

УДК 004.312

*А.В. ЛАПУТЕНКО¹, Х.Е. ЛОПЕЗ², Н.В. ЕВТУШЕНКО¹***ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ ПРИ ВЕРИФИКАЦИИ КОМПОНЕНТОВ ФИЗИЧЕСКИХ СИСТЕМ: ОЦЕНКА КАЧЕСТВА ТЕСТОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ***

Данная статья является продолжением исследований по построению качественных верификационных тестов для проверки цифровых компонентов, используемых в различных физических системах, таких, как, например, встраиваемые датчики и элементы, используемые для передачи данных. Для экспериментальных исследований выбраны схемы из пакета бенчмарков B01-B10 (ITC'99 benchmarks (Second Release)), которые представляют собой компоненты физических систем, спроектированных для различных приложений. Тестовые последовательности строятся, в первую очередь, для обнаружения наиболее широко известных мутаций эталонного описания с использованием трех различных подходов к синтезу тестов. В качестве мутаций рассматриваются одиночные константные неисправности, неисправности «перемычек» и неисправности, незначительно изменяющие поведение одного логического элемента цифровой схемы. Поскольку наибольший интерес представляют короткие тестовые последовательности, обеспечивающие определенные гарантии при положительном результате тестирования, экспериментально исследуются различные подходы к построению тестовых последовательностей, обнаруживающих заданное множество мутаций. В первой серии экспериментов сравниваются два подхода к построению так называемых полных тестов. В первом случае для каждой мутации находится обнаруживающая ее кратчайшая тестовая последовательность, во втором случае – тестовая последовательность генерируется псевдослучайным образом, т.е., вообще говоря, является более длинной. Тем не менее после минимизации теста, т.е. удаления последовательностей, обнаруживающих одно и то же множество мутаций, длина теста, построенного вторым способом, может быть существенно уменьшена, что в ряде случаев опровергает необходимость направленного синтеза кратчайшей последовательности, обнаруживающей одну мутацию. Однако проведенные эксперименты также показывают, что построение тестовых последовательностей исключительно случайным способом не дает возможности обнаружить все мутации заданного класса: начиная с определенной длины, происходит насыщение и полнота теста не увеличивается. Вместе с тем комбинация «чисто» случайного и псевдослучайного моделирования входных воздействий цифровых схем позволяет добиться высокой полноты относительно коротких тестов для перечисленных классов неисправностей.

Ключевые слова: цифровые компоненты, верификация, область неисправности, мутация, тестовые последовательности, проверяющий тест.

Введение

В настоящее время цифровые компоненты являются неотъемлемой частью физических систем; примерами могут служить встроенные датчики, портативные средства связи для передачи полученных данных и т.п.; соответственно такие компоненты должны быть тщательным образом верифицированы/протестированы. К сожалению, полнота верификационных тестов, построенных по спецификациям достаточно высокого уровня, таким, как расширенные (полу-) автоматы, марковские цепи и др., оказывается очень низкой [1], и построенные тесты не обнаруживают большое количество ошибок в реализациях цифровых компонентов, в том числе ошибок проектирования. Значительно более эффективными являются тесты, построенные по модели логической схемы, в которую можно (автоматически) перевести поведенческую модель более высокого уровня. Последнее объясняется достаточно просто: именно логические схемы являются наиболее близкой моделью при реализации цифровых устройств. Качество верификационного теста в значительной степени определяется его возможностями обнаруживать различные мутации эталонной логической схемы; наиболее известные мутации моделируются одиночными константными неисправностями и неисправностями «перемычек»; в последнее время к множеству мутаций добавляются так называемые трудно обнаружимые неисправности [2], которые соответствуют незначительному изменению поведения одного из логических элементов схемы. Рассматриваемые классы неисправностей, в том числе, соответствуют ошибкам, которые может совершить инженер-разработчик, описывая поведение цифровой схемы в языке логического уровня Verilog или VHDL [3], и соответственно такие тесты могут рассматриваться как верификационные тесты высокого

* Работа выполнена при частичной поддержке Российского научного фонда, грант № 16-49-03012.

качества для проверки соответствия синтезированной цифровой схемы необходимым функциональным требованиям. Если для синтезированной схемы получены положительные ответы на все тесты, построенные для описанных выше мутаций, то с большой вероятностью синтезированная схема соответствует/удовлетворяет необходимым функциональным критериям/требованиям.

Данная статья посвящена экспериментальному сравнению различных подходов к построению верификационных тестов. В частности, мы рассматриваем полноту случайно сгенерированных тестов, а также исследуем свойства тестов, построенных на основе кратчайших различающих последовательностей для обнаружения мутаций в описании поведения логической схемы (что, вообще говоря, требует довольно больших усилий для (сверх-) больших интегральных схем), и таких последовательностей, сгенерированных псевдослучайным образом с использованием системы логического синтеза и верификации. Мы экспериментально показываем, что практически для всех схем (и всех наборов мутаций) сгенерированная случайным образом тестовая последовательность может обнаружить более 70 % мутаций; однако далее происходит «насыщение», и дальнейшее увеличение длины случайной тестовой последовательности не приводит к повышению полноты такого теста. Более того, последовательности, нацеленные на обнаружение определенных мутаций, оказываются намного эффективнее, т.е. достигают практически такой же полноты при значительно меньшей длине.

Эксперименты проводились для набора контрольных примеров (бенчмарков) V01-V10 (США, 2000) [4], которые представляют собой компоненты физических систем, спроектированных для различных приложений; эти компоненты включают схемы, направленные на обработку данных, полученных от сенсорных датчиков, системы распределения нагрузки и др. В первой серии экспериментов сравниваются два подхода к построению так называемых полных тестов, т.е. тестов, обнаруживающих все возможные мутации заданного класса для эталонного описания логической схемы. В первом случае для каждой мутации находится обнаруживающая ее кратчайшая тестовая последовательность, в другом случае – тестовая последовательность генерируется псевдослучайным образом и, как правило, является более длинной. Тем не менее после минимизации теста, т.е. удаления последовательностей, обнаруживающих одно и то же множество мутаций, тест, построенный вторым способом, достаточно часто оказывается короче, что говорит о том, что во многих случаях нет смысла «тратить время и силы» на построение кратчайшей последовательности, обнаруживающей одну мутацию. Однако проведенные эксперименты также показывают, что построение тестовых последовательностей исключительно случайным способом (без нацеливания на обнаружение определенной мутации или групп мутаций) не дает возможности обнаружить все мутации заданного класса: начиная с определенной длины, происходит насыщение и полнота теста не увеличивается при увеличении длины случайной последовательности. Лучшие результаты по скорости синтеза теста и его полноте показывает совмещение двух подходов: случайной и «нацеленной» генераций с последующей минимизацией теста.

Структура работы следующая: в п. 1 кратко описываются схемы, предъявленные к экспериментам, а в п. 2 приводятся полученные экспериментальные результаты.

1. Описание схем, предъявленных к экспериментам

Логическая сеть (иногда *схема*) состоит из $(m, 1)$ -полюсников [5], каждый из которых имеет m входных полюсов (входов), один выходной полюс (выход) и реализует некоторую логическую функцию. Элемент может быть комбинационным (*вентилем*), т.е. реализовывать некоторую элементарную булеву функцию; в противном случае элемент является элементом задержки, который часто называется *триггером*. В работе рассматриваются схемы с синхронным функционированием, т.е. предполагается исследование специального класса логических сетей, в которых следующее состояние и выходной вектор вычисляются за один такт. Подобные логические схемы могут быть описаны по-разному (в различных форматах). В данной работе, подобно [2], для представления логической схемы используется формат BLIF Berkley Logic Interchange Format (BLIF). В этом случае каждый из логических элементов описывается соответствующей таблицей истинности, в которой явно перечисляются единичные наборы соответствующих булевых функций; остальные наборы доставляют значение нуль на выходе вентиля.

Для схем в формате BLIF мы рассматриваем неисправности трех типов: одиночные константные неисправности (тип А), неисправности типа «перемычка» (тип В) и одиночные неисправности функционирования логического элемента схемы, так называемые трудно обнаружимые неисправ-

ности (тип С). Подробное описание вносимых неисправностей можно найти в [2], где, в частности, иллюстрируется, что тесты, обнаруживающие все неисправности одного из типов, оказываются недостаточно полными относительно неисправностей других типов, поэтому под *полным верификационным тестом* мы понимаем объединение всех тестовых множеств.

В работе исследуются схемы из пакета ИТС'99, которые представляют собой компоненты физических систем, спроектированных для различных приложений; эти компоненты включают схемы, направленные на обработку данных, полученных от сенсорных датчиков, системы распределения нагрузки и др. Параметры схем из этого пакета приведены в табл. 1.

Таблица 1
Выбранные схемы из пакета ИТС'99

Схема	Число элементов	Число триггеров	Число внешних входов	Число внешних выходов
b01	42	2	2	5
b02	23	4	1	1
b03	126	30	4	4
b06	45	9	2	6
b08	153	21	9	4
b09	141	28	1	1
b10	178	17	11	6

Эксперименты по оценке полноты тестов для схем из пакета ИТС'99 проводились на виртуальной машине Intel(R) Core(TM) i5-3210M с частотой 2.50 ГГц и 2.5 Гбайт оперативной памяти при использовании 64-битной операционной системы CentOS 7.2 GNU/Linux с ядром 3.10.0-327.4.5.el7.x86_64. Экспериментальное исследование включает несколько этапов, которые описаны ниже.

2. Экспериментальные результаты

2.1. Эффективность построения кратчайшей различающей последовательности для эталонных схем и их мутантов

Сравним два подхода к синтезу тестов с гарантированной полнотой для логических схем. Первый подход основан на использовании команды *dprove* системы ABC [6], возвращающей различающую последовательность для двух последовательностных схем на основе подходящего (последовательностного) майтера. Во втором подходе майтер строится для комбинационных эквивалентов рассматриваемых последовательностных схем, с последующим решением задачи выполнимости для булевой функции, реализуемой данным майтером. Иными словами, в первом случае реализуется некоторая псевдослучайная различающая последовательность, во втором случае находится кратчайшая различающая последовательность, если таковая существует. Безусловно, второй способ требует больше времени для генерации тестовых последовательностей (табл. 2), но при этом он доставляет более короткие различающие последовательности.

Таблица 2
Среднее время генерации одной различающей последовательности

Схема	Подход 1, с	Подход 2, с
b01	0.14	0.26
b02	0.12	0.34
b03	0.15	0.44
b06	0.13	0.27
b08	0.15	1.48
b09	0.12	1.44
b10	0.13	0.45

Построенные тесты подвергались оптимизации, т.е. сокращению, с сохранением их полноты; результаты оптимизации приведены на рис. 1 и 2 соответственно.

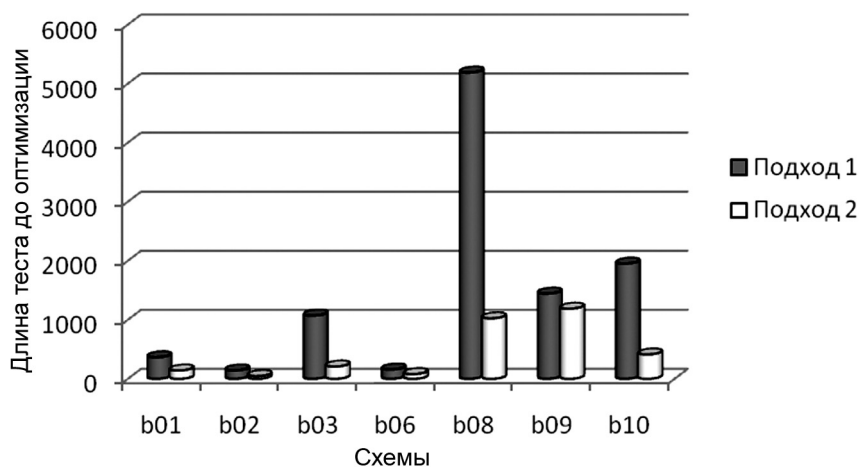


Рис. 1. Длина тестов (количество тестовых наборов) до оптимизации

Сравнив диаграммы на рис. 1 и 2, можно заметить, что после минимизации длины полных тестов оказались практически одинаковыми, а для больших схем, таких, как b08 и b09, тест, основанный на поиске кратчайшей последовательности, оказался даже длиннее. Таким образом, вообще говоря, не очевидно, нужно ли «тратить силы и время» на построение кратчайшей последовательности, обнаруживающей каждую мутацию, или же более эффективно использовать процесс минимизации всего теста, возможно, итеративно.

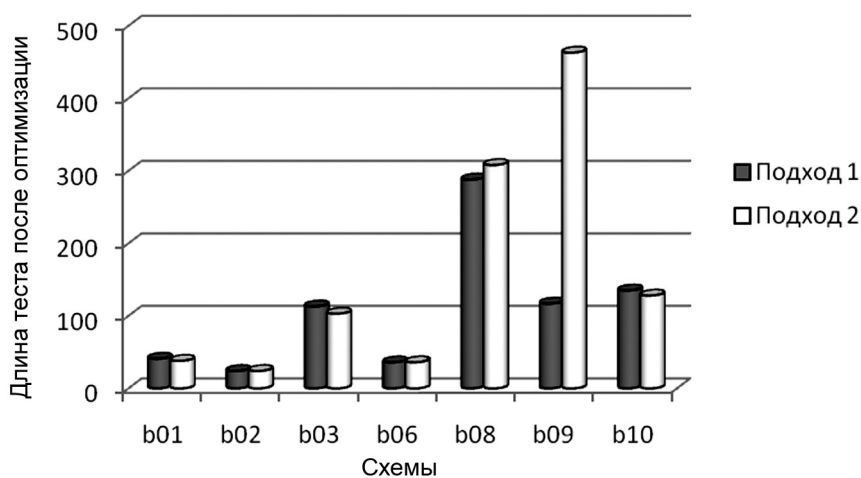


Рис. 2. Длина тестов после оптимизации

2.2. Полнота случайных тестовых последовательностей

Несмотря на то, что псевдослучайные тесты, ориентированные на обнаружение определенной неисправности, после минимизации оказались лучше (в смысле длины), чем тесты, построенные по кратчайшим последовательностям, использование только случайных тестов является не слишком эффективным (рис. 3). Для достаточно небольших схем b01, b02, b06 из выбранного пакета были проведены эксперименты по оценке полноты случайно сгенерированных тестовых последовательностей относительно всех рассматриваемых видов неисправностей.

Результаты, приведенные на рис. 3, показывают, что, начиная с некоторого значения длины случайной тестовой последовательности, достигается максимально возможная полнота. Дальнейшее увеличение длины тестовой последовательности не приводит к увеличению полноты (относительно любого класса рассматриваемых неисправностей), т.е. использование только случайных тестов оказывается неэффективным для обнаружения всех мутаций любого из рассматриваемых

классов. Тем не менее случайная генерация может быть эффективно использована на первом этапе синтеза тестов для логических схем.

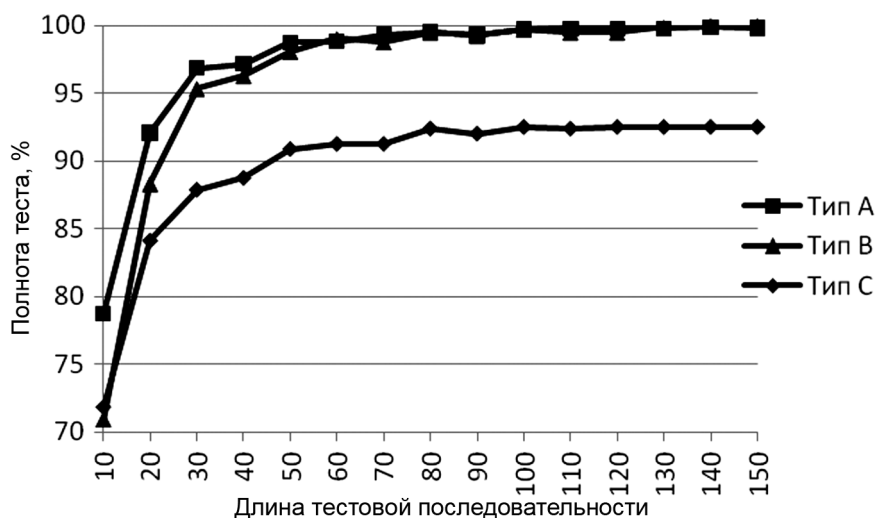


Рис. 3. Полнота случайно сгенерированных тестовых последовательностей для схемы b01

2.3. Полнота «нацеленных» псевдослучайных тестовых последовательностей

Из рис. 4, где приведены результаты экспериментов по оценке полноты тестов, полученных с использованием майтера в системе ABC («нацеленная» псевдослучайная генерация), видно, что последовательности, направленные на обнаружение некоторого мутанта, имеют лучшую полноту для коротких последовательностей. Для схемы b01 полнота 80–85 % достигается уже на длине 10, в то время как для случайных последовательностей эта длина больше 20. Таким образом, «нацеленная» псевдослучайная генерация может быть эффективно использована на втором этапе синтеза полных проверяющих тестов для логических схем после генерации случайной тестовой последовательности подходящей длины.

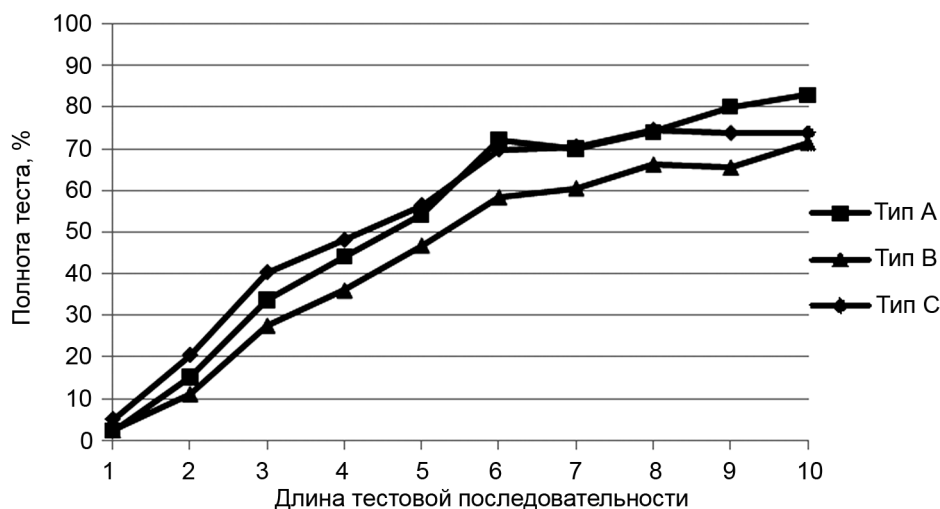


Рис. 4. Полнота «нацеленных» псевдослучайных тестовых последовательностей для схемы b01

2.4. Оценка длины теста при комбинированном подходе

В качестве третьего подхода для синтеза полных тестов был рассмотрен комбинированный подход (подход 3), в котором на первом этапе запускается случайная генерация; далее тест «достраивается» различающимися последовательностями для обнаружения не обнаруженных ранее неисправностей.

Как показывают проведенные эксперименты, длина полного теста, построенного с использованием комбинированного подхода, оказывается минимальной (рис. 5).

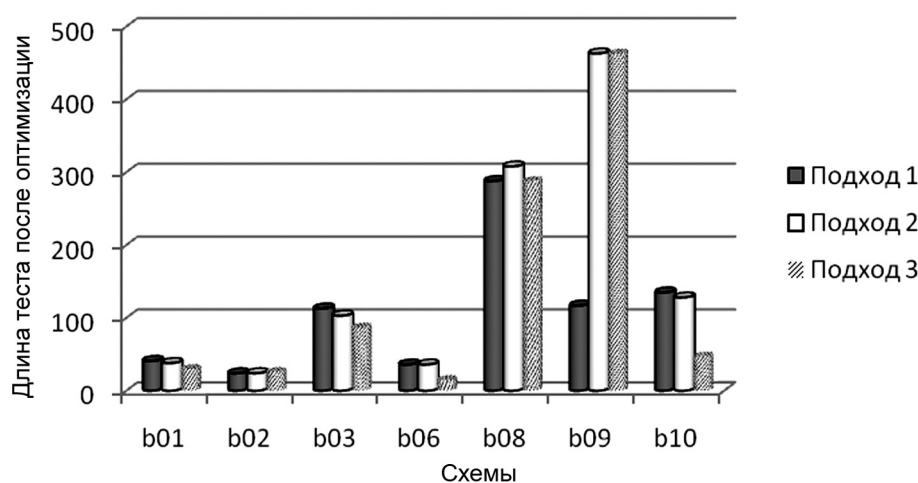


Рис. 5. Длина тестов, построенных по комбинированному методу (подход 3), после оптимизации

Заключение

В работе исследованы свойства тестовых последовательностей, построенных для верификации логических компонентов, используемых в физических системах, относительно наиболее распространенных моделей ошибок трех типов, а именно одиночных константных неисправностей, неисправностей «перемычек» и неисправностей при замене одного вентиля схемы (или одного единичного набора подходящей булевой функции). Экспериментальные результаты показывают, что в большинстве случаев нет смысла строить кратчайшие последовательности для мутаций – более эффективно провести минимизацию построенного теста, возможно, итеративно. Использование только случайных последовательностей при тестировании/верификации логических компонентов, используемых в физических системах, не дает хороших результатов: начиная с некоторого значения длины тестовой последовательности, достигается максимально возможная полнота (относительно каждого класса мутаций). Дальнейшее увеличение длины тестовой последовательности не приводит к увеличению полноты. Если последовательности строятся «нацеленно», то полнота теста существенно увеличивается – в этом случае даже короткие последовательности обнаруживают достаточно большое количество неисправностей. Соответственно наиболее перспективный подход представляет собой последовательное выполнение двух шагов: а) генерации случайных тестовых наборов для первой части теста; б) «нацеленной» генерации для «достраивания» теста подходящими различающимися последовательностями.

СПИСОК ЛИТЕРАТУРЫ

1. Smolov S.A., López J., Kushik N., et al. // Proc. IEEE East-West Design & Test Symposium (EWDTS'2016). – 2016. – P. 189–192.
2. Кушик Н.Г., Лопез Х.Е., Евтушенко Н.В. // Изв. вузов. Физика. – 2016. – Т. 59. – № 8. – С. 134–139.
3. Бибило П.Н. Основы языка VHDL: учеб. пособие. – 7-е изд. – М.: Книжный дом «ЛИБРОКОМ», 2016. – 328 с.
4. Corno F., Reorda M.S., and Squillero G. // IEEE Design Test Comput. – 2000. – V. 17. – P. 44–53.
5. Агибалов Г.П., Оранов А.М. Лекции по теории конечных автоматов. – Томск: Изд-во Том. ун-та, 1984. – 185 с.
6. Brayton R. and Mishchenko A. // Proc. CAV. – LNCS 6174. – 2010. – P. 24–40.

¹ Национальный исследовательский Томский государственный университет, г. Томск, Россия

Поступила в редакцию 14.07.17.

² Телеком Южный Париж / Университет Париж Сакле, г. Еври, Франция