

Что касается других статистик алгоритмов, представленных в табл. 1, 2, то для обоих правил выбора операций стандартные отклонения S_{Δ} с ростом размера задачи N и отношения $\frac{n}{m}$ имеют тенденцию к уменьшению, что свидетельствует о более интенсивной концентрации разброса случайных погрешностей Δ вокруг среднего значения и, таким образом, более «кучному» решению задач. С ростом отношения $\frac{n}{m}$ для обоих правил алгоритма наблюдается также интенсивное снижение максимального выброса погрешностей Δ_{\max} , что свидетельствует о более «точном» решении задач с большим отношением $\frac{n}{m}$.

Таким образом, экспериментально показано, что предложенный новый алгоритм решения задачи теории расписаний с правилом выбора операций на основании прогноза текущей длины расписания по величине погрешности решения превосходит алгоритм с лучшим правилом выбора операций NOPNR. На этом основании его можно рекомендовать к практическому использованию в разработках программного обеспечения систем компьютерного составления производственных расписаний.

Список литературы: 1. Костикова М.В., Пьянида В.А. Алгоритмы решения задач теории расписаний на основе прогноза. Часть 1 // АСУ и приборы автоматики. 2007. 2. Конвей Р.В., Максвелл В.П., Миллер Л.В. Теория расписаний. М.: Наука, 1975. 360 с. 3. Канцедал С.А. Вычислительные алгоритмы решения задач теории расписаний // Изв. АН СССР. Техническая кибернетика. 1982. – №3. С. 42–51. 4. Канцедал С.А. Эффективные алгоритмы упорядочения работ в многостадийных производственных системах дискретного типа. Автореф. дисс. д-ра техн. наук. Харьков: Институт проблем машиностроения АН УССР. 1991. 32 с.

Поступила в редколлегию 07.09.2007

Костикова Марина Владимировна, канд. техн. наук, доцент кафедры информатики Харьковского национального автомобильно-дорожного университета. Научные интересы: математическое моделирование, теория расписаний и ее применение. Адрес: Украина, 61002, Харьков, ул. Петровского, 25, тел. 707-37-74.

Пьянида Виктор Александрович, преподаватель кафедры прикладной математики и информатики Запорожского государственного института экономики и управления. Научные интересы: теория расписаний в моделировании производственных и управленческих процессов. Адрес: Украина, 51400, Павлоград, ул. Парковая, 1а, тел. 311-95.

УДК 519.713:681.326

В.И. ХАХАНОВ, М.А.КАМИНСКАЯ, С.А.ЗАЙЧЕНКО

ВЕРИФИКАЦИЯ ЦИФРОВЫХ УСТРОЙСТВ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ АНАЛИЗА ТЕСТОПРИГОДНОСТИ И АССЕРЦИОННЫХ БИБЛИОТЕК

Предлагается метод анализа тестопригодности для операционных и управляющих автоматов, представленных на системном уровне, на основе использования ассерционных библиотек. Описывается методология выбора контрольных точек в устройстве для дальнейшего внедрения ассерций в описание устройства.

1. Введение

В связи с интенсивным развитием цифровых устройств и ростом их сложности верификация цифровых устройств становится все более актуальной темой. Актуальность определяется необходимостью повышения быстродействия средств моделирования, улучшения

качества теста и уменьшения его размерности для цифровых систем на кристаллах, имеющих миллионы вентиляей. Высокие затраты, обусловленные трудоемкостью верификации функционально и структурно сложных схем, могут достигать 70% от общего времени разработки проекта.

До настоящего времени в связи с надежностью используемых программ синтеза и методик проектирования, самым низким уровнем детализации цифровых устройств для обработки был уровень регистровых передач [1]. Основными элементами для обработки на этом уровне являются регистры и соединения между ними. Проектирование на уровне регистровых передач подобно программированию, например на С. Однако есть небольшие отличия. Например, такие моменты как синхронизация, timing, concurrency, должны быть рассмотрены на этапе программирования. Одним из решений проблемы верификации является использование ассерций [2] – OpenVera Assertions (OVA), PSL/Sugar, Open Verification Libraries (OVL) и C/C++/SystemC™ конструкции как в программном моделировании, так и аппаратно в виде синтезируемых конструкций. Языком описания устройства могут служить такие языки как VHDL, System Verilog, С.

Существует ряд стандартов использования ассерций, таких как стандарты AVM [1], OVM [3], VVM.

Ассерции моделируются отдельно от остального HDL кода. Результат работы ассерций можно увидеть в программе просмотра функционального покрытия (Functional Coverage viewer).

Внедрение ассерций для сложных цифровых устройств – трудоемкий и времязатратный процесс. В настоящее время нет определенного метода выбора критических мест в устройстве для введения ассерций – проектировщик интуитивно внедряет ассерции в программный код. Именно поэтому был разработан метод анализа цифровых устройств на системном уровне описания с дальнейшей выборкой проблемных мест в коде для внедрения ассерций.

Цель работы: существенное уменьшение времени верификации, синтеза тестов и/или повышение степени покрытия неисправностей для заданных входных наборов путем модификации структуры цифрового автомата на основе анализа его тестопригодности.

Объект исследования – цифровой автомат, представленный в виде ориентированного графа.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Задание модели цифрового автомата.
2. Определение модели неисправностей.
3. Оценка тестопригодности графа автомата.
4. Разработка алгоритма модификации графа автомата.
5. Верификация и тестирование метода анализа тестопригодности на реальных цифровых автоматах.

2. Структура операционного устройства

По принципу академика В.М. Глушкова любое устройство обработки цифровой информации можно разделить на операционный и управляющий блоки. Такой подход упрощает проектирование, а также облегчает понимание процесса функционирования вычислительного устройства.

Операционный автомат может быть представлен системным уровнем (досинтезное представление устройства, его внутренняя структурная модель), уровнем регистровых передач, а также вентильным уровнем. Для проведения анализа тестопригодности операционный автомат может быть представлен совокупностью регистров, сумматоров и других узлов, производящих прием из внешней среды и хранение кодов, их преобразование и выдачу результатов работы во внешнюю среду, а также выдачу в управляющий блок и внешнюю среду оповещающих сигналов $U = \{u_1, u_2, \dots, u_n\}$, входных-выходных сигналов, модулей памяти, набором счетчиков, триггеров, блоков АЛУ и взаимосвязями между этими компонентами.



Рис.1. Операционный и управляющий блоки цифрового устройства

Управляющий автомат вырабатывает распределенную во времени последовательность управляющих сигналов $V = \{v_1, v_2, \dots, v_m\}$ порождающих в операционном блоке нужную последовательность микроопераций.

Последовательность управляющих сигналов определяется сигналами Z кода операции, поступающими в управляющий блок извне, и сигналами V , зависящими от операндов и промежуточных результатов преобразований.

Декомпозиция цифрового вычислительного устройства представлена на рис. 1.

Пример содержательного графа переходов, синтезируемого в среде Active-HDL [4] автомата со всеми типами операционных вершин, представлен на рис. 2.

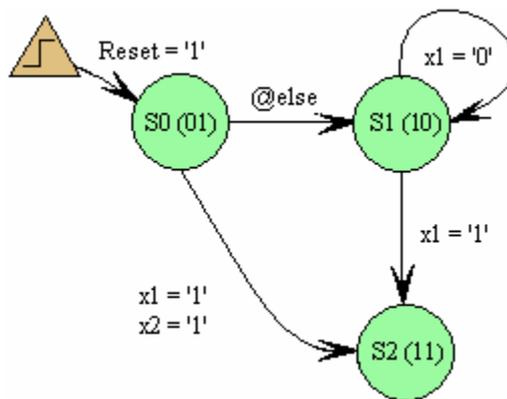


Рис. 2. Граф переходов автомата в среде Active-HDL

В табл. 1 задано кубическое покрытие данного графа в двухтактном алфавите A^2 .

Для представления двухтактного алфавита используются следующие обозначения: $A^2 = \{Q = 00, E = 01, H = 10, J = 11, O = \{Q, H\}, I = \{E, J\}, A = \{Q, E\}, B = \{H, J\}, S = \{Q, J\}, P = \{E, H\}, C = \{E, H, J\}, F = \{Q, H, J\}, L = \{Q, E, J\}, V = \{Q, E, H\}, Y = \{Q, E, H, J\}, U\}$ – двухтактный алфавит описания состояний/переходов автоматных переменных [4]. Символ U – пустое множество \emptyset .

3. Определение модели неисправностей

Граф переходов конечного управляющего автомата представляет собой поведенческую модель цифрового устройства. Данное обстоятельство предполагает введение адекватных моделей неисправностей, соответствующих данному уровню. Не затрагивая физическую сущность вводимых далее дефектов, рассмотрим основные неисправности для каждого уровня описания объекта.

Определение 1. Модель неисправности вентиля F^G, функционального F^F, алгоритмического или поведенческого F^B уровней описания цифрового автомата называется имплицативной,

Таблица 1

| Вход | Переход | Выход |
|--------------------|---------|-------|
| Reset = '1' | 00 – 01 | QE |
| Reset = '0' | 01 – 10 | EH |
| x1 = '0' | 10 – 10 | JQ |
| x1 = '1' | 10 – 11 | JE |
| x1 = '1', x2 = '1' | 01 – 11 | EJ |

если она определяется посредством элементарных термов, формирующих его функцию с выполнением условий:

$$M^G \Rightarrow F^G = (L_j \in L, A = \{0,1\}); M^F \Rightarrow F^F = (C_{ij} \in C, A^2);$$

$$M^B \Rightarrow F^B = (T_{ij} \in T, V = \{V_1, V_2, \dots, V_j, \dots, V_n\});$$

где $M = \{M^G, M^F, M^B\}$ – модели вентильного, функционального, алгоритмического уровней описания цифрового автомата; $L_j \in L$ – переменная или линия, принимающая двоичное значение из алфавита существенных символов $A = \{0,1\}$; $C_{ij} \in C$ – координата кубического покрытия, определенная в существенных символах двухтактного алфавита A^2 ; $T_{ij} \in T$ – множество переходов автомата, определенное на векторе вершин $V = \{V_1, V_2, \dots, V_j, \dots, V_n\}$ графа.

Определение 2. Графу переходов цифрового автомата соответствует обобщенная модель неисправностей $F = (F^G, F^F, F^B)$, где F^G – модель неисправности вентильного, F^F – функционального, F^B – алгоритмического или поведенческого уровней описания цифрового автомата. Это связано с тем, что помимо поведенческого уровня представления самого графа, в модели автомата существуют предикатные функции возбуждения, которые могут быть описаны на вентильном и/или функциональном уровнях.

Определение 3. Импликативная неисправность графа автомата есть переход $T_{ir} \in T$, который выполняется вместо $T_{ij} \in T$ при условии, что в графе существуют переходы $\{T_{ij}, T_{ir}\} \in T$.

Импликативная модель дефектов $F = (F^G, F^F, F^B)$ не может увеличить пространство состояний конечного автомата S , и находится с последним в отношении $S^F \subseteq S$ [4].

Таким образом, представленная модель дефектов графа автомата может быть использована для проверки неправильных переходов в цифровом изделии.

Одной из распространенных форм исходного описания специализированного цифрового вычислительного устройства является граф переходов автомата. Достоинства упомянутой формы заключаются в наглядности и технологичности представления функций, которые просто преобразуются в табличную форму внутреннего описания автомата в виде таблиц или кубических покрытий. Поэтому такой способ представления проекта имеется во всех системах проектирования ведущих фирм мира. Однако под такую реализацию цифрового устройства практически отсутствуют средства синтеза тестов для верификации и диагностирования проекта. Аналогами тест-генераторов для граф-схем могут служить системы: State-CAD фирмы Visual Software Solutions, Escalade фирмы Mentor Graphics, TetraMax фирмы Synopsys. Поэтому проблема генерации проверяющих тестов для цифровых автоматов, описанных в виде графов переходов и реализуемых на кристаллах ПЛИС, является весьма актуальной для EDA (Electronic Design Automation) рынка.

Другое решение связано с использованием методов анализа тестопригодности и дальнейшей модификацией управляющего либо операционного блока, представленного в виде графа автомата.

Метод анализа тестопригодности основан на топологическом анализе ориентированного графа и его модификации путем разделения тестового и функционального режимов работы в целях улучшения тестопригодности и упрощения решения задач тестирования [5].

4. Метод TGA – Testability Graph Analysis

Анализ тестопригодности разрабатываемой модели системы необходимо проводить на всех стадиях проектирования. Методы анализа тестопригодности разрабатывались для различных уровней описания цифрового изделия, в зависимости от тенденций развития

средств проектирования. Первые публикации в этой области принадлежат Рутману [6], Стефенсону [7] и Грасону [8]. Наиболее известные методы анализа тестопригодности на вентильном уровне – это CAMELOT, метод Питерсона, SCOAP, TADATPG [9]. Другие исследования ученых в этой области рассмотрены в работах [10-15].

При этом самый адекватный анализ соответствует наиболее точной модели, которая определяется вентильным уровнем описания, поскольку структура устройства здесь представлена максимально детализированно. Тем не менее, анализ тестопригодности на более высоких уровнях описания, где модель проекта отражает лишь структуру взаимосвязанных компонентов, имеет место, поскольку здесь трудоемкость процедуры анализа минимальна, но оценки тестопригодности и последующая модификация проекта на основе технологий граничного сканирования могут существенно повлиять на стоимость диагностического обеспечения и обслуживания (временные и материальные затраты на синтез тестов, моделирование неисправностей и диагностирование дефектов для каждой стадии проектирования).

На системном уровне устройство представлено в виде совокупности взаимосвязанных компонентов (взаимосвязанных функциональностей устройства – в этом случае для каждого функционального блока устанавливается ассерция и системой диагностирования определяется, в каком именно функциональном блоке существует неисправность. Далее к рассмотрению и дальнейшему тестированию подлежит тот функциональный блок, в котором произошел сбой) или операционного и управляющего автоматов (граф-схема алгоритма). Далее рассматриваются модели проекта системного уровня, представленные в виде граф-схем.

Здесь основная сложность их тестирования заключается в том, что в FSM должны быть проверены все труднодостижимые состояния; тупиковые ситуации или коллизии двух потоков данных; места локализации ветвлений и обратных связей в коде (if, case, loop). Относительно управляющих автоматов можно выделить следующее правило: Каждый FSM должен содержать ассерции, которые проверяют кодирование состояний и переходов [2]. Примеры использования ассерций для верификации FSM:

assert_bits, assert_next_state, assert_value, assert_cycle_sequence, assert_code_distance, assert_next, assert_transition, assert_one_hot, assert_one_cold, assert_zero_one_hot.

Предлагаемый в работе метод заключается в вычислении значений достижимостей вершин, формирующих оценку тестопригодности. Проводится процедура анализа на графе, который строится по размеченной ГСА автомата Мили, Мура. Автомат Мура выбран по причине возможности наблюдения большего количества внутренних состояний, чем в ГСА автомата Мили.

Достижимость (accessibility) – равно управляемость начальной вершины $A(a_1) = 1$ или 100%. Значение управляемости каждой последующей вершины зависит от достижимости предыдущей вершины и коэффициента достижимости. Достижимость может принимать относительное значение, лежащее в интервале $[0;1]$. Значение $A(a_i) = 0$ имеет вершина, которая не достижима ни по какому пути в графе. Практически значения достижимости большинства вершин находятся в границах интервала $[0;1]$:

$$A(a_j) = \sum_{i=1}^m A(a_{i-1}) \times \frac{1}{n} \times w_{ij},$$

где n – количество возможных наборов, с помощью которых можно попасть в вершину a_i ,

w_{ij} – вес дуги; $w_{ij} = \frac{1}{k_i}$; k_i – число исходящих дуг из вершины, m – количество управляющих вершин (входящих дуг).

На рис. 3 представлена отмеченная граф-схема алгоритма автомата Мура.

Достижимость всех вершин ($a_i, i = \overline{1,8}$) данного графа представлена следующими оценками:

$A(a_1) = 1$; $A(a_2) = 1$;
 $A(a_3) = 0,2863533467$; $A(a_4) = 0,2863533467$;
 $A(a_5) = 0,035794183$; $A(a_6) = 0,008948545$;
 $A(a_7) = 0,0363545$; $A(a_8) = 0,045301$;

Тестопригодность схемы вычисляется по формуле:

$$A_{total} = \frac{1}{m} \sum_{i=1}^m A(a_i),$$

где m – количество вершин.

Исходная тестопригодность схемы (до модификации) равна $A_{total} = 0,333455$.

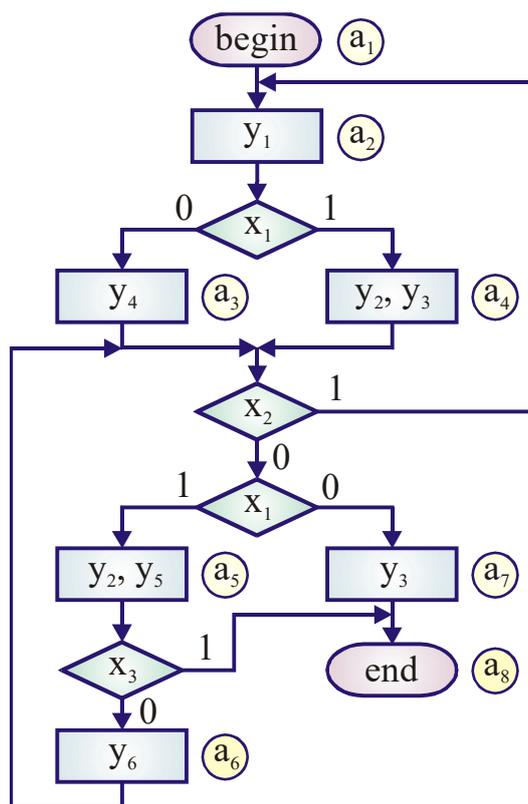


Рис. 3. Отмеченная ГСА автомата Мура

5. Стратегия выбора контрольных точек и введения ассерций в код устройства

Стратегия выбора точек для модификации графа состоит в следующем: выбираются 3% линий с минимальными значениями достижимости A . При этом к выбранным линиям добавляются другие, имеющие оценки, равные максимальному из 3% выбранных, если таковые имеются. Обычно линий с одинаковыми значениями показателей сравнительно мало – это особенность метода анализа тестопригодности. 3%-ная квота линий была выбрана из накладываемых ограничений на количество внешних дополнительных контактов в устройстве – не более 5%.

Далее с полученным множеством пересекается множество контрольных точек в коде, выбранных на основе общепринятых правил установки ассерций в коде. В [2] представлен набор правил, предположений и рекомендаций по использованию ассерций. Всего таких правил порядка сорока. Например, для устройств АЛУ ассерции должны проверять правильность выполнения операций, переполнение разрядной сетки и др. Каждый модуль памяти, регистры FIFO, LIFO должны быть проверены ассерциями.

Формула, по которой в коде устройства могут быть выбраны контрольные точки для внедрения ассерций, выглядит следующим образом:

$$Z = \{Y_{TY}\} \cap \{A_{Rules}\} / \{Z_{TY}\},$$

где $\{Y_{TY}\}$ – множество точек, выбранных по методу анализа тестопригодности; $\{A_{Rules}\}$ – множество контрольных точек в устройстве, выбранных на основе правил использования ассерций в коде устройства; $\{Z_{TY}\}$ – множество точек в устройстве, которые были выбраны по методу анализа тестопригодности, но не могут быть использованы в качестве ассерций.

Модификация структуры графа. При выполнении процедуры обхода всех вершин графа и выбора минимального пути для построения теста возникает проблема достижимости состояний. Из-за наличия дополнительных внутренних переменных в функции возбуждения полученный путь может не существовать для содержательного графа. В этом случае необходима модификация пути. Предложенная стратегия модификации состоит в разделении режимов тестирования и нормального функционирования схемы. Для этого на каждую выбранную по стратегии вершину в графе ставится условная вершина, которая предполагает 100% управляемость выбранной вершины графа. Ожидается, что такой подход к модификации устройства позволит при небольших аппаратных затратах существенно повысить тестопригодность разрабатываемого устройства.

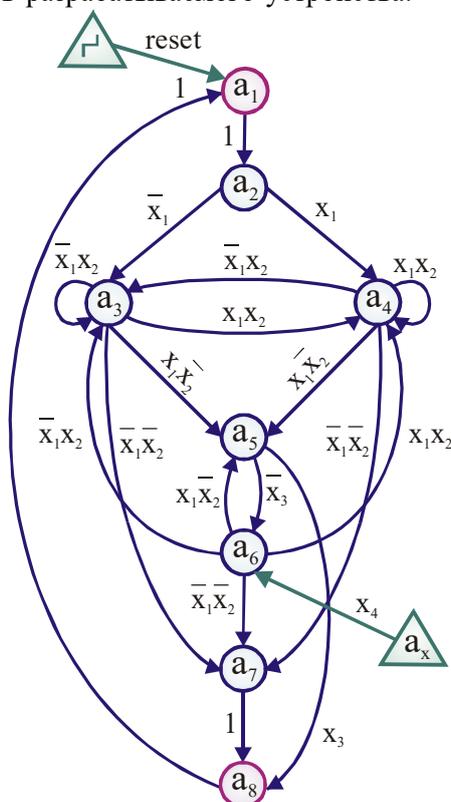


Рис. 4. Модифицированный граф автомата Мура

На примере (рис. 4) видно, что наихудший показатель достижимости имеет вершина a_6 . Поэтому в структуру графа, представленного на рис. 4, встраивается вершина a_x , повышающая управляемость узла a_6 .

При пересчете показателей достижимости при наличии управляющей вершины получаются значения:

$$\begin{aligned} A(a_1) &= 1; & A(a_2) &= 1; \\ A(a_3) &= 0,2863533467; & A(a_4) &= 0,2863533467; \\ A(a_5) &= 0,035794183; & A(a_6) &= 0,05894; \\ A(a_7) &= 0,039462; & A(a_8) &= 0,0484105; \\ A(a_x) &= 1. \end{aligned}$$

Общая тестопригодность схемы после модификации равна $A_{total} = 0,41725$.

Аналогичный анализ может быть проведен как на системном уровне, так и на уровне регистровых передач и на вентильном уровне.

6. Экспериментальные исследования.

Вычисление показателей достижимости и применение стратегии модификации автомата производилось на примерах FSM, представленных в табл. 2. В качестве управляемой вершины выбиралась вершина с наименьшим показателем достижимости.

Таблица 2. Результаты анализа тестопригодности

| FSM | A_{BM} | A_{AM} | Differ | Increase |
|------------|----------|----------|--------|----------|
| CA_M | 0,23306 | 0,3133 | 0,08 | 23,9% |
| Multiplier | 0,44271 | 0,5508 | 0,11 | 25% |
| CSK_M | 0,38360 | 0,5232 | 0,14 | 36,8% |
| MPA_Ma | 0,24702 | 0,4001 | 0,16 | 64,7% |
| Traf_light | 0,33346 | 0,4173 | 0,08 | 23,9% |

Здесь A_{BM} и A_{AM} – показатели достижимости до и после модификации графа.

Введение одной дополнительной вершины в структуру графа позволило повысить качество покрытия неисправностей для вершин графа в среднем на 15–30%, что иллюстрируется данными, представленными в табл.3. Здесь от-

сутствует оценка качества теста для примера Multiplier, поскольку покрытие неисправностей до модификации уже было равно 100%.

Таблица 3. Результаты анализа покрытия дефектов

| FSM | $Q_{BM}, \%$ | $Q_{AM}, \%$ | Increase, % |
|------------|--------------|--------------|-------------|
| CA_M | 50,00 | 62,5 | 12,5 |
| CSK_M | 71,43 | 85,71 | 14,28 |
| MPA_M | 37,5 | 75,0 | 37,5 |
| Traf_light | 62,5 | 87,5 | 25 |

7. Заключение

Научная новизна: разработан метод расчета показателей тестопригодности, а также стратегии выбора точек для модификации устройства и способ модификации содержательного графа FSM в целях увеличения его тестопригодности и повышения качества покрытия неисправностей тестом.

Практическая значимость и преимущества:

1) Разработанный метод анализа тестопригодности позволяет при небольших аппаратных затратах (до 20%) повысить качество покрытия (на 10-30%) дефектов уже существующего теста.

2) Уменьшает времени синтеза тестов путем введения дополнительных переменных при разделении режимов тестирования и функционирования.

3) Метод позволяет проводить анализ устройства на самых ранних стадиях проектирования и повышать Yield Ratio – выход годных изделий.

Недостатком предложенного метода можно считать то, что анализ на вентильном уровне дает более точные показатели тестопригодности для дальнейшей модификации схемы, чем на более высоких уровнях описания устройства. Также в схему вводится избыточность в виде дополнительных вершин и дуг, что приводит к появлению дополнительных переменных.

Список литературы: 1. *Advanced Verification Methodology Cookbook*, Version 2.0, Mark Glasser, Adam Rose, Tom Fitzpatrick, Dave Rich, Harry Foster, July 24, 2006. 2. *Verification Methodology Manual for System Verilog*, Janick Bergeron, Eduard Cerny, Alan Hunter, Andrew Nightingale, 2006 Synopsys, Inc. and ARM Limited. 528 p. 3. *OVM Class Reference*, Version 1.0.1, February 2008. 286 p. 4. *Хаханов В.И., Ковалев Е.В., Ханько В.В., Масуд М.Д. Мехеду. Система генерации тестов для проектирования цифровых автоматов в среде Active-HDL // АСУ и приборы автоматики. Харьков. 2000. Вып. 111. С. 15-22.* 5. *Кривуля Г.Ф., Хаханов В.И., Ковалев Е.В. Проектирование тестов для цифровых устройств на основе FPGA, CPLD // Информационно-управляющие системы на железно-дорожном транспорте. 2000. № 4. С. 120-121.* 6. *Rutman R. A. Fault Detection Test Generation for Sequential Logic Heuristic Tree Search.* IEEE Computer Repository Paper No. R-72-187. 1972. 7. Grason J., "TMEAS - A Testability Measurement Program," Proc. 16th Design Automation Conf. P. 156-161, June, 1979. 8. J. Grason and A. W. Nagel, "Digital Test Generation and Design for Testability." Journal Digital Systems, Vol.5, No. 4. P. 319-359, 1981. 9. *Кулак Э.Н., Каминская М.А. Эвристический метод анализа тестопригодности для тестирования цифровых схем детерминированным тестом // Радиоэлектроника и информатика. 2005. № 3. С. 113-119.* 10. *Parker K.P., McCluskey E.J. Probabilistic Treatment of General Combinational Networks // IEEE Trans. on Computers. vol. C-24. no. 6. 1975. P. 668-670.* 11. Grason J. TMEAS - A Testability Measurement Program // Proc. 16th

Design Automation Conf. 1979. P. 156-161. **12.** Larsson E., Peng Z. A Behavioral-Level Testability Enhancement Technique // IEEE European Test Workshop. Constance.– Germany. 1999. **13.** Larsson E., Peng Z. Testability Analysis of Behavioral-Level VHDL Specifications // IEEE European Test Workshop.– 1998. **14.** Flottes M. L., Pires R., Rouzeyre B. Analyzing Testability from Behavioral to RT Level // Proc. European Design&Test Conf. 1997. P.158-165. **15.** Zdenek Kotasek, Richard Ruzicka, Josef Strnadel, Jan Hlavicka. Interactive Tool for Behavioral Level Testability Analysis // Proceedings of the IEEE ETW2001.- Stockholm.- SE. 2001. P. 117-119.

Поступила в редколлегию 20.09.2007

Хаханов Владимир Иванович, декан факультета КИУ, доктор технических наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование цифровых систем. Увлечения: футбол, горные лыжи, путешествия. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua

Каминская Марина Александровна, аспирантка кафедры АПВТ ХНУРЭ, инженер по тестированию компании ИП «Интспей-Украина». Научные интересы: тестопригодное проектирование, моделирование и верификация цифровых систем на кристаллах. Увлечения: литература, музыка. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 7021326. E-mail: maryna4329@kture.kharkov.ua.

Зайченко Сергей Александрович, аспирант кафедры АПВТ ХНУРЭ, начальник отдела разработки компании Aldec-Kharkov Ltd. Научные интересы: системы автоматизированного проектирования, моделирования и верификации цифровых систем на кристаллах. Увлечения: литература, музыка, футбол. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (097)-367-62-93. E-mail: Sergei.Zaychenko@aldec.com.

УДК 004.652

В.В. ЕВСЕЕВ, Ю.В. ШОВКОПЛЯС, Н.В. САМОЙЛЕНКО

МОДЕЛИРОВАНИЕ РБД С ПРОИЗВОЛЬНОЙ ТОПОЛОГИЕЙ

Рассматривается задача моделирования распределенных баз данных (РБД) с произвольной топологией, а также его аналитические и имитационные методы. Описываются результаты, которые подтверждают эффективность использования данных методов в зависимости от различных исходных данных. Даются рекомендации по дальнейшему использованию предложенных методов моделирования.

Введение

В настоящее время в связи с использованием РБД в различных технических, экономических и социальных системах, имеющих разные топологические структуры, топология самих РБД может также существенно варьироваться. Можно выделить следующие основные топологические структуры РБД: полносвязную, ячеистую, шинную, структуру «звезда» и кольцевую [1]. Таким образом, актуальной является задача формализации процесса моделирования распределенных баз данных независимо от их топологической структуры в реальных условиях функционирования, т.е. в условиях, предполагающих отказы оборудования, линий связи, случайный характер возникновения запросов пользователей.

1. Постановка задачи

Для различных топологических структур РБД принято разрабатывать различные математические модели, отличающиеся как необходимой исходной информацией об РБД, так и методами моделирования [2]. В данном исследовании предполагается разработка моделей РБД с произвольной топологией. Это обусловлено тем, что такой подход позволяет создать достаточно универсальные модели, описывающие широкий класс РБД.

Модели функционирования РБД предназначены для количественного описания зависимостей между параметрами, определяющими объемно-временные характеристики потоков входной информации, основные качественные характеристики РБД [3], эффективность преобразования входных потоков информации в выходные. В настоящее время применяются в основном имитационные и аналитические методы моделирования РБД [4].