

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ И КОНТРОЛЯ В РАДИО- И МИКРОЭЛЕКТРОНИКЕ

УДК 62-507 : 681.326

Г. В. ЗИЗИН, В. В. КАШТАНОВ, И. Е. ЛОБОВ,
В. Е. МЕЖОВ, Ю. А. ЧЕВЫЧЕЛОВ

(Воронеж)

ПРИНЦИПЫ И АЛГОРИТМ НАПРАВЛЕННОЙ ГЕНЕРАЦИИ ТЕСТОВ В СИСТЕМЕ ПРАЦИС-ТМ

Введение. Основными характерными особенностями рассматриваемого в данной статье способа генерации тестов являются: использование ускоренного алгоритма синтеза тестов для последовательностных схем, построенного на базе модифицированного FAN-алгоритма; повышение адекватности базовой тестовой модели посредством целевого задания задержек логических элементов и наличие аппарата, автоматизирующего процесс верификации тестовой модели; применение средств, позволяющих получить рекомендации по повышению контролепригодности тестируемой схемы как на этапе проектирования, так и в процессе синтеза тестов; предварительный экспресс-анализ полноты получаемых тестов, сокращающий время синтеза тестов и длину тестовых последовательностей; двухэтапная процедура верификации синтезированных тестов, уменьшающая общее время генерации тестов и их верификации и улучшающая общую структуру получаемых тестов.

Алгоритм синтеза тестов. Основным алгоритмическим компонентом данной системы является процедура синтеза тестов для комбинационных схем, построенных на базе FAN-алгоритма [1]. В настоящее время FAN-алгоритм еще не применяется в отечественных системах генерации тестов. Поэтому в данной статье мы сочли целесообразным привести его краткое описание.

Основные характерные особенности FAN-алгоритма (в отличие от *D*-алгоритма [2]) — совмещение процедур *D*-продвижения и обратного продвижения, а также управление состоянием схемы и сосредоточение процедуры возврата на узловых линиях (головных линиях и точках разветвления). Поясним терминологию. Граничными линиями мы будем называть выходы точек разветвления, а также все линии — последователи граничных. Тогда головную линию можно определить как предшественника какой-либо граничной линии, не являющегося граничной линией. Иными словами, головная линия — это последняя линия, лежащая на пути, соединяющем какой-либо внешний вход схемы с некоторой граничной линией. Остальные линии, лежащие на таких путях, будем называть свободными линиями. Алгоритм предусматривает поиск теста относительно головных линий, полностью определяющих состояние граничных линий схемы; доопределение свободных линий и окончательное построение внешнего входного вектора происходит на заключительном этапе построения теста. В этом отношении FAN-алгоритм подобен РОДЕМ-алгоритму [3], но в отличие от последнего при синтезе вектора особое внимание уделяется точкам разветвления как источникам возможных противоречий при обеспечении требуемого состояния. Анализ состояния точек разветвления ускоряет процесс поиска теста за счет

более раннего выявления противоречий и сокращения времени вычислений между возвратами.

Еще одна процедура, обеспечивающая ускорение процесса поиска теста,— процедура предварительной активизации пути. Она состоит в активизации всего или части D -пути в случае, если этот путь является единственно возможным для распространения неисправного эффекта (такая ситуация складывается, если на каком-либо этапе синтеза теста D -фронт состоит только из одного элемента).

Выбор элемента из D -фронта и обратное продвижение осуществляются, как и в D -алгоритме, с использованием параметров управляемости и наблюдаемости. Взаимодействие процедур D -продвижения и обратного продвижения в FAN-алгоритме происходит следующим образом: после выбора из D -фронта очередного элемента производится процедура обратного продвижения, в результате которой узлам схемы присваиваются некоторые значения, а сами узловые линии заносятся в список линий для выполнения процедуры возврата.

Определение состояния всех линий схемы, а также D -продвижение происходят в процессе выполнения процедуры импликации от узловых линий схемы. Точке разветвления значение присваивается в случае возникновения противоречивых требований к ее состоянию в процессе обратного продвижения. После очередного присвоения осуществляется процедура импликации от соответствующей узловой линии, в рамках которой выявляются противоречия, проводится анализ D -фронта и делается заключение о возможности дальнейшего D -продвижения или необходимости выполнения процедуры возврата.

Процедура возврата состоит в присвоении альтернативного значения последней занесенной в список узловой линии с последующей импликацией. Если альтернативное значение уже было выбрано, то текущей узловой линии присваивается неопределенное значение, а предшествующей узловой линии — альтернативное значение.

Для сокращения времени синтеза тестов (уменьшения количества возвратов) и расширения класса тестируемых схем в базовый алгоритм введен ряд существенных дополнений и изменений. Так, значительно расширена библиотека логических элементов: введены элементы с состоянием высокого импеданса и элементы монтажных соединений, позволяющие тестировать схемы, имеющие шинную организацию, расширен алфавит логических состояний, введены функциональные элементы памяти (триггеры), значительно ускоряющие процесс обработки итеративной модели и повышающие качество синтезируемых тестов за счет увеличения адекватности тестовой модели. Анализ и моделирование состояний на линиях схемы производится в семизначной логике: $0, 1, Z, D, \wedge D, X/[0, 1, Z, D, \wedge D], U$, где 0 — значение логического нуля; 1 — значение логической единицы; Z — состояние высокого импеданса; D — неисправный эффект (1 — исправный, 0 — неисправный); $\wedge D$ — неисправный эффект (0 — исправный, 1 — неисправный); $X/[0, 1, Z, D, D]$ — неопределенное значение, доопределяемое на множестве указанных значений или на каком-либо его подмножестве; U — непредсказуемое значение без возможности доопределения.

Неопределенное значение с доопределением на множестве — это неопределенное состояние на линии, которая при текущем состоянии схемы может быть установлена в одно из значений указанного множества. Непредсказуемое значение указывает на неопределенное состояние на выходе элемента, если при текущем состоянии схемы состояние на данной линии не может быть изменено. Расширение алфавита при помощи введения неопределенных состояний с доопределением на множестве значений и непредсказуемых значений позволяет сократить количество возвратов посредством более раннего обнаружения противоречий при синтезе тестов.

Следует отметить некоторые особенности и недостатки FAN-алгоритма при тестировании последовательностных схем. Так, процедура

обратного продвижения и присвоения значений точкам разветвления содержит определенный элемент случайности, т. е. обеспечение активизации выбранного пути распространения неисправного эффекта носит вероятностный характер, что может привести к непредсказуемым отклонениям при поиске оптимального решения, определяемого параметрами управляемости и наблюдаемости. Интуитивно влияние этого фактора при поиске теста на комбинационные схемы не должно быть значительным. Однако при синтезе тестов на последовательностные схемы такое алгоритмическое решение может оказать значительное отрицательное влияние. Это согласуется с тем соображением, что для последовательностных схем различия между оптимальными и наилучшими решениями в плане сложности построения теста могут быть значительно более существенными, чем для комбинационной логики.

Другой недостаток FAN-алгоритма применительно к тестированию последовательностных схем большой размерности (свыше 1000 вентилей) связан с возможностью некоторой избыточности при синтезе вектора, т. е. при обеспечении требуемого состояния могут быть произведены избыточные присвоения узловым линиям, не являющиеся в данной ситуации необходимыми.

К сожалению, в настоящий момент дать более точные сведения о степени влияния указанных факторов на эффективность работы системы не представляется возможным, так как исследования в данном направлении еще не закончены.

Общие принципы синтеза тестов для последовательностных схем. Для возможности использования алгоритма при тестировании последовательностных схем производится приведение схемы к условно-комбинационному виду и построение итеративной тестовой модели. При этом процедуры *D*-продвижения и обратного продвижения условно разделяются на две части: процедуры для комбинационной логики и для элементов памяти. Данное деление обусловлено особенностями построения тестовой модели схемы.

Моделирование поведения осуществляется традиционно путем создания последовательности копий тестируемой схемы, в которых, наряду с внешними входами и выходами, имеются также псевдовходы и псевдовыходы; состояние последних соответствует состоянию элементов памяти в предшествующий и текущий моменты времени. Моделирование переходов схемы из одного состояния в другое при подаче очередного входного вектора происходит посредством многократного последовательного моделирования комбинационной части схемы с учетом изменившегося состояния элементов памяти (состояния на псевдовходах и псевдовыходах очередной копии схемы), а результаты моделирования отражаются в последовательности создаваемых при этом копий схемы, называемых итерациями. Таким образом, процедуры *D*-продвижения и обратного продвижения разделяются на внутриитерационные, производящие обработку комбинационной логики, и межитерационные, обеспечивающие анализ и моделирование состояний элементов памяти.

В остальном процедуры *D*-продвижения и обратного продвижения выполняются обычным образом с использованием параметров управляемости и наблюдаемости. При этом процесс распространения неисправного эффекта происходит по принципу «вглубь», т. е. в качестве очередного элемента для *D*-продвижения выбирается элемент с наименьшей наблюдаемостью. Оптимизация процедуры обратного продвижения производится согласно параметрам управляемости.

В общем случае (при тестировании последовательностных схем) процесс синтеза тестов можно разделить на два этапа: активизацию неисправности и трансляцию неисправного эффекта на внешние выходы схемы.

Активизация неисправности заключается в построении последовательности входных векторов, приводящей схему в состояние, при кото-

ром значение на линии, ассоциируемой с данной неисправностью, в исправной схеме и при наличии этой неисправности было бы различно. В данном случае приведение схемы в требуемое состояние производится из любого (неопределенного) предыдущего состояния схемы. Таким образом, активизация неисправности неявно включает в себя установку схемы. Это придает всей системе синтеза тестов (в отличие от метода случайной генерации) необходимую гибкость, позволяющую сочетать различные методы синтеза тестов, а также модификацию полученной тестовой последовательности без потери тестирующих способностей всей довательности входных векторов, переводящих схему в состояние, при которых неисправный эффект, распространяясь по схеме, в конечном итоге проявится на одном из внешних выходов схемы или в любом другом месте, доступном для контроля.

В алгоритмическом плане при трансляции неисправного эффекта можно выделить две фазы: активизацию пути распространения неисправного эффекта (*D*-продвижение); обеспечение распространения неисправного эффекта по выбранному пути (обратное продвижение).

Однако в отличие от *D*-алгоритма, как отмечалось ранее, в данном методе эти процедуры совмещены по времени, т. е. процедура обратного продвижения производится после каждого шага *D*-продвижения (изменения *D*-фронта), при этом полное обратное продвижение (до внешних входов схемы) осуществляется при достижении эффекта неисправности выхода какого-либо элемента памяти, т. е. при запоминании неисправного эффекта. При распространении неисправного эффекта через комбинационную логику обратное продвижение производится в пределах текущей итерации.

Процесс трансляции неисправного эффекта является многотактным. Это означает, что при достижении неисправного эффекта какого-либо элемента памяти происходит загрузка соответствующего элемента памяти, после чего этот элемент памяти переводится в режим хранения, а условия, обеспечивающие предшествующую активизацию пути распространения неисправного эффекта, отменяются. В дальнейшем цикл повторяется. Таким образом, распространение неисправного эффекта производится последовательно от одного элемента памяти к другому с поочередным переводом соответствующих элементов памяти в режим хранения. Возможен также одноктактный режим, т. е. поддержание всего или части пути распространения неисправного эффекта в активном состоянии. Переход к этому режиму осуществляется автоматически в процессе перебора вариантов построения теста.

В итоге процесс построения тестов для последовательных схем выглядит следующим образом. На первом этапе активизируется неисправность, результатом чего является построение последовательности входных векторов, обеспечивающей не только активизацию неисправности, но также и распространение неисправного эффекта до ближайшего элемента памяти и запоминание в нем. Таким образом, на первом этапе производится *D*-продвижение от места возникновения (источника) неисправности до ближайшего элемента памяти и обратное продвижение с построением последовательности входных векторов, т. е. полное обеспечение активизации фрагмента *D*-пути посредством обратного продвижения до внешних входов схемы.

На втором этапе (этапе трансляции) взаимодействие процедур *D*-продвижения и обратного продвижения аналогично описанному выше, т. е. после активизации *D*-пути до очередного элемента памяти про-

изводится полное обеспечение активизации фрагмента D -пути с построением группы входных векторов. Таким образом, многотактный режим трансляции неисправного эффекта происходит при полном обеспечении активизации очередных фрагментов D -пути.

Результатом данного подхода к построению теста является тот факт, что даже в случае неудачи построения теста (невозможности трансляции по какой-либо причине неисправного эффекта на внешний выход схемы) производится синтез последовательности входных векторов, обеспечивающих распространение неисправного эффекта до выходов соответствующих элементов памяти. Это дает возможность автоматизированной генерации множества контрольных точек, с использованием которых осуществима регистрация сигнала на указанных линиях, выработать рекомендации по повышению контролепригодности тестируемых схем.

Еще одним важным следствием применяемого подхода является изменение стратегии поиска теста, т. е. стратегии возвратов и перебора вариантов построения теста. При этом значительно проще реализуется возможность создания гибкой технологии возвратов, что при дальнейшем развитии системы в значительной степени скажется на эффективности ее работы и, по нашему мнению, может придать ей черты интеллектуальной направленности при синтезе тестов.

Тестовая модель и способы повышения ее адекватности. В данной системе для синтеза тестов используется синхронная итеративная модель. Ее применение предполагает синхронное переключение элементов памяти, входящих в состав схемы, т. е. изменение состояния элементов памяти является результатом подачи очередного вектора на внешние входы схемы, а не переключения какого-либо элемента памяти. Система может быть использована также и для синтеза тестов на асинхронные схемы, однако в этом случае неадекватность поведения тестовой модели может привести к снижению качества синтезируемых тестов. Другая причина снижения качества синтезируемых тестов как для синхронных, так и для асинхронных схем — отсутствие учета задержек логических элементов.

Для повышения адекватности тестовой модели и качества тестов в данной системе предусмотрено выполнение ряда мероприятий. При описании элементов памяти (триггеров) предпочтение отдается функциональному описанию, так как на структурном уровне даже синхронные триггеры могут иметь асинхронный тип переключения. В этой системе возможно также целевое задание задержек в случае, если отсутствие учета какой-либо задержки приводит к неадекватному поведению тестовой модели. Величины задержек измеряются в условных относительных единицах, определяющих только отношение времени переключения различных логических элементов. Задание задержек производится посредством условного разрыва линии выхода соответствующего элемента с введением дополнительного псевдовхода и псевдовыхода, что для тестовой модели соответствует задержке в одну условную единицу. Для автоматизации целевого задания задержек предусмотрена процедура выявления элементов, поведение которых отличается в тестовой модели и при моделировании с реальными задержками.

Другим альтернативным вариантом повышения качества синтезируемых тестов является приведение схемы к синхронному виду посредством введения контрольных точек. Для этого в системе осуществляется автоматизированное формирование списка линий, распространение сигнала по которым может привести к появлению асинхронных переключений в схеме. Разрыв этих линий или размещение на них контрольных точек, обеспечивающих принудительную установку и регистрацию сигнала, не только приводит схему к синхронному виду, но также значительно повышает контролепригодность всей схемы.

Формирование множества неисправностей. В общем случае для обеспечения необходимой полноты проверки схем формируется полное

множество неисправностей заданного класса (данная система ориентирована на выявление класса одиночных константных неисправностей $\equiv 0$ и $\equiv 1$), т. е. возможность неисправности предполагается на каждой линии схемы.

В системе формируется характеристическое множество неисправностей (множество эквивалентных неисправностей, построение тестов на которые обеспечивает проверку схемы на полном множестве неисправностей), состоящее из неисправностей на внешних входах схемы и выходах точек разветвления. При описании схем с использованием функциональных элементов памяти исходное множество неисправностей дополняется неисправностями на выходах элементов памяти. Правомерность такого решения может быть легко доказана, если учесть, что любая линия схемы достигается от какой-либо линии характеристического множества, а также то, что при построении теста на какую-либо неисправность будут выявлены все неисправности, лежащие на пути распространения неисправного эффекта. С целью ускорения процесса синтеза тестов производится упорядочивание характеристического множества неисправностей. При этом неисправности в списке располагаются по мере их удаления от внешних входов (в порядке убывания параметра наблюдаемости соответствующих линий). Такое упорядочивание обеспечивает выявление наибольшего количества неисправностей на начальной стадии синтеза тестов.

Экспресс-анализ полноты синтезируемых тестов. Процесс синтеза тестов происходит до полного исчерпания списка неисправностей. При этом верификация синтезируемых тестов осуществляется после завершения построения всей тестовой последовательности. Такой режим повышает эффективность синтеза тестов, однако может привести к значительной избыточности при тестировании и в конечном итоге к снижению эффективности и качества работы всей системы. Избыточность связана с тем, что тест, построенный с целью выявления какой-либо конкретной неисправности, может обеспечивать также выявление целого ряда других неисправностей. Для устранения этого недостатка и исключения повторного синтеза тестов на выявленные неисправности в данной системе служит процедура экспресс-анализа полноты синтезируемых тестов, осуществляемая следующим образом.

После построения теста на очередную «целевую» неисправность производится формирование списка «сопутствующих» неисправностей. Формирование списка происходит методом поиска линий, лежащих на критических путях, т. е. линий, смена значений сигнала на которых приводит к изменению состояния на каком-либо внешнем выходе. Этот метод обеспечивает выявление неисправностей, лежащих не только на пути распространения неисправного эффекта, но также и на других линиях, что значительно повышает эффективность работы системы и предельно уменьшает избыточность тестовой последовательности.

Верификация тестов. Даже при выполнении всех мероприятий по повышению адекватности тестовой модели и обеспечению качества синтезируемых тестов получаемая последовательность входных векторов является лишь кандидатом в тест и требует верификации.

В описываемой системе верификация проходит в два этапа. На первом этапе производится проверка качества кандидата в тест посредством анализа тестирующих способностей относительно одной неисправности из группы неисправностей, для выявления которых предназначена данная последовательность входных векторов (целевой неисправности). После проверки качества выполняется процедура сохранения синтезированной последовательности с созданием результирующей тестовой последовательности, содержащей тесты, проверка качества которых дала положительный результат. Остальные кандидаты также сохраняются и могут быть использованы для определения причин неадекватного поведения тестовой модели. На втором этапе производится окончательная верификация результирующей тестовой последовательности.

сти и определение полноты проверки полученным методом тестов. При неудовлетворительной полноте после выполнения мероприятий по повышению адекватности тестовой модели и контролепригодности схемы процесс синтеза тестов может быть повторен.

Процедура верификации синтезированной последовательности входных векторов в два этапа сокращает общее время верификации за счет более раннего выявления входных векторов, имеющих низкое качество, так как окончательная верификация производится при моделировании в «неисправном режиме» и требует больших вычислительных затрат (верификация качества производится в режиме «исправного моделирования» с фальш-условием, т. е. методом двукратного моделирования с учетом неисправности и последующим сравнением результатов).

СПИСОК ЛИТЕРАТУРЫ

1. Hideo Fujiwara, Takeshi Shimono. On the acceleration of test generation algorithm // IEEE Trans. Comput.—1983.—С-32, N 12.— P. 1135.
2. Roth J. P., Bourieus W. G., Schneider P. R. Programmed algorithms to compute tests to detect and distinguish between failures in logic circuits // IEEE Trans. Comput.—1967.— EC-16, N 5.— P. 567.
3. Goel P. An implicit enumeration algorithm to generate test for combinational logic circuits // IEEE Trans. Comput.—1981.—С-30, March.— P. 215.

Поступила в редакцию 11 октября 1988 г.

УДК 681.5

С. Г. ДЕМЕНТЬЕВ, В. Г. СЕЛИВАНОВ
(Томск)

ГРАФИЧЕСКИЙ СПОСОБ ВВОДА ИНФОРМАЦИИ О ПРИНЦИПИАЛЬНОЙ СХЕМЕ В САПР ПЕЧАТНЫХ ПЛАТ

Системы автоматизированного проектирования (САПР) печатных плат, разрабатываемые в настоящее время, основываются на широком использовании средств автоматической или диалоговой трассировки и автоматического размещения элементов радиоэлектронной аппаратуры (РЭА) [1—3].

Исходными данными для программ трассировки и размещения являются список межэлементных связей (цепей) и перечень элементов, входящих в схему, которые составляются и вводятся, как правило, вручную. Подготовка и ввод этой информации в ЭВМ — трудоемкий и утомительный процесс, требующий значительных затрат времени. Такая процедура ввиду ее однообразия чревата появлением многих ошибок в синтаксических конструкциях входного языка и в описании топологии схем, причем если неточности синтаксиса можно определить с помощью программной диагностики, то ошибки в описании топологии зачастую выявляются уже в процессе настройки готового модуля РЭА [4]. В последнее время создаются системы, в которых этот процесс автоматизируется, и возникает сквозная автоматизированная технологическая цепочка получения фотошаблонов печатных плат [5, 6]. Совершенствуется и ввод графической информации о топологии печатных плат. В дополнение к традиционным кодировщикам, программное обеспечение для которых продолжает развиваться [7], появились устройства автоматического считывания координат изображения [8]. В некоторых системах используется ввод информации о схеме с помощью графических дисплеев [9], однако программная реализация и используемые