

УДК 004.3

А.А. Корченко (аспірант),
Ю.Е. Зинченко (канд. техн. наук, доц.)
Донецкий национальный технический университет
zinchenko@donntu.edu.ua

ПАРАЛЛЕЛЬНАЯ ГЕНЕРАЦИЯ АДАПТИВНЫХ ПСЕВДОСЛУЧАЙНЫХ ТЕСТОВ

Рассматривается одна из основных задач диагностики – построение теста цифрового устройства. Для адаптивного подхода построения теста, описанного авторами ранее, предлагается модификация для параллельных систем. Параллельная версия алгоритма позволяет сократить время на построение теста, а также повысить его качество. Приведены результаты экспериментальных исследований для набора схем пакета ISCAS'89.

Ключевые слова: псевдослучайное тестирование, цифровая схема, граф состояний, параллельные вычисления.

Введение

Как известно, процесс построения псевдослучайных тестов (ПСТ) связан с большим количеством вычислений, объем которых растет экспоненциально с увеличением сложности тестируемой схемы. Идея использования параллельных алгоритмов в тестировании для повышения быстродействия известна давно. При этом используются несколько основных подходов распределения вычислительной нагрузки – разделение списка неисправностей [1,3], эвристическое распараллеливание [2], разделение поискового пространства [2,3], алгоритмическое разделение [3] и топологическое разделение [4].

Предложенный авторами адаптивный подход генерации ПСТ [5] позволяет повысить качество теста для схем с памятью за счет использования графа состояний схемы. Однако вопрос быстродействия по-прежнему остается актуальным, в связи с чем предлагается оптимизация адаптивного подхода на основе распределенных вычислений. В данной статье рассматривается проблема эффективности распределенных вычислений применительно к адаптивному подходу построения ПСТ, и в частности к обработке адаптивной составляющей алгоритма.

Целью данной статьи является увеличение качества и скорости построения псевдослучайного теста для последовательностных схем. Для

этого решается задача параллельной генерации адаптивных псевдослучайных тестов.

Постановка задачи

В диагностике традиционно используются несколько способов распараллеливания процесса построения теста [6]:

- разделение списка неисправностей – полный список неисправностей делится на несколько частей по количеству используемых процессоров;
- эвристическое распараллеливание – для одной и той же неисправности каждый процессор строит тест по своему алгоритму;
- разделение поискового пространства – способ организации работы процессоров, при котором они сообща решают задачу построения теста для одной и той же неисправности;
- алгоритмическое разделение – разделение алгоритма на независимые части, которые могут выполняться одновременно;
- топологическое разделение – разделение схемы на части и присвоение каждому процессору своей части.

Перечисленные способы могут комбинироваться в том или ином виде и для разного рода диагностических задач показывать различные результаты.

В то же время, основной вычислительной нагрузкой при построении ПСТ являются генерация псевдослучайной последовательности и моделирование схемы в исправном/неисправном состояниях. Для адаптивного подхода генерации к этим пунктам добавляется еще и обработка графа состояний.

В ходе построения теста алгоритм адаптивной генерации предусматривает следующие операции, связанные с графом состояний:

- 1) построение графа, добавление дуг и вершин;
- 2) поиск оптимального пути из вершины e_i к вершине e_j ;
- 3) идентификация тупиковых ветвей.

Ускорение параллельной работы алгоритма можно оценить по закону Амдала:

$$S_N = \frac{1}{\alpha - \frac{1-\alpha}{N}}$$

где N – количество процессоров, α – доля кода, которая выполняется последовательно.

Таким образом, задача исследования заключается в разработке структуры вычислительной системы и применении известных подходов распараллеливания вычислений к адаптивному алгоритму построения ПСТ для максимизации ускорения алгоритма S_N .

Решение проблемы

Основой адаптивного подхода генерации [5] является граф состояний и переходов схемы $G = (V, E)$, где $V = \{v_1, v_2, \dots, v_p\}$ – множество состояний схемы и $E = \{e_1, e_2, \dots, e_q\}$ – множество переходов между состояниями. Переходы между состояниями схемы происходят под воздействием наборов входных сигналов схемы $X = \{x_1, x_2, \dots, x_n\}$, где $n = 2^{ki}$, а ki – количество входов схемы. При этом формируются наборы выходных сигналов схемы $Y = \{y_1, y_2, \dots, y_m\}$, где $m = 2^{ko}$, а ko – количество выходов схемы. Очевидно, что граф G обладает следующими свойствами:

1) граф является ориентированным, поскольку его ребра, представляющие собой переходы из состояния в состояние, являются ориентированными, т.е. в общем случае $v_i, v_j \neq v_j, v_i$;

2) граф может содержать изолированные вершины $V_z \square V$, которые соответствуют состояниям, недостижимым в ходе функционирования схемы;

3) граф может содержать петли v_i, v_i , которые обозначают отсутствие перехода в другое состояние при данном наборе входных сигналов x_i ;

4) граф может содержать кратные ребра, что соответствует переходам схемы под воздействием различных наборов входных сигналов;

5) граф является взвешенным, его дугам присваивается оценочная функция $f_e = f(F_{cov})$, где F_{cov} – количество обнаруженных неисправностей для данного перехода, которая показывает степень полезности перехода при построении теста.

Существует два основных вида архитектуры параллельных систем – с общей памятью и с отдельной памятью (пакетным обменом). Каждая из подобных систем подходит для решения своего класса задач. Так, системы с общей памятью позволяют использовать единое адресное пространство для всех процессоров, однако при этом возникает проблема синхронизации доступа к данным. В свою очередь, системы с пакетным обменом предполагают наличие собственного адресного пространства для каждого процессора, что позволяет им работать относительно независимо, периодически обмениваясь пакетами между собой.

Учитывая специфику алгоритма адаптивной генерации теста, целесообразно использовать архитектуру параллельной системы с пакетным обменом, которая может быть реализована как для многопроцессорной платформы, так и для локальной компьютерной сети (рис. 1).

В такой структуре каждый процессор работает с одной и той же схемой и выполняет построение теста для полного списка неисправностей, но генерация псевдослучайных последовательностей происходит различным образом. За счет этого схемы на разных процессорах

принимают разные последовательности состояний, а, следовательно, и графы состояний этих схем будут различными.

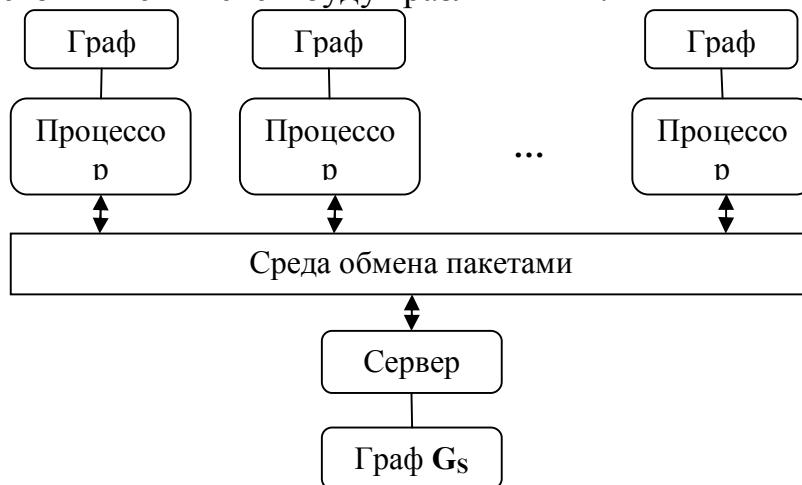


Рисунок 1 – Структура вычислительной системы

На рис. 2 приведен общий алгоритм работы параллельной вычислительной системы при построении теста с помощью адаптивного подхода.

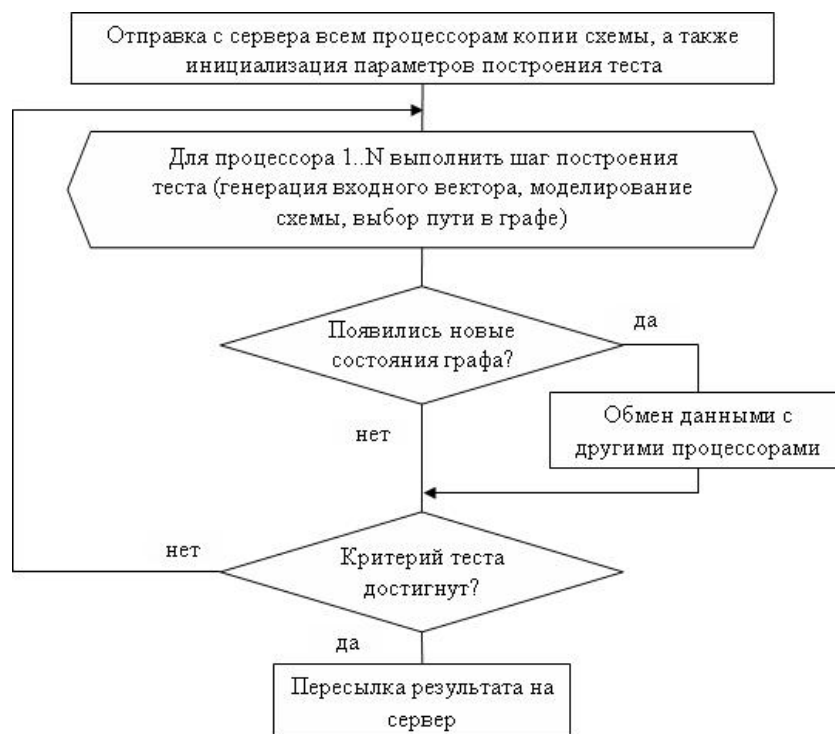


Рисунок 2 – Алгоритм построения теста в параллельной системе

Для обеспечения эффективности использования параллельных ресурсов каждому процессору присваиваются уникальные параметры алгоритма генерации псевдослучайной последовательности, а также параметры алгоритма выбора пути в графе. Благодаря этому распределенные вычисления охватывают значительно большее число вариантов построения теста, что положительно сказывается на качестве результата и скорости его получения.

Параметры алгоритма выбора пути в графе задаются таким образом, чтобы каждый процессор просчитывал только часть пути, разделяя полученные результаты с другими участниками вычислительной системы. Так, на рисунке 3 показан пример части графа состояний, в котором присутствуют два основных пути: $v_0-v_1-v_2-v_3$ и $v_0-v_4-v_5-v_6$. Кроме того, в процессе накопления состояний графа могут появиться дополнительные пути (показаны пунктиром): $v_2-v_4-v_5-v_6$ и v_2-v_6 . Просчет приведенных цепочек состояний может быть проведен независимо друг от друга, после чего результаты вычислений становятся доступны всем узлам вычислительной сети. Наличие альтернативных путей в графе, которые могут быть просчитаны независимо друг от друга, позволяет говорить о том, что алгоритм обладает большой долей параллельного кода. А это, в свою очередь, влияет на увеличение ускорения параллельной работы алгоритма S_N .

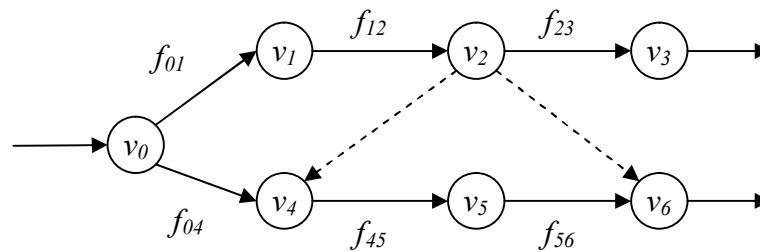


Рисунок 3 – Пример части графа состояний

Построение графа состояний происходит в ходе построения теста. Основным критерием для добавления информации о паре состояний v_i, v_j и переходе e_{ij} между ними является значение оценочной функции f_e . Добавление в граф возможно в следующих случаях:

- 1) если $f_e > 0$;
- 2) если $f_e \geq 0$ и $v_j \notin V$;
- 3) если $f_e \geq 0$ и $\square e_{ij} \notin E$.

Сервер выполняет функцию накопления значений глобального графа состояний, принимая пакеты от процессоров с локальными частичными версиями графа, объединяя части и отсылая обратно процессорам данные о глобальном графе состояний.

Следует отметить, что размер полного графа состояний с увеличением размера схемы увеличивается экспоненциально, поэтому в реальных расчетах используется только часть полного графа $G_S \subseteq G$. Обозначим локальные графы состояний, получаемых на N процессорах как G_1, G_2, \dots, G_N . Тогда процесс построения серверной части графа на шаге t можно выразить как $G_S^t = G_1^{t-1} \cup G_2^{t-1} \cup \dots \cup G_N^{t-1}$. В свою очередь, каждый процессор получает на следующем шаге версию графа, хранящегося на сервере $G_i^t = G_S^t$. Этот процесс продолжается итерационно до завершения построения теста.

Таким образом, параллельные вычисления для адаптивного алгоритма построения ПСТ позволяют повысить скорость и качество теста за счет использования особенностей графа состояний схемы. Степень ускорения S_N при этом может отличаться в зависимости от структуры схемы, что показано при проведении экспериментальных исследований.

Экспериментальные исследования

Предложенная модификация метода псевдослучайной генерации тестов была реализована в составе программного комплекса AGAT, в структуру которого входят система моделирования неисправностей, генератор псевдослучайных последовательностей, модуль управления графом состояний и другие компоненты. Система позволяет загружать модель ЦУ и осуществляет построение теста с заданной полнотой покрытия неисправностей.

Экспериментальные исследования проводились на схемах набора ISCAS'89 [7]. В таблице 1 приведен результат для некоторых схем при построении теста в одно- и двухпроцессорной системах (Intel Core2Duo E6750@2.66ГГц), где F_{cov} – покрытие неисправностей, t – время построения теста.

Таблица 1 – Результаты параллельного построения ПСТ

Схема	ЛЭ/Триггеры	1 процессор		2 процессора			
		$F_{cov}, \%$	t, s	$F_{cov}, \%$	t, s	$F_{cov}, \%$	t, s
s820	289/5	64.09	600	64.09	473	73.15	600
s838	390/32	72.71	600	72.71	600	72.71	600
s953	395/29	80.54	600	80.54	513	82.33	600
s1196	529/18	97.23	600	97.23	468	97.49	600
s1423	957/74	83.96	600	83.96	524	85.29	600
s1494	647/6	67.16	600	67.16	492	71.80	600

Для набора схем проводилось три эксперимента:

- 1) построение теста с помощью одного процессора в течение 10 минут;
- 2) построение теста с помощью двух процессоров до достижения полноты покрытия неисправностей, полученной в предыдущем эксперименте;
- 3) построение теста с помощью двух процессоров в течение 10 минут.

Результаты показывают прирост скорости и качества теста при использовании параллельных вычислений для адаптивного подхода генерации ПСТ. Следует отметить, что эффективность параллельного алгоритма становится наиболее заметной при приближении полноты

покрытия неисправностей к своему максимуму. Это связано с тем, что в большинстве схем легко определяющиеся неисправности покрываются в первую очередь, а построение теста для оставшихся неисправностей происходит с гораздо меньшей скоростью.

Для двухпроцессорной системы прирост производительности составляет 10-15% по сравнению с однопроцессорной системой, что объясняется случайным характером метода псевдослучайного построения теста – просчетом большого числа вариантов, которые не используются в результирующем тесте.

Заключение

В статье предложен алгоритм проведения параллельных вычислений для адаптивного подхода построения тестов ЦУ, который позволяет повысить скорость и качество теста, что подтверждается результатами экспериментальных исследований. Результат достигается за счет разделения вычислительной нагрузки на основе данных графа состояний ЦУ.

Практическая значимость алгоритма заключается в сокращении вычислительных затрат на генерацию теста на 10-15%, а также в увеличении качества теста на 5-7%.

Дальнейшие исследования планируется проводить вокруг проблемы сочетания детерминированного и псевдослучайного подходов построения тестов Banerjee P. Fault Partitioning Issues in an Integrated Parallel Test Generation ЦУ.

Список литературы

1. Patil S. Fault Simulation Environment / S. Patil // Proc. 1989 Int'l Test Conf., CS Press, Los Alamitos, Calif. – 1989. – P. 718-726.
2. Klenke R.H. Parallel-processing techniques for automatic test pattern generation / R.H. Klenke, R.D. Williams, J.H. Aylor // IEEE Computer. – 1992. – V. 25, Issue 1, Jan. – P. 71-84.
3. A Parallel Scheme for Test-Pattern Generation / Motohara A., Nishimura K., Fujiwara H., Shirakawa I. // IEEE International Conference on Computer-Aided Design. – 1986. – P. 156-159.
4. Klenke R.H. Parallelization Methods for Circuit Partitioning Based Parallel Automatic Test Pattern Generation / R.H. Klenke, R.D. Williams, J.H. Aylor // Proceedings of the IEEE VLSI Test Symposium. – 1993. – April. – P. 71-78.
5. Зинченко Ю.Е. Адаптивный подход к генерации псевдослучайных тестов цифровых устройств / Ю.Е. Зинченко, А.А. Корченко // Наукові праці Донецького національного технічного університету. Серія «Проблеми моделювання та автоматизації проектування» (МАП-2011). – 2011. – Вип. 9 (179). – С. 360-365.
6. Bushnell M.L. Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits / M.L. Bushnell, V.D. Agrawal. – Kluwer academic publishers, 2001. – 690 p.

7. Brgles F. Combinational profiles of sequential benchmark circuits / F. Brgles, D. Bryan, K. Kozminski // International symposium of circuits and systems, ISCAS-89. – 1989. – P. 1929-1934.

Надійшла до редакції 03.11.2011.

Рецензент: канд. техн. наук, доц. Дяченко О.Н.

О.О. Корченко, Ю.Є. Зінченко

Донецький національний технічний університет

Паралельне генерування адаптивних псевдовипадкових тестів. Розглядається одна з основних задач діагностики – побудова тесту цифрового пристрою. Для адаптивного підходу побудови тесту, описаного авторами раніше, пропонується модифікація для паралельних систем. Паралельна версія алгоритму дозволяє скоротити час на побудову тесту, а також підвищити його якість. Наведено результати експериментальних досліджень для набору схем пакета ISCAS'89.

Ключові слова: псевдовипадкове тестування, цифрова схема, граф станів, паралельні обчислення.

A.A. Korchenko, Y.E. Zinchenko

Donetsk National Technical University

Parallel Generation of Adaptive Pseudorandom Test. We considered one of the main diagnostic problems – building digital devices test. We proposed a modification for parallel systems of adaptive approach to construct a test that had been described by the authors previously. A parallel version of the algorithm reduces the time to build a test and improve its quality. The experimental results for a set package of schemes ISCAS'89 are included.

Keywords: pseudo-random testing, digital circuit, the graph of states, parallel computing.