

На правах рукописи

Чибисов Петр Александрович

**ВСТРЕЧНОЕ ТЕСТИРОВАНИЕ  
ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ МИКРОПРОЦЕССОРОВ**

Специальность 05.13.11 –

математическое и программное обеспечение вычислительных машин,  
комплексов и компьютерных сетей

**АВТОРЕФЕРАТ**  
диссертации на соискание ученой степени  
кандидата технических наук

Москва  
2013

Работа выполнена в Федеральном государственном бюджетном учреждении науки Научно-исследовательском институте системных исследований РАН.

Научный руководитель: Бобков Сергей Геннадьевич,  
доктор технических наук, профессор

Официальные оппоненты: Петросянц Константин Орестович, доктор технических наук, профессор, заведующий кафедрой электроники и наноэлектроники Федерального государственного автономного образовательного учреждения высшего профессионального образования «Национальный исследовательский университет «Высшая школа экономики»

Краснюк Андрей Анатольевич, кандидат технических наук, доцент, заместитель заведующего кафедрой электроники Национального исследовательского ядерного университета «МИФИ» (НИЯУ МИФИ)

Ведущая организация: Открытое акционерное общество «Центральный научно-исследовательский институт автоматики и гидравлики»

Защита диссертации состоится 5 декабря 2013 г. в 15:00 на заседании диссертационного совета Д 002.087.01 при Федеральном государственном бюджетном учреждении науки Институте системного программирования Российской академии наук по адресу: 109004, Москва, ул. Александра Солженицына, д.25, конференц-зал (комн. 110).

С диссертацией можно ознакомиться в библиотеке Федерального государственного бюджетного учреждения науки Института системного программирования Российской академии наук.

Автореферат разослан 1 ноября 2013 г.

Ученый секретарь  
диссертационного совета,  
кандидат физ.-мат. наук



/Прохоров С.П./

## Общая характеристика работы

### Актуальность темы

Число транзисторов в современных высокопроизводительных микропроцессорах достигает нескольких миллиардов. Большие объемы СБИС неизбежно приводят к наличию ошибок в микропроцессорах. Непрерывная погоня за повышением производительности и требование сохранения программной совместимости приводит к неизбежному усложнению микропроцессоров и, соответственно, к появлению всё новых и новых ошибок. В микропроцессоре Intel Core i7-600 на декабрь 2011 г. найдено 123 ошибки, из которых не планируется исправлять 118, неизвестны пути обхода 62 ошибок. Таким образом, задача повышения эффективности тестирования микропроцессоров на этапе их создания является актуальной задачей, сложность которой растет по мере усложнения микропроцессоров. При этом функциональная верификация сложна с одной стороны тем, что она не может быть выполнена с гарантией отсутствия ошибок в проекте ни за какое осмысленное время, с другой стороны, цена имеющейся ошибки в уже изготовленной СБИС может превышать стоимость разработки и изготовления самой СБИС. Особенно это справедливо в тех случаях, когда речь идет об ответственных применениях вычислительных и управляющих модулей, в основе которых лежит разработанный микропроцессор, а также в случаях, требующих повышенной надежности функционирования микропроцессора, когда стоимость аппаратуры превышает стоимость всего цикла разработки микропроцессора.

Множество разнообразных теоретических и практических методов было предложено и применяется как в академических исследовательских работах, так и в крупных производственных центрах. Однако, проводимые академические исследования редко достигают масштабов проектов индустрии (здесь и далее «проектом» мы называем высокоуровневую

спецификацию логики микропроцессора – High Level Design, HDL - например, на языке Verilog). В публикациях ведущих мировых лидеров отрасли просматривается тенденция не разглашать информацию о текущих методиках и средствах, либо распространять информацию с задержкой, что объясняется соображениями конкуренции.

Особое внимание при проектировании микропроцессоров уделяется тестированию их RTL-модели, так как: 1) поиск ошибок ведется параллельно с продвижением разработки, 2) стоимость ошибки на этом этапе проектирования минимальна.

Все методы верификации моделей на этапе RTL-проектирования можно разделить на формальные и динамические. Формальные методы позволяют математически доказать правильность работы какого-либо блока микропроцессора, но являются крайне трудоемкими. Например, при разработке микропроцессора Pentium 4 фирмой Intel была проведена формальная верификация модуля вещественной арифметики (FPU), модуля декодирования инструкций, а также модуля динамического планирования потока инструкций. При этом на эту работу было затрачено 60 человеко-лет.

Динамические методы верификации (имитационное тестирование) моделей микропроцессоров широко применяются для верификации RTL-моделей. Как правило, качество (полноту) тестирования оценивают по величине покрытия кода разрабатываемой RTL-модели (так называемое структурное покрытие). Однако даже 100% покрытия не гарантирует отсутствие ошибок.

При проектировании микропроцессоров специального назначения процесс разработки аппаратной и программной частей происходит, как правило, одновременно. При этом алгоритмы программ могут оптимизироваться под аппаратуру для повышения реальной (в отличие от пиковой) производительности, и, наоборот, создаваемая аппаратура учитывает назначение аппаратно-программного комплекса и особенности возможных программ пользователей. Такой подход известен как «встречная

оптимизация» аппаратно-программных комплексов (предложен в НИИСИ РАН). Так как уже на начальных этапах проектирования, когда цена исправления ошибки в модели микропроцессора минимальна, уже известны и доступны программы и приложения пользователя, полезно использовать знания о коде выделенных приложений для повышения качества тестирования. В связи с этим представляется, что разработка метода функциональной верификации RTL-моделей микропроцессоров, учитывающего результаты анализа функционального тестового покрытия выделенных приложений (метод «встречного» тестирования), является актуальной темой исследования.

### **Цель и задачи исследования**

Целью диссертационной работы является разработка метода встречного тестирования микропроцессоров, их моделей и прототипов для повышения эффективности верификации и системного тестирования микропроцессоров ответственного применения и их RTL-моделей. Для достижения указанной цели в диссертационной работе поставлены и решены лично автором следующие основные задачи:

1. провести анализ существующих методов и средств функциональной верификации и тестирования современных микропроцессоров;
2. разработать методику тестирования микропроцессоров и их RTL-моделей существующим пользовательским программным обеспечением (ПО) под операционной системой (ОС) Linux;
3. разработать метод встречного тестирования для улучшения качества тестирования проектируемых микропроцессоров ответственного применения для выделенного класса пользовательских задач и повышения производительности микропроцессоров.

## **Научная новизна исследования**

Научной новизной обладают следующие результаты исследования:

1. методика запуска кода готовых приложений под ОС, а также загрузки самой ОС на ранних стадиях проектирования RTL-модели;
2. метод тестирования микропроцессора с привлечением контрольных задач от потенциальных пользователей («встречное» тестирование);
3. методика выбора набора тестов, учитывающая специфику определенного класса задач.

## **Практическая значимость**

Разработанные автором методы и методики использованы при проектировании в НИИСИ РАН микропроцессоров 1890ВМ5Ф, 1890ВМ6Я, а также успешно используются для выполнения ОКР «Процессор-5» и «Процессор-6» для создания микропроцессоров 1890ВМ8Я и 1890ВМ9Я. Благодаря комплексному применению созданных средств тестирования в серийно выпускаемых вычислительных системах на базе микропроцессоров 1890ВМ5Ф и 1890ВМ6Я не выявлено ни одной ошибки в выделенных областях применения. Результаты работы могут использоваться в НИИ системных исследований РАН, ЗАО «МЦСТ», ФГУП НИИМА «Прогресс», а также в других научных и промышленных организациях, занимающихся разработкой высокопроизводительных микропроцессоров.

## **Апробация**

Основные положения диссертации доложены на следующих конференциях и семинарах:

- VI, VII, IX и XIII научно-технические конференции "Электроника, Микро- и Нанозлектроника" (г. Нижний Новгород, 2004, 2005, 2007, 2011 гг.);
- Проблемы разработки перспективных микро- и нанозлектронных систем (МЭС-2005, 2010, 2012, г. Истра Московской области);

- Семинарах НИИ системных исследований РАН (г. Москва, 2006, 2011, 2013 гг.);
- Семинаре Института системного программирования РАН (г.Москва, 2013г.).

## **Публикации**

По теме диссертации автором опубликованы 14 работ, в том числе 6 в рецензируемых изданиях, входящих в перечень рекомендованных ВАК, полно отражающих основные результаты диссертации.

## **Структура и объем работы**

Работа состоит из введения, четырех глав, заключения и списка литературы (114 наименований). Основной текст диссертации (без приложений и списка литературы) занимает 163 страницы.

## **Содержание работы**

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

**В первой главе** приведен обзор и анализ существующих в настоящее время методов функциональной верификации микропроцессоров. Описываются достоинства и ограничения различных методов. Также даются определения основных терминов, связанных с верификацией и тестированием микропроцессоров. Основное внимание уделяется динамическому имитационному моделированию на RTL-уровне (*pre-silicon verification*), а также системному функциональному тестированию, как основным методам проверки корректности работы микропроцессоров.

В разделе 1.2 рассматриваются методы, применяемые для верификации микропроцессоров с использованием ПЛИС-прототипа. В разделе 1.3 приводится обзор методов тестирования СБИС микропроцессоров (*post-*

*silicon verification*) и их тестовых кристаллов. В разделе 1.4 рассматриваются метрики тестового (функционального) покрытия, применяемые при верификации моделей микропроцессоров.

В конце главы приводится анализ текущего состояния изучаемой области (верификация моделей микропроцессоров повышенной надежности для ответственных применений), а также уточняются цели и задачи диссертационной работы.

**Во второй главе** рассмотрена предлагаемая автором методика тестирования микропроцессоров и их моделей существующим программным обеспечением под управлением операционной системы на разных уровнях абстракции модели. Особое внимание уделяется тестированию RTL-моделей (модель на уровне регистровых передач) микропроцессоров.

*Методика запуска кода готовых приложений пользователя под ОС*, а также загрузки самой ОС на ранних стадиях проектирования RTL-модели микропроцессора для верификации проекта включает в себя следующие действия:

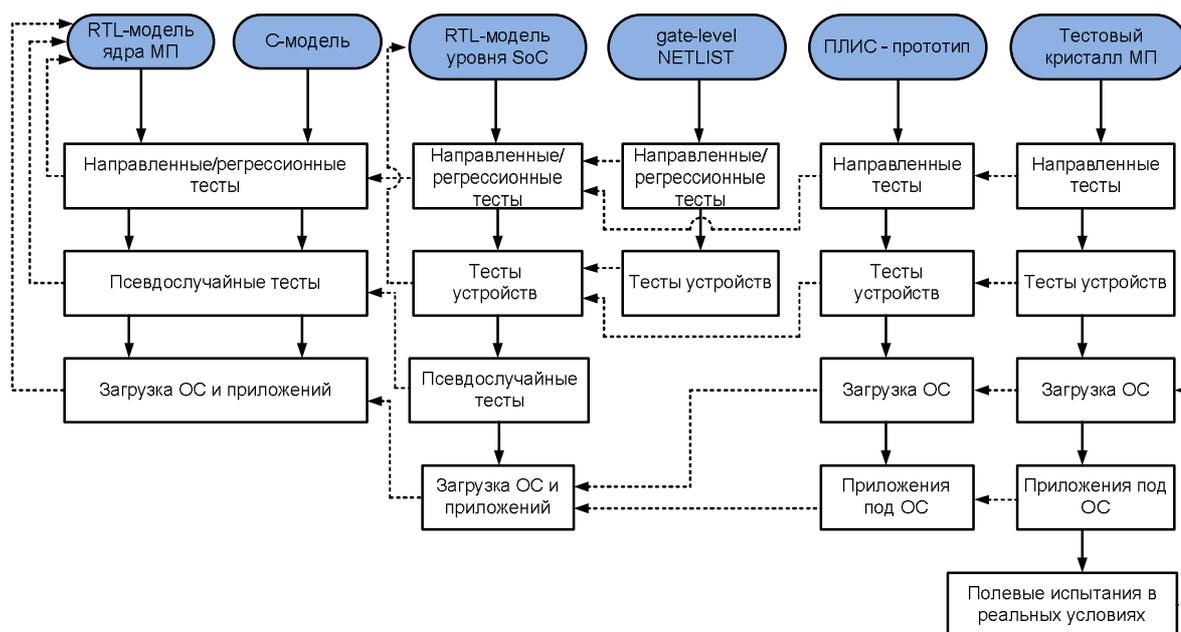
1. создание и отладка образцового (эталонного) эмулятора (C-модель);
2. отладка ядра микропроцессора и его окружения с помощью имеющихся направленных тестов, а также с помощью тестов, создаваемых с помощью генератора псевдослучайных тестов;
3. портирование ядра ОС Linux на данную процессорную архитектуру;
4. добавление в образцовый эмулятор возможности сохранения состояния эмулируемого микропроцессора в требуемый момент времени в виде файла (файлов), а также обратной функции восстановления состояния из файла (файлов) состояния с последующим возобновлением работы эмулятора с данного момента времени;
5. разработка управляющих программ, позволяющих автоматически получать с помощью эмулятора «срезы» состояния в произвольные моменты времени с последующим конвертированием формата файлов состояния (если требуется) для применения файлов в качестве исходных для загрузки RTL-

модели. Это необходимо для обеспечения возможности запуска ОС Linux на RTL-модели с произвольного момента времени загрузки, а также для автоматизации всего подхода, когда различные участки общей трассы ОС выполняются на модели с использованием различных вычислительных ресурсов (таких как персональные ЭВМ сотрудников или многопроцессорные вычислительные кластеры);

6. автоматизированный запуск ОС Linux с выбранными тестовыми приложениями с сохранением состояния модели микропроцессора через равные промежутки времени с последующим созданием лог-файлов и сравнением их на одном узле вычислительного кластера с целью выявления ошибок или минимизации тестового набора.

Так как скорость выполнения программ на RTL-модели микропроцессора крайне мала, для повышения эффективности тестирования целесообразно использовать ПЛИС-прототип, а также тестовые кристаллы микропроцессоров. В работе анализируется взаимосвязь объектов тестирования, а также рассматривается степень применимости обсуждаемой методики к каждому из них в контексте общего плана верификации.

На рис.1 рассмотренная методика тестирования RTL-модели, ПЛИС-прототипа и тестового кристалла представлена в графической форме. Сплошными линиями показаны переходы между уровнями тестирования для разных уровней абстракции проекта, пунктирными – переходы в случае нахождения ошибки. Методика охватывает практически все уровни абстракции представления модели; при этом, в случае нахождения ошибки, например, в ПЛИС-прототипе, требуется воспроизвести и исправить ее на более низком уровне, после чего повторить цикл тестирования. Под полевыми испытаниями понимается изучение стабильности работы СБИС микропроцессора вне лабораторных (нормальных) условий, то есть в полном рабочем диапазоне температур, в составе многопроцессорных кластеров под продолжительной вычислительной нагрузкой и тому подобное.



**Рис.1. Методика тестирования микропроцессора на разных стадиях (уровнях) разработки кодом приложений пользователя под ОС**

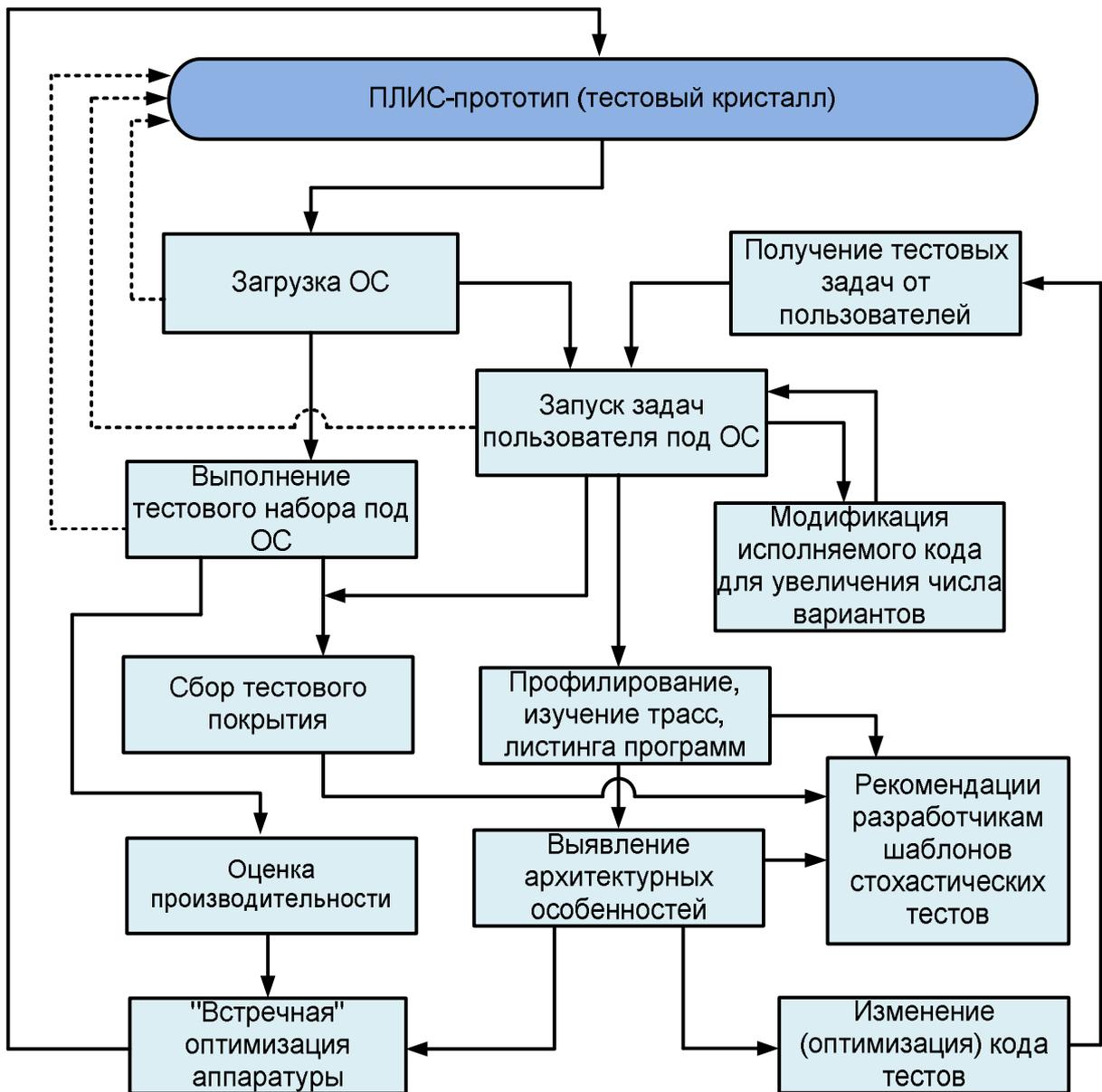
Подбор тестов для данной методики осуществляется среди свободно распространяемых программ с открытым исходным кодом. Требуется определить конечный набор тестов, который является достаточным для покрытия выбранных функциональных метрик с заданным уровнем покрытия. Для изучения покрытия тестового набора применяются как стандартные метрики, основанные, в первую очередь, на структурном покрытии HDL-кода RTL-модели, так и функциональные метрики, основанные на покрытии задаваемых инженером-верификатором пространств комбинаций инструкций и их аргументов.

Исходный тестовый набор строится из доступных тестов разнообразного назначения и функциональности, а также из реальных пользовательских приложений, которые предполагается запускать на разрабатываемой системе на кристалле. Так как проектируемая система на кристалле создается совместно (и одновременно) с программным обеспечением для нее, то используя подход ко-верификации, можно заранее получить представление о предполагаемой сложности и функциональности кода. Особенно это справедливо для специальных систем ответственного назначения.

В конце главы описывается предлагаемая автором методика поиска и локализации ошибок в ПЛИС-прототипе (тестовом кристалле).

В **третьей главе** предложен *метод тестирования микропроцессора с привлечением контрольных (тестовых) задач от потенциальных пользователей («встречное» тестирование)*. Метод является развитием подхода *ко-верификации* для микропроцессоров повышенной надежности для ответственных применений. Рассматриваемый в главе комплексный подход к системной верификации учитывает информацию о пользовательских приложениях, разрабатываемых для данной системы, а также принимает во внимание результаты измерения покрытия кода задач пользователя с помощью разрабатываемых метрик функционального покрытия.

Таким образом, метод встречного тестирования микропроцессоров, их моделей и прототипов требует включения в рассмотренный выше маршрут тестирования дополнительных этапов, связанных с получением тестовых (контрольных) задач от потенциальных пользователей, компиляцией и запуском этих программ на прототипе процессора с последующей оптимизацией аппаратной и программной частей будущего комплекса для повышения производительности микропроцессора, а также для повышения качества тестирования, что схематически отражено на рис.2.



**Рис.2. Метод тестирования микропроцессора с привлечением контрольных (тестовых) задач от потенциальных пользователей («встречное» тестирование)**

Рассмотрим предлагаемый метод встречного тестирования. Пусть  $A$  - множество всех инструкций микропроцессора, определяемое его спецификацией (порядка 100 инструкций). Предположим, что буфер инструкций исследуемого микропроцессора считывает из памяти или кэш-памяти инструкций по 4 инструкции (окно выполнения), выдаваемые на

декодирование и дальнейшее суперскалярное исполнение в конвейере. Протестировать все возможные комбинации инструкций из всего набора  $A$ , очевидно, не представляется возможным (да и не требуется) за обозримое время, поэтому проведем редуцирование множества  $A$  к множеству  $B$ , опустив из рассмотрения инструкции, не создаваемые (либо создаваемые крайне редко в строго определенных ситуациях, поддающихся переборному тестированию) компиляторами программ, написанных на языках программирования высокого уровня (речь идет в основном о C, C++ и Fortran). Удаленные из рассмотрения инструкции и их комбинации тестируются обычными методами.

Проведем факторизацию групп инструкций множества  $B$  по классам эквивалентности, основываясь на спецификации микропроцессора, а также используя рекомендации группы разработчиков RTL-модели. Результатом факторизации будет множество  $C$  классов инструкций, в которое, например, будут входить 6-7 классов (такие как класс арифметических команд или класс команд ветвления), каждый из которых состоит из 1-4 групп инструкций, обозначенных как эквивалентные (в таком масштабе рассмотрения, инструкция сложения ADD будет эквивалентна инструкции вычитания SUB). Объединим получившиеся группы в множество  $D$ .

В итоге, наша исследовательская (отправная) модель пространства покрытия будет состоять из всех возможных комбинаций по 4 инструкции, каждая из которых берется из множества  $D$ . Такие комбинации являются, по сути, элементарными тестовыми воздействиями для модели микропроцессора, причем все такие комбинации регистрируются, используя тестовое покрытие, задаваемое на уровне инструкций. Так как у каждой инструкции может быть множество явных и неявных аргументов и параметров (так, для инструкции LOAD - это сегмент виртуальной памяти, адрес, смещение, разрядность загружаемых данных, попадание либо промах в кэш-память L1, L2, политика обращения в кэш-память и другие), то с их учетом суммарное количество подлежащих тестированию комбинаций в

одном окне выполнения инструкций микропроцессора будет слишком велико. Для ограничения сверху количества тестовых ситуаций и уменьшения избыточности в рамках встречного тестирования множество комбинаций параметризуется (уточняется) с учетом результатов, полученных в результате анализа листингов и трасс задач пользователя под ОС. Затем, исходя из заданных ограничений, создается метрика функционального покрытия, основанная на группах инструкций из рассмотренного выше множества  $D$  с учетом выбранных аргументов и параметров.

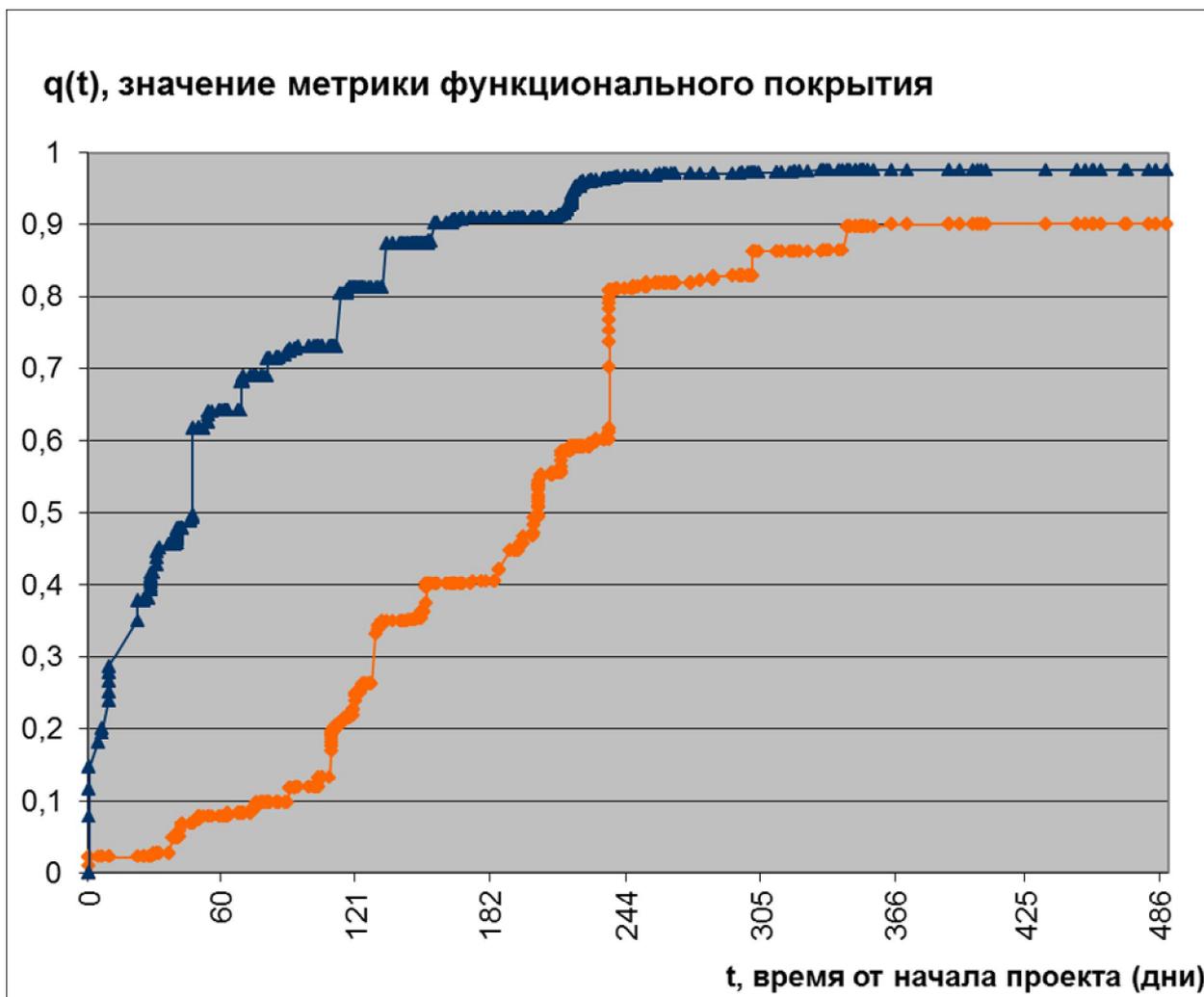
Метрика задаётся следующим образом. Обозначим  $\Omega$  - пространство всех допустимых элементарных тестовых воздействий в рамках нашей исследовательской модели. Пусть при прохождении набора тестов  $T$  на тестируемую модель подаётся совокупность тестовых воздействий  $\Omega_T$ . Тогда значение заданной этим пространством метрики будет составлять  $|\Omega_T| / |\Omega|$ . То есть мы получили функцию, ставящую в соответствие набору тестов число от 0 до 1. Тогда за один из критериев завершённости тестирования можно взять достижение этой метрикой некоторого порогового значения, например, 0,95 или 0,99.

С учетом созданной метрики функционального покрытия создаются шаблоны для генерации стохастических (псевдослучайных) тестов, в результате запуска которых могут быть найдены новые ошибки. В рамках одной выбранной модели покрытия накапливаются данные по функциональному покрытию, использующиеся для замыкания первого контура (обозначен «I» на рис.3) обратной связи (по измеряемым значениям покрытия). Данные по функциональному покрытию, получаемые непосредственно в процессе выполнения задач пользователя, также учитываются.



**Рис.3. Получение метрики функционального покрытия для «встречного» тестирования модели микропроцессора**

Как отмечалось выше, вводится и второй путь обратной связи, замыкающийся через расширение и уточнение модели функционального покрытия по результатам интерпретации результатов измерений (обозначен «II» на рис.3).



**Рис.4. Значение одной из метрик функционального покрытия в течение времени**

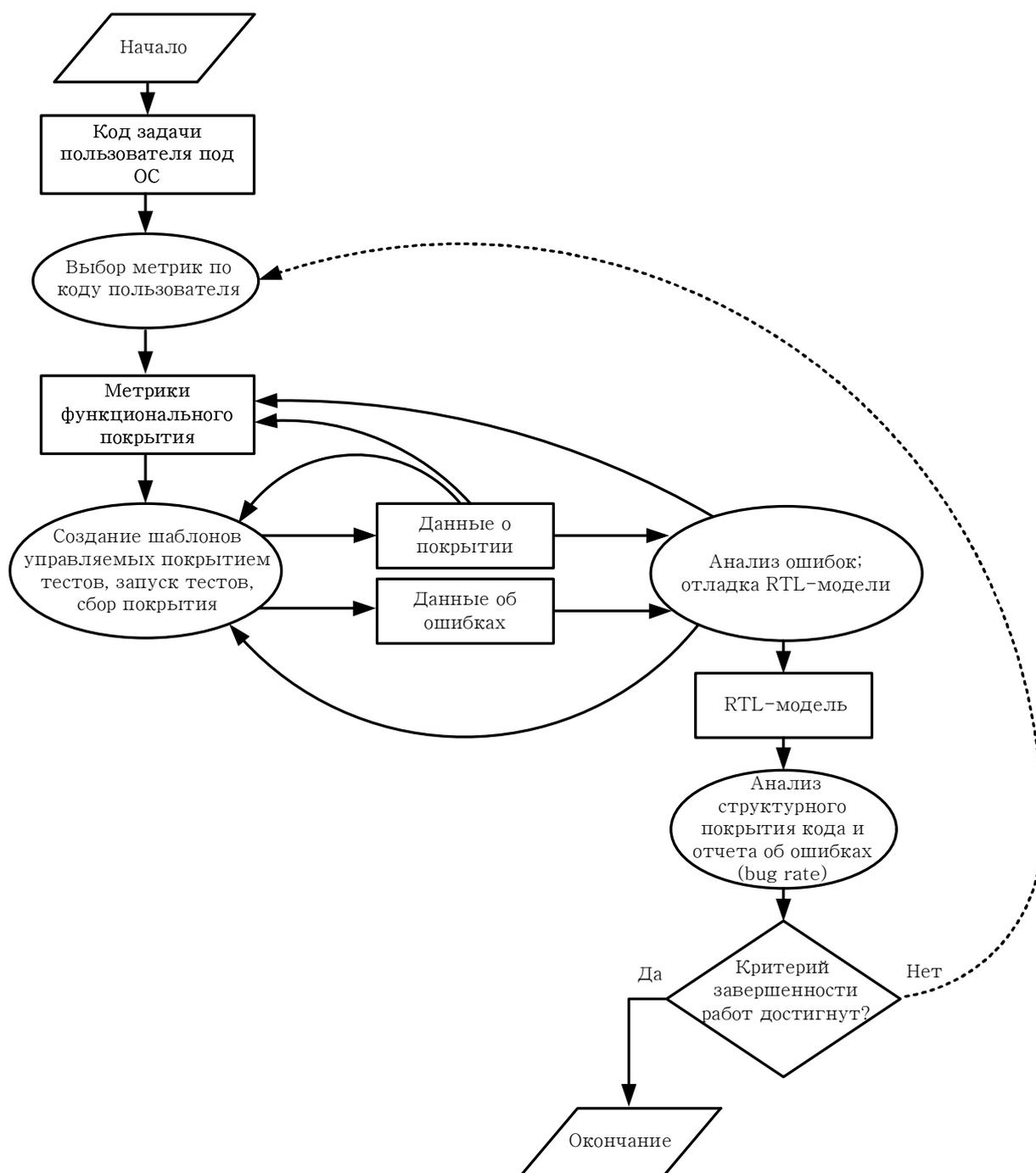
Результаты применения метода встречного тестирования на практике приведены на рис.4 (▲ (черным цветом) - тестирование с учетом рассматриваемого подхода (встречное тестирование), серым - без учета подхода встречного тестирования). Данные, представленные на рис.4, свидетельствуют о значительном сокращении времени, требуемого для достижения заданного значения метрики функционального покрытия (более чем в два раза), а также о повышении достижимого функционального покрытия при применении предлагаемого подхода.

Процесс (методика) выбора набора тестов с учетом специфики определенного класса задач пользователя включает в себя следующие этапы:

1. получение информации о назначении комплекса;

2. классификация задач, решаемых с помощью комплекса в целом;
3. оценка разделения решаемых системой на кристалле задач на аппаратные и программные;
4. изучение спецификации на проектируемую аппаратуру;
5. получение исходных кодов ОС и приложений пользователя под ОС, выбор инструментальных средств (компиляторов, кросс-компиляторов, отладчиков и так далее);
6. сборка тестов, изучение кода (листинги, запуск на эталонном эмуляторе с получением фрагментов трасс выполнения, профилирование);
7. выявление особенностей данных приложений, тщательный анализ потенциальных фрагментов с повышенной вероятностью возникновения ошибок, изучение ошибок в RTL-модели;
8. построение функциональных метрик покрытия на основе наборов инструкций микропроцессора (с учетом классов эквивалентности инструкций) по представленной на рис.5 схеме;
9. пополнение базы тестов имеющимися в свободном доступе программами со схожим назначением и характеристиками для использования их в качестве дополнительных тестов, а также псевдослучайными тестами для повышения покрытия по заданным метрикам;
10. детализация метрик, разработка детального тестового плана.

Имеются работы, указывающие на то, что распределение Рэлея хорошо описывает ожидаемый уровень поступления ошибок с течением времени проектирования, суммарное количество ожидаемых ошибок, а также дает прогноз времени окончания работ по верификации. Предшествующие работы были посвящены анализу динамики выявления ошибок в ПО. Автор провел анализ применимости распределения Рэлея для процессов верификации аппаратуры.



**Рис.5. Схема процедуры построения функциональных метрик**

Функция плотности распределения вероятности возникновения ошибок в RTL-модели (или суммарно на разных уровнях абстракции) имеет вид:

$$f(t) = \frac{6K}{t_d^2} t e^{-3\frac{t^2}{t_d^2}},$$

где  $K$  – коэффициент, определяющий суммарное, найденное за время работ по верификации количество ошибок, оценивается сверху эмпирическим путем и для достаточно опытных групп разработчиков составляет  $K \approx \frac{\text{количество строк кода модели}}{150}$ ;

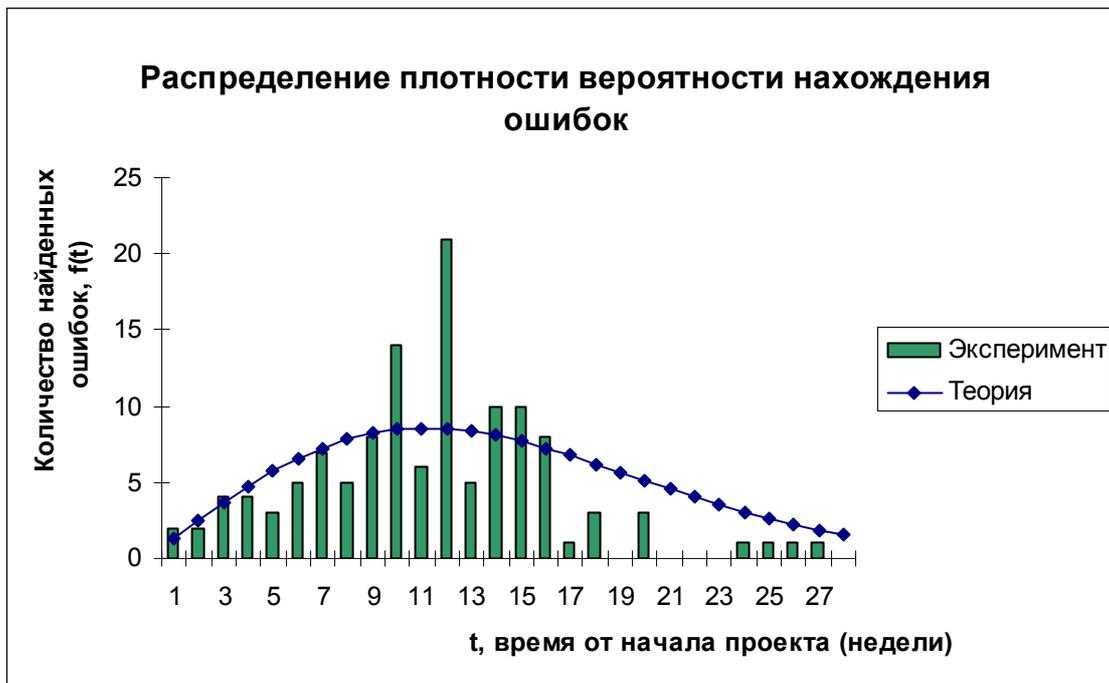
$t_d$  – общее время длительности проекта, то есть такое время, за которое выявлено 95% всех ошибок.

Интегральная функция распределения числа найденных ошибок по неделям определяется следующей формулой:

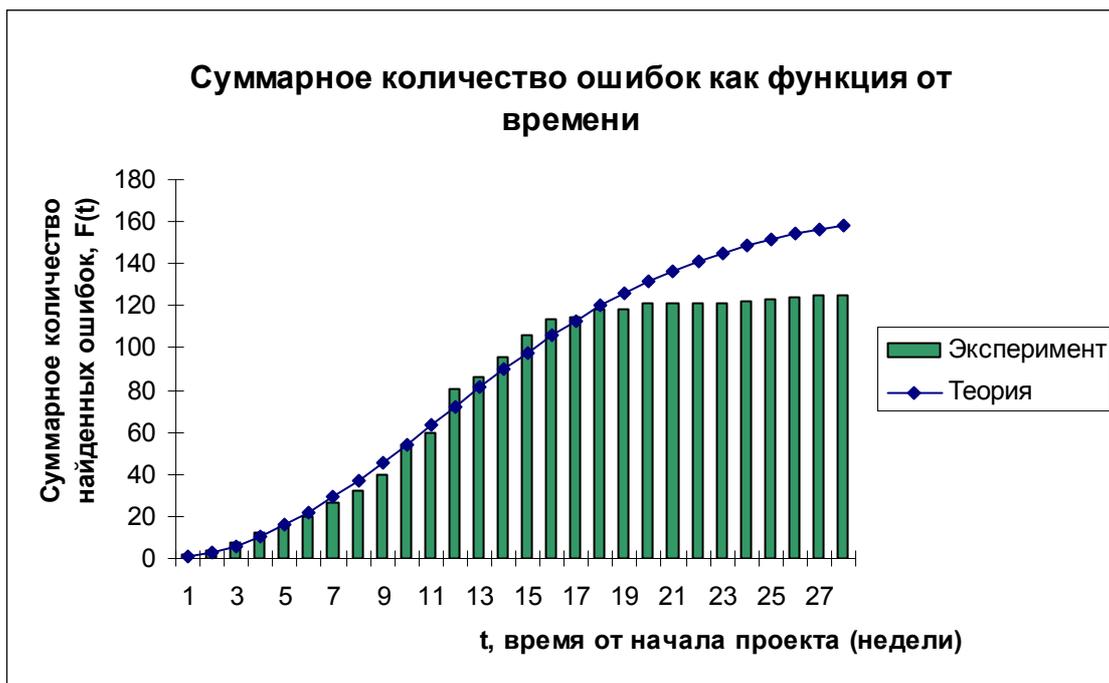
$$F(t) = K \left( 1 - e^{-3 \frac{t^2}{t_d^2}} \right).$$

При тестировании микропроцессора 1890ВМ5Ф, разрабатываемого в НИИ системных исследований РАН, информация о найденных ошибках протоколировалась в системе сбора ошибок. Эти данные («эксперимент»), собранные за 28 недель работ по проектированию и верификации RTL-модели микропроцессора, были отмечены на графике (см. рис. 6 и рис. 7 - в виде гистограммы), затем была получена оценочная кривая в виде распределения Рэля («теория»). Из рис.6 и рис.7 можно сделать вывод о том, что выбранный вид кривой позволяет оценить количество находимых ошибок с течением времени. Теоретические оценки распределения плотности вероятности нахождения ошибок, полученные при тестировании микропроцессора 1890ВМ6Я, хорошо согласуются с экспериментальными данными.

На основе полученных данных можно сделать вывод о возможности применения распределения Рэля для прогнозирования времени окончания работ по верификации микропроцессора.



**Рис. 6. Распределение плотности вероятности нахождения ошибок**



**Рис. 7. Суммарное количество найденных ошибок как функция от времени**

**В четвертой главе** описываются результаты практического использования предлагаемого автором метода встречного тестирования разрабатываемых в НИИ системных исследований микропроцессоров. Также приводится краткая спецификация исследуемых микропроцессоров и

результаты применения разработанных методик на практике. Показаны результаты сравнения производительности микропроцессоров 1890ВМ5Ф и 1890ВМ6Я на ряде стандартных тестов. Дается практическое обоснование того, что предлагаемый автором метод встречного тестирования микропроцессоров соответствует поставленным в работе целям.

В **заключении** перечисляются основные результаты работы.

## **Основные результаты работы**

Основные научные и практические результаты диссертационной работы, **выносимые на защиту**, заключаются в следующем:

1. предложена методика запуска кода готовых приложений под ОС, а также загрузки самой ОС на ранних стадиях проектирования RTL-модели;
2. предложен метод тестирования микропроцессора с привлечением контрольных (тестовых) задач от потенциальных пользователей («встречное» тестирование);
3. разработана методика выбора набора тестов, учитывающая специфику определенного класса задач;
4. разработаны инструменты, поддерживающие предложенные методы.

Перечисленные результаты получены лично автором и были успешно апробированы при создании микропроцессоров 1890ВМ5Ф и 1890ВМ6Я в НИИСИ РАН, а также при выполнении ОКР «Процессор-5» и «Процессор-6» по созданию высокопроизводительных микропроцессоров 1890ВМ8Я и 1890ВМ9Я.

## **Работы автора по теме диссертации**

1. Чибисов П.А., Николина Н.В. Функциональная верификация RTL-модели суперскалярных микропроцессоров. Электроника, микро- и наноэлектроника // Сборник научных трудов. М.: МИФИ, 2004. С. 213–216.

2. Аряшев С.И., Николина Н.В., Чибисов П.А. Тесты аттестации архитектуры RTL-модели 64-разрядного суперскалярного микропроцессора // Проблемы разработки перспективных микроэлектронных систем - 2005. Сборник научных трудов / под общ. ред. А.Л.Стемпковского. М.:ИППМ РАН, 2005. С. 257-262.
3. Аряшев С.И., Краснюк А.А., Чибисов П.А. Адаптация тестов для оценки производительности 64-разрядного универсального суперскалярного микропроцессора // Проблемы разработки перспективных микроэлектронных систем - 2005. Сборник научных трудов / под общ. ред. А.Л.Стемпковского. М.:ИППМ РАН, 2005. С. 263-268.
4. Аряшев С.И., Зубковский П.С., Николина Н.В., Чибисов П.А. Основные подходы к верификации блока вещественной арифметики // Проблемы разработки перспективных микроэлектронных систем - 2005. Сборник научных трудов / под общ. ред. А.Л.Стемпковского. М.:ИППМ РАН, 2005. С. 269-274.
5. Аряшев С.И., Барских М.Е., Чибисов П.А. Оценка покрытия тестами машин состояний RTL-модели процессора // Электроника, микро- и нанoeлектроника. Сборник научных трудов / под ред. В.Я.Стенина.- М.:МИФИ, 2007, с. 153-156.
6. Аряшев С.И., Николина Н.В., Чибисов П.А. Этапы тестирования моделей микропроцессора со встроенным системным контроллером. // Электроника, микро- и нанoeлектроника. Сборник научных трудов. М: МИФИ, 2007. – С. 179-182.
7. Чибисов П.А., Трубицын Д.А., Баранов С.В. Алгоритмы тестирования памяти при проведении радиационных испытаний микропроцессорного модуля // Проблемы разработки перспективных микро- и нанoeлектронных систем - 2010. Сборник трудов / под общ. ред. академика А.Л.Стемпковского. М.:ИППМ РАН, 2010. С. 257-260.
8. Николина Н.В., Зубковский П.С., Чибисов П.А. Сопроцессоры вещественной и комплексной арифметики и их тестирование // Проблемы

разработки перспективных микро- и нанoeлектронных систем - 2010. Сборник трудов / под общ. ред. академика А.Л.Стемпковского. М.:ИППМ РАН, 2010. С. 360-363.

9. Аряшев С.И., Трубицын Д.А., Чибисов П.А. Оптимизация потребляемого тока, частоты функционирования и напряжения питания суперскалярных 64-разрядных микропроцессоров (на примере 1890BM5Ф) // Электроника, микро- и нанoeлектроника. Сборник научных трудов. М: МИФИ, 2011, с.61-64.

10. Ровинский Е.В., Чибисов П.А. Запуск ОС Linux как этап функционального тестирования микропроцессоров // Проблемы разработки перспективных микро- и нанoeлектронных систем - 2012. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. М.: ИППМ РАН, 2012. С. 125-128.

11. Николина Н.В., Чибисов П.А., Аряшев С.И. Современные тенденции оценки и контроля производительности микропроцессоров на стадии их разработки // Проблемы разработки перспективных микро- и нанoeлектронных систем - 2012. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. М.: ИППМ РАН, 2012. С. 489-494.

12. Чибисов П.А. Тестирование микропроцессоров и их RTL-моделей приложениями пользователя под ОС Linux. // Программные продукты и системы, №3, 2012, с. 112-116.

13. Аряшев С. И., Николина Н. В., Чибисов П. А. Организация регрессионного процесса тестирования RTL-моделей микропроцессоров // Параллельные вычисления и задачи управления. Сборник трудов конференции РАСО'2012, с. 231-237.

14. Бобков С.Г., Чибисов П.А. Повышение качества тестирования высокопроизводительных микропроцессоров методами встречного тестирования с анализом функционального тестового покрытия выделенных приложений. // Информационные технологии, №8, 2013, с. 26-33.