

УДК 658.5:681.3

В.А. АНДРИЯНОВ, А.Н. МАРТЫНЮК

Одесский национальный политехнический университет, Украина

МОДИФИКАЦИЯ СТРУКТУРНОГО ГЕНЕРАТОРА ТЕСТОВ FAN

Работа посвящена исследованию структурных алгоритмов генерации тестов. Целью исследования является модификация существующего алгоритма генерации тестов за счет сокращения числа откатов выполненных алгоритмом. Для этого выполнено уточнение процедуры расчета контролируемости. В работе также выполнен анализ алгоритмов, разработаны модификация алгоритма FAN с улучшенной эвристикой, использованной для управления обратной трассировкой, и прототип системы генерации тестов на базе исследуемых структурных алгоритмов: D, PODEM и FAN.

Ключевые слова: константные неисправности, активизация путей, тест, алгоритм FAN, процедура SCOAP.

Введение

В современных компьютерных технологиях по-прежнему большое значение уделяется разработке цифровых устройств. Требования, предъявляемые к надежности таких устройств высоки, так как ошибки и сбои в них могут привести к некорректной работе систем, в которые эти устройства входят. В то же время число критичных применений самих компьютерных систем высоко.

Как правило, цифровые устройства достаточно сложны и их производство следует начинать только после тщательного проектирования. Так как исправление ошибок в готовом устройстве требует больших затрат, ошибки важно выявить и исправить на этапе проектирования. Вместе с тем, всевозможные виды моделирования, верификации и тестирования на этом этапе не исключают возможности появления бракованных изделий. Кроме того, в процессе эксплуатации ошибки появляются в результате целого ряда причин, начиная от естественной деградации материала микросхем и заканчивая экстремальными условиями использования. Все это обуславливает необходимость развития и эффективной реализации разнообразных методов тестирования и, следовательно, синтеза самих тестов.

Среди алгоритмов синтеза тестов для сложных комбинационных схем и схем с памятью в качестве наиболее значимых выделяют алгоритмы D, PODEM [1] и FAN [2].

В общем случае обычно определяют два вида тестов [3]: параметрические и функциональные. Функциональное тестирование использует входные сигналы для получения выходных реакций. Генерация тестовых векторов и сценариев – одна из наибо-

лее важных составляющих классических задач анализа, синтеза, контроля и диагноза вычислительных систем. Вместе с тем, она до сих пор не является в полной мере разрешенной, в частности, из-за того, что темпы развития аппаратуры существенно превышают темпы развития генерации тестов. Проблема состоит в том, что все алгоритмы генерации тестов относятся к классу NP-полных задач и не всегда возможно ответить на вопрос, сколько времени требуется для построения теста для трудно проверяемых ошибок и сбоев [3]. В этой связи, в частности, актуальна задача сокращения времени поиска тестового решения, выполняемого за счет уточнения промежуточных решений, которые принимаются алгоритмами генерации тестов.

1. Анализ основных составляющих алгоритмов генерации тестов

Генераторы тестов решают задачу поиска наборов входных сигналов, организованных в вектора и структуры (сценарии), которые выявляют разницу в работе исправных и неисправных устройств. Кроме поиска тестов, алгоритмы генерации тестов имеют еще несколько применений, а именно [3]:

- нахождение избыточных частей в цифровых устройствах;
- доказательство соответствия разных реализаций цифрового устройства друг другу.

Для решения задачи поиска теста генератор теста должен иметь описание тестируемого устройства с заданным уровнем детализации. Чем выше уровень детализации, тем больше разных ошибок и сбоев могут быть промоделированы. Для моделирования, в частности, «const1» и «const0» необходи-

мо описать устройство в виде набора элементов и связей между ними. Если тестируются временные сбои, описание расширяется информацией о задержках распространения сигналов. Эти описания могут быть прямыми и непрямыми.

Простейший тест – исчерпывающий. В этом случае поиск тест-векторов не требуется, так как тест включает все возможные входные вектора (и их последовательности для устройств с памятью). Однако исчерпывающий тест обладает избыточностью, которую следует уменьшить. Например, предположение единичности ошибок и сбоев позволяет выполнить их сжатие на основе их оптимального покрытия тестовыми векторами с учетом их различимости (эквивалентности).

Все генераторы тестов используют структуры данных, представляющих пространство поиска тестовых векторов. Например, часто используются бинарные деревья поиска. При этом полный генератор может при поиске тестового вектора обойти все дерево. Не тестируемый сбой не выявляется даже после полного обхода дерева, это означает, устройство работает корректно даже в присутствии сбоя.

Алгебра генераторов тестов обычно имеет расширенную булеву нотацию, единую для корректной и дефектной схемы, что обуславливает один проход для определения значений. Простейшая алгебра – пятизначная $\{0, 1, X, D, \bar{D}\}$.

Используемые алгоритмы генераторов тестов для цифровых схем обычно псевдослучайные и детерминированные – на основе активизации путей.

В генераторах на основе активизации путей для константных неисправностей активизация и распространение обычно выполняются с помощью тех или иных процедур [4]:

- импликации;
- определения целей;
- обратной трассировки;
- определения возможности продолжения поиска;
- смены последнего решения.

Вместе эти процедуры образуют алгоритмы «согласования и связи» (“branch and bound” algorithm) [4], предназначенные для поиска и его прерывания на возможно более ранних стадиях, как только ясно, что путь не ведет к желаемому результату. Как следствие, уменьшаются вычислительные затраты.

Прямая или обратная импликация обеспечивает расчет полностью определенных сигналов в схеме. Удобная структура для импликации стек, где записываются наборы выбранных генератором сигналов в выбранных узлах, например, входах, схемы. Остальные процедуры реализуют алгоритм «согласования и связи».

Так, например, процедура определения целей выбирает промежуточные цели, существенно влияющие на объем вычислений, а процедура обратной трассировки преобразует промежуточные цели в узлы дерева выбора, используемого для поиска.

Процедура смены последнего решения меняет одно из решений, принятых ранее предыдущими процедурами, если формируется вывод о невозможности достижения требуемых результатов. От процедуры смены последнего решения также существенно зависит объем вычислений. Обычно смена решения происходит, если:

- фронт стал пустым (нет распространения);
- сигнал одновременно и «0», и «1», что невозможно.

Для поиска тестового вектора в прямом или обратном виде используются активизация и распространение сбоя. Активизация обеспечивает условия проявления сбоя, распространение обеспечивает возможность наблюдения проявления на выходах схемы [4].

2. Постановка задачи исследования

Задача определяется как модификация эффективных алгоритмов генерации тестов с целью возможно более раннего сокращения числа неверных решений этих алгоритмов.

Для решения следует решить подзадачи:

- анализа эффективных, программно реализуемых алгоритмов генерации тестов;
- разработка методики уточнения меры контролируемости на основе алгоритма SCOAP [5];
- модификации основных процедур, ориентированной на эффективную программную реализацию;
- разработки среды сопоставления алгоритмов и их реализаций, в частности PODEM и FAN.

3. Модификация алгоритма FAN

Рассмотрим часть схемы показанной на рис.1. Эта часть схемы состоит из трех элементов «E1», «E2», «E3». Для линий «A», «B», «C», «G» рассчитаны контролируемости нуля и единицы. Для активизации сбоя «константа 1» на линии «E» необходимо обеспечить логический ноль на выходе «E2». Этого можно достичь двумя способами:

- установить $C=0$ и $G=0$;
- установить $C=1$ и $G=1$.

Пользуясь наперед заданными мерами управляемости, алгоритм FAN выберет $C=0$ и $G=0$, потому что суммарная контролируемость входов элемента E2 по нулям меньше, чем контролируемость по единицам (27 против 82).

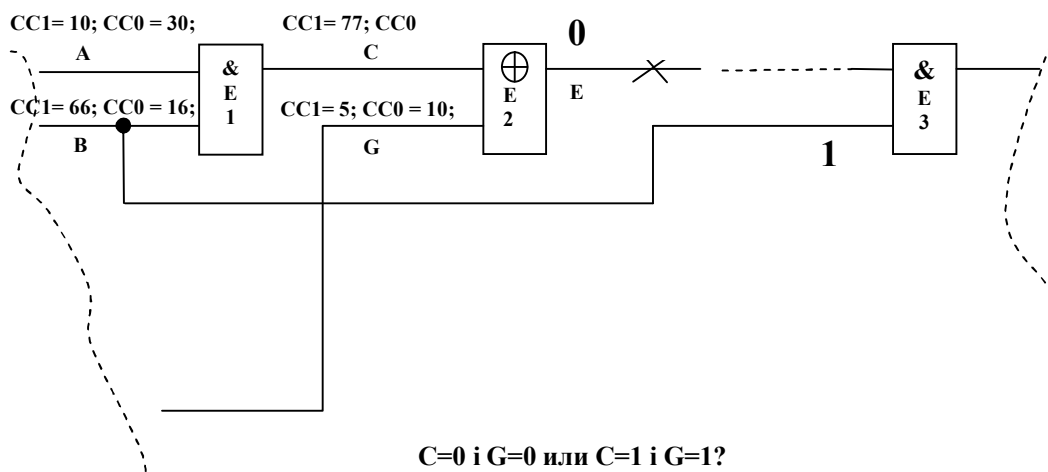


Рис. 1. Выбор необходимых значений сигналов

Однако благодаря процедуре уникальной активизации, значение сигнала на линии «В» установлено равным единице, поэтому для обеспечения $C=0$ понадобится установка $A=0$. Поскольку $CC1(A) < CC0(A)$, то можно сделать вывод, что решение о выборе $C=0$ и $G=0$ – неверное. В данном случае выгоднее выбрать $C=1$ и $G=1$, потому что несмотря на большое значение контролируемости, установка $B=1$ необходимо в любом случае, а $A=1$ обеспечить легче чем $A=0$.

Для исключения неточности описанной в предшествующем пункте необходимо решить две задачи:

- отложить расчет контролируемости до момента, когда будут известны части схемы которые подходят для уникальной активизации.
- выполнить модификации процедуры расчета контролируемости, которая будет исключать из рассмотрения линии, на которых сигналы уникально определены.

Решение первой задачи не требует значительных усилий, поскольку процедура поиска участков схемы для уникальной активизации и процедура расчета контролируемости не зависят одна от другой. Для решения другой задачи необходимо выполнить модификацию процедуры расчета контролируемости таким образом, чтобы исключить из рассмотрения линии, на которых сигнал уникально определен. В качестве базовой процедуры расчета контролируемости используем процедуру SCOAP.

Для примера схемы показанной на рис. 2, в случае применения модифицированной процедуры SCOAP, значение контролируемости единицы на линии C, устанавливается равным 1. Алгоритм FAN выбирает набор $C=1$, $G=1$ и следствием этого выбора становится тестовый вектор $(Y, W, U, V, Z, X) = (1, 1, 0, 1, 1, 1)$ получаемый без выполнения откатов. Если не применять модифицированную процедуру SCOAP, то алгоритм FAN выберет набор $C=0$, $G=0$.

Поскольку E уникально активизирована значением 1, то потребуется установка значения $A=0$. Значения сигнала A не может быть установлено равным 0, что в результате приводит к исчезновению условий активизации сбоя. При этом будет выполнен один откат и будет выбран набор $C=1$, $G=1$.

4. Программная реализация

Разработанная в результате исследования алгоритмов генераторов тестовых векторов программная среда генерации тестов определяет:

- собственные структуры данных для описания цифровых схем;
- собственный язык описания цифровых схем на базе XML, а также синтаксически управляемый транслятор этих описаний в динамический код для сборки соответствующих цифровых схем;
- графический редактор цифровых схем на базе тонкого клиента;
- алгоритмы моделирования работы цифровых схем;
- алгоритмы для поиска и устранения эквивалентных сбоев;
- алгоритмы для нахождения минимального набора тестовых векторов;
- утилита для создания комбинационных схем с определенными параметрами;
- алгоритмы PODEM и FAN;

Собственные структуры данных для описания схем включают в себя:

- простой элемент;
- сложный элемент;
- связь.

Собственный язык описания схем на базе XML дает возможность добавления в систему описаний новых устройств. При синтезе схем вместо интерпретации XML описания выполняется его трансляция в ма-

шинно-независимый язык низкого уровня СЛ. Это позволяет значительно сократить время синтеза схемы и увеличить производительность системы в целом.

Система позволяет выполнять моделирование цифровых схем. Для моделирования работы про-

стых логических элементов используются таблицы или альтернативные графы их функционирования. Для сложных элементов, представленных в виде «черного ящика», моделирование выполняется с помощью специальных процедурных описаний.

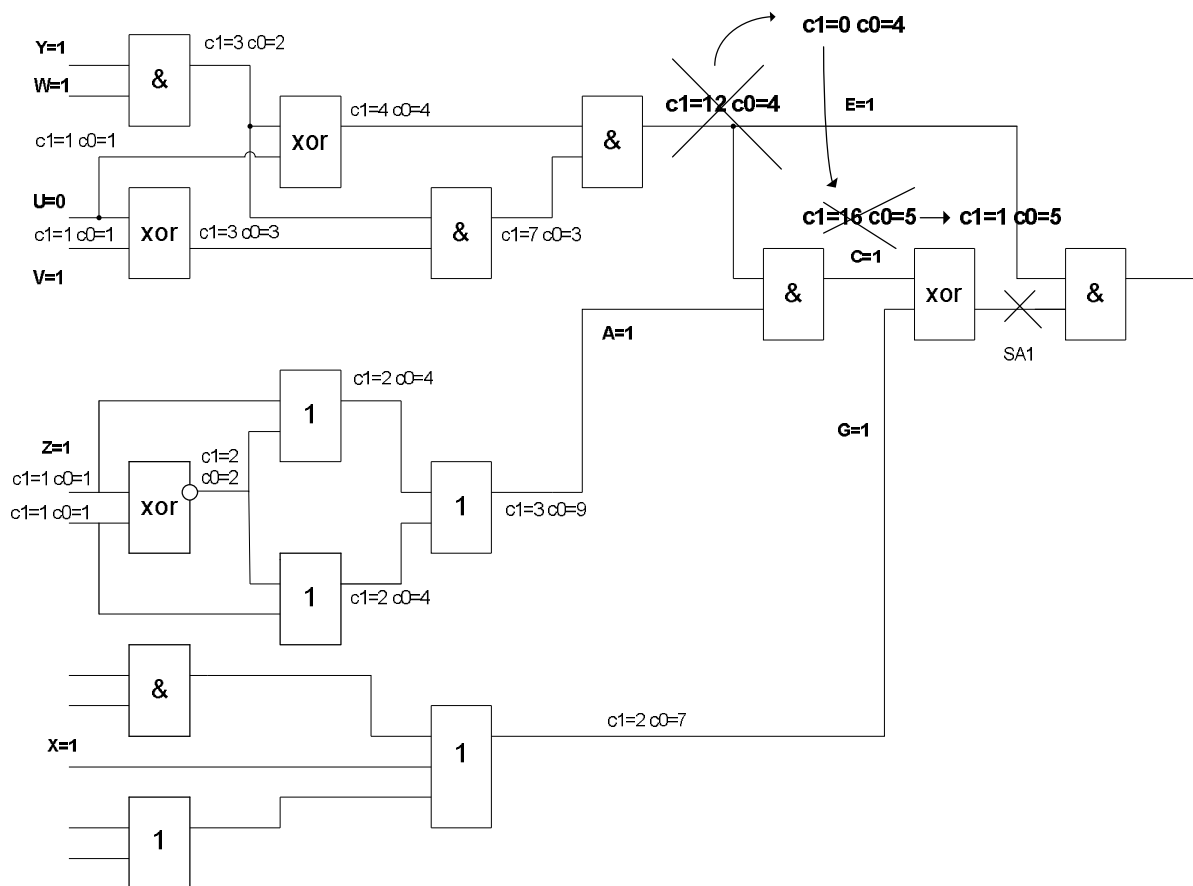


Рис. 2. Пример нахождения тестового набора без откатов

В системе используются алгоритмы поиска минимального набора тестовых векторов, что позволяет устранить поглощаемые тестовые вектора. Система позволяет собирать статистику и контролировать процесс поиска тестов с помощью различных алгоритмов. Статистика представляется в системе в виде следующих графиков зависимости:

- количества откатов от параметров устройства;
- покрытия от разрешенного количества откатов на один сбой;
- времени работы от требуемого покрытия.

Система предоставляет информацию о среднем времени, затраченном на один сбой, а также информацию о приблизительном времени нахождения теста.

Заключение

В результате проведенного исследования выполнен анализ алгоритмов генерации тестовых

векторов, разработана модификация алгоритмов PODEM и FAN с улучшенной эвристикой, использованной для управления обратной трассировкой, реализован программный прототип системы генерации тестов на базе исследуемых алгоритмов (D, PODEM, FAN), а также модифицированного алгоритма FAN. По сравнению с оригинальным алгоритмом FAN, модификация позволяет сократить время поиска тестов, для комбинационных схем с большим количеством разветвлений и уникально активизированных путей.

Предложенный прототип программной системы генерации тестов позволяет выполнять сравнения различных алгоритмов генерации на различных схемах и выполнять сбор статистики за счет незначительного увеличения времени подготовки к поиску следующего теста.

Литература

1. Goel P. An implicit enumeration algorithm to generate tests for combinational logic circuits // *IEEE Transactions on Computers*. – 1981. – Vol. 30, No.3. – P. 215-222.
2. Fujiwara H. On the acceleration of test generation algorithms/ H.Fujiwara, T.Shimono // *IEEE Transactions on Computers*. – 1983. – Vol. C-32, No.12. – P. 1137-1144.
3. Bushnell M.L. Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits / M.L. Bushnell, V.D. Agrawal. – New York, Boston, Dordrecht, London, Moscow: Kluwer Academic Publishers, 2002. – P.156-206.
4. Scheffer L. EDA for IC system design, verification, and testing / L. Scheffer, L. Lavagno, G.Martin. – New York : CRC Press Taylor & Francis Group, 2006. – P. 221-227.
5. Goldstein L.H. SCOAP: Sandia controllability /observability analysis program / L.H. Goldstein, E.L. Thigpen // *Proceedings of the 17th conference on Design automation. June 23-25, 1980, Minneapolis, Minnesota, United States*. – 1980. – P. 190-196.

Поступила в редакцію 10.02.2010

Рецензент: д-р техн. наук, проф. А.В. Дрозд, Одесский национальный технический университет, Украина.

МОДИФІКАЦІЯ СТРУКТУРНОГО ГЕНЕРАТОРА ТЕСТІВ FAN

В.О. Андріянов, О.М. Мартинюк

Робота присвячена дослідженню структурних алгоритмів генерації тестів. Метою дослідження є модифікація існуючого структурного алгоритму генерації тестів за рахунок скорочення числа змін рішень, що виконуються алгоритмом. Для цього виконано уточнення процедури розрахунку контрольованості. Також в роботі виконано аналіз алгоритмів, розроблені модифікація алгоритму FAN з покращеною евристикою, яка використовується для керуванням зворотним трасуванням, і прототип системи синтезу тестів на базі досліджуваних структурних алгоритмів: D, PODEM і FAN.

Ключові слова: константні несправності, активізація шляхів, тест, алгоритм FAN, процедура SCOAP.

MODIFICATION OF STRUCTURAL TEST PATTERN GENERATION ALGORITHM FAN

V.O. Andriyanov, O.M. Martinyuk

This paper is devoted to research of structural test pattern generation algorithms. Our purpose is to accelerate existing one structural test generation algorithm by decreasing amount of backtracks. To do this, we made controllability calculation procedure more accurate. Also in this paper we made analysis of existing algorithms, described the modification of FAN algorithm with improved heuristics and prototype of automated test pattern generation system based on structural test generation algorithms D, PODEM and FAN.

Keywords: stuck-at faults, path sensitization, test, FAN algorithm, SCOAP procedure.

Андріянов Вадим Александрович – ассистент кафедры компьютерных интеллектуальных систем и сетей, Одесский национальный политехнический университет, Одесса, Украина, e-mail: Vadim.Andriyanov@live.com.

Мартинюк Александр Николаевич – канд. техн. наук, доцент, доцент кафедры компьютерных интеллектуальных систем и сетей, Одесский национальный политехнический университет, Одесса, Украина, e-mail: martinyuk@ics.opu.ua.