

На правах рукописи

Зубковская Наталья Владимировна

МЕТОД ТЕСТИРОВАНИЯ ПРОИЗВОДИТЕЛЬНОСТИ И  
КОРРЕКТНОСТИ МИКРОПРОЦЕССОРОВ ПРИ ПОМОЩИ  
НАЦЕЛЕННЫХ ТЕСТОВЫХ ПРОГРАММ

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 Российской академии наук.

Научный руководитель: кандидат технических наук  
Аряшев Сергей Иванович

Официальные оппоненты: Шагурин Игорь Иванович, профессор,  
доктор технических наук, заместитель  
заведующего кафедрой «Микро- и  
нанозлектроники» Федерального  
государственного автономного  
образовательного учреждения высшего  
профессионального образования  
«Национального исследовательского  
ядерного университета «МИФИ»

Эйсымонт Леонид Константинович,  
кандидат физико-математических наук,  
научный консультант Федерального  
государственного унитарного предприятия  
«Научно-исследовательского института  
«Квант»

Ведущая организация: Открытое акционерное общество «Научно-  
исследовательский институт  
микроэлектронной аппаратуры «Прогресс»

Защита состоится 5 декабря 2013 г. в 16:00 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 (комната 110).

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

Автореферат разослан 01 ноября 2013

Ученый секретарь  
диссертационного совета  
кандидат физ.-мат. наук



/Проخورов С.П./

## Общая характеристика работы

### Актуальность темы

Усложнение задач, решаемых на современных вычислительных системах, ведет к повышению требований к надежности и производительности микропроцессоров, входящих в состав вычислительных комплексов, что, в свою очередь, ведет к необходимости выполнения углубленного контроля функциональности и производительности, как уже изготовленных микросхем, так и микросхем в процессе разработки.

Решение задач повышения надежности и производительности микропроцессоров диктует необходимость использования разнообразных техник тестирования и моделирования и их комбинаций. Так, для целей повышения производительности в архитектуре современных микропроцессоров используются механизмы типа кэш-памяти нескольких уровней, конвейера инструкций, суперскалярного и спекулятивного выполнения инструкций. Проектирование таких сложных механизмов нуждается в инструментальной поддержке, которая в данном случае предоставляется средствами разработки и анализа RTL-модели (модели уровня регистровых передач) микропроцессора или системы на кристалле.

Для оценки производительности могут применяться как приложения, которые планируется использовать на разрабатываемом микропроцессоре, так и специальные тестовые программы, называемые бенчмарками (от английского benchmarks).

Однако не все подходы и техники тестирования и моделирования сочетаются и могут использоваться на всех фазах разработки. Так, применение готовых тестовых программ для оценки производительности на стадии разработки RTL-модели сталкивается со следующими проблемами. Во-первых, время работы обычных тестов производительности становится неприемлемо большим, так как симуляция на RTL-модели на шесть порядков медленнее работы реальной микросхемы. Во-вторых, используя обычные тесты производительности, трудно локализовать дефекты

производительности из-за того, что в процессе выполнения таких тестов задействуются сразу несколько блоков микропроцессора, и при получении низкого значения оценки производительности разработчикам приходится выяснять, работа какого именно блока приводит к этому результату. Существующие методы адаптации готовых тестовых программ для запуска на RTL-модели, позволяющие разбить выполнение программы на части, не решают проблемы локализации дефектов производительности.

Следовательно, имеется необходимость в разработке метода исследования и контроля производительности RTL-моделей, позволяющего локализовать проблемы производительности, контролировать производительность в течение всего времени проектирования, а также автоматизировать процесс создания тестовых ситуаций.

Как правило, для контроля функциональной корректности микропроцессоров используют другие методы построения или генерации тестов, отличные от тестов производительности. Так, для верификации RTL-моделей микропроцессоров широко применяется имитационное тестирование. Для построения тестовых последовательностей применяют такие методы как ручная разработка на основе типовых сценариев, случайная генерация и генерация тестов на основе конечных автоматов или других систем переходов. Применение совокупности данных методов позволяет уменьшить число логических ошибок, однако, помимо ошибок RTL-моделей готовая микросхема может содержать ошибки, проявившиеся при переходе на вентилярный уровень (NETLIST). Предлагаемые на рынке средства формальной верификации, позволяющие проверить эквивалентность NETLIST-модели и RTL-модели, не дают гарантии отсутствия всех временных нарушений. Поэтому при отладке NETLIST-модели также используют имитационное тестирование. Из-за большей детализации скорость работы тестов на NETLIST-модели на несколько порядков замедляется по сравнению со скоростью их работы на RTL-модели. Следовательно, необходим инструмент, позволяющий проверить все узлы блоков микропроцессора при помощи компактных тестов, не требующих больших временных ресурсов.

Кроме того, существует необходимость в создании методики, позволяющей нацеливать тест на определенную ситуацию, что позволит локализовать ошибки, обнаруженные на стадиях тестирования ПЛИС-прототипа (на базе программируемых логических интегральных схем), тестового кристалла или готовой СБИС, для последующего регрессионного тестирования RTL-моделей следующих поколений микропроцессоров или систем на кристалле.

Тем самым показано, что для решения как задач тестирования производительности, так и задач функционального тестирования в процессе проектирования и разработки современных микропроцессоров имеется потребность в методе построения наборов тестовых программ, нацеленных на заданные механизмы микропроцессоров или на заданные ситуации, возникающие в процессе их работы, при этом дополнительным преимуществом метода является его пригодность как для тестирования RTL-моделей, так и для NETLIST-моделей, что позволяет сократить время разработки СБИС.

### **Цель и задачи работы**

Целью диссертационной работы является разработка метода и средств исследования и контроля производительности и корректности функционирования моделей микропроцессоров при помощи нацеленных тестовых программ (микротестов), а также в разработке на основе этих же средств методики локализации ошибок RTL-модели и NETLIST-модели, обнаруженных на стадиях тестирования ПЛИС-прототипа, тестового кристалла или готовой СБИС.

В соответствии с этим были определены следующие основные задачи диссертации:

1. Анализ и классификация методов функциональной верификации и тестирования производительности микропроцессоров.
2. Разработка методики построения критериев полноты тестовых воздействий для оценки производительности, основанной на анализе трасс

исполняемых программ с учетом особенностей реализации архитектуры и микроархитектуры процессора.

3. Разработка средств генерации микротестов, позволяющих автоматизировать процесс их создания.

4. Разработка методики и технологического процесса регрессионного тестирования, обеспечивающего контроль производительности и функциональности.

### **Научная новизна**

Научной новизной обладают следующие результаты работы:

1. Методика построения критериев полноты тестовых воздействий для оценки производительности, основанная на анализе трасс исполняемых программ.

2. Методика локализации ошибок RTL-модели и NETLIST-модели, обнаруженных на стадиях тестирования ПЛИС-прототипа, тестового кристалла или готовой СБИС.

**Практическая значимость.** Предложенные в работе методы и методики могут использоваться для решения задач оценки производительности и корректности функционирования разрабатываемых микропроцессоров. На основе предложенного метода оценки и контроля производительности и функциональности микропроцессоров была реализована система регрессионного тестирования. Система использовалась при разработке отечественной микросхемы 1890ВМ6Я, выпускаемой серийно, а также продолжает использоваться для разработки микросхем 1890ВМ8 и 1890ВМ9 (на данном этапе разработки получен тестовый кристалл 1890ВМ8).

Проведенные экспериментальные исследования и испытания микросхемы 1890ВМ6Я и тестового кристалла 1890ВМ8 позволили оценить точность результатов исследования производительности на стадии разработки RTL-модели, а введенные на этапе проектирования оптимизации позволили повысить производительность микропроцессора.

**Апробация.** Основные положения диссертации докладывались на научно-технических конференциях и семинарах:

- Научно-техническая конференция «Электроника, микро- и наноэлектроника», 2008 г. (г. Вологда), 2011 г. (г. Суздаль), 2012 г. (г. Суздаль).

- Всероссийская научно-техническая конференция "Проблемы разработки перспективных микро- и наноэлектронных систем", 2005 г. (г. Истра), 2010 г. (г. Истра), 2012 г. (г. Истра).

- Международная конференция «Параллельные вычисления и задачи управления», 2012 г. (г. Москва).

- На семинарах НИИСИ РАН.

**Публикации.** По теме диссертационной работы опубликованы 17 печатных работ, из них 4 в изданиях по перечню ВАК. Получено 2 патента Российской Федерации на изобретение.

**Объем и структура работы.** Диссертация состоит из введения, четырех глав, заключения и списка литературы (115 наименований). Основной текст диссертации (без приложений и списка литературы) занимает 114 страниц машинописного текста.

## **Содержание работы**

Во **введении** обоснована актуальность темы, определена цель диссертации. Сформулирована постановка задачи и основные положения, выносимые на защиту.

**Первая глава** посвящена обзору существующих методов функциональной верификации и оценки производительности проектируемых микропроцессоров и систем на их основе. Описываются достоинства и ограничения различных методов. В конце главы делается анализ «узких мест» в функциональной верификации и оценке производительности моделей микропроцессоров, уточняются цели и задачи диссертационной работы.

**Вторая глава** посвящена методу исследования и контроля корректности функционирования и производительности моделей микропроцессоров при помощи микротестов, а также методике локализации ошибок RTL-модели и NETLIST-модели, обнаруженных на стадиях тестирования ПЛИС-прототипа, тестового кристалла или готовой

СБИС. В главе описывается общая схема процесса разработки и верификации микропроцессоров, а также открытые проблемы повышения эффективности тестирования, которые выбраны в качестве объекта исследования, а именно: тестирование функциональности и производительности микропроцессоров на стадии разработки. В рамках работ по верификации в НИИСИ РАН разработан подход встречного тестирования микропроцессоров. До настоящего времени подход позволял существенно сократить число логических ошибок в готовой микросхеме на выделенном классе задач, но не решал проблемы тестирования NETLIST-модели из-за низкой скорости симуляции, а также не позволяет оценивать и контролировать производительность на микроархитектурном уровне. Для решения этой задачи в диссертационной работе предложен метод исследования и контроля производительности и корректности функционирования моделей микропроцессоров при помощи микротестов.

Общая схема метода такова. Применению метода предшествует некоторая подготовительная работа, которая выполняется для каждой новой архитектуры процессора (для нового набора команд с учетом особенностей кэш-памяти, конвейера и прочего). Подготовительная работа включает анализ параметров (характеристик), влияющих на скорость работы каждой из команд. Параметры также характеризуют то, какие элементы микроархитектуры будут задействованы при выполнении данных команд. В соответствии с этими параметрами и набором команд пополняется инициализирующая библиотека (библиотека, в набор функций которой входят программы, подготавливающие состояние микропроцессора к выполнению тестовой ситуации).

Для примера рассмотрим параметры, влияющие на скорость выполнения команды загрузки данных из набора команд MIPS64 *lw rt, offset(base)*. Такими параметрами являются:

- Значения тактовых частот процессора и памяти (*CPU\_freq*, *MEM\_freq*);
- Параметры, характеризующие состояние кэш-памятей при выполнении инструкции, а именно:

- промах или попадание данных в кэш-память разных уровней ( $L1\_hit$ ,  $L2\_hit$ );
- состояние бита Writeback (обратная запись) в кэш-памяти второго уровня ( $WB$ );
- Параметры, характеризующие состояние буфера трансляции адресов ( $JTLB\_hit$ ) и состояние промежуточного буфера, содержащего записи, по которым были последние обращения данных в буфер трансляции адресов ( $mTLB\_hit$ );
- Поле  $base$  задает адрес загружаемых данных, который определяет номер банка динамической памяти, в который происходит обращение ( $NBank$ );
- Параметры, характеризующие промах или попадание в буферы данных ( $L2\_buff\_hit$ ,  $WB\_buff\_hit$  и прочее).

Критерий полноты для рассмотренной инструкции строится на основе метрики, определенной декартовым произведением множеств значений перечисленных параметров, а именно:

$\{L1\_hit, L2\_hit, mTLB\_hit, WB, CPU\_freq, MEM\_freq, L2\_buff, WB\_buff$   
и прочее}, где

$L1\_hit = \{\text{истина, ложь}\};$

$L2\_hit = \{\text{истина, ложь}\};$

$mTLB\_hit = \{\text{истина, ложь}\};$

$WB = \{0, 1\};$

$L2\_buff\_hit = \{\text{истина, ложь}\};$

$WB\_buff\_hit = \{\text{истина, ложь}\};$

$CPU\_freq$ ,  $MEM\_freq$  – могут принимать любые интересующие значения.

В методе реализуется перебор всех значимых сочетаний параметров для выделенных последовательностей инструкций ограниченной длины. Алгоритм определения всех значимых параметров приведен в диссертации. Для получения набора микротестов на вход генератору подается файл с параметрами генерации, а также правилами, в соответствии с которыми должен осуществляться перебор. Так для параметра  $L1\_hit = \{\text{истина}\}$ , нет

необходимости перебирать значения параметров  $WB$ ,  $L2\_buff\_hit$ ,  $WB\_buff\_hit$  и  $MEM\_freq$ , так как на время выполнения инструкции загрузки попавшей в кэш-память первого уровня данные параметры не влияют. Результатом автоматизированного перебора параметров является набор спецификаций, содержащий все значимые комбинации параметров.

На этом подготовительный этап заканчивается, далее метод можно разделить на две фазы:

1. Создание системы микротестов, нацеленных на проверку элементов микроархитектуры для проверки корректности и производительности процессора.
2. Создание дополнительных микротестов, для выявления проблем производительности важных в условиях эксплуатации целевых приложений.

По результатам этих фаз вносятся корректировки в дизайн. Итерации тестирования и корректировки проводятся до тех пор, пока не будут получены приемлемые показатели корректности и производительности. При переходе из одной фазы в другую пополняется инициализирующая библиотека.

Первая фаза метода исследования и контроля производительности и корректности функционирования моделей микропроцессоров состоит из создания системы микротестов, нацеленных на проверку элементов микроархитектуры для проверки корректности и производительности процессора. Для этой цели для разных блоков микропроцессора разработаны шаблоны, включающие инициализирующую часть, тело тестовой ситуации и проверку результата. Тело тестовой ситуации включает в себя инструкции, способные задействовать различные элементы микроархитектуры тестируемых блоков, в зависимости от параметров, с которыми они выполняются.

Рассмотрим, как устроены тестовые программы, полученные в рамках предлагаемого метода. Каждая тестовая программа имеет следующую структуру:

$$\varphi_i = (\varphi_{init}[p_i] \cdot \psi \cdot x_i[p_i] \cdot \gamma_i[p_i])_{i=1..n}, \text{ где:}$$

- ✓  $\varphi_{init}[p_i]$  – программа подготовки к тестовой ситуации: содержит вспомогательные инструкции, предназначенные для инициализации микропроцессора, а также для подготовки к тестовым ситуациям;
- ✓  $p_i$  – параметры, управляющие генерацией тестовых программ: указывают, какую именно подготовку нужно реализовать для той или иной тестовой ситуации;
- ✓  $\psi x_i[p_i]$  – тестовая ситуация ( $i = 1 \dots n$ )
  - $\psi$  – программа инициализации счетчиков производительности (только для оценки производительности);
  - $x_i[p_i]$  – тестовая ситуация: тестируемая последовательность инструкций, где  $p_i$  – это параметры, управляющие генерацией;
- ✓  $\gamma_i[p_i]$  – функция выдачи результата тестовой ситуации (либо результат оценки производительности, либо проверка корректности состояния микропроцессора после выполнения тестовой ситуации);
- ✓  $n$  – количество тестовых ситуаций (тестовых программ).

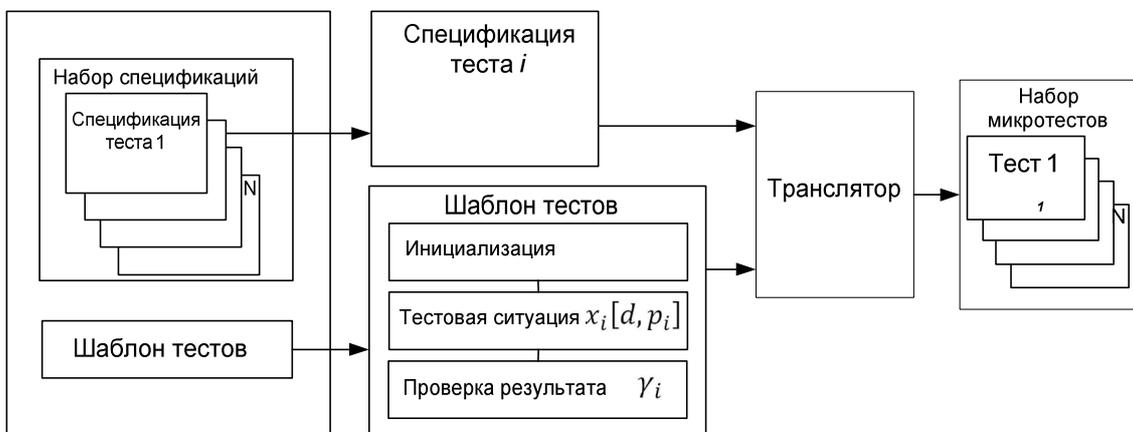


Рис. 1. Сборка тестовых программ

На рис. 1 показан процесс сборки тестовых программ с шаблоном тестов, содержащим процедуру инициализации, саму тестовую ситуацию и проверку результата.

Функция проверки результата в случае оценки производительности выдает результат измерения скорости выполнения тестовой ситуации, а в случае функционального контроля – результат сравнения состояния микропроцессора после выполнения теста, реализующего тестовую ситуацию, с эталонными значениями из таблицы результатов. Таблица формируется по результатам прохождения теста на образцовом эмуляторе. Для этого в шаблон теста встроена функция распознавания модели, на которой был запущен тест (C-модель или RTL-модель), в зависимости от чего тест либо формирует таблицу результатов, либо производит их сравнение с эталонными значениями.

Первая фаза завершается запуском сгенерированных тестов, анализом их результатов и внесением корректировок в дизайн в соответствии с выявленными проблемами. По завершении первой фазы происходит переход ко второй фазе метода, заключающейся в создании дополнительных микротестов, для выявления проблем производительности важных в условиях эксплуатации целевых приложений. Для реализации второй фазы используется метод, основанный на подходе «встречной оптимизации», предложенном в НИИСИ РАН. Ограничением метода «встречной оптимизации» является то, что он позволяет оптимизировать процессор на архитектурном уровне, и не применим для оптимизации микроархитектуры процессора.

Вторая фаза метода состоит из следующих этапов:

1. Получить результаты профилирования (Сбор динамических характеристик, позволяющих провести анализ ОС и приложений, запущенных под управлением ОС). Результатом этапа является набор выделенных целевых фрагментов кода;
2. При помощи программных средств построить критерии полноты для выделенных последовательностей инструкций;
3. На основе критериев с использованием инициализирующих библиотек запустить программу построения микротестов. В процессе генерации набора микротестов могут быть выявлены недостижимые наборы параметров инициализации. Анализ

подобных ситуаций позволяет обнаружить узкие места в микроархитектуре процессора;

4. Построить таблицу результатов (результатом прохождения тестов является скорость выполнения последовательности инструкций в тактах). Вместе с таблицей результатов разработчику предоставляется описание ситуации в терминах значений параметров инициализации. На данном этапе пополняется таблица эталонных значений для последующего регрессионного тестирования.

Метод оценки производительности также предполагает использование фрагментов кода реальных приложений для более детального исследования производительности. Это могут быть либо реализованные библиотечные функции, либо основные циклы программ оценки производительности, таких как Dhrystone, Whetstone и прочее.

Рассмотрим подробнее методику построения критериев полноты тестовых воздействий для оценки производительности. Методика основана на анализе трасс исполняемых программ с учетом особенностей реализации архитектуры и микроархитектуры процессора. Необходимым инструментом для реализации методики определения базовых шаблонов и параметров является покомандный программный эмулятор работы микропроцессора. В эмулятор интегрирована как возможность сохранять трассы выполняемых инструкций в лог-файлы, так и возможность сохранять и восстанавливать состояние эмулируемого микропроцессора в требуемый момент времени.

Методика построения критериев полноты тестовых воздействий, необходимых для оценки производительности, включает в себя следующие действия:

1. Загрузка ядра ОС на эмуляторе (C-модели) без сбора информации о состоянии процессора после выполнения инструкций (без сохранения трасс выполняемых инструкций). Сохранение состояния эмулируемого микропроцессора после загрузки ОС.

2. Запуск задач пользователя, тестов производительности под ОС на эмуляторе с сохранением трасс выполняемых инструкций, с предшествующим восстановлением состояния эмулируемого микропроцессора.
3. Разработка средств, позволяющих собрать статистику по исполняемому коду (лог-файлам с трассами выполненных на эмуляторе инструкций) и определить фрагменты программы, повторяющиеся наибольшее число раз.
4. Профилирование с использованием разработанных средств сбора статистики.
5. По результатам профилирования создание шаблонов тестов, включающих выделенные последовательности инструкций.
6. Определение параметров, влияющих на время выполнения каждой инструкции в шаблоне теста.

По результатам оптимизации архитектуры пополняется система микротестов первой фазы метода.

Вторая фаза завершается генерацией тестов и пропуском их на модели, анализом результатов и внесением корректировок в дизайн в соответствии с выявленными проблемами. По завершении второй фазы пополняется набор шаблонов параметров первой фазы метода.

Также во второй главе рассматривается методика локализации ошибок RTL-модели и NETLIST-модели, обнаруженных на стадиях тестирования ПЛИС-прототипа, тестового кристалла или готовой СБИС.

В рамках работ по верификации в НИИСИ РАН была разработана методика поиска и локализации ошибок в ПЛИС-прототипе и тестовом кристалле. Применение методики требовало наличия образцового эмулятора, в котором реализация поведенческой модели внешних устройств является трудоемкой задачей. Кроме того, созданные по данной методике тесты были не пригодны для регрессионного тестирования.

Новая методика локализации, предложенная в диссертации, основана на использовании инициализирующей библиотеки, применяемой в методе

исследования и контроля производительности и корректности функционирования моделей микропроцессоров.

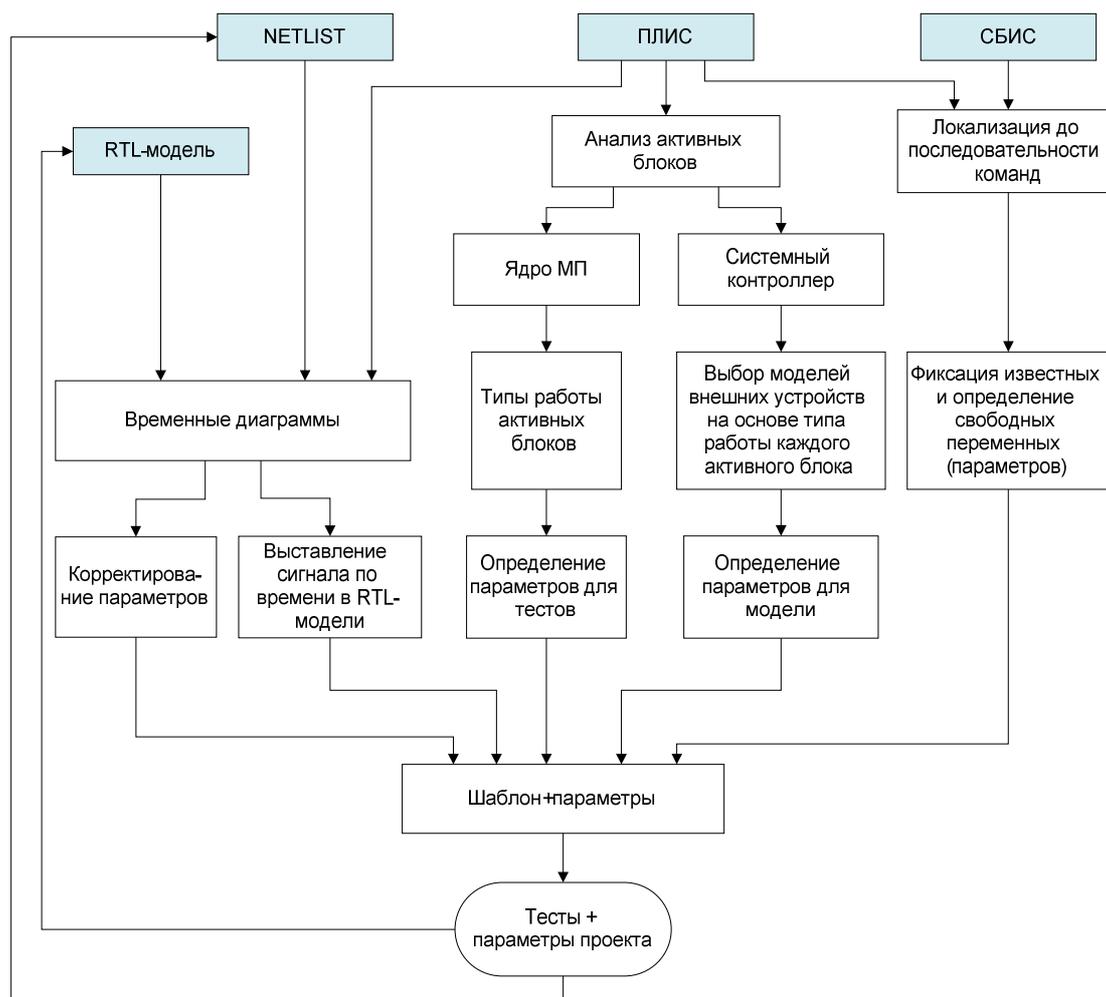


Рис. 2. Методика локализации ошибок RTL-модели и NETLIST-модели

Для реализации методики локализации ошибок RTL-модели и NETLIST-модели необходимо провести следующие подготовительные действия:

- Добавить в RTL-модель модели внешних устройств для имитации циклов чтения/записи по PCI и каналу прямого доступа к памяти. Параметризовать частоту рассмотренных циклов.
- Параметризовать установку сигналов в моделях МП по времени.

Схематически методика локализации ошибок RTL-модели и NETLIST-модели представлена на рис.2.

В **третьей главе** рассмотрены средства, позволяющие реализовать метод оценки производительности и функциональности моделей микропроцессора и методики локализации, а также технологический процесс регрессионного тестирования.

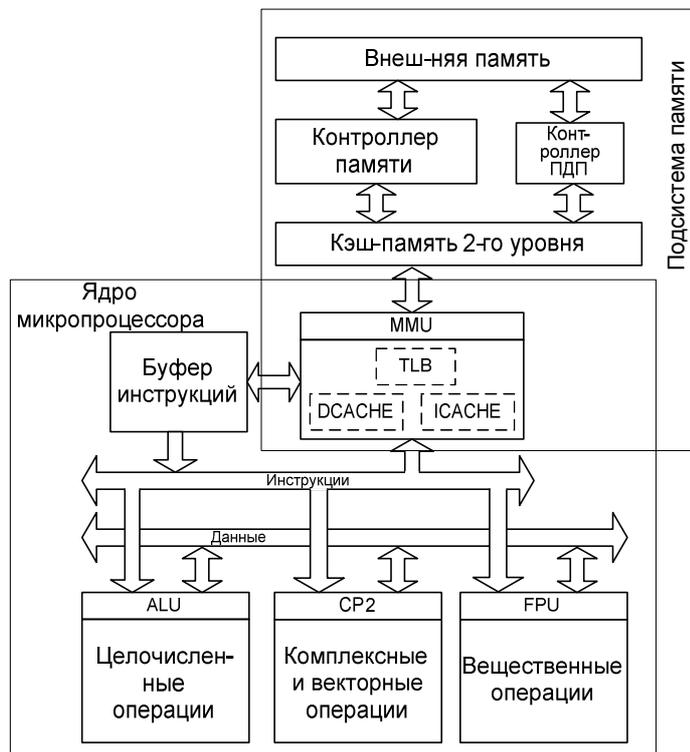


Рис. 3. Структурная схема микропроцессора, разрабатываемого в НИИСИ РАН

В рамках работ по оценке производительности разрабатываемого в НИИСИ РАН 64-разрядного суперскалярного микропроцессора были разработаны наборы микротестов для блока вещественной арифметики, блока комплексной арифметики, блока выборки/выдачи инструкций (буфера инструкций) и подсистемы памяти, реализованные для MIPS-подобной архитектуры. Структурная схема рассматриваемого микропроцессора представлена на рис. 3.

Разработанная система автоматизированного тестирования представляет собой набор тестов, разбитых на группы, а также набор управляющих программ, позволяющих осуществить запуск выбранных групп тестов и собрать полученные результаты. Управляющие программы написаны на языках интерпретатора BourneShell и PERL.

Для каждого блока и подсистемы микропроцессора, представленных на рис.3 сформированы наборы микротестов с различными тестовыми ситуациями. Изоляция каждого тестового случая в отдельном микротесте удобна тем, что разработчик имеет возможность его рассмотрения с использованием временных диаграмм и не тратит времени на поиск данной ситуации в большом и сложном тесте. На рис. 4 показана схема функционирования системы регрессионного тестирования.

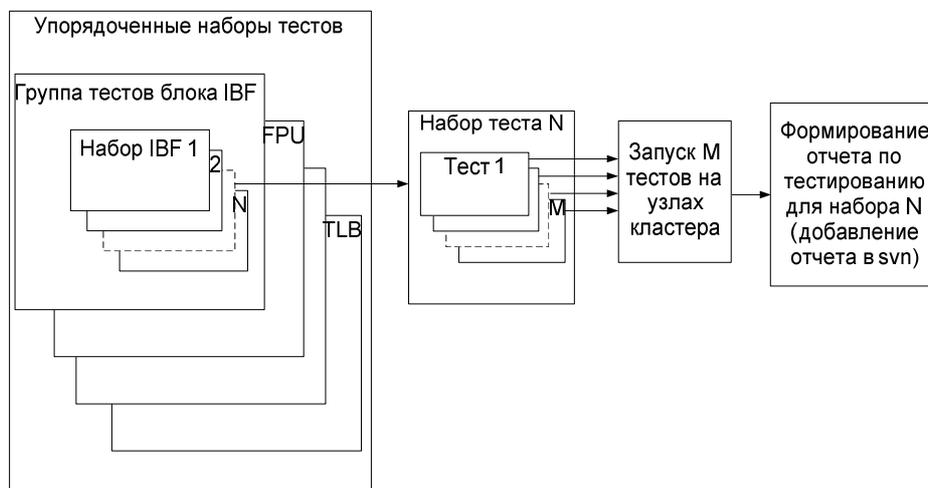


Рис. 4. Схема функционирования системы оценки производительности

Для ускорения процесса регрессионного тестирования база тестов запускается параллельно на узлах Linux-кластера, для этого потребовалось доработать управляющие программы для запуска микротестов на кластере.

Результатом регрессионного процесса измерения производительности является наглядная таблица с возможностью ее сравнения с предыдущими данными. Такая таблица содержит название теста, характеризующее реализованную в нем ситуацию, а также количество тактов, требуемое на ее выполнение. В качестве вспомогательного инструмента для сравнения результатов был использован инструмент отслеживания ошибок Trac. Результаты измерения производительности выдаются в wiki-формате и сохраняются на странице в Trac со ссылкой на версию тестируемого проекта в системе управления версиями. Таким образом, разработчик может увидеть итог прохождения тестов и, при необходимости сравнить текущий результат с одним из предыдущих. Благодаря наличию ссылки на проект в системе управления

версиями, существует возможность увидеть, какие именно изменения в проекте привели к полученному результату. Кроме того, предусмотрен механизм, выдающий отчет об отличии полученных данных от эталонных. Эталонными данными может быть как максимально хороший результат, полученный ранее, так и желаемый результат, которого стремится добиться разработчик.

Достоинством предложенной тестовой системы является гибкость, возможность использования общих механизмов (шаблонов, транслятора для различных шаблонов, библиотек инициализации) как для реализации метода тестирования производительности и корректности функционирования моделей микропроцессоров, так и для методики локализации ошибок RTL-модели и NETLIST-модели, обнаруженных на стадиях тестирования ПЛИС-прототипа, тестового кристалла или готовой СБИС.

В **четвертой** главе приводится опыт практического применения предлагаемой методики на микропроцессорах 1890ВМ6Я и 1890ВМ8.

В результате оценки производительности методом применения микротестов в RTL-моделях микропроцессоров 1890ВМ5Ф, 1890ВМ6Я и 1890ВМ8 были выявлены «узкие места» архитектуры, задержки в выполнении команд, которые являются устранимыми путем оптимизации данной архитектуры.

В работе блока вещественной арифметики микротестами обнаружено, что в режиме 32-х разрядной совместимости независимые команды выполняются в 4-5 раз медленнее, чем в 64-х разрядном режиме. При рассмотрении данного теста на временной диаграмме выяснилось, что после вещественной команды умножения с накоплением (даже если за ней не стояла зависящая от нее инструкция) следовала задержка до записи полученного значения в регистровый файл. После устранения данной задержки тестовые программы в режиме совместимости стали проходить за то же время, что и в 64-х разрядном режиме.

При оценке производительности буфера инструкций были обнаружены задержки в выполнении коротких циклов в зависимости от

адреса команды перехода. То есть, если слот задержки инструкции перехода находится в следующей кэш-линии, то короткий цикл не распознается, команда перехода предсказывается обычным образом (с задержкой в один такт). Выявлено, что на производительность коротких циклов влияла команда безусловного перехода, стоящая после короткого цикла, в данном случае короткий цикл также не распознавался.

Система микротестов позволяет оценить задержку в выполнении команды перехода в зависимости от ее положения в считанной из кэш-памяти восьмерке инструкций. На рис. 5 представлено сравнение задержки от команды перехода для микропроцессоров VM5 и VM6. Видно, что время выполнения зависит не только от положения команды перехода в восьмерке считанных из кэш-памяти инструкций, но и от того, какое количество инструкций находилось в буфере на момент запроса – четное или нечетное.

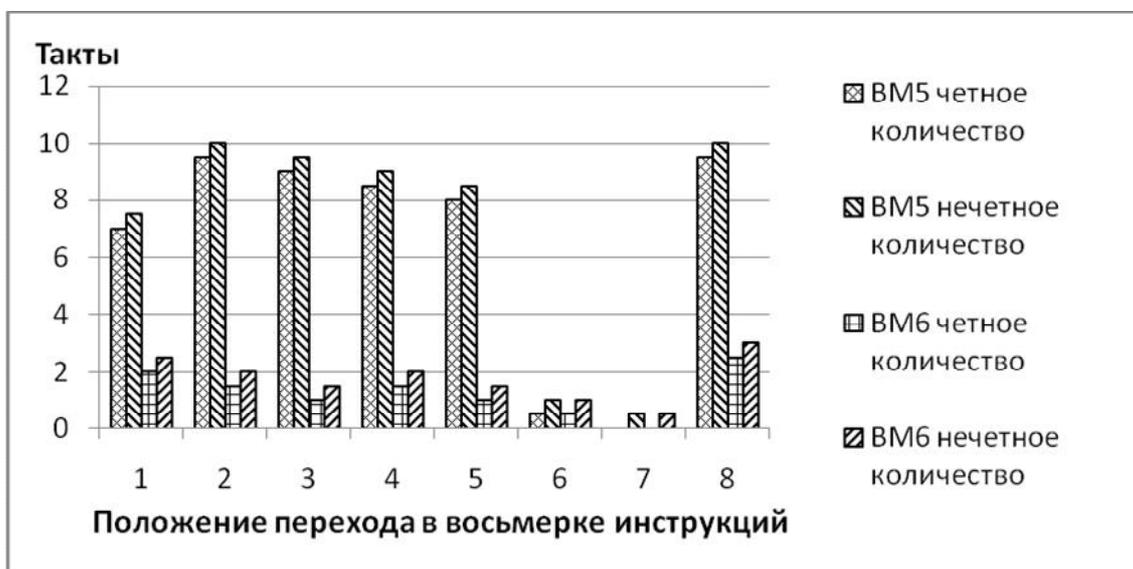


Рис. 5. Задержка выполнения команд перехода в тактах для микропроцессора 1890VM8 с буфером инструкций на 12 команд

С помощью системы микротестов можно исследовать влияние различных архитектурных решений на производительность микропроцессора. Например, была исследована работа буфера инструкций разной глубины, а именно: буфер инструкций на 12 и 16 команд.

Микротесты позволили оценить вклад буферов данных в производительность подсистемы памяти: были исследованы варианты с различным количеством и объемом буферов данных.

На рис. 6 показана скорость загрузки данных в тактах в зависимости от размера буферов предварительного вычитывания данных для микропроцессоров VM5Ф, VM6Я и VM8. Обращения в память с шагом 128 байт по длительности обработки соответствует случайному доступу к памяти, так как промахиваются в кэш-память и буферы предварительного вычитывания. Из рис. 6 видно, что в процессоре VM6Я буфер предварительного вычитывания замедлял чтение данных, т.к. происходила выборка лишней информации.

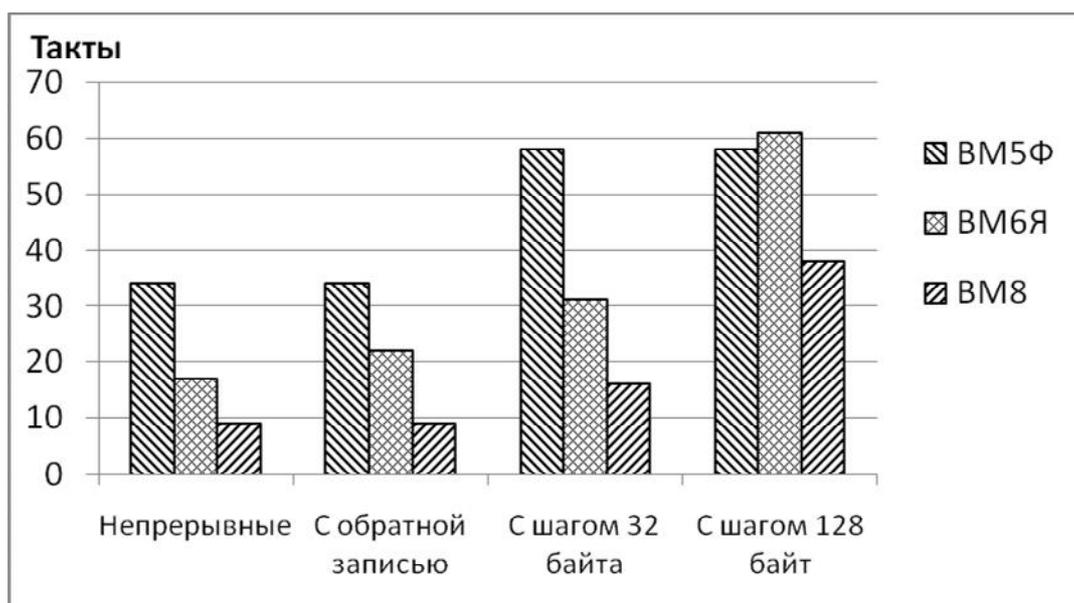


Рис. 6. Время выполнения команд загрузки в тактах (частота работы памяти равна частоте работы процессора)

С помощью системы микротестов была обнаружена неэффективность адресации банка динамической памяти: банк памяти адресовался только битами индекса, причем тот же индекс адресовал и строку кэш-памяти второго уровня. Таким образом, любое обращение к памяти, вызывающее обратную запись, обращается в один банк с самой обратной записью. Для адресации банка памяти автором было предложено использовать не только биты индекса, но и биты тэга, что сократило количество таких ситуаций. Схема работы блока перестановки битов адреса оформлена в виде патента.

С помощью системы микротестов проводятся исследования влияния частоты работы процессора или памяти на скорость загрузки/сохранения или копирования данных.

Методика автоматизированной генерации микротестов позволила упростить отладку модели микросхемы вентиляционного уровня после топологического проектирования (NETLIST). Время локализации ошибок сократилось более чем в 2 раза. Кроме того, данная методика показала свою эффективность для поиска сложных ошибок RTL-моделей, NETLIST-моделей и СБИС.

Основным итогом главы является практическое обоснование соответствия предлагаемого метода оценки производительности и функциональности, а также методики локализации ошибок поставленным в работе целям.

В **заключении** приводятся основные результаты работы.

## **Основные результаты работы**

Основные результаты диссертационной работы заключаются в следующем:

1. Разработан метод исследования и контроля производительности и функциональности микропроцессора на основе микротестов.

2. Разработана методика построения тестового набора для оценки производительности, основанная на анализе трасс исполняемых программ. Методика позволяет определить базовые шаблоны для тестовых ситуаций и параметры, влияющие на скорость выполнения инструкций в целевом тесте.

3. Разработана методика локализации ошибок RTL-моделей, NETLIST-моделей и СБИС.

4. На основе проведенного исследования разработана система автоматической генерации тестовых программ и система регрессионного тестирования для оценки и контроля производительности и функциональности микропроцессора, позволяющая за короткое время обнаружить дефекты в производительности и функциональности, а также представить результат в удобном для дальнейшего анализа виде.

Использование предложенных метода и методик позволило автору предложить несколько решений по оптимизации архитектуры

микропроцессора, которые были реализованы в микросхеме, а также оформлены в виде патентов РФ на изобретение.

Разработанная методика локализации ошибок RTL-моделей, NETLIST-моделей и СБИС позволила локализовать 4 ошибки, найденные на готовой микросхеме, 6 ошибок тестового кристалла и 23 ошибки, обнаруженные при тестировании модели микропроцессора на ПЛИС-прототипе.

Разработанные в диссертации методики оценки и контроля производительности микропроцессоров и система регрессионного тестирования были использованы при разработке отечественной микросхемы 1890ВМ6Я, выпускаемой серийно, а также продолжают использоваться для разработки микросхем 1890ВМ8 и 1890ВМ9 (на данном этапе разработки получен тестовый кристалл 1890ВМ8).

Проведенные экспериментальные исследования и испытания микросхемы 1890ВМ6Я и тестового кристалла 1890ВМ8 позволили оценить точность результатов исследования производительности на стадии разработки RTL-модели, а введенные на этапе проектирования оптимизации позволили повысить производительность микропроцессора.

## **Работы автора по теме диссертации**

1. Зубковская Н.В. Оценка и контроль производительности и работоспособности проектируемых микропроцессоров на базе микротестов // Электроника, микро- и нанoeлектроника. Сб. научн. трудов. - М.: МИФИ, 2013. - С. 188-195.
2. Аряшев С.И., Зубковская Н.В., Корниленко А.В., Рогаткин Б.Ю. Имитационное моделирование подсистемы памяти микропроцессора со встроенным системным контроллером // Электроника, микро- и нанoeлектроника. Сб. научн. трудов. - М.: МИФИ, 2013. - С. 118-125.
3. Аряшев С.И., Корниленко А.В., Зубковская Н.В., Саяпин П.В. Повышение производительности подсистемы памяти методом буферизации данных // Информационные технологии. - 2013. - №6 - С. 11-17.

4. Пат. 2480820 РФ, МПК G06F 15/00 G06F 12/12. Устройство обработки информации / Аряшев С.И., Корниленко А.В., Зубковская (Николина) Н.В., Петров К.А. - Опубл. 27.04.2013. - Бюл. № 12. – 7 С.
5. Пат. 2475817 РФ, МПК G06F9/30. Устройство буферизации потоков данных, считываемых из ОЗУ / Аряшев С.И., Корниленко А.В., Зубковская (Николина) Н.В. - Опубл. 20.02.2013. - Бюл. № 5. – 6 С.
6. Зубковская (Николина) Н.В. Микротесты для оценки производительности RTL-моделей микропроцессоров // Программные продукты и системы. – 2012, №3. - С. 103 – 107.
7. Аряшев С.И., Зубковская (Николина) Н.В., Чибисов П.А. Организация регрессионного процесса тестирования RTL-моделей микропроцессоров // Доклады шестой международной конференции РАСО-2012. - С. 231-237.
8. Зубковская (Николина) Н.В., Чибисов П.А., Аряшев С.И. Современные тенденции оценки и контроля производительности микропроцессоров на стадии их разработки // Сборник научных трудов «Проблемы разработки перспективных микроэлектронных систем – 2012» / Под ред. А.Л. Стемповского. - М.: ИПМРАН, 2012. - С. 489 – 494.
9. Аряшев С.И., Барских М.Е., Зубковская (Николина) Н.В. Оценка производительности буфера инструкций на стадии разработки RTL-модели микропроцессора // Электроника, микро- и нанoeлектроника. Сб. научн. трудов. - М.: МИФИ, 2012. - С. 158-165.
10. Аряшев С.И., Корниленко А.В., Зубковская (Николина) Н.В., Петров К.А. Повышение сбоеустойчивости и быстродействия подсистемы внешней динамической памяти микропроцессорной системы // Электроника, микро- и нанoeлектроника. Сб. научн. трудов. - М.: МИФИ, 2012. - С. 173-178.
11. Рогаткин Б.Ю., Зубковская (Николина) Н.В., Аряшев С.И. Проблемы проектирования подсистемы памяти ядра микропроцессора со встроенным системным контроллером / 16-я Международная телекоммуникационная конференция молодых ученых и студентов «Молодежь и наука», 2012.
12. Аряшев С.И., Зубковская (Николина) Н.В. Влияние буферизации данных на производительность подсистемы памяти // Электроника, микро- и нанoeлектроника. Сб. научн. трудов. - М.: МИФИ, 2011. - С. 164-167.
13. Зубковская (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 РАН А.Л. Стемпковского. – М.: ИППМ РАН, 2010. – С. 360-363.
14. Аряшев С.И., Зубковская (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.
  15. Аряшев С.И., Зубковский П.С., Зубковская (Николина) Н.В., Чибисов П.А. Основные подходы к верификации блока вещественной арифметики. Проблемы разработки перспективных микроэлектронных систем – 2005. Сборник научных трудов. – М: ИППМ РАН, 2005. – С. 269-274.
  16. Аряшев С.И., Зубковская (Николина) Н.В., Чибисов П.А. Тесты аттестации архитектуры RTL-модели 64-разрядного суперскалярного микропроцессора. Проблемы разработки перспективных микроэлектронных систем – 2005. Сборник научных трудов. – М: ИППМ РАН, 2005. – С. 257-262.
  17. Чибисов П.А., Зубковская (Николина) Н.В. Функциональная верификация RTL-модели суперскалярных микропроцессоров. Электроника, микро- и нанoeлектроника. Сборник научных трудов. – М: МИФИ, 2004. – С. 213–216.