На правах рукописи

Пантелеев Алексей Юрьевич

Высокопроизводительные сопроцессоры для параллельной обработки данных в формате с плавающей точкой в системах цифровой обработки сигналов

05.13.05 − Элементы и устройства вычислительной техники и систем управления

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Автор: 

Москва, 2013

Диссертация выполнена в Национальном исследовательском ядерном университете «МИФИ».

Научный руководитель:

доктор технических наук, профессор
Шагурин Игорь Иванович.

Официальные оппоненты:

доктор технических наук, профессор
Корнеев Виктор Владимирович,
заместитель директора ФГУП НИИ «Квант» по научной работе;

кандидат технических наук
Осипенко Павел Николаевич,
директор департамента проектирования ОАО «Байкал Электроникс».

Ведущая организация:

Научно-исследовательский институт системных исследований Российской академии наук (НИИСИ РАН).

Защита диссертации состоится 15 апреля 2013 г. в 15 часов 00 минут на заседании диссертационного совета Д 212.130.02 в Национальном исследовательском ядерном университете «МИФИ», расположенном по адресу: 115409, г. Москва, Каширское шоссе, 31, тел. (499) 324-87-66.

С диссертацией можно ознакомиться в библиотеке НИЯУ «МИФИ».

Автореферат разослан « » марта 2013 г.

Просим принять участие в работе совета или прислать отзыв в одном экземпляре, заверенный печатью организации.

Ученый секретарь
диссертационного совета
д.т.н., профессор

П.К. Скоробогатов

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы

Требования к производительности систем цифровой обработки сигналов постоянно растут: появляются новые стандарты связи с широкими полосами пропускания, используется цифровая модуляция радиосигналов, требуется обрабатывать многоканальные сигналы в одном устройстве, и так далее. В настоящее время основным способом увеличения производительности является применение параллельных вычислений при условии эффективного использования аппаратных ресурсов. В связи с этим актуальной является разработка новых способов построения устройств, выполняющих задачи цифровой обработки сигналов, используя параллельные вычисления, в частности, программируемых сопроцессоров, предназначенных для применения в составе СБИС класса «система на кристалле».

Цель диссертационной работы

Повышение производительности СБИС класса «система на кристалле» при выполнении цифровой обработки сигналов путем применения архитектурных решений, позволяющих эффективно организовать параллельную обработку данных в формате с плавающей точкой. Для достижения данной цели в диссертации решаются следующие задачи:

1) Исследование возможностей и проблем построения сопроцессоров для цифровой обработки сигналов с использованием многопоточной архитектуры с целью определения факторов, ограничивающих производительность, способов повышения производительности, а также определения целесообразности применения многопоточной архитектуры для построения цифровых сигнальных процессоров.

2) Разработка системы векторной памяти, которая бы эффективно работала при использовании характерных для цифровой обработки сигналов режимов адресации.

3) Разработка модели исполнения и системы команд сопроцессора, которая бы позволяла обрабатывать задачи различного размера с минимальным числом циклов и вспомогательных инструкций в программах и способствовала достижению масштабируемости архитектуры.

4) Разработка структуры вычислительного конвейера сопроцессора, позволяющего достичь высокой эффективности реализации основных алгоритмов цифровой обработки сигналов.

5) Разработка структуры и RTL-модели сопроцессора, в котором используются решения, предлагаемые в диссертации, с целью подтверждения достоверности и практической применимости результатов, достигнутых в ходе выполнения вышеозначенных задач.

Научная новизна

1) Предложен способ организации векторной памяти с использованием конвертируемых режимов адресации, который позволяет размещать в памяти матрицы и обеспечивать параллельный доступ к элементам строк или столбцов матриц без транспонирования, а также реализовать другие режимы адресации, характерные для алгоритмов цифровой обработки сигналов.

2) Предложено использовать в программируемых сопроцессорах, предназначенных для цифровой обработки сигналов, векторные вычисления с поддержкой векторов переменной длины. Это позволяет исключить внутренние циклы в программах и обеспечить детерминированное время выполнения программ.

3) Предложен способ построения планировщика выполнения векторных инструкций для сопроцессора с поддержкой векторов переменной длины, основанный на расчете длительности блокировок, используемых для разрешения зависимостей по данным в программе. Планировщик, построенный таким образом, обеспечивает высокое быстродействие сопроцессора в широком диапазоне параметров выполняемой программы (длина векторов, длина последовательностей независимых инструкций) и потребляет небольшое количество аппаратных ресурсов.

4) Предложен способ построения быстродействующего конвейерного арифметического блока, выполняющего составные вычислительные операции в действительных и комплексных числах формата с плавающей точкой, такие как умножение со сложением и вычитанием, путем различного соединения арифметических блоков, выполняющих по отдельности сложение и умножение действительных чисел того же формата. Применение такого блока обеспечивает небольшую длину конвейера при выполнении простых операций, минимальное количество обращений к памяти при выполнении сложных операций, и быструю смену типа выполняемых операций.

Практическая значимость

1) Разработана RTL-модель векторного сопроцессора для цифровой обработки сигналов, которая может быть использована в качестве сложнофункционального блока при реализации СБИС класса «система на кристалле». Разработанная модель обеспечивает возможность эффективной реализации основных алгоритмов цифровой обработки сигналов (быстрое преобразование Фурье, свертка, умножение вектора на матрицу и др.) с применением чисел формата с плавающей точкой. Функционирование данной модели проверено путем выполнения на ней набора функциональных и алгоритмических тестов с использованием различных симуляторов Verilog HDL.

2) Проведен логический синтез и получена логическая структура сопроцессора с 4 вычислительными конвейерами и 96 КБ внутренней памяти данных, которая при изготовлении СБИС по технологическому процессу с проектной нормой 40 нм занимает на кристалле площадь 1,88 мм^2 и может работать при тактовой частоте до 1 ГГц. Эффективность работы сопроцессора при реализации алгоритмов цифровой обработки сигналов достигает 98% и при увеличении числа вычислительных конвейеров снижается незначительно, что позволяет при необходимости получить логическую структуру сопроцессоров, имеющих в 2 или в 4 раза большую производительность.

Внедрение результатов диссертации

В ЗАО НТЦ «Модуль» проводилась ОКР «Процессор-1-Модуль» по разработке архитектуры СФ-блока NM-кластера на основе матрично-векторных процессорных ядер. В состав процессорного ядра NM-кластера входит программируемый сопроцессор векторной обработки данных в формате с плавающей точкой, в котором использовались результаты разработки модели сопроцессора М3. Применение такого сопроцессора в составе NM6407 позволяет организовать выполнение алгоритмов цифровой обработки сигналов в числах стандартного 32-битного формата с плавающей точкой с высокой производительностью – до 20 операций за такт при тактовой частоте в 1 ГГц при реализации по технологическим нормам 40 нм. В частности, данный сопроцессор позволяет выполнять быстрое преобразование Фурье размерностью в 1024 точки за 3130 процессорных тактов, что существенно превосходит существующие процессоры цифровой обработки сигналов. Сопроцессор эффективно реализует и другие функции, такие как КИХ-фильтрация и вычисление фрагментов нейросетей.

Основные положения, выносимые на защиту

1) Архитектура и структура масштабируемого векторного сопроцессора, предназначенного для применения в составе СБИС класса «система на кристалле» для цифровой обработки сигналов.

2) Способ организации и адресации векторной памяти с несколькими раздельно адресуемыми банками, который позволяет размещать в памяти матрицы произвольного размера и получать доступ к строкам или к столбцам матрицы без транспонирования, а также упрощать реализацию нематричных алгоритмов цифровой обработки сигналов на векторных процессорах.

3) Способ построения планировщика вычислений для процессора с поддержкой векторов переменной длины. Применение векторов переменной длины позволяет упростить программы и сократить объем кода, требуемый для реализации основных алгоритмов цифровой обработки сигналов по сравнению с процессорами, использующими вектора фиксированной длины, а также обеспечивает детерминированное время выполнения программ.

4) Структура многофункционального векторного вычислительного блока, производящего операции с действительными и комплексными числами при помощи изменения порядка соединения вычислительных блоков.

Апробация работы

Основные результаты диссертации докладывались и обсуждались на Всероссийских научно-технических конференциях «Проблемы разработки перспективных микро- и наноэлектронных систем» (Москва, 2010, 2012), 19-й Всероссийской межвузовской научно-технической конференции студентов и аспирантов «Микроэлектроника и информатика – 2012» (Москва, МИЭТ, 2012), Научных сессиях НИЯУ МИФИ (Москва, 2008–2012),

По результатам диссертации опубликовано 6 статей (из них 4 – в изданиях, входящих в Перечень периодических изданий, рекомендованных ВАК РФ), 8 тезисов докладов.

Достоверность результатов

Достоверность представленных в диссертации данных о производительности разработанных моделей сопроцессоров подтверждается путем поведенческой симуляции данных моделей при выполнении тестовых программ, включающих реализации алгоритмов ЦОС, с проверкой корректности результатов работы этих программ. Возможность аппаратной реализации со-

процессоров M1 и M3 подтверждается путем логического синтеза их RTL-моделей с использованием проверенных современных средств (Synopsys Design Compiler, Xilinx XST) и поведенческой симуляцией моделей, полученных в результате такого синтеза.

Структура и объем диссертации

Диссертация состоит из введения, 4 глав, заключения, списка литературы, включающего 131 наименование, и 3 приложений, содержащих исходные тексты реализации различных алгоритмов БПФ и акт о внедрении результатов диссертационной работы. Содержание диссертации изложено на 154 страницах машинописного текста, включая 36 рисунков и 23 таблицы.

СОДЕРЖАНИЕ РАБОТЫ

Архитектура систем и алгоритмы цифровой обработки сигналов

Для реализации задач цифровой обработки сигналов (ЦОС) часто применяются специализированные цифровые сигнальные процессоры (ЦСП), позволяющие реализовать типичные алгоритмы с минимальными издержками, используя вычислительные ресурсы устройства близко к их максимальным возможностям. Большинство ЦСП имеют характерные особенности архитектуры, позволяющие им эффективно выполнять алгоритмы цифровой обработки сигналов, отличающиеся высокой плотностью вычислений и обилием циклов. Такими особенностями являются поддержка операции умножения с накоплением, специальных режимов адресации памяти, выполнения нескольких разнородных операций за такт с применением архитектуры типа VLIW (Very Long Instruction Word).

Кроме отдельных ЦСП, часто применяются специализированные системы на кристалле (СнК), в состав которых входят различные блоки, характерные для проектируемого устройства. Такими блоками, среди прочих, могут являться сопроцессоры, предназначенные для выполнения определенных алгоритмов с высокой производительностью и/или энергоэффективностью. Существует множество вариантов реализации таких блоков. Самый простой и эффективный вариант – когда блок выполняет один алгоритм для входных данных фиксированной размерности, например, фильтр с конечной импульсной характеристикой (КИХ-фильтр) или быстрое преобразование Фурье (БПФ). Реже применяются программируемые сопроцессоры. Они способны

выполнять более широкий набор задач, чем блоки с фиксированной функциональностью, однако их производительность обычно ниже.

Современные программируемые устройства, применяемые для высокопроизводительной цифровой обработки сигналов, обладают достаточно схожей структурой. Обычно это системы на кристалле или многопроцессорные системы, где процессорные ядра обладают архитектурой типа VLIW. Явно параллельная, масштабируемая обработка данных типа SIMD (Single Instruction Multiple Data) применяется относительно редко, и процессорные ядра поддерживают только короткие вектора (до 64 битов в случае распространенных процессоров).

Некоторые распространенные модели ЦСП поддерживают вычисления с плавающей точкой, но производительность таких вычислений значительно (в несколько раз) ниже, чем производительность вычислений с фиксированной точкой или с целыми числами. При этом разработка алгоритмов для вычислений с плавающей точкой значительно проще, чем для вычислений с фиксированной точкой за счет того, что числа с плавающей точкой позволяют закодировать значительно больший диапазон значений и обеспечивают равномерную относительную погрешность представления во всем диапазоне.

Многие алгоритмы ЦОС строятся на основе совместно используемых базовых алгоритмов. Такими базовыми алгоритмами являются БПФ, свертка и корреляция, векторные и матричные операции, в частности, умножение вектора на матрицу. Для базовых алгоритмов имеются способы эффективной параллельной реализации. Из алгоритмов БПФ оптимальным с точки зрения параллельной реализации является алгоритм многомерного разложения, но возможно также применение других алгоритмов. Параллельная реализация алгоритмов свертки и корреляции достаточно проста, но требует высокой пропускной способности памяти при использовании векторного доступа с произвольным смещением. Параллельная реализация умножения вектора на матрицу требует высокой пропускной способности памяти при использовании простой векторной адресации. В целом, алгоритмы ЦОС требовательны к пропускной способности системы памяти при использовании характерных способов адресации.

Исследование проблем построения параллельных сопроцессоров для ЦОС с использованием многопоточной архитектуры

Процессоры, архитектура которых предусматривает одновременное выполнение большого количества независимых потоков команд, представляют перспективное направление развития параллельных вычислительных машин. Использование многопоточной модели исполнения требует от программистов решения задач в явно параллельном виде, что значительно облегчает распределение нагрузки между параллельно работающими вычислительными устройствами. К данной категории процессоров относятся, в основном, современные графические процессоры (Graphics Processing Unit, GPU), которые обладают значительно большей вычислительной мощностью и энергетической эффективностью, чем другие программируемые устройства общего назначения. Имеются различные средства программирования GPU для решения вычислительных (т.е. не графических) задач, одним из которых является открытый стандарт OpenCL. Он не ограничивается только лишь графическими процессорами, а предусматривает использование в качестве сопроцессоров различных классов устройств, отвечающих определенным требованиям, основные из которых – наличие собственной памяти и поддержка выполнения множества независимых потоков команд.

Для того чтобы оценить сложность и перспективность применения многопоточных архитектур для построения систем и сопроцессоров ЦОС, были разработаны две модели сопроцессоров M1 и M2, частично поддерживающих стандарт OpenCL. Они характеризуется следующими функциональными возможностями:

1) Одновременное выполнение достаточно большого числа независимых потоков команд (например, 64), объединенных в группы по числу вычислительных конвейеров процессора (например, 4) – такая группа называется «**варп**».

2) Поддержка независимых операций ветвления в каждом варпе.

3) Наличие отдельной области памяти у каждого потока команд. Эта память поддерживает только доступ по постоянному адресу, заданному в программе.

4) Наличие разделяемой памяти, доступной всем активным потокам команд. Адресация разделяемой памяти осуществляется независимо из каждого потока. Такой режим адресации называется косвенной векторной адресацией.

5) Поддержка операций барьерной синхронизации потоков команд для организации доступа различных потоков к разделяемой памяти.

Полная поддержка стандарта OpenCL требует наличия некоторых блоков, выполняющих функции, которые не требуются при реализации алгоритмов ЦОС. В частности, сопроцессор не имеет доступа к глобальной памяти: предполагается, что данные предварительно загружаются в разделяемую память при помощи контроллеров прямого доступа к памяти, входящих в состав СнК. Кроме того, поддержка независимого ветвления программы в каждом отдельном потоке заменена на независимые ветвления в каждом варпе, а вычисления трансцендентных функций не поддерживается.

Модель М2 имеет следующие отличия от модели М1:

1) Поддержка одновременного нахождения в конвейере нескольких последовательных команд из одного потока (при условии независимости данных инструкций).

2) Поддержка одновременного запуска до 3 разнородных команд на каждом такте посредством явного указания выполняемых таким образом команд (т.е. использование архитектуры типа VLIW).

3) Модель М2 реализована в виде программного потактового симулятора, тогда как модель М1 реализована на уровне регистровых передач (т.е. является синтезируемой RTL-моделью).

В диссертации приведены структурные схемы сопроцессоров М1 и М2. Также описан принцип функционирования блоков, реализующих функции, необходимые для поддержки многопоточной модели исполнения: планировщика, который производит выбор варпа и инструкции для выполнения на каждом такте, а также блока доступа к разделяемой памяти. Представлены результаты синтеза блока доступа к разделяемой памяти при различных количествах конвейеров сопроцессора и банков разделяемой памяти, и показано, что данный блок плохо масштабируется: с ростом числа конвейеров или банков число логических элементов, требуемых для реализации блока, возрастает квадратично, а максимальная тактовая частота падает. Этот блок занимает б**о**льшую часть логических элементов, требуемых для реализации сопроцессора, и ограничивает его тактовую частоту.

Для сопроцессоров М1 и М2 реализованы программы вычисления БПФ по алгоритму Cooley-Tukey, БПФ по алгоритму двумерного разложения, блочной свертки и умножения вектора на матрицу. Исследована производительность данных программ при различных размерах задач и при различных

параметрах сопроцессоров (главными из которых являются число параллельно работающих конвейеров и число стадий конвейера), определены факторы, ограничивающие производительность. По результатам экспериментов с использованием разработанных моделей и алгоритмов сделан ряд выводов, основными из которых являются:

1) При выполнении сопроцессором задач, размер которых характерен для ЦОС (например, БПФ от вектора размерностью 1024 отсчета), одного лишь переключения выполняемых потоков команд на каждом такте недостаточно для того, чтобы скрыть латентность вычислений и обеспечить эффективное использование вычислительных блоков. Поддержка нахождения нескольких инструкций из одного потока команд в конвейере («полностью конвейерный режим») позволяет существенно повысить производительность сопроцессоров при выполнении алгоритмов ЦОС и производит наибольший эффект (ускорение в 2 и более раз) при вычислении БПФ по алгоритму двумерного разложения.

2) Применение архитектуры типа VLIW позволяет существенно повысить эффективность использования вычислительных блоков сопроцессора, но имеет недостатки, связанные с увеличением объема программ и сложностью коммутации банков памяти и вычислительных блоков сопроцессора. Использование VLIW является целесообразным, если необходимо обеспечить высокую производительность для широкого класса задач, т.е. когда отсутствует возможность реализовать архитектуру процессора, оптимизированную для реализации определенных алгоритмов.

3) Одной из наиболее важных проблем построения сопроцессоров, реализующих многопоточную архитектуру, является сложность аппаратной реализации косвенной векторной адресации памяти: площадь, занимаемая блоком доступа к такой памяти на кристалле, квадратично зависит от количества вычислительных конвейеров сопроцессора и банков разделяемой памяти. Другая важная проблема заключается в том, что время выполнения программы при использовании множества независимых потоков команд существенно зависит от порядка выполнения команд из различных потоков. Выбор потока команд для исполнения реализуется эвристическим способом, и поэтому время выполнения программы является труднопредсказуемым, что затрудняет применение таких сопроцессоров в составе систем, работающих в условиях реального времени.

Применение конвертируемой адресации памяти

Поддержка доступа в разделяемую память при наличии конфликтов банков является необязательной для реализации рассматриваемых алгоритмов ЦОС. Исключение блока определения конфликтов банков позволит значительно упростить сопроцессор. Однако наличие такого блока является необходимым для корректной реализации операции косвенной векторной адресации памяти, следовательно, исключение такого блока приведет к необходимости выбора другой модели исполнения, не использующей такую адресацию. Для реализации рассматриваемых алгоритмов ЦОС необходимо реализовать возможность выполнения следующих операций:

1) Транспонирование матриц произвольного размера, что необходимо для реализации БПФ по алгоритму двумерного разложения.

2) Обращение к векторной памяти по некратному смещению, которое необходимо для реализации свертки.

3) Использование скалярных значений в векторных операциях.

Для решения этой проблемы предлагается использовать векторную память с несколькими банками и с поддержкой специализированных режимов адресации. Такая память автоматически генерирует множество необходимых адресов при поступлении одного запроса в соответствии с заданным режимом адресации, что дает возможность гарантировать отсутствие конфликтов банков, а также упростить схемы коммутации банков и вычислительных конвейеров сопроцессора.

Применение конвертируемых режимов адресации может ускорить реализацию БПФ по алгоритму двумерного разложения и уменьшить объем ее кода. Это связано с тем, что для выполнения промежуточного транспонирования матрицы данных не нужно использовать разделяемую память – вместо этого используется одна инструкция изменения режима адресации. Для оценки того, насколько изменится производительность БПФ, разработана программа вычисления БПФ вектора размерностью 256 отсчетов по такому упрощенному алгоритму. Время выполнения алгоритма при использовании конвертируемых режимов адресации сократилось на 7–11%, в зависимости от числа конвейеров сопроцессора. Число обращений к разделяемой памяти сократилось на 40%, а объем программы – на 7%.

Для размещения данных матриц в памяти и доступа к строкам и столбцам матриц без транспонирования предлагается использовать скошенную адресацию. Пример расположения матриц в векторной памяти показан на

рис. 1. Изображена матрица размерностью 8 строк на 8 столбцов, расположенная в 8 банках памяти. Строки обозначены латинскими буквами A–H, столбцы – цифрами 0–7; столбцы 0 и 4 выделены цветом фона. При использовании прямой адресации каждая строка матрицы занимает ячейки с одинаковыми адресами во всех банках, причем столбец 0 всегда располагается в банке 0. При использовании скошенной адресации каждая строка также занимает ячейки с одинаковыми адресами, но в других банках: вся строка циклически сдвигается вправо (или влево) со смещением, равным номеру строки.

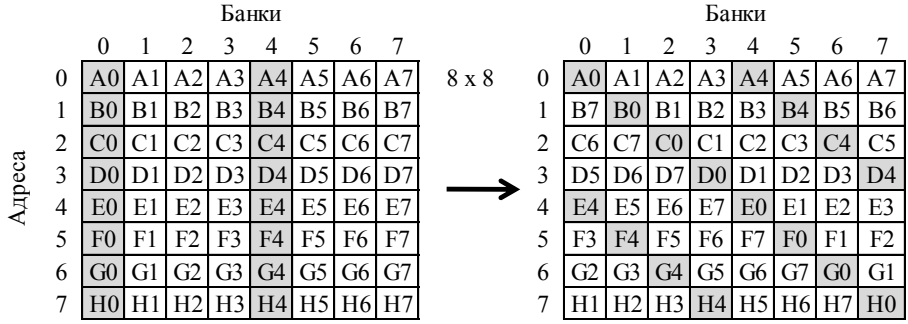

Рис. 1. Расположение матриц в векторной памяти с использованием прямой и скошенной адресации

Применение конвертируемых режимов адресации памяти позволяет аппаратно реализовать режимы доступа к памяти, необходимые для реализации основных алгоритмов ЦОС. Благодаря этому можно исключить из программ инструкции, производящие вычисление адресов, а если использование целочисленной арифметики не требуется для вычисления результатов алгоритмов, то такие инструкции можно исключить и из набора команд сопроцессора. К тому же, применение конвертируемых режимов адресации никогда не приводит к возникновению конфликтов банков при доступе к памяти, вследствие чего блоки определения конфликтов можно также исключить, а коммутаторы данных, входящие в состав блока доступа к разделяемой памяти, можно заменить на более простые блоки циклического сдвига.

Применение векторов переменной длины

Поддержка сопроцессором раздельного ветвления потоков команд внутри варпа не является обязательным условием для реализации рассматриваемых алгоритмов ЦОС. Инструкции ветвления в тестовых программах для

сопроцессоров M1 и M2 используются либо для организации циклов, одинаково обрабатываемых всеми потоками, либо для задания числа потоков, выполняющих алгоритм или его часть. Отсюда можно сделать вывод, что поддержка раздельного ветвления параллельных потоков команд для реализации рассматриваемых алгоритмов ЦОС вообще не требуется, и многопоточную модель исполнения можно заменить использованием векторов переменной длины.

Программа, работающая с векторами переменной длины, не является многопоточной, в отличие от программы, соответствующей стандарту OpenCL. Процессор выполняет один поток команд последовательно, где каждая команда управляет выполнением одинаковых операций над множеством элементов данных, т.е. используется архитектура типа SIMD.

Распространенные процессоры, в набор инструкций которых входят векторные операции, используют вектора фиксированной длины. Если требуется обработка более длинных векторов, приходится использовать множество инструкций в цикле. Для того чтобы избавиться от циклов в подобных случаях, целесообразно использовать процессор с поддержкой векторов переменной длины, которые позволяют выполнять циклы аппаратными средствами, без использования дополнительных инструкций.

Длина векторов в таких процессорах обычно задается один раз для группы инструкций, например, с помощью специального регистра. При заданной длине каждый вектор разделяется на множество фрагментов, каждый из которых обрабатывается физически параллельно, на разных вычислительных блоках процессора. Такой фрагмент далее называется **векторным элементом**.

Вектора (т.е. данные, с которыми работают инструкции) наиболее эффективно располагать в произвольно адресуемой статической памяти, шина данных которой имеет ширину, равную одному векторному элементу, а различные векторные элементы хранятся по разным адресам. Чтобы обеспечить достаточную пропускную способность памяти для выполнения инструкций (например, чтение 2 и запись 1 векторного элемента каждый такт), можно использовать память с несколькими портами, что приводит к увеличению площади, занимаемой памятью на кристалле, в расчете на 1 бит. Другим вариантом является разделение памяти на несколько страниц, что может создавать структурные конфликты и усложняет составление программы и планирование выполнения инструкций, однако является более эффективным реше-

нием с точки зрения соотношения производительности и площади, занимаемой на кристалле. В диссертации подробно описан способ построения планировщика выполнения инструкций при использовании такой многостраничной системы памяти и приведены аналитические расчеты производительности такого планировщика.

Предлагаемый способ построения планировщика основан на расчете времени блокировок, требуемых для разрешения зависимостей по данным между последовательными инструкциями в программе. Планировщик хранит информацию о нескольких инструкциях, находящихся в конвейере сопроцессора в каждый момент времени. При поступлении на вход планировщика новой инструкции производится проверка наличия зависимостей по данным между выполняемыми инструкциями и новой инструкцией. При наличии таких зависимостей производится расчет числа тактов, в течение которых требуется блокировать выполнение новой инструкции. Исходными данными для такого расчета являются заданная длина векторов и длительность обработки каждого векторного элемента новой и предшествующей инструкций. По истечении времени блокировки новая инструкция отправляется на выполнение, но только при условии, что вход вычислительного конвейера свободен, т.е. запуск выполнения предыдущих инструкций произведен полностью.

Применение векторов переменной длины дает возможность сделать время выполнения программ детерминированным, в отличие от планирования выполнения многопоточных программ, которое зависит от труднопредсказуемых событий. Эта характеристика крайне важна для систем обработки данных в реальном масштабе времени, к которым относятся многие ЦОС-системы.

Основным преимуществом использования векторов переменной длины вместо векторов постоянной длины является то, что инструкции, работающие с векторами переменной длины, заменяют внутренние циклы в программах. Это приводит к упрощению кода и отсутствию необходимости применения предсказания ветвлений и спекулятивного выполнения для достижения высокой эффективности работы сопроцессоров.

Применение инструкций, работающих с комплексными числами

При обработке сигналов часто применяются комплексные числа. В связи с этим многие специализированные устройства включают схемы, работающие с комплексными числами, а в архитектуру процессоров включают соот-

ветствующие дополнительные инструкции. Применение таких инструкций позволяет сократить число инструкций в программах в несколько раз. Используя набор инструкций сопроцессора M2, комплексное умножение с накоплением можно реализовать, используя 4 инструкции действительного умножения с накоплением, а основную операцию БПФ – «бабочку» – используя 8 различных инструкций, работающих с действительными числами. При наличии специальных команд в наборе инструкций сопроцессора эти и другие подобные операции могут быть реализованы при помощи всего одной инструкции. В целом, применение инструкций, работающих с комплексными числами, может сократить объем программ в 2–8 раз (до 10 раз по сравнению с набором инструкций, не включающим действительное умножение с накоплением). Важным следствием из сокращения числа инструкций является сокращение числа обращений к регистровой памяти. Сокращение числа операций чтения и записи напрямую отражается на энергопотреблении устройства. Помимо этого, применение таких инструкций позволяет повысить производительность модели сопроцессора M2 в 2,5 раза без увеличения количества вычислительных конвейеров, т.е. без потери эффективности работы.

В диссертации описаны 2 варианта построения многофункционального вычислительного блока, который может эффективно выполнять как операции над комплексными, так и над действительными числами. Первый вариант основан на построении конвейера с изменяемой структурой на базе обычных блоков, производящих сложение и умножение чисел формата с плавающей точкой. Глубина конвейера зависит от выполняемой операции и принимает значения от 2 стадий (для выполнения простой пересылки входных данных на выход) до 24 стадий (для выполнения умножения со сложением и вычитанием в комплексных числах).

Структурная схема такого вычислительного блока показана на рис. 2. При реализации наиболее сложной операции – комплексного умножения со сложением и вычитанием – блок функционирует следующим образом. Комплексные операнды A, B, C через ряд мультиплексоров в течение двух тактов передаются на блоки умножения MUL1 и MUL2 и блоки задержки. Затем результаты умножения поступают на блок сложения ADD1, и далее – на блоки сложения ADD2 и ADD3. Результаты сложения переупорядочиваются и выводятся из конвейера (OUT). При реализации более простых операций часть конвейера пропускается при помощи мультиплексоров.

Для управления таким конвейером используется специальный блок, состоящий из линии задержки, по которой проходят коды выполняемых инструкций, и 4 блоков комбинационной логики. Каждый блок комбинационной логики управляет рядом мультиплексоров на одной стадии конвейера, а также операциями, выполняемыми блоками сложения или умножения. На вход каждого такого блока подаются коды инструкций, запущенных столько тактов назад, сколько им необходимо для того, чтобы дойти до данной стадии конвейера при любых возможных вариантах структуры конвейера.

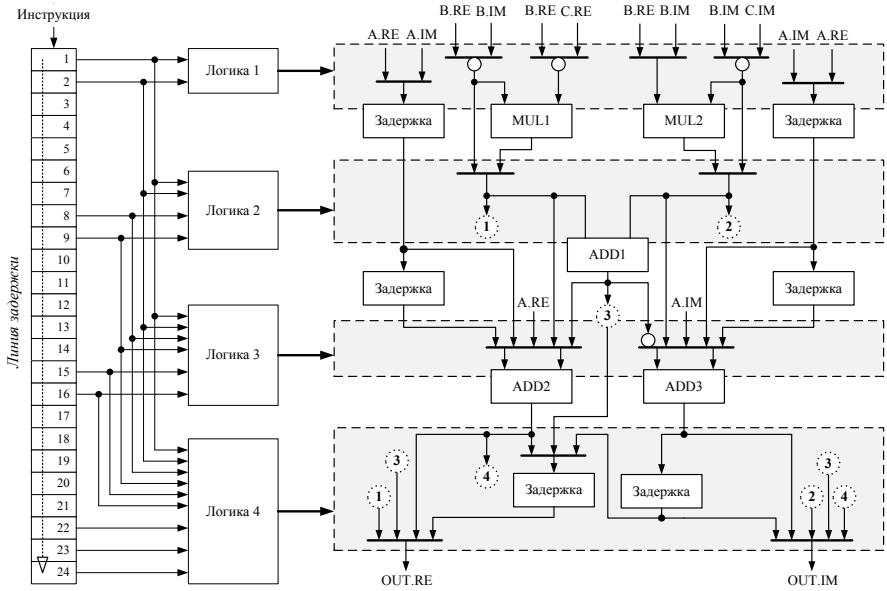

Рис. 2. Структурная схема вычислительного блока с изменяемой структурой

Второй вариант построения вычислительного блока основан на применении сумматора с тремя операндами. Применение такого блока может позволить повысить точность вычислений за счет отсутствия операций промежуточного округления. Проведенные на программной модели эксперименты показывают снижение погрешности вычисления БПФ на 7–14% при переходе от конвейера, состоящего из отдельных вычислительных блоков, к совмещенной схеме на базе сумматора 3 операндов.

Разработка высокопроизводительного векторного сопроцессора с поддержкой векторов переменной длины и потоковой модели вычислений

Сопроцессор М3 реализован в виде архитектурной модели и синтезируемой RTL-модели. Он поддерживает вычисления в комплексных числах и конвертируемые режимы адресации памяти. Помимо этого, в нем реализована поддержка векторов переменной длины и поддержка команд обработки потоков данных.

Структурная схема сопроцессора М3 с 4 вычислительными конвейерами представлена на рис. 3. Бо́льшую часть сопроцессора составляют три одинаковых страницы векторной памяти данных объемом по 32 КБ каждая. Значение 32 КБ было выбрано, чтобы имелась возможность реализации БПФ по алгоритму двумерного разложения на 4096 отсчетов, что предусматривает размещение всех исходных данных в одной странице. Порты памяти данных соединяются через коммутаторы с векторным вычислительным блоком и блоками ввода-вывода, которые обеспечивают интерфейс с контроллерами прямого доступа к памяти СнК. Запросы на чтение и запись в память поступают от планировщика вычислений и блоков ввода-вывода.

Сопроцессор М3 обрабатывает данные, загружаемые во внутреннюю память по двум 64-битным входным шинам, и выдает полученные результаты по одной выходной шине такой же разрядности. Допускается загрузка и выгрузка данных одновременно с обработкой другой задачи.

Сопроцессор имеет двухуровневую систему команд. Команды нижнего уровня (вычислительные команды) определяют вычисления над данными, находящимися во внутренней векторной памяти. Команды верхнего уровня (потоковые команды) управляют размещением данных в векторной памяти, передачей данных между сопроцессором и внешними устройствами, а также запуском вычислительных программ. Потоковые команды подаются в планировщик потоков от управляющего процессора СнК по входной шине команд. Этот блок определяет возможные зависимости между потоковыми командами и запускает выполнение команд тогда, когда это не приведет к неопределенным результатам (например, не допускается одновременная загрузка и выгрузка данных для одной области памяти).

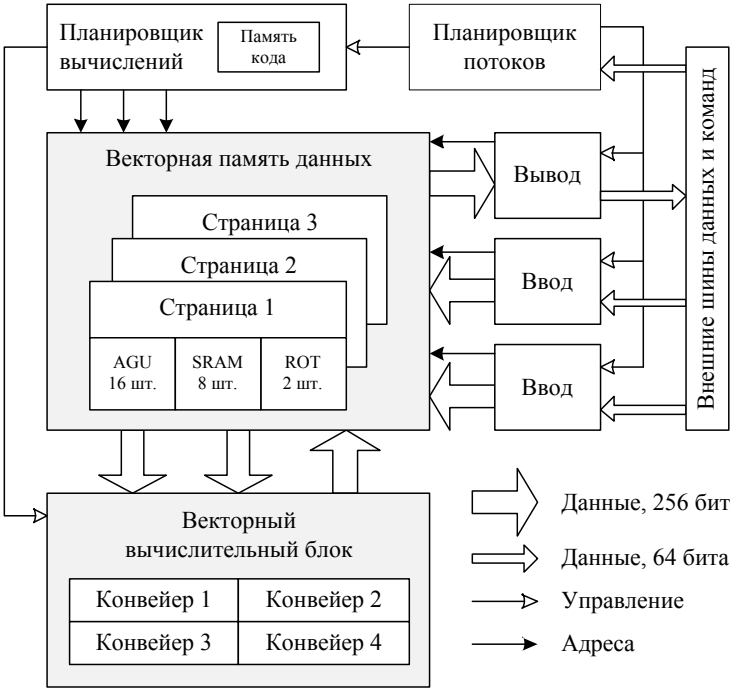

Рис. 3. Общая структура сопроцессора М3

Вычисления производятся в соответствии с программой, заранее загружаемой во внутреннюю память кода, которая находится в планировщике вычислений. Эта программа состоит из инструкций, которые могут описывать только последовательные вычисления; условных переходов в вычислительной программе быть не может. Инструкции работают с данными, находящимися в векторной памяти и представленными в виде векторов переменной длины. Каждый вектор в памяти данных называется **регистром**. Предусмотрено всего 10 вычислительных инструкций, которые выполняют операции только над числами формата с плавающей точкой. Тип данных – действительный или комплексный – задается при запуске программы и является общим для всех выполняемых инструкций.

Регистры разделены на 8 независимых сегментов. Каждый сегмент регистров может находиться в любой из трех страниц памяти данных по любому

адресу, кратному одному векторному элементу, и иметь размер, равный 2^k векторных элементов. У каждого сегмента может быть отдельно установлен режим адресации, который влияет на то, какие ячейки памяти соответствуют различным номерам регистров. Реализуется 5 режимов адресации, которые делятся на две группы: первая группа содержит простой, скалярный и сверточный режимы, а вторая – два матричных режима.

Простой режим. Память в простом режиме представляется в виде матрицы, где каждая строка является отдельным регистром. Регистры независимы друг от друга.

Скалярный режим предназначен только для чтения. При чтении регистра значение одной ячейки памяти используется во всех элементах прочитанного вектора. Такой режим предназначен для замены непосредственных констант в коде программы, например, коэффициентов внутри БПФ.

Сверточный режим предназначен только для прямой реализации алгоритма свертки. В этом режиме регистры занимают перекрывающиеся адреса в памяти, т.е. например элементы 1–7 регистра 0 - это то же самое, что элементы 0–6 регистра 1.

Прямой матричный режим и **транспонированный матричный режим** реализуют размещение в памяти матриц с использованием скошенной адресации. Прямой режим ставит в соответствие каждому векторному регистру строку матрицы, а транспонированный режим – столбец. Таким образом, транспонирование матрицы сводится к переключению режима адресации соответствующего сегмента.

Для сопроцессора М3 реализован ряд тестовых программ, включающих базовые алгоритмы ЦОС в действительных и комплексных числах. Данные о производительности вариантов сопроцессора М3 с различным числом конвейеров при выполнении этих программ приведены в табл. 1. Производительность сопроцессора М3 значительно превышает производительность моделей сопроцессоров М1 и М2 за счет применения вышеописанных архитектурных решений. Из приведенных данных также видно, что при решении большинства задач сопроцессор получает значительное ускорение при переходе на большее число конвейеров, т.е. его архитектура хорошо масштабируется. В диссертации приведены зависимости производительности сопроцессора М3 от глубины конвейера и от различных параметров планировщика вычислений (числа записей в таблице зависимостей и применения раннего запуска инструкций).

Таблица 1. Производительность сопроцессора М3 с различным числом конвейеров при реализации основных алгоритмических тестов

Алгоритм	Тип данных	Размер задачи	Средняя длина вектора	4 конвейера		8 конвейеров		16 конвейеров	
				Загрузка выч. блока	Кол-во тактов	Кол-во тактов	Ускорение отн. 4 конв.	Кол-во тактов	Ускорение отн. 8 конв.
Свертка	Компл.	32 x 32	32.0	52.1%	982	974	1%	970	0%
		128 x 32	128.0	98.3%	2084	1060	97%	982	8%
	Действ.	32 x 32	16.0	17.9%	715	713	0%	712	0%
		128 x 32	64.0	70.4%	727	719	1%	715	1%
БПФ	Компл.	64	8.0	40.7%	305	281	9%		
		128	11.0	62.2%	450	359	25%		
		256	16.0	89.5%	706	480	47%	427	12%
		512	21.9	95.0%	1466	770	90%	574	34%
		1024	32.0	97.6%	3130	1602	95%	838	91%
		2048	43.7	98.9%	6698	3386	98%	1730	96%
		4096	64.0	99.5%	14378	7226	99%	3650	98%
Умножение вектора на матрицу	Компл.	8 x 8	8.0	33.9%	112	109	3%		
		16 x 16	16.0	69.3%	202	153	32%	138	11%
		32 x 32	32.0	91.1%	571	330	73%	217	52%
		64 x 64	64.0	98.3%	2084	1083	92%	586	85%
	Действ.	8 x 8	4.0	12.0%	92				
		16 x 16	8.0	33.1%	139	129	8%		
		32 x 32	16.0	62.2%	251	209	20%	200	4%
		64 x 64	32.0	81.7%	656	411	60%	305	35%
Редук. "+"	Компл.	2048	22.8	45.8%	1169	672	74%	431	56%
Редук. "×"	Компл.	2048	22.8	86.3%	1240	728	70%	495	47%
	Действ.	2048	12.0	42.1%	672	431	56%	376	15%

Синтез RTL-модели сопроцессора М3 для реализации в составе «системы на кристалле» проводился с использованием САПР Cadence и библиотеки стандартных ячеек с технологическими нормами 40 нм. В табл. 2 приведены результаты синтеза RTL-модели сопроцессора М3 с 4 конвейерами и 96 КБ внутренней памяти данных.

Основную долю площади (86%), занимаемой сопроцессором М3 на кристалле, составляют блоки статической памяти данных; они же потребляет 42% мощности, и еще 46% мощности потребляет основной вычислительный блок. Из этого можно сделать вывод, что блоки, реализующие основные особенности архитектуры – переменную длину векторов, конвертируемые режимы адресации – не оказывают существенного влияния на аппаратные характеристики сопроцессора, но помогают эффективно загрузить вычислительный блок работой.

В диссертации приведено сравнение производительности рассматриваемых сопроцессоров с другими устройствами при вычислении БПФ. Сопроцессор М3 при реализации в составе «системы на кристалле» по указанной

выше технологии превосходит по производительности операций с числами формата с плавающей точкой рассмотренные серийные ЦСП в несколько раз. В частности, один из самых быстрых ЦСП, ADI TigerSHARC ADSP-TS201S с тактовой частотой 600 МГц, производит вычисление БПФ-1024 за 15,6 мкс, т.е. в 4,98 раза медленнее, чем сопроцессор М3.

Таблица 2. Результаты синтеза RTL-модели сопроцессора М3 с 4 конвейерами для реализации в составе СнК

Параметр	Значение
Максимальная тактовая частота	1 ГГц
Площадь, занимаемая на кристалле	1,88 мм^2
Количество логических ячеек	115 968
Потребляемая мощность (оценка синтезатора)	144,3 мВт
Макс. число операций с числами формата с плавающей точкой за такт	20
Пиковая производительность	20 GFLOPS
Время вычисления одного БПФ-1024	3,13 мкс
Производительность БПФ-1024	3.27×10^8 отсчетов/с

ЗАКЛЮЧЕНИЕ

Основной результат диссертации заключается в решении актуальной задачи повышения производительности систем ЦОС, работающих с числами формата с плавающей точкой, путем применения новых способов построения блоков сопроцессоров и специализированной модели исполнения программ, позволяющих получить высокую эффективность работы параллельных программируемых сопроцессоров при решении основных задач ЦОС. Рассматривалась реализация алгоритмов быстрого преобразования Фурье, вычисления свертки, умножения вектора на матрицу и редукции.

Частные научные результаты

1) Предложен способ построения системы векторной памяти, реализующей конвертируемые режимы адресации, которая позволяет размещать в памяти матрицы и обеспечивать параллельный доступ к элементам строк или столбцов матриц без транспонирования, а также реализовать другие режимы

адресации, характерные для алгоритмов цифровой обработки сигналов. По сравнению с векторной памятью, реализующей универсальную операцию косвенной векторной адресации, предлагаемая система характеризуется лучшей масштабируемостью, а также отсутствием необходимости производить программное вычисление адресов. При реализации на ПЛИС, такая система с 16 банками памяти требует в 2 раза меньше логических ячеек и может работать на тактовой частоте в 2,3 раза большей, чем система памяти с поддержкой косвенной векторной адресации.

2) Предложено использовать в программируемых сопроцессорах, предназначенных для ЦОС, векторные вычисления с поддержкой векторов переменной длины. По сравнению с использованием векторов фиксированной длины, характерных для большинства используемых в настоящее время процессоров, применение такой модели исполнения позволяет исключить внутренние циклы в программах. По сравнению с использованием многопоточной модели исполнения, характерной для устройств с поддержкой OpenCL, использование векторов переменной длины позволяет обеспечить детерминированное время выполнения программ, что важно при построении систем, работающих в условиях реального времени. В совокупности с применением системы векторной памяти, аппаратно реализующей требуемые режимы адресации, такая модель исполнения позволяет существенно сократить набор инструкций сопроцессора за счет исключения инструкций, используемых для вычисления адресов и организации циклов.

3) Разработан планировщик выполнения векторных инструкций для сопроцессора с поддержкой векторов переменной длины, основанный на расчете длительности блокировок, используемых для разрешения зависимостей по данным в программе. Такой планировщик обеспечивает высокое быстродействие сопроцессора в широком диапазоне параметров выполняемой программы (длина векторов, длина последовательностей независимых инструкций) и потребляет небольшое количество аппаратных ресурсов.

4) Предложен способ построения быстродействующего конвейерного арифметического блока, выполняющего составные вычислительные операции в действительных и комплексных числах формата с плавающей точкой, такие как умножение со сложением и вычитанием, путем различного соединения арифметических блоков, выполняющих по отдельности сложение и умножение действительных чисел того же формата. Блок, построенный таким образом, обеспечивает небольшую латентность простых операций (от 1

такта) и быструю смену типа выполняемых операций (более чем в половине случаев переключение операций выполняется без задержки). Также предложен способ построения конвейерного арифметического блока, основанного на использовании ассоциативного сумматора 3 чисел формата с плавающей точкой, который характеризуется большей точностью получаемых результатов, что приводит к уменьшению погрешности вычисления БПФ на 7–14%, но такой блок существенно сложнее в реализации.

Основной практический результат диссертации состоит в разработке и внедрении RTL-модели сопроцессора М3, в котором использованы решения, предложенные в диссертации. Такой сопроцессор обеспечивает возможность эффективной реализации основных алгоритмов цифровой обработки сигналов с применением чисел формата с плавающей точкой. В частности, при реализации алгоритма БПФ от 1024 точек производительность сопроцессора с 4 вычислительными конвейерами достигает 97,6% от максимальной, а при вычислении свертки в комплексных числах с размерностью ядра 32 точки – 98,3%. При увеличении числа конвейеров эффективность работы сопроцессора при достаточно больших размерах решаемых задач снижается незначительно.

Проведен логический синтез RTL-модели сопроцессора М3 и получена логическая структура сопроцессора с 4 вычислительными конвейерами и 96 КБ внутренней памяти данных, которая при изготовлении СБИС по технологическому процессу с проектной нормой 40 нм занимает на кристалле площадь 1,88 мм^2 и может работать при тактовой частоте до 1 ГГц. С учетом вышеуказанных показателей эффективности реализации алгоритмов, данный сопроцессор существенно (более чем в 4 раза) превосходит по производительности существующие серийные ЦСП.

Публикации по теме диссертации

Публикации в изданиях, рекомендованных ВАК РФ

1. Пантелеев А.Ю., Шагурин И.И., Деревянко Д.А. Применение технологии OpenCL для проектирования структуры СБИС векторных процессоров // Проблемы разработки перспективных микро- и наноэлектронных систем – 2010. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. – М.: ИППМ РАН, 2010. – С. 336–341.

2. Пантелеев А.Ю., Шагурин И.И. Применение конвертируемых режимов адресации для повышения производительности сопроцессоров цифровой обработки сигналов в составе многоядерной СнК // Проблемы разработки перспективных микро- и наноэлектронных систем – 2012. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. – М.: ИППМ РАН, 2012. – С. 389–394.

3. Пантелеев А.Ю. Планирование выполнения инструкций для векторных процессоров с переменной длиной векторов // Проблемы разработки перспективных микро- и наноэлектронных систем – 2012. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. – М.: ИППМ РАН, 2012. – С. 395–398.

4. Пантелеев А.Ю. Цифровая обработка сигналов на современных графических процессорах // Цифровая обработка сигналов. – 2012 – № 3. – С. 65–71.

Статьи и материалы конференций

5. Пантелеев А.Ю., Литвинов Е.И. Верификация и тестирование сложнофункциональных СБИС. Часть 2 // Электронные компоненты. – 2012. – № 8. – С. 101–105.

6. Пантелеев А.Ю., Литвинов Е.И. Верификация и тестирование сложнофункциональных СБИС. Часть 1 // Электронные компоненты. – 2012. – № 7. – С. 20–24.

7. Пантелеев А.Ю., Деревянко Д.А., Иванов П.Ю. Высокопроизводительный векторно-конвейерный сопроцессор для цифровой обработки сигналов // Микроэлектроника и информатика – 2012. 19-я Всероссийская межвузовская научно-техническая конференция студентов и аспирантов: Тезисы докладов. – М.: МИЭТ, 2012. – С. 185.

8. Пантелеев А.Ю., Шагурин И.И. Векторный сопроцессор для реализации DSP-алгоритмов в числах с плавающей точкой // Научная сессия МИФИ 2012: сборник научных трудов, том 1. – М.: МИФИ, 2012. – С. 80–81.

9. Пантелеев А.Ю., Деревянко Д.А. Организация векторной памяти DSP-процессоров для размещения матриц с прямым и транспонированным доступом // Научная сессия МИФИ 2012: сборник научных трудов, том 1. – М.: МИФИ, 2012. – С. 81.

10. Пантелеев А.Ю., Иванов П.Ю. Конвейерный блок вычислений в комплексных числах с плавающей точкой // Научная сессия МИФИ 2012: сборник научных трудов, том 1. – М.: МИФИ, 2012. – С. 81.

11. Пантелеев А.Ю. Масштабируемая архитектура векторного процессора для эффективной реализации БПФ // Научная сессия МИФИ 2011: сборник научных трудов, том 1. – М.: МИФИ, 2011. – С. 133.

12. Пантелеев А.Ю. Обработка расходящихся потоков в векторном процессоре, предназначенном для использования в составе СБИС типа «система на кристалле» // Научная сессия МИФИ 2010: сборник научных трудов, том 2. – М.: МИФИ, 2010. – С. 113–114.

13. Пантелеев А.Ю., Шалтырев В.А. Компилятор последовательного языка программирования в синтезируемый конечный автомат // Научная сессия МИФИ 2009: сборник научных трудов, том 2. – М.: МИФИ, 2009. – С. 77–78.

14. Пантелеев А.Ю., Шалтырев В.А. Процессорное ядро с изменяемым набором инструкций // Научная сессия МИФИ 2008: сборник научных трудов, том 8. – М.: МИФИ, 2008. – С. 192–194.