

ОТЗЫВ ОФИЦИАЛЬНОГО ОППОНЕНТА

на диссертационную работу

Татарникова Андрея Дмитриевича

«Автоматизация конструирования генераторов тестовых программ для микропроцессоров на основе формальных спецификаций»,
представленную на соискание ученой степени кандидата технических наук по специальности 05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Рассматриваемая диссертационная работа посвящена актуальной теме – исследованиям в области функциональной верификации микропроцессоров. Под функциональной верификацией понимается комплекс мер, нацеленный на обеспечение корректности проектируемых микропроцессоров. Одним из основных подходов к решению этой задачи является исполнение тестовых программ на проектной модели микропроцессора и анализ полученных трасс исполнения.

Тестовые программы создаются автоматически при помощи специальных инструментов, известных как генераторы тестовых программ. Большинство подобных инструментов разработано для конкретных систем команд и реализует ограниченные наборы техник построения тестов. Поскольку системы команд и техники генерации эволюционируют, перед разработчиками генераторов тестовых программ стоит задача расширения их возможностей. Решение этой задачи требует значительных усилий, так как описание системы команд зачастую неотделимо от логики техник генерации и для поддержки новых возможностей требуются существенные изменения в реализации инструмента.

Таким образом, актуальной темой исследований является разработка методов, которые бы позволили создавать расширяемые генераторы тестовых программ, легко адаптируемые к новым системам команд.

В диссертационной работе А.Д. Татарникова получены следующие основные результаты:

1. Предложен метод автоматизации конструирования генераторов тестовых программ для микропроцессоров на основе формальных спецификаций.

Применение данного метода сводит разработку генератора тестовых программ к написанию формальных спецификаций системы команд микропроцессора.

2. Разработан предметно-ориентированный язык для описания шаблонов тестовых программ, позволяющий задавать сценарии тестирования на основе различных техник генерации. Данный язык адаптируется к системе команд, описанной при помощи формальных спецификаций, и допускает расширение пользовательскими конструкциями.
3. Предложена архитектура генераторов тестовых программ для микропроцессоров, позволяющая интегрировать несколько различных техник генерации и допускающая расширение функциональности компонентами, реализующими пользовательские техники генерации.
4. Разработан программный инструмент MicroTESK, который использует предложенный автором метод для конструирования генераторов тестовых программ. Сконструированные генераторы имеют предложенную автором архитектуру и используют предложенный язык описания шаблонов тестовых программ. Генераторы тестовых программ для систем команд MIPS64 и ARMv8, созданные при помощи инструмента MicroTESK, применяются в промышленных проектах по верификации микропроцессоров.

Все основные результаты диссертации опубликованы и апробированы на семинарах и конференциях. Автореферат полно и правильно отражает содержание диссертации.

Диссертация состоит из введения, четырех глав, заключения и приложения.

Введение раскрывает актуальность темы диссертационной работы, формулируют ее цели и задачи, обосновывает ее теоретическую и практическую значимость.

Глава 1 содержит обзор существующих техник и инструментов генерации тестовых программ для функциональной верификации микропроцессоров. На основе анализа их возможностей и присущих им ограничений автором делается вывод об актуальности разработки методов, которые позволили бы минимизировать трудозатраты на разработку генераторов тестовых программ и обеспечили бы возможность интеграции разнообразных техник генерации.

Глава 2 содержит описание предлагаемого автором метода автоматизации конструирования генераторов тестовых программ на основе формальных спецификаций, которые включают в себя описание системы команд на языке nML и описание логики подсистемы памяти на языке mmuSL. Также в главе

описывается предметно-ориентированный язык, основанный на Ruby, который автор предлагает использовать для описания тестовых сценариев. Помимо этого глава содержит описание предлагаемой автором архитектуры генераторов тестовых программ, которая позволяет интегрировать разные техники генерации и допускает расширение набора поддерживаемых техник.

Глава 3 содержит описание инструмента MicroTESK, реализующего предлагаемый автором метод автоматизации конструирования генераторов тестовых программ. В главе рассматривается архитектура инструмента и конструируемых с его помощью генераторов тестовых программ. Отдельное внимание уделено расширяемости инструмента.

Глава 4 описывает результаты применения инструмента MicroTESK для создания генераторов тестовых программ для систем команд MIPS64, ARMv8, PowerPC и RISC-V. В главе приводятся характеристики перечисленных систем команд и сведения о трудоемкости разработки их формальных спецификаций. Также в главе перечисляются возможности сконструированных генераторов тестовых программ. На основе этого делается вывод о применимости полученных генераторов для использования в промышленных проектах по верификации микропроцессоров.

Заключение содержит перечисление основных результатов работы.

Полученные результаты позволяют упростить разработку генераторов тестовых программ для микропроцессоров, сведя ее к написанию формальных спецификаций на основе руководства по системе команд. Предложенный язык описания шаблонов тестовых программ позволяет описывать разнообразные тестовые сценарии для любых систем команд. Расширяемая архитектура генераторов тестовых программ позволяет интегрировать разные техники генерации в рамках единого инструмента. В совокупности предложенный подход, его реализация и ее успешное применение в промышленных проектах, определяет высокую значимость для науки и производства полученных автором диссертации результатов.

К числу недостатков работы можно отнести следующие:

1. В работе широко используется понятие архитектуры процессора и архитектурной независимости (инструментов, методик и т. п.). При этом не делается попытки как-то описать множество поддерживаемых архитектур, или, другими словами, уточнить понятие архитектуры процессора применительно к данной работе. Очевидно, что все рассмотренные в Главе 4 архитектуры в это подразумеваемое множество входят. Так же очевидно, что всегда может появиться процессор, система команд которого окажется настолько самобытной, что понятие «любой архитектуры» окажется к нему

- не применимым. Сказанное тем более важно, что, согласно стр. 47, речь идет не только о системе команд как таковой, но и о микроархитектуре, то есть о скрытых внутренних состояниях процессора, контроль над которыми не доступен программисту непосредственно. Для сколько-нибудь основательного покрытия этих состояний тестированием просто необходимо иметь модель того, что мы, собственно, тестируем. Следующие два замечания являются уточнением и развитием настоящего замечания, то есть также посвящены проблеме покрытия тестированием внутренних состояний процессора, не имеющих прямого отражения в системе команд.
2. В работе ничего не говорится о техниках генерации тестовых программ, нацеленных на проверку конвейерного выполнения команд. Конвейеры в процессорах общего назначения относятся к той самой микроархитектуре, о которой шла речь в предыдущем замечании. Хотелось бы увидеть каким образом компоненты, реализующие подобные техники, могут быть интегрированы в сконструированные при помощи предлагаемого автором метода генераторы тестовых программ.
 3. Автор утверждает, что предложенный метод позволяет создавать генераторы тестовых программ для любых микропроцессорных архитектур. Однако в работе представлены результаты практического применения метода только для архитектур семейства RISC. Хотелось бы увидеть оценку его применимости для семейств CISC и VLIW. Для последних характерна исключительная сложность и обилие опять же тех самых внутренних, «микроархитектурных» состояний, которые не доступны программисту прямо и непосредственно, но которые надо тестировать.
 4. В работе описываются возможности предложенного автором языка описания шаблонов тестовых программ, но отсутствуют примеры готовых шаблонов и тестовых программ, сгенерированных на их основе. Примеры, иллюстрирующие результаты работы инструмента, упростили бы читателю понимание предлагаемых автором методов.
 5. Хотелось бы увидеть в работе оценку уровня покрытия, которого можно достичь в результате применения генераторов тестовых программ, сконструированных при помощи предложенного автором подхода.
- Указанные недостатки не умаляют ценности и качества проведенной диссертантом работы.

Диссертация является законченным научным исследованием и удовлетворяет всем требованиям, предъявляемым ВАК к кандидатским диссертациям, а ее автор, А.Д. Татарников, заслуживает присуждения ему ученой

степени кандидата технических наук по специальности 05.13.11 – математическое
и программное обеспечение вычислительных машин, комплексов и
компьютерных сетей.

Официальный оппонент,
заведующий сектором ИПМ им. М.В. Келды
доктор физико-математических наук

А.О. Лацис

Подпись А.О. Лациса удостоверяю,
Ученый секретарь ИПМ им. М.В. Келдыша
кандидат физико-математических наук

А.И. Маслов

29 сентября 2017г.