

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.052.42

Щепанский
Валентин Евгеньевич

Верификация автоматных VHDL-моделей взаимодействующих цифровых
систем

АВТОРЕФЕРАТ

на соискание степени магистра технических наук
по специальности 1-40 80 01 «Элементы и устройства вычислительной
техники и систем управления»

Научный руководитель
Бибило Петр Николаевич
Профессор, доктор технических наук

Минск 2019

ВВЕДЕНИЕ

При проектировании сложной цифровой аппаратуры наиболее часто используется модель цифрового устройства с памятью в виде модели конечного автомата. Степень интеграции современных сверхбольших интегральных схем (СБИС) растет экспоненциально. Это позволяет с каждым годом размещать на кристалле всё более сложные проекты. При описании таких проектов, для более удобного восприятия используют иерархические автоматы.

Метод проверки того, что аппаратная или программная система соответствует заявленной спецификации (обладает необходимыми свойствами) называется верификацией. К сожалению, полностью верифицировать систему обычно намного сложнее, чем создать её проект. Поэтому в не очень ответственных системах верификация не всегда оправдана, и проще исправлять ошибки по мере их обнаружения во время работы системы. Однако существуют такие системы, в которых ошибки нельзя допускать или они могут обойтись слишком дорого. Например, системы управления транспортом, медицинское оборудование, военные программы, финансовые программы и многие другие области, ошибки в которых могут привести к гибели людей или слишком большим убыткам.

Из-за таких систем верификация стала одной из основных проблем проектирования устройств. Под верификацией понимается проверка правильности исходного VHDL-описания, т. е. проверка соответствия составленного синтезируемого VHDL-описания проектируемой цифровой системы спецификациям на проектирование.

Верификация отдельных автоматов является хорошо изученным направлением и для неё уже есть множество методик и функций симуляторов. В то время как верификации иерархических автоматов или систем автоматов исследованы слабо.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования

Современные системы автоматизированного проектирования (САПР) цифровых устройств на базе программируемых логических интегральных схем (ПЛИС) и заказных цифровых СБИС позволяют провести сквозной процесс проектирования – от описаний алгоритмов функционирования устройств до получения файлов конфигурирования ПЛИС либо до описаний топологии заказных СБИС. В практике проектирования

разработаны эффективные методы верификации проектных данных на различных этапах проектирования. На первый план при проектировании в настоящее время выступают проблемы верификации исходных спецификаций, представленных на высокоуровневых языках проектирования, например, на языке VHDL [1]. Для такой верификации разрабатываются различные подходы, в том числе и функциональная верификация, позволяющая оценить полноту осуществления переходов в компонентных автоматах и общем графе переходов между состояниями взаимодействующих цифровых систем, описываемых в терминах автоматных моделей.

Цель и задачи исследования

Целью данного исследования является разработка методики и тестирующей программы для выполнения верификации исходных спецификаций взаимодействующих цифровых устройств и систем.

В соответствии с поставленной целью, в работе сформулированы и решены следующие задачи:

1. Разработать методику функциональной верификации исходных спецификаций цифровых устройств, представленных на языке VHDL.
2. Разработать тестирующие программы для примеров VHDL описаний цифровых устройств.
3. Оценить полноту функционального покрытия для выполненной верификации с помощью аппарата ассертов.

Объектом исследования являются VHDL-описания взаимодействующих цифровых устройств, способы их моделирования и верификации в современных САПР.

Предметом исследования являются способы написания тестирующих программ для верификации цифровых устройств, организации верификации и оценки полноты верификации. В качестве примеров цифровых устройств предполагается использовать достаточно сложные примеры взаимодействующих цифровых устройств из практики проектирования.

Область исследования и содержание диссертационной работы соответствует образовательному стандарту высшего образования второй ступени (магистратуры) специальности 1-40 80 01 «Элементы и устройства вычислительной техники и систем управления».

Научная новизна заключается в разработке методики верификации систем на основе моделирования и построения компактных тестов по

результатам моделирования.

Положения, выносимые на защиту

1. Способ написания тестирующей программы для получения потактовых состояний устройств: конечных автоматов, сетей автоматов и иерархических автоматов.

2. Алгоритм и программа построения компактных тестов по графам переходов состояний.

Апробация результатов диссертации

Основные положения и результаты диссертационной работы докладывались и обсуждались на следующих конференциях: 54-я научная конференция аспирантов, магистрантов и студентов БГУИР (Минск, 2018).

Опубликованность результатов исследования

По результатам исследований, представленных в диссертации, опубликован тезис в сборниках и материалах научной конференции.

Структура и объем диссертации

Структура диссертационной работы обусловлена целью, задачами и логикой исследования. Работа состоит из введения, пяти глав и заключения, библиографического списка и приложений. Общий объем диссертации – 104 страниц. Работа содержит 14 таблицы, 16 рисунков. Библиографический список включает 15 наименования, графический материал включает 10 слайдов презентации (Приложение Б).

СОДЕРЖАНИЕ РАБОТЫ

Во **введении** рассмотрено современное состояние проблемы верификации конечных автоматов, а также дается обоснование актуальности темы диссертационной работы.

В **общей характеристике работы** сформулированы ее цель и задачи, показана связь с научными программами и проектами, даны сведения об объекте исследования и обоснован его выбор, представлены положения, выносимые на защиту, приведены сведения о личном вкладе соискателя, апробации результатов диссертации и их опубликованность, а также, структура и объем диссертации.

В **первой главе** даётся определение основным понятиям.

Автоматное программирование – это парадигма программирования, при использовании которой программа или ее фрагмент осмысливается как система автоматизированных объектов управления. Особенность такого программирования заключается в явном выделении состояний и переходов между ними.

Формальная верификация представляет собой процесс доказательства с помощью формальных методов корректности или некорректности алгоритмов, программ и систем в соответствии с заданным описанием их свойств.

Для формальной верификации требуются:

1. Модель системы, обычно содержащая множество состояний, которые хранят информацию о значениях переменных, программных счетчиках и т. п., и отношение переходов, которое описывает, как система переходит из одного состояния в другое.

2. Метод спецификации для выражения требований в формальном виде.

3. Множество правил доказательства, позволяющих определить, удовлетворяет ли модель сформулированным требованиям.

Система синхронных автоматов – это система конечных автоматов связанных таким образом, что изменение состояния всех автоматов происходит одновременно – общий синхросигнал.

Иерархический конечный автомат – это система конечных автоматов связанных таким образом, что любой модуль может быть вызвана из любого другого модуля включая этот модуль.

Конечные автоматы используют при проектировании относительно простых схем. Для более сложных устройств лучше всего подходит иерархические конечные автоматы. Иерархический конечный автомат можно рассматривать на разных уровнях абстракции таким образом, что модули автомата, которые образуют иерархию, могут быть самостоятельно

реализованы, протестированы и отлажены. Этот метод позволяет справиться с растущей сложностью цифровых схем и систем.

Были выдвинуты следующие требования к примерам автоматов, над которыми будет проводиться верификация:

Число компонентных автоматов – не более 4.

1. Число состояний компонентного автомата – не более 40.
2. Число входных (выходных) сигналов автомата – не более 10.
3. Тип входных данных – `std_logic_vector`.

Во **второй главе** представлены актуальные методы решения проблем верификации.

В настоящее время для формальной верификации программного обеспечения применяются два основных подхода: дедуктивная верификация (верификация на основе логического вывода) и верификация на модели.

Дедуктивная верификация трудоемка и требует высококвалифицированных специалистов в области доказательства теорем и логического вывода.

Верификация на модели состоит из четырех основных этапов.

1. Построение модели программы.
2. Задание требований в терминах выбранного типа темпоральной логики.
3. Верификация модели с целью проверки выполнения формализованных требований.
4. Анализ контрпримера в случае несоответствия программы требованиям.

Выполнение первого этапа верификации в общем случае достаточно трудоемко в связи с необходимостью построения модели, адекватной верифицируемой программе. При этом полученная модель должна иметь конечное число состояний, так как аппарат анализа моделей с бесконечным числом состояний разработан только для отдельных классов систем (например, для вполне структурированных систем помеченных переходов). Отметим, что для эффективной проверки модели число состояний в ней должно быть не слишком большим. Однако существующие методы построения моделей для программ, написанных традиционным способом, приводят к очень большому числу состояний, так как в традиционных программах обычно не разделяются управляющие и вычислительные состояния.

Затруднение вызывает также и выполнение второго этапа верификации, так как для верификации требования должны быть сформулированы в терминах модели. При этом также встает вопрос об адекватности формально записанных требований исходным. По сравнению с первыми двумя этапами, третий этап верификации достаточно хорошо автоматизируется. Известны

инструментальные средства для верификации моделей (верификаторы), в том числе свободные, например, NuSMV, SPIN и Bogor. На четвертом этапе для программ общего вида при нахождении ошибки в модели часто возникают проблемы при переносе контрпримера в верифицируемую программу.

В **третьей главе** описаны разработанные примеры различных автоматов.

Для примера конечного автомата был взят интерфейс JTAG.

JTAG — название рабочей группы по разработке стандарта IEEE 1149. Позднее это сокращение стало прочно ассоциироваться с разработанным этой группой специализированным аппаратным интерфейсом на базе стандарта IEEE 1149.1. Официальное название стандарта Standard Test Access Port and Boundary-Scan Architecture. Интерфейс предназначен для подключения сложных цифровых микросхем или устройств уровня печатной платы к стандартной аппаратуре тестирования и отладки.

Стандарт JTAG позволяет контролировать и управлять состояниями портов ввода-выводов. Это происходит за счет встраивание специальной архитектуры в современные чипы, обеспечивающей доступ к выводам с помощью 4-х проводного последовательного интерфейса.

Автомат имеет 16 состояний. Управление интерфейсом осуществляется путем воздействия на автомат посредством сигнала TMS. Переходы происходят по переднему фронту сигнала TCK. Чтение и запись данных происходят одновременно.

В качестве примера синхронной сети автоматов была выбрана модель сети автоматов – это обобщенное описание протокола MSI, обеспечивающего когерентность распределенной памяти.

Для примера иерархического конечного автомата был взят банкоматом, в задачи которого входит:

- проверка пароля при вводе карты;
- просмотр баланса;
- снятие денег со счёта.

Структура работы банкомата состоит из:

1. Автомат – основная часть банкомата, в которой формируются состояния автомата и выдаются команды на остальные модули. С этой частью мы будем работать в дальнейшем.

2. Механическая часть банкомата – модуль банкомата связанный с наличием карточки в банкомате, её выдачи, проверки наличности в банкомате, выдача денег.

3. Монитор – интерфейс вывода данных банкомата для передачи информации пользователю.

4. Регистр данных – модуль банкомата, в котором хранится временная информация текущих операций: номер банковского счёта, введённый пин код, баланс, сумма на снятие денег.

5. Связь с банком – модуль банкомата отвечающий за все операции связанные с банком: проверка пин кода, запрос на снятие средств, проверка баланса.

6. Циферблат – интерфейс ввода для работы с банком пользователю.

В **четвёртой главе** описывается разработанная на языке Matlab программа BypassGraph, представлен пример её работы.

Программа BypassGraph выполняет следующие задачи:

1. На основании файлов состояний (State.tst) и входных сигналов (IN.tst) построить матрицу смежности и ориентированный граф, вершинам которого соответствуют состояния, а каждая дуга является переход из одного состояния в другое.

2. Обойти ориентированный граф и записать пройденный путь.

3. Минимизировать полученный путь.

4. В файл теста (res.tst) записывается набор входных сигналов, при тестировании на которых исследуемый автомат пройдёт все свои состояния.

В **пятой главе** представлены результаты моделирования примеров автоматов на случайно сгенерированных входных наборах и на наборах полученных программой BypassGraph. Проверена их полнота при помощи возможностей системы Questa Sim, а именно распознавания FSM конечного автомата и директив языка PSL. Полученные тысты по

В приложениях представлен результат на проверку плагиата, графический материал, исходное описание разработанных примеров автоматов на языке VHDL с их тестами и исходное описание разработанной программы на языке Matlab

ЗАКЛЮЧЕНИЕ

В современно Мире сложность цифровых устройств постоянно растёт. Тем самым сложность написания проектов и их верификация становится всё сложнее и требует всё больше времени. VHDL является мощным средством для проектирования цифровых устройств.

В диссертации решается проблема верификации при помощи средств моделирования: системы Questa Sim и языка PSL. Для выделения состояний устройств по тактам были написаны тестирующие программы на языке VHDL. Для поиска минимизированного пути покрытия графов состояний цифровых систем была написана программа. При помощи написанной программы были получены тесты для конечного автомат, сети автоматов и иерархического конечного автомата. Автоматы были промоделированы на полученных тестах. При помощи системы Questa Sim, где она позволила выделить автоматы, и директив покрытия программы PSL было показали, что тесты являются полными, т.е. покрывают все переходы между состояниями графов автоматов. Можно сделать вывод, что программа работает корректно и решает поставленную перед ней задачу.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

[1] Щепанский, В. Е. Верификация VHDL-моделей конечных автоматов / В. Е. Щепанский // Компьютерные системы и сети: материалы 54-й научной конференции аспирантов, магистрантов и студентов. – Минск: БГУИР, 2018. – С. 255-256.