

Российская академия наук
Институт системного программирования

УДК 519.68

На правах рукописи



Камкин Александр Сергеевич

**МЕТОД АВТОМАТИЗАЦИИ ИМИТАЦИОННОГО ТЕСТИРОВАНИЯ
МИКРОПРОЦЕССОРОВ С КОНВЕЙЕРНОЙ АРХИТЕКТУРОЙ НА
ОСНОВЕ ФОРМАЛЬНЫХ СПЕЦИФИКАЦИЙ**

Специальность 05.13.11 –
математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата физико-математических наук

Москва
2009

Работа выполнена в Институте системного программирования РАН.

Научный руководитель: доктор физико-математических наук
Петренко Александр Константинович

Официальные оппоненты: кандидат технических наук
Груздов Федор Анатольевич
доктор физико-математических наук
Лазутин Юрий Михайлович

Ведущая организация: **Московский государственный институт
электроники и математики**

Защита диссертации состоится « 24 » апреля 2009 г. в 15 часов на заседании диссертационного совета Д.002.087.01 при Институте системного программирования РАН по адресу:

109004, Москва, ул. А. Солженицына (бывшая Б. Коммунистическая), д.25,
Институт системного программирования РАН, конференц-зал (комн. 110).

С диссертацией можно ознакомиться в библиотеке Института системного программирования РАН.

Автореферат разослан « » марта 2009 г.

Ученый секретарь
диссертационного совета
кандидат физ.-мат. наук



/Прохоров С.П./

Общая характеристика работы

Актуальность темы

В современном мире большое значение отводится разработке и производству *микропроцессоров* — программно-управляемых устройств, предназначенных для решения задач цифровой обработки данных. Требования, предъявляемые к надежности микропроцессоров, очень высоки. Ошибки в них могут приводить к некорректной работе вычислительных систем, в состав которых они входят, в то время как число критически важных приложений микропроцессорных систем достаточно велико.

Микропроцессоры являются сложными электронными устройствами, производство которых можно начать только после тщательного проектирования. Проектирование микропроцессоров выполняется на специализированных языках описания аппаратуры (HDL, Hardware Description Languages). Поскольку исправление ошибок в уже изготовленных микропроцессорах возможно только в исключительных случаях и требует колоссальных затрат, ошибки крайне желательно выявить и исправить еще на этапе проектирования.

Одним из основных методов обеспечения надежности микропроцессоров является тестирование проектной модели. Такое тестирование называется *имитационным (simulation-based verification)*¹. По различным данным, тестирование микропроцессоров составляет 50-80% от общего объема трудозатрат на их разработку. Для большей эффективности обнаружения ошибок и упрощения их локализации тестирование микропроцессоров включает в себя тестирование модулей по отдельности. Таким образом, тестирование осуществляется на двух основных уровнях: *модульном (unit-level)* и *системном (core-level)*.

¹ Далее для краткости будем называть имитационное тестирование просто тестированием.

Большинство микропроцессоров, разрабатываемых в настоящее время, имеют *конвейерную архитектуру*. Это означает, что в один и тот же момент времени микропроцессор может обрабатывать сразу несколько инструкций. Выполнение инструкций в таких микропроцессорах разбивается на несколько стадий, при этом разные стадии разных инструкций могут выполняться параллельно. Конвейерная организация значительно усложняет устройство микропроцессора, добавляя в него механизмы планирования и синхронизации выполнения инструкций, которые в совокупности называются *управляющей логикой (control logic)*. Сложная организация и огромное пространство состояний микропроцессоров с конвейерной архитектурой делают их тестирование очень сложной задачей, которую невозможно решить без применения *методов автоматизации тестирования*.

Анализ ошибок в микропроцессоре MIPS R4000 PC/SC (ревизия 2.2)², проведенный Ричардом Хо (Richard Ho) и др.³, говорит, что большинство ошибок (93.5%) связаны с управляющей логикой конвейера, причем значительную их часть очень сложно обнаружить. Приведенный выше анализ ошибок относится к середине 1990-ых гг., но, следует отметить, что ошибки в управляющей логике и по сей день являются трудно обнаруживаемыми. Список ошибок, датируемый 2006 г., широко используемого в аэрокосмической промышленности микропроцессора TSC695F компании Atmel с архитектурой SPARC v7, включает ошибку (одну из трех), связанную с работой конвейера⁴.

В последнее время определенный прогресс в автоматизации тестирования микропроцессоров достигнут в исследованиях в области *тестирования на основе формальных спецификаций (specification-based testing)*. Основная идея этого направления заключается в том, что в процесс

² MIPS R4000PC/SC Errata, Processor Revision 2.2 and 3.0. MIPS Technologies Inc., May 10, 1994.

³ R.C. Ho, C.H. Yang, M.A. Horowitz, and D.L. Dill. Architecture Validation for Processors. ISCA'1995: International Symposium on Computer Architecture, 1995.

⁴ TSC695 Errata Sheet. Atmel Corporation, July, 2006.

разработки тестов включаются формальные спецификации тестируемого компонента и формальные модели тестов. Спецификации и модели используются для автоматической проверки правильности поведения, автоматической генерации тестовых последовательностей и оценки полноты тестирования. Их использование позволяет сократить трудоемкость разработки тестов и повысить «глубину» тестирования.

Прямое применение существующих методов автоматизации тестирования на основе формальных спецификаций к промышленным проектам сталкивается с проблемами. Разработка спецификаций и моделей для сложных микропроцессоров и их модулей является трудоемкой задачей, и если метод не предоставляет механизмы эффективной поддержки тестов, то его применение в условиях постоянного уточнения требований, постоянных доработок интерфейса и реализации, которые имеют место в промышленных проектах, потребует колоссальных затрат. Существующие методы автоматизации тестирования либо не масштабируются на сложные микропроцессоры, либо не устойчивы к изменениям в реализации, что приводит к существенной переработке разработанных описаний даже при небольших изменениях в реализации.

Одна из причин, по которой методы на основе формальных спецификаций не достаточно эффективно применяются в промышленных проектах, состоит в наличии двух независимых описаний: формальной спецификации, которая используется для оценки правильности поведения микропроцессора или его модуля, и модели теста, используемой для генерации тестовых последовательностей и оценки полноты тестирования. Другая причина заключается в неразвитых средствах формальной спецификации для микропроцессоров с конвейерной архитектурой, что не позволяет достичь высокого уровня автоматизации тестирования. Представляется перспективным разработать метод формальной спецификации конвейера и метод автоматизации тестирования на основе спецификаций предлагаемого вида, лишенный указанных недостатков.

Цель и задачи работы

Целью диссертационной работы является разработка метода автоматизации тестирования микропроцессоров на основе формальных спецификаций, который масштабируется на сложные микропроцессоры с конвейерной архитектурой и является пригодным для использования в промышленных проектах. Для достижения цели работы были поставлены следующие задачи:

1. Провести анализ существующих методов автоматизации тестирования микропроцессоров с конвейерной архитектурой.
2. Разработать методы автоматизации модульного и системного тестирования микропроцессоров с конвейерной архитектурой, которые обеспечивают высокий уровень автоматизации разработки тестов и повторного использования спецификаций и тестов.
3. Провести апробацию разработанных методов в промышленных проектах.

Научная новизна

Научной новизной обладают следующие результаты работы:

1. Метод формальной спецификации модулей микропроцессоров на основе пред- и постусловий стадий выполнения операций.
2. Метод избыточного описания тестового покрытия с помощью тестовых ситуаций и зависимостей.
3. Метод генерации тестовой последовательности на основе обобщенной автоматной модели конвейера, построенной по формальной спецификации модуля и описанию тестового покрытия.
4. Метод генерации тестовых программ для системного тестирования микропроцессоров на основе формальной спецификации системы команд и тестового покрытия.

Практическая значимость

Практическая значимость работы подтверждается эффективностью применения предложенных методов для тестирования отдельных модулей и

подсистем микропроцессора Комдив64-СМП, системного тестирования микропроцессора Комдив64 и системного тестирования арифметического сопроцессора Комдив128, разрабатываемых в НИИ системных исследований РАН.

Результаты работы могут быть использованы в исследованиях, которые ведутся в Институте системного программирования РАН, Московском государственном институте электроники и математики, НИИ системных исследований РАН, Институте точной механики и вычислительной техники им. С.А. Лебедева РАН, Институте проблем информатики РАН и других научных и промышленных организациях.

Апробация и публикации

Основные положения работы докладывались на следующих конференциях и семинарах:

- Научной конференции «Тихоновские чтения», проводимой на факультете ВМиК МГУ им. М.В. Ломоносова (г. Москва, 2005 г.);
- Международном симпозиуме по усиливающим приложениям формальных методов, верификации и валидации (ISoLA: International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, г. Пафос, 2006 г.);
- Первом весеннем коллоквиуме молодых исследователей в области программной инженерии (SYRCoSE: Spring Young Researchers Colloquium on Software Engineering, г. Москва, 2007 г.);
- Международном симпозиуме «Восток-Запад: проектирование и тестирование» (EWDTS: East-West Design & Test Symposium, г. Ереван, 2007 г. и г. Львов, 2008 г.);
- Форуме аспирантов, проводимом в рамках международной конференции «Проектирование, автоматизация и тестирование в Европе» (DATE: Design, Automation & Test in Europe, г. Мюнхен, 2008 г.);

- Семинарах Института системного программирования РАН (г. Москва, 2007-2008 гг.);
- Семинаре НИИ системных исследований РАН (г. Москва, 2008 г.).

По теме диссертации автором опубликовано 14 работ (из них 1 в изданиях по перечню ВАК), список которых приведен в конце автореферата.

Структура и объем диссертации

Работа состоит из введения, четырех глав, заключения и списка литературы (109 наименований). Основной текст диссертации (без приложений и списка литературы) занимает 172 страницы.

Краткое содержание диссертации

Во **введении** обосновывается актуальность темы работы, определяются ее цели и задачи, раскрывается практическая значимость.

Первая глава содержит обзор методов, которые в настоящее время используются для функциональной верификации микропроцессоров. Наряду с тестированием рассматриваются методы формальной верификации, такие как проверка эквивалентности, проверка моделей и автоматизированное доказательство теорем. Описываются достоинства и ограничения различных методов. Основное внимание уделяется методам модульного и системного тестирования, поскольку это основные методы проверки корректности микропроцессоров и других сложных программных и аппаратных систем. В конце главы делается анализ существующих методов автоматизации тестирования микропроцессоров с конвейерной архитектурой, уточняются цели и задачи диссертационной работы.

Во **второй главе** описывается предлагаемый автором *метод автоматизации модульного тестирования микропроцессоров*. В начале главы вводятся основные понятия теории расширенных автоматов, на основе которых формально определяется модель модуля с конвейерной организацией, называемая *автоматной моделью конвейера*. Для наглядности сначала рассматривается случай конвейеров без блокировок, затем введенные понятия обобщаются на случай конвейеров с блокировками. Для

каждого из этих случаев описывается метод формальной спецификации модулей, названный *контрактной спецификацией конвейера*.

Метод является обобщением контрактных спецификаций в форме *пред- и постусловий* на случай модулей с конвейерной организацией. Поскольку в конвейерных модулях выполнение операций разбито на *стадии*, в предлагаемом методе спецификации используется понятие *функции композиции операций*, которая каждому *стимулу* ставит в соответствие последовательность стадий. Для описания управляющей логики модуля используется *состояние управления*, которое моделирует выполняемые на конвейере операции. Спецификация отдельной операции включает спецификации стимула и стадий операции. Спецификация стимула состоит из *предусловия стимула*, которое определяет допустимость подачи операции на выполнение в зависимости от значений входных параметров операции, состояния данных (контекста) и состояния управления. Спецификация каждой стадии включает *охранный предикат стадии* и *постусловие стадии*. Охранный предикат стадии определяется на основе состояния управления. Невыполнимость предиката означает блокировку стадии.

Семантика контрактной спецификации конвейера определяется с помощью формализма расширенных автоматов. Для этого используются два основных оператора: *оператор сдвига конвейера* и *оператор связывания стадий*. Оператор сдвига конвейера определяет изменение состояния управления при подаче стимула в зависимости от значений охранных предикатов стадий. Оператор связывания стадий возвращает множество незаблокированных стадий состояния управления. Автомат, лежащий в основе расширенного автомата, интерпретирующего контрактную спецификацию конвейера, называется *управляющим автоматом*. Управляющий автомат является формальной моделью управляющей логики модуля.

Между расширенными автоматами и контрактными спецификациями вводится *отношение соответствия*, которое определяется с использованием функций соответствия стимулов, реакций, состояний и контекста.

Исследуется выразительная мощность контрактных спецификаций конвейера. Результатом исследований являются *теоремы об адекватности и полноте спецификаций*:

Теорема (об адекватности спецификаций). Пусть инициальный расширенный автомат соответствует контрактной спецификации конвейера, тогда соответствующий спецификационный подавтомат является автоматной моделью конвейера.

Теорема (о полноте спецификаций). Для любой автоматной модели конвейера существует точно описывающая ее контрактная спецификация конвейера.

На практике распространены модули, имеющие память (набор ресурсов), которую можно адресовать. Для компактного описания таких модулей вводится понятие *контрактной спецификации конвейера с ресурсами*. Идея состоит в том, что каждая стадия операции параметризуется адресами ресурсов, которые она использует. Адреса ресурсов вычисляются по значениям входных параметров стимула. В состояние управления добавляется информация об адресах ресурсов, используемых в стадиях обрабатываемых операций. Охранные предикаты стимулов и стадий определяются на основе расширенного состояния управления. Семантика контрактной спецификации конвейера с ресурсами определяется с помощью ее преобразования к форме определенной ранее контрактной спецификации конвейера с блокировками. Тем самым, эти формализмы имеют одинаковую выразительную мощность.

Далее в главе описывается, как контрактные спецификации конвейера могут быть использованы для решения задач тестирования: проверки правильности поведения, оценки полноты тестирования и генерации тестовой последовательности.

Описывается *схема проверки правильности поведения модулей*, на основе которой производится классификация ошибок в модулях. Выделяются следующие типы ошибок: *ошибки несоответствия состояния, ошибки*

несоответствия реакций и ошибки несоответствия контекстов. Основной интерес работы представляют ошибки несоответствия состояний, поскольку они формализуют ошибки в управляющей логике модуля.

Для оценки полноты тестирования предлагается *метод неизбыточного описания тестового покрытия*, который заключается в декомпозиции описания тестового покрытия на описания *тестовых ситуаций* и *зависимостей*. Тестовые ситуации определяются для отдельных стимулов и описывают ограничения на значения входных параметров и контекст контрактной спецификации. Зависимости ограничивают значения входных параметров для пары стимулов. В работе предполагается, что множества тестовых ситуаций и зависимостей являются *полными*, т.е. полностью покрывают области определений соответствующих стимулов.

Цель тестирования определяется как покрытие всевозможных *обобщенных состояний конвейера* — состояний управления, размеченных тестовыми ситуациями и зависимостями. Такие состояния не просто моделируют выполняемые на конвейере операции, но и содержат для каждой из них сведения о тестовой ситуации и информацию о том, как операция зависит от других выполняемых на конвейере операций.

В работе предлагается конструктивный способ описания тестового покрытия, позволяющий тестовой системе автоматически генерировать значения входных параметров стимулов на основе информации о тестовой ситуации и зависимостях. Тестовые ситуации и зависимости снабжаются специальными функциями, называемыми *конструкторами*, которые полностью или частично конструируют значения входных параметров соответствующего стимула. При подаче стимула для определенной тестовой ситуации и множества зависимостей от выполняющихся на конвейере операций применяются соответствующие конструкторы, результатом которых является набор значений входных параметров стимула.

Предлагаемый метод генерации тестовой последовательности основан на обходе графа состояний *обобщенной автоматной модели конвейера*, которая строится на основе управляющего автомата контрактной

спецификации конвейера и описания тестового покрытия. Состояниями обобщенной автоматной модели конвейера являются обобщенные состояния, стимулами — стимулы контрактной спецификации конвейера, размеченные тестовыми ситуациями и зависимостями. Показывается, что граф состояний обобщенной автоматной модели конвейера является конечным, детерминированным и сильно-связным, что позволяет применять избыточные алгоритмы обхода, исследованные в работах И.Б. Бурдонова, А.С. Косачева, В.В. Кулямина и А.В. Хорошилова.

Доказывается *теорема о полноте тестов*, которая утверждает, что при определенных ограничениях на класс ошибок, спецификацию и тестовое покрытие тестовая последовательность, построенная с помощью предложенного метода, позволяет обнаружить все ошибки в тестируемом модуле. Во-первых, предполагается, что тестируемый модуль можно смоделировать расширенным автоматом, причем модуль считается ошибочным тогда и только тогда, когда его автоматная модель не соответствует спецификации. Во-вторых, предполагается, что в тестируемом модуле все ошибки являются следствием ошибок в управляющей логике (ошибки несоответствия состояний). В-третьих, тестовые ситуации и зависимости должны детерминированным образом определять выполнимость охранных предикатов и идентифицировать состояние управления тестируемого модуля.

Теорема (о полноте тестов). *Пусть тестовые ситуации и зависимости определяют охранные предикаты контрактной спецификации конвейера и состояние реализации. Тогда, если в реализации существует ошибка, она будет найдена тестовой последовательностью, построенной с помощью обхода графа состояний обобщенной автоматной модели конвейера.*

В конце главы описывается разработанная автором библиотека PIPE для инструмента CTESK, предназначенная для поддержки создания спецификаций и тестов для модулей с конвейерной организацией.

В **третьей главе** описывается предлагаемый автором *метод автоматизации системного тестирования микропроцессоров*. Метод заключается в автоматической генерации тестовых программ на основе формальной спецификации системы команд и описания тестового покрытия. Для спецификации инструкций используется модель архитектуры микропроцессора, отражающая подсистемы микропроцессора, используемые типы данных и состояние микропроцессора. Спецификация инструкции включает следующие составляющие:

- *интерфейс инструкции* — описывает операнды инструкции. Для каждого операнда указывается его имя, способ передачи значения (непосредственно или через регистр, в последнем случае также указывается тип регистра), тип данных и информация о том, является операнд входным или выходным;
- *предусловие инструкции* — определяет ситуации, в которых определено поведение микропроцессора при выполнении данной инструкции. Автоматически создаваемой частью условия является проверка совместимости значений входных операндов соответствующим типам данным;
- *функция выполнения инструкции* — обновляет модельное состояние микропроцессора на основе значений входных операндов инструкции и текущего состояния микропроцессора;
- *ассемблерный формат инструкции* — определяет формат записи инструкции и ее операндов на языке ассемблера.

Совместно с методом можно использовать два способа проверки правильности поведения: с помощью *сравнения трасс* и используя *тесты со встроенными проверками (self-checking tests)*. Сравнение трасс основано на ко-симуляции, когда сгенерированные тестовые программы выполняются на тестируемой модели микропроцессора и симуляторе. Результатами выполнения программ являются *трассы* — последовательности событий, каждое из которых включает данные о выполненной инструкции, значениях ее выходных регистров, состоянии счетчика адреса и, возможно, некоторую

другую информацию. Вердикт об ошибке выносится на основе сравнения полученных трасс на соответствие. В случае тестов со встроенными проверками, как видно из названия, проверки правильности поведения вставляются в тестовую программу. Такие тесты можно использовать для тестирования проектных моделей микропроцессоров без использования вспомогательного симулятора, а также для проверки готовых микросхем.

Как и в описанном методе модульного тестирования метод системного тестирования использует *неизбыточное описание тестового покрытия* в форме *тестовых ситуаций* и *зависимостей*. Тестовые ситуации описывают ограничения на значения входных операндов инструкций и состояние микропроцессора. Зависимости ограничивают значения входных операндов для пары инструкций. В методе выделяется два типа зависимостей: *зависимости по регистрам* и *зависимости по данным*. Зависимости по регистрам подразделяются на зависимости типа «определение-использование» («чтение после записи»), когда выходной регистр одной инструкции является входным регистром следующей за ней инструкции⁵, зависимости типа «определение-определение» («запись после записи»), когда выходной регистр одной инструкции также является выходным регистром следующей за ней инструкции, а также зависимости типа «использование-определение» («запись после чтения»).

Зависимости по данным представляют собой ограничения на значения операндов пары инструкций. Распространенным случаем таких зависимостей являются зависимости по адресам, когда значения операндов имеют смысл адресов данных в памяти. В качестве примеров зависимостей по адресам можно привести следующие: совпадение виртуальных адресов, используемых различными инструкциями; совпадение физических адресов при несовпадающих виртуальных адресах; попадание в одну и ту же запись буфера трансляции адресов; попадание в одну и ту же строку кэш-памяти.

⁵ Между зависимыми по регистру инструкциями могут находиться другие инструкции, но эти инструкции не должны переопределять регистр.

Генерируемые тестовые программы имеют следующую структуру:

$\pi = \pi_{start} \cdot \{\langle \pi_i, x_i[s_i, d_i] \rangle\}_{i=1,n} \cdot \pi_{stop}$, где:

- π_{start} — *инициализирующая программа*
префикс тестовой программы, который содержит инструкции, предназначенные для инициализации микропроцессора;
- $\langle \pi_i, x_i[s_i, d_i], post_i \rangle$ — *тестовый вариант* ($i = 1, \dots, n$):
 - π_i — *программа подготовки тестового воздействия*
последовательность инструкций, осуществляющая инициализацию регистров и памяти микропроцессора;
 - $x_i[s_i, d_i]$ — *тестовое воздействие*
специально подготовленная последовательность инструкций, предназначенная для тестирования микропроцессора, где s_i — *множество тестовых ситуаций*, а d_i — *множество зависимостей*;
 - $post_i$ — *тестовый оракул*
последовательность инструкций, содержащая проверку корректности состояния микропроцессора после выполнения тестового воздействия. Тестовый оракул является необязательным элементом тестовой программы.
- π_{stop} — *завершающая программа*
постфикс тестовой программы, который содержит инструкции, предназначенные для завершения работы микропроцессора;
- n — *размер тестовой программы*
число тестовых воздействий в программе.

В методе реализуется направленный перебор всевозможных сочетаний тестовых ситуаций и зависимостей для последовательностей инструкций ограниченной длины. Как и все комбинаторные методы, он подвержен «комбинаторному взрыву» — размер тестов резко возрастает при увеличении длины тестируемых последовательностей инструкций, числа тестовых ситуаций и других параметров генерации. Для сокращения размера тестов

используются дополнительные *эвристики*, применяемые на основе экспертных оценок:

- *классификация инструкций* — инструкции системы команд микропроцессора разбиваются на группы; при генерации указывается максимальное число различных групп инструкций в тестовом воздействии.
- *факторизация инструкций* — некоторые инструкции внутри одной группы объединяются в классы эквивалентности; тестовые шаблоны, в которых на одинаковых позициях находятся одинаковые инструкции, считаются эквивалентными — для тестирования достаточно использовать любой из них.
- *ограничение числа или глубины зависимостей* — при генерации указывается максимальное число зависимостей или максимальная глубина зависимостей между инструкциями.

В конце главы описывается архитектура генератора тестовых программ для микропроцессоров, рассматривается устройство разработанного инструмента.

В **четвертой главе** описываются результаты практической апробации предлагаемого автором метода для тестирования промышленных микропроцессоров, их модулей и подсистем, а именно: буфера трансляции адресов, модуля кэш-памяти второго уровня и подсистемы управления памятью микропроцессора Комдив64-СМП, микропроцессора Комдив64 и арифметического сопроцессора Комдив128, разрабатываемых в НИИ системных исследований РАН. Приводятся данные о трудоемкости разработки спецификаций и тестов, объеме разработанных описаний и возможности их повторного использования для тестирования микропроцессоров или их модулей с близкой функциональностью, в частности, для тестирования других версий тех же компонентов. Основным результатом главы является практическое обоснование того, что предлагаемый автором метод автоматизации тестирования микропроцессоров с конвейерной архитектурой соответствует поставленным

в работе целям. Обобщенные данные по методам модульного и системного тестирования приведены в таблицах 1 и 2 соответственно.

Таблица 1. Данные по методу модульного тестирования.

| Показатель | Буфер трансляции адресов | Модуль кэш-памяти второго уровня |
|---|---------------------------------|---|
| Объем реализации, строк кода | 3500 | 3000 |
| Число тестируемых операций | 5 | 6 |
| Число требований | 99 | 166 |
| Объем спецификаций и тестов, строк кода | 3500 | 4700 |
| Трудоемкость разработки тестов, чел.-мес. | 2.5 | 4 |
| Число найденных ошибок в проектной модели | 10 | 4 |

Таблица 2. Данные по методу системного тестирования.

| Показатель | Подсистема управл. памятью | Микропроцессор Комдив64 | Сопроцессор Комдив128 |
|---|-----------------------------------|--------------------------------|------------------------------|
| Число тестируемых инструкций | 4 | 221 | 109 |
| Объем спецификаций и тестов, строк кода | 5560 | 47190 | 20610 |
| Трудоемкость разработки тестов, чел.-мес. | 2 | 9 | 4 |
| Число найден. ошибок в эталонном симуляторе | 0 | 6 | 5 |
| Число найден. ошибок в проектной модели | 1 | 9 | <i>Нет данных</i> |

Применение метода модульного тестирования в проектах по тестированию буфера трансляции адресов и модуля кэш-памяти второго уровня показывает, что трудоемкость разработки тестов пропорциональна числу требований, накладываемых на тестируемый модуль, и составляет около 2-х требований за 1 человеко-день. Для метода системного тестирования получены данные, что в среднем разработка тестов для 1-ой инструкции микропроцессора занимает 0.7–0.8 человеко-дня. Линейная

зависимость трудоемкости разработки тестов от числа требований и инструкций позволяет утверждать, что предлагаемые методы являются масштабируемыми. Данные о трудоемкости разработки тестовых программ для микропроцессора Комдив64-СМП показывают, что предлагаемый метод автоматизации системного тестирования, позволяет сократить затраты на создание системных тестов в 2–2.5 раза.

В заключении перечисляются основные результаты работы.

Основные результаты работы

Основные научные результаты, полученные в диссертационной работе и выносимые на защиту, состоят в следующем:

1. Разработан метод формальной спецификации модулей микропроцессоров на основе пред- и постусловий стадий выполнения операций, названный контрактной спецификацией конвейера. Формальные спецификации предлагаемого вида могут использоваться для автоматизации проверки правильности поведения конвейерных модулей, на их основе можно определять тестовое покрытие и оценивать полноту тестирования. Разработанный метод позволяет строить формальные спецификации для модулей со сложной управляющей логикой, которая допускает блокировки конвейера.
2. Разработан метод избыточного описания тестового покрытия с помощью тестовых ситуаций и зависимостей, который позволяет декомпозировать сложную структуру целей тестирования на набор относительно простых описаний, определяемых для отдельных операций или пар операций. Разработанный метод позволяет сократить трудоемкость разработки тестов и увеличить эффективность их поддержки.
3. Разработан метод генерации тестовой последовательности на основе контрактной спецификации конвейера и описания тестового покрытия, состоящий в автоматическом построении обобщенной автоматной модели конвейера и обходе ее графа состояний. Метод позволяет

сократить трудоемкость разработки тестов и увеличить устойчивость тестов к изменениям реализации за счет того, что отпадает необходимость в ручной разработке модели теста.

4. Разработан метод генерации тестовых программ на основе формальных спецификаций системы команд микропроцессора и описания тестового покрытия, состоящий в целенаправленном переборе всевозможных сочетаний тестовых ситуаций и зависимостей для последовательностей инструкций ограниченной длины и использовании дополнительных эвристик для сокращения размера тестов. Метод позволяет генерировать тестовые программы со встроенными проверками, что делает возможным их использование для проверки готовых микросхем.

На основе предложенных методов автоматизации модульного и системного тестирования были реализованы инструментальные средства имитационного тестирования микропроцессоров. Результаты апробации методов и инструментов показывают их применимость для сложных промышленных микропроцессоров. Предложенные методы автоматизации модульного и системного тестирования микропроцессоров с конвейерной архитектурой соответствуют поставленным в работе целям.

Работы автора по теме диссертации

1. Иванников В.П., Камкин А.С., Косачев А.С., Кулямин В.В., Петренко А.К. Использование контрактных спецификаций для представления требований и функционального тестирования моделей аппаратуры // Программирование, № 5, 2007. С. 47–61.
2. Иванников В.П., Камкин А.С., Кулямин В.В., Петренко А.К. Применение технологии UniTESK для функционального тестирования моделей аппаратного обеспечения. Препринт № 8. М.: ИСП РАН, 2005. 26 с.
3. Камкин А.С. Тестирование в условиях неполной информации. Подход к разработке спецификаций и генерации тестов // Труды ИСП РАН, т. 10. М., 2006. С. 143–166.

4. Kamkin A. The UniTESK Approach to Specification-Based Validation of Hardware Designs // Proceedings of ISoLA 2006: The 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, 2006. P. 52–58.
5. Камкин А.С. Использование формальных спецификаций для функционального тестирования модулей микропроцессоров // Тезисы конференции «Микроэлектроника и информатика – 2007», 2007. С. 188.
6. Kamkin A. Contract Specification of Pipelined Designs: Application to Testbench Automation // Proceedings of SYRCoSE 2007: The 1st Spring Young Researchers Colloquium on Software Engineering, v. 2. 2007. P. 7–13.
7. Аряшев С.И., Камкин А.С., Рогаткин Б.Ю. Тестирование RTL-моделей аппаратуры с помощью технологии UniTESK на примере блока преобразования адресов микропроцессора // Сборник научных трудов конференции «Электроника, микро- и нанoeлектроника – 2007», 2007. С. 183–187.
8. Камкин А.С. Использование контрактных спецификаций для автоматизации функционального тестирования моделей аппаратного обеспечения // Труды ИСП РАН, т. 13, ч. 1. М., 2007. С. 123–142.
9. Kamkin A. Testbench Automation for Pipelined Designs Based on Contract Specifications // Proceedings of EWDTs 2007: The 5th East-West Design & Test Symposium, 2007. P. 348–353.
10. Камкин А.С., Чупилко М.М. Тестирование модулей арифметики над числами с плавающей точкой микропроцессоров на соответствие стандарту IEEE 754 // Труды ИСП РАН, т. 14, ч. 2. М., 2008. С. 7–22.
11. Камкин А.С. Генерация тестовых программ для микропроцессоров // Труды ИСП РАН, т. 14, ч. 2. М., 2008. С. 23–63.

12. Chupilko M., Kamkin A., Vorobyev D. Methodology and Experience of Simulation-Based Verification of Microprocessor Units Based on Cycle-Accurate Contract Specifications // Proceedings of SYRCoSE 2008: The 2nd Spring Young Researchers Colloquium on Software Engineering, v. 2. 2008. P. 25–31.
13. Kamkin A. Coverage-Directed Verification of Microprocessor Units Based on Cycle-Accurate Contract Specifications // Proceedings of EWDTTS 2008: The 6th East-West Design & Test Symposium, 2008. P. 84–87.
14. Камкин А.С. Комбинаторная генерация тестовых программ для микропроцессоров на основе моделей. Препринт № 21. М.: ИСП РАН, 2008. 18 с.