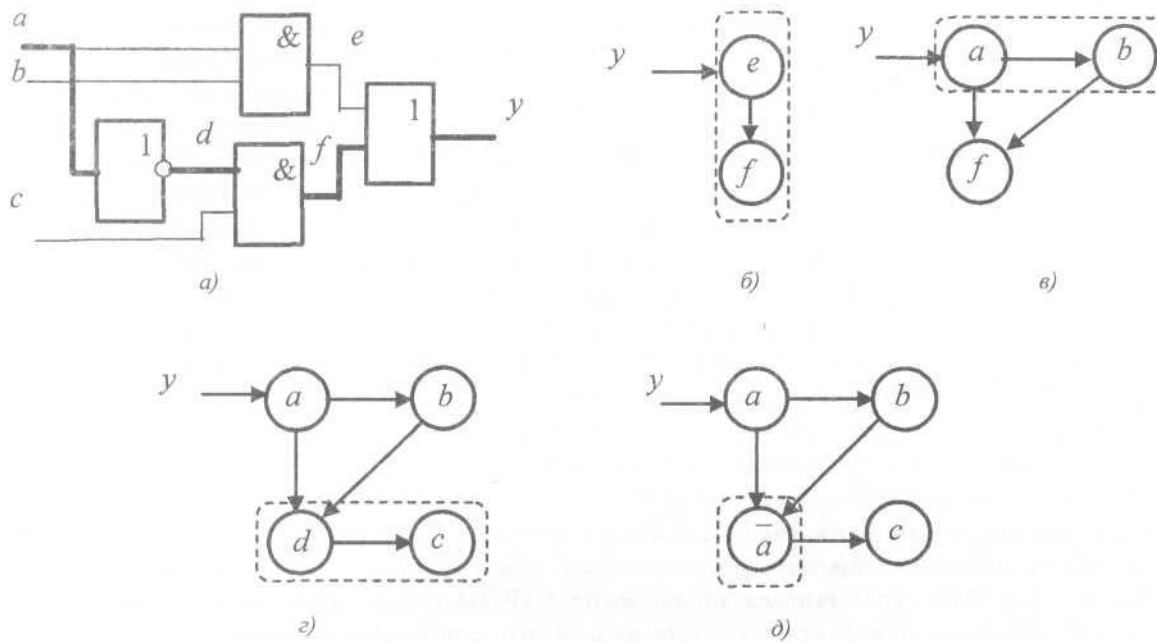


Рис. 1. Фрагмент структуры и соответствующее BDD-представление: а) логическая схема; б) – д) последовательные этапы построения BDD

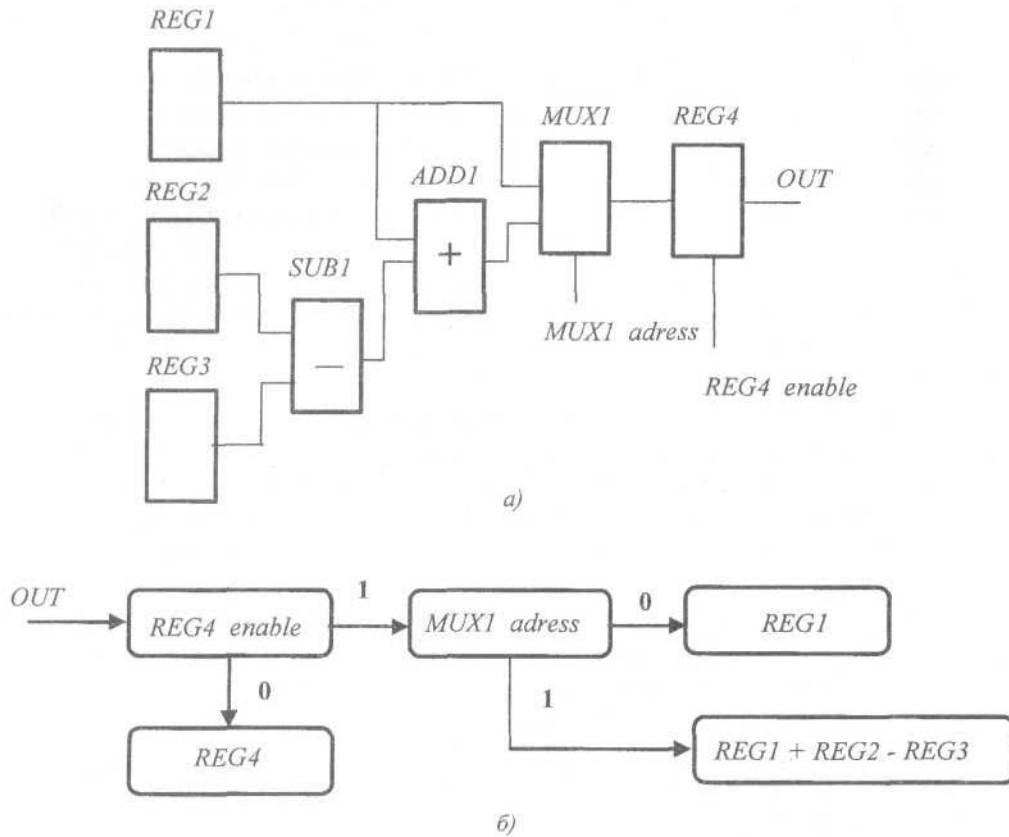


Рис. 2. Применение SSBDD-представления на уровне RTL: а) структурная схема; б) SSBDD-представление схемы

Рассмотрим правила построения SSBDD по вентиляльному представлению объекта [11] (рис. 1). В соответствии с соглашением стрелка вправо от узла соответствует 1, а стрелка вниз – 0. Терминальные узлы, соответствующие логическим состояниям 0 и 1, опущены. Построение SSBDD начинается от вентиля, который является источником выходного сигнала. Затем каждый элемент заменяется его функцией в виде BDD. На рис. 1, б показано SSBDD-представление для элемента ИЛИ, на рис. 1, в, д – процесс последовательного построения SSBDD-представления фрагмента структуры. На каждом этапе один элемент заменяется его представлением в виде BDD. На SSBDD можно проследить непосредственные отношения между узлами и путями сигнала в соответствующей схеме (рис. 1, д).

На рис. 2, а изображена схема преобразования и передачи информации между функционально сложными блоками некоторого цифрового объекта (представление на уровне межрегистровых передач) [13]. Пути преобразования информации в схеме определяются признаками *REG4_enable* и *MUX1_adress*, которые в рассматриваемом случае принимают булевы значения. Следует отметить важное свойство BDD и SSBDD, которое позволило описать в виде SSBDD данную схему (рис. 2, б): все переменные принимают булевы значения. Сложность применения SSBDD в общем случае обусловлена тем, что при переходе на уровень RTL значения переменных могут быть различными (целочисленными, многозначными и т. д.). Это ограничивает применение SSBDD рассмотрением уровня вентиляльной структуры. Поэтому для цифровой системы в целом неприемлемо представление как в виде BDD, так и в виде SSBDD по той причине, что далеко не все переменные имеют логическое представление информации.

4. Направленное построение тестов контроля на основе DD потока данных и DD потока управления

Задача построения тестов на уровне системного представления объекта, а также на уровне RTL, в отличие от функционально-логического структурного представления объекта, требует разработки математической платформы для построения модели, эффективной для работы с программным кодом (например, с кодом на языке VHDL). В качестве математической платформы для описания цифровой системы на поведенческом уровне или уровне RTL будем использовать описание в виде диаграмм принятия решений: DD (Decision Diagram) потоков данных и DD потока управления [13]. Это соответствует тому, что традиционно на уровне RTL цифровая система рассматривается как две подсистемы: операционная, выполняющая преобразование данных в соответствии с заданными алгоритмами, и управляющая, реализующая управление операционной частью (частями) системы.

Путь данных распространяется через функциональные блоки, которые могут представлять собой комбинационные или последовательностные схемы. При описании системы каждая выходная и внутренняя переменная описывается в виде DD потоков данных. Нетерминальные вершины DD потока данных содержат значения сигналов управления, а терминальные (оконечные) узлы описывают сигналы путей данных, т. е. первичных входов, регистров, операций и констант.

Управляющая часть описывается конечным автоматом, таблица переходов которого представляется единственной DD. Нетерминальные узлы представляют текущее состояние и входные условия (состояния) управляющей части, а терминальные – следующие логические состояния, которые управляют распространением и обработкой информации в системе.

Заметим, что описание объекта на языке VHDL на системном уровне и уровне RTL не несет информации о структуре объекта. Тем не менее, на этапе структурного синтеза каждый фрагмент описания будет реализован в виде определенного «механизма» [14], т. е. в виде некоторой структурной схемы в элементном базисе технологической библиотеки проектирования. Рассмотрим пример DD-представления фрагмента поведенческого описания проекта на языке VHDL (рис. 3). В данном случае диаграмма принятия решений представляет собой ориентированный граф без петель, в котором нетерминальные вершины представляют логические состояния (условия), терминальные вершины – операции (действия), а

связи указывают подмножество узлов (состояний), для которых будет выбираться узел приемника. В отличие от бинарных диаграмм принятия решений, сигналы или переменные нетерминальных узлов могут быть любыми булевыми (описание флагов, логических условий (состояний) и др.) или целыми числами (описание команд, областей (полей) контроля и т. д.). Переменные или сигналы терминальных узлов помечены константами, переменными (булево или целое число) или выражениями для вычисления значения целого числа. На поведенческом уровне DD потока данных генерируется для каждой внутренней и выходной переменной проекта. Такая DD потока данных имеет столько ветвей, сколько раз переменная появляется слева в операторе назначения. Кроме того, генерируется DD блока управления, которая описывает последовательность условий (состояний) активизации пути. Диаграммы, полученные на основе VHDL-кода, который описывает цифровой объект на системном уровне или уровне RTL, будем использовать как вычислительную модель при иерархическом построении тестов.

Рассмотрим задачу построения теста в следующей постановке. Объект представлен на уровне описания поведения или уровне RTL. Структурная реализация объекта отсутствует, но известны используемые технологические библиотеки синтеза. Необходимо построить тест контроля объекта, при этом модели рассматриваемых неисправностей должны соответствовать моделям неисправностей, свойственных реальному физическому объекту.

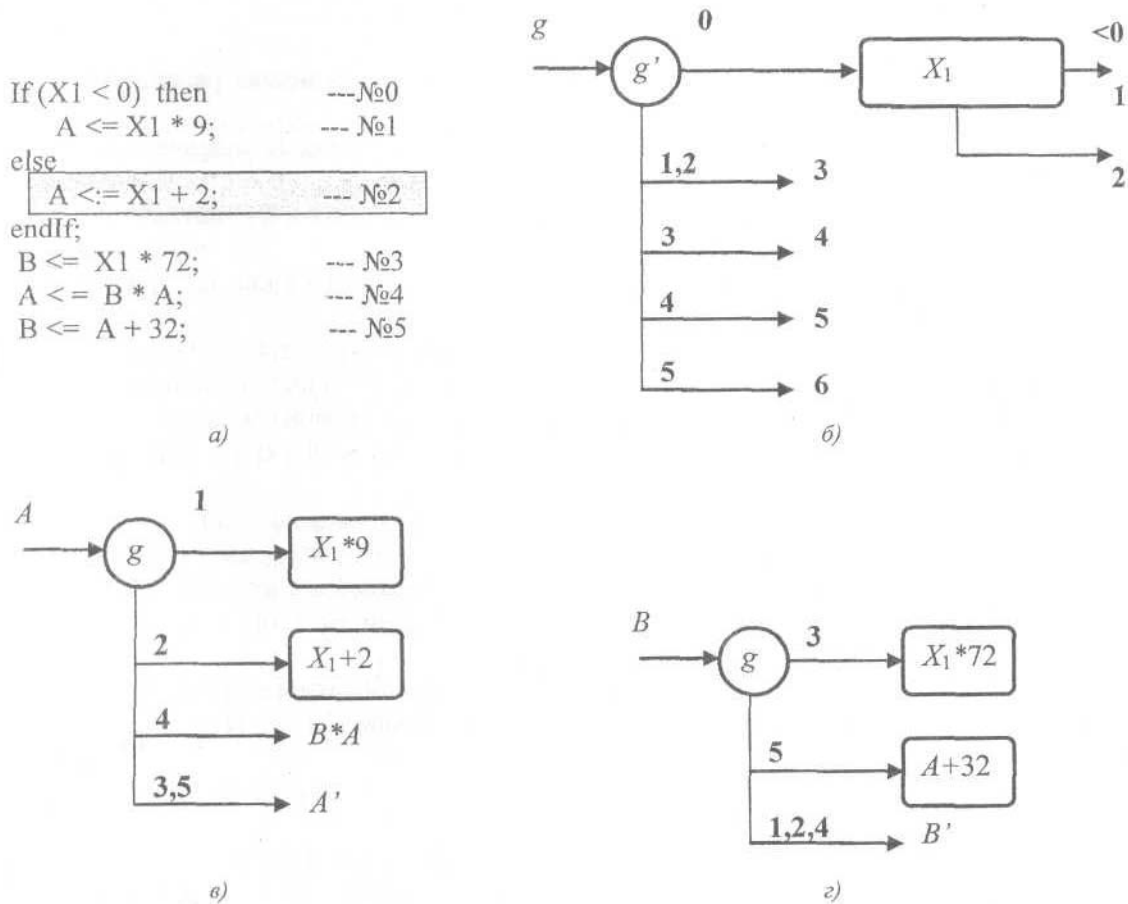


Рис. 3. Представление фрагмента программного кода в виде DD потока управления и потоков данных: а) фрагмент кода программы с описанием некоторого объекта на уровне RTL; б) DD потока управления; в) DD потока данных для переменной A; г) DD потока данных для переменной B

Воспользуемся известной идеей Рота [17] активизации одномерного пути для построения тестов контроля цифровых устройств на структурном уровне и ее развитием в направлении активизации многомерного пути. Применим подход Рота к разработке алгоритма построения тестов

та контроля цифровой системы на уровне поведенческого представления или RTL-уровне. Как и на структурном уровне, для объекта, представленного в виде DD потоков данных и DD потока управления, при построении теста методом направленного поиска необходимо активизировать неисправность, распространить ее влияние хотя бы к одному выходу и доопределить значения входных переменных.

По VHDL-описанию строим DD потоков данных и DD потока управления [13, 15], которые переводим в множество предикатов изменения состояния системы и ограничений активизации пути [16]. В последующем эти предикаты будут использоваться на этапе разрешения ограничений при активизации и распространении неисправностей.

Во фрагменте кода (рис. 3, а) входной переменной является $X1$, внутренними и/или выходными сигналами являются A и B . Через g' обозначено предыдущее состояние системы. Под состоянием системы будем понимать состояние всех ее сигналов и переменных.

Положим, что необходимо проверить правильность выполнения оператора № 2 – $A \leftarrow X1 + 2$ (выделен на рис. 3, а), т. е. правильность функционирования «механизма» [14], обеспечивающего структурную реализацию функции данного оператора. Для этого прежде всего следует привести систему к тому состоянию, которое она должна иметь непосредственно перед выполнением данного оператора ($g = 2$). Условиями активизации являются $g' = 0$, а также $X1 \geq 0$. Для распространения эффекта неисправного оператора на выходы системы необходимо обеспечить условия транспортировки:

- 1) $B \neq 0$ (выведено из выражения $A := B * A$);
- 2) $X1 \neq 0$ (выведено из выражения $B := X1 * 72 \neq 0$).

При невыполнении данных условий неисправность «механизма» рассматриваемого оператора не может быть обнаружена.

Ниже приведен общий алгоритм построения теста на уровне поведенческого представления объекта или уровне RTL:

1. По VHDL-описанию объекта строим DD потоков данных и DD потока управления (используется синтезируемое подмножество языка VHDL).

2. Выбираем очередной узел DD (оператор программного кода, контролируемый «механизм»).

3. Для выбранного «механизма» применяем очередную функциональную неисправность. Методика построения моделей функциональных неисправностей для «механизмов» исполнения конструкции if-then-else приведена в работе [18]. Вопрос построения моделей функциональных неисправностей для «механизмов» реализации арифметико-логических операций рассматривается в работе [19].

4. Извлекаем функциональные ограничения для активизации данной неисправности и распространения ее эффекта к выходам системы. Необходимо обеспечить следующее условие: хотя бы один бит вектора выходных переменных в «механизме» с внесенной неисправностью должен отличаться от соответствующего бита в исправном «механизме».

5. Если ограничения неразрешимы, переход к п. 3.

6. Извлекаем функциональные ограничения на этапе обратного прохода для расширения фиксации входных переменных, при разрешении ограничений сохраняем входные состояния. Переход к п. 3 или к п. 2.

7. В противном случае переход к п. 3, или к п. 2, или на выход при окончании процесса либо по истечении времени.

При разработке теста на первом этапе будем проверять ошибки выбора ветви (нетерминальные узлы в DD потока управления). На рис. 4 схематически показано соответствие между ошибкой оператора if-then-else (функциональной неисправностью моделируемой системы) и неисправностью константного типа некоторого «механизма» его структурной реализации.

В исправном состоянии мультиплексор реализует функцию $Z = X2 \wedge Y + X1 \wedge 'Y$. При неисправности типа const 0 на втором входе вентиля $M2$ на выходе схемы реализуется функция $Z = X2 \wedge Y$. Данной функции соответствует модель функциональной неисправности, показанной на рис. 4, в.

Для получения моделей функциональных неисправностей применяются имеющиеся средства построения тестов контроля цифровых устройств на функционально-логическом уровне. Генерация тестов производится в классе одиночных неисправностей константного типа.

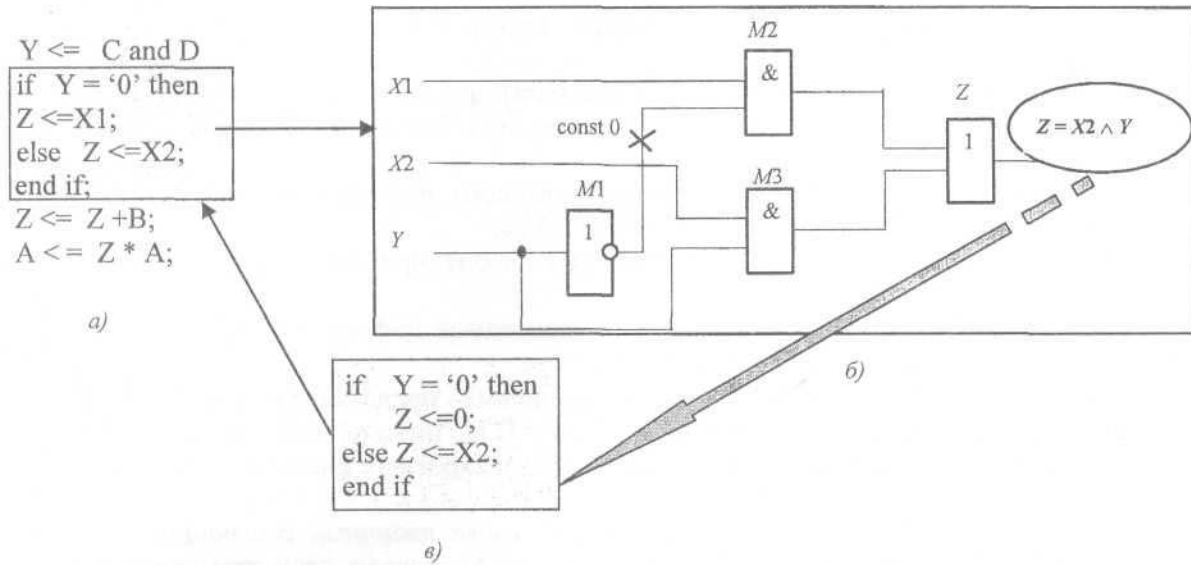


Рис. 4. Схема соответствия между функциональной неисправностью и неисправностью вентильной структуры: а) фрагмент кода описания объекта; б) одна из возможных схемных реализаций выделенного блока описания в виде мультиплексора; в) функциональная неисправность, соответствующая физической неисправности *const 0* на выходе вентили *M1*

Предложенный метод построения теста работает с высокоуровневым представлением объекта контроля, что на порядок уменьшает размерность задачи построения теста по сравнению с его функционально-логическим представлением на структурном уровне. Метод позволяет использовать тесты контроля блоков, которые покрывают их реальные физические неисправности, в среде генерации тестов на высоком уровне. Иерархический подход к генерации тестов позволяет на самом высоком уровне анализировать способность системы к тестированию. При этом можно сравнивать контролепригодность объекта в зависимости от используемой технологической элементной базы. Можно генерировать тесты для различных возможных реализаций проектов (различных технологических библиотек) и выбирать решение, которое является лучшим с точки зрения способности к тестированию.

Заключение

В работе предложен метод направленного построения теста контроля цифровой системы при ее описании на поведенческом уровне или уровне межрегистровых передач. В основе метода лежит идея активизации многомерного пути, которая получила теоретическое и практическое решение при направленном построении теста на уровне структурного представления объекта. Особенность предложенного подхода заключается в использовании моделей реальных неисправностей объекта [18] при формировании множества рассматриваемых функциональных неисправностей.

Список литературы

1. Основы технической диагностики ; под ред. П.П. Пархоменко. – М. : Энергия, 1976. – 463 с.
2. Ibarra, O.H. Polynomially Complete Fault Detection Problems / O.H. Ibarra, S. Sahn // IEEE Transactions on Computers. – 1975. – Vol. C-24, № 3. – P. 242–249.

3. Murray, B.T. Hierarchical Test Generation Using Precomputed Tests for Modules / B.T. Murray, J.P. Hayes // International Test Conference. – Washington, 1988. – P. 221–229.
4. Lee, J. Architectural Level Test Generation for Microprocessors / J. Lee, J.H. Patel // IEEE Trans. CAD. – 1994. – Vol. 13, № 10. – P. 1288–1300.
5. Ubar, R. Fast Test Pattern Generation for Sequential Circuits Using Decision Diagram Representations / R. Ubar // Journal of Electronic Testing: Theory and Applications (JETTA). – 2000. – Vol. 16, № 3. – P. 213–226.
6. Золоторевич, Л.А. Переключательное квазистатическое моделирование СБИС. Сравнение методов по точности моделей / Л.А. Золоторевич, Д.И. Юхневич // Автоматика и телемеханика. – 1998. – № 9. – С. 130–141.
7. Интеллектуальные системы автоматизированного проектирования БИС и СБИС / В.А. Мищенко [и др.]. – М. : Радио и связь, 1988. – 272 с.
8. Проектирование самотестируемых СБИС : в 2 т. / В.Н. Ярмолик [и др.]. – Минск : БГУИР, 2001. – Т. 2. – 163 с.
9. Ярмолик, В.Н. Проектирование контролепригодных цифровых устройств / А.А. Иванов, В.Н. Ярмолик. – Минск : Бестпринт, 2006. – 296 с.
10. Бибило, П.Н. Оптимизация многоуровневых представлений систем булевых функций на основе диаграмм двоичного выбора / П.Н. Бибило, П.В. Леончик // Шестая Междунар. конф. «Автоматизация проектирования дискретных систем». – Минск, 2007. – С. 162–169.
11. Золоторевич, Л.А. Построение тестов на основе двоичных решающих диаграмм / Л.А. Золоторевич, В.А. Куликович // Материалы Пятой Междунар. науч.-техн. конф. «Информационные технологии в промышленности». – Минск, 2008. – С. 105–106.
12. Ubar, R. Test Synthesis with Alternative Graphs / R. Ubar // IEEE Design and Test of Computers. – 1996. – Vol. 13, № 1. – P. 48–57.
13. Jervan, G. Test Cost Minimization for Hybrid BIST / G. Jervan, Z. Peng, R. Ubar // IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT'00). – Japan, 2000. – P. 283–291.
14. Белкин, В.В. Тестовое диагностирование современных микропроцессоров с использованием функциональных моделей / В.В. Белкин // Автоматика и телемеханика. – 2008. – № 8. – С. 139–152.
15. Inoue, M. Test synthesis for datapath using datapath-controller functions / M. Inoue, K. Suzuki, H. Okamoto // Proc. of the 12th Asian Test Symposium (ATS'03). – China, 2003. – P. 294–299.
16. Georgiadis, L. Finding dominators revisited / L. Georgiadis, R.E. Tarjan // Proc. 15th ACM-SIAM Symp. on Discrete Algorithms. – New Orleans, 2004. – P. 862–871.
17. Roth, J.P. Diagnosis of automata failures: a calculus and a method / J.P. Roth // IEEE Trans. on Computers. – 1966. – Vol. 15, № 7. – P. 278–291.
18. Золоторевич, Л.А. Моделирование неисправностей СБИС на поведенческом уровне на языке VHDL / Л.А. Золоторевич // Информатика. – 2005. – № 3 (7). – С. 135–144.
19. Hansen, M.C. High-Level Test Generation using Physically-Induced Faults / M.C. Hansen, J.P. Hayes // Proc. VLSI Test Symposium. – Princeton, NJ, USA, 1995. – P. 20–28.

Поступила 22.06.09