На правах рукописи

Хилько Дмитрий Владимирович

ИССЛЕДОВАНИЕ И РАЗРАБОТКА ПОТОКОВОЙ РЕКУРРЕНТНОЙ АРХИТЕКТУРЫ ДЛЯ ЭФФЕКТИВНОЙ РЕАЛИЗАЦИИ ПАРАЛЛЕЛИЗМА В ОБЛАСТИ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Специальность 2.3.2– «Вычислительные системы и их элементы»

АВТОРЕФЕРАТ

диссертации на соискание ученой степени

кандидата технических наук

Москва – 2023

Работа выполнена в Федеральном государственном учреждении «Федеральный исследовательский центр «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН)

Научный руководитель: **Степченков Юрий Афанасьевич**
кандидат технических наук, ведущий научный сотрудник ФИЦ ИУ РАН

Официальные оппоненты: **Бобков Сергей Геннадьевич**
доктор технических наук,
Институт проблем проектирования в микроэлектронике Российской академии наук, заместитель директора

Тюрин Сергей Феофентович
доктор технических наук, профессор,
заслуженный изобретатель РФ,
Пермский национальный исследовательский политехнический университет, профессор кафедры Автоматика и телемеханика

Ведущая организация: Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук»

Защита диссертации состоится 06 декабря 2023 года в 15 часов 00 минут на заседании диссертационного совета 24.1.224.04 при ФИЦ ИУ РАН по адресу: 119333, г. Москва, ул. Вавилова, 44, корп. 2.

С диссертацией можно ознакомиться в библиотеке ФИЦ ИУ РАН по адресу: 119333, г. Москва, ул. Вавилова, 44, корп. 2 и на сайте официальном сайте ФИЦ ИУ РАН http://www.frccsc.ru.

Отзывы на автореферат в двух экземплярах, заверенные печатью учреждения, высылать по адресу: 119333, г. Москва, ул. Вавилова, 44, корп. 2, ученому секретарю диссертационного совета 24.1.224.04.

Автореферат разослан _____ 2023 года.

Ученый секретарь
диссертационного совета 24.1.224.04 Р.В. Разумчик

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность исследования. Одна из ключевых характеристик любой вычислительной системы – это ее производительность. На сегодняшний день большинство вычислительных систем основаны на традиционной архитектуре, предложенной фон-Нейманом. Долгое время высокого показателя производительности фон-Неймановских систем удавалось достигать путем повышения тактовой частоты. Но уже на уровне частот выше 3 ГГц проблемы питания и охлаждения процессора становятся определяющими. Поэтому ведутся исследования в области разработки высокопроизводительных архитектур вычислительных систем. Одним из основных способов повышения производительности является реализация параллельных вычислений на различных уровнях: команд, данных и задач. Современные фон-Неймановские процессоры реализуют широкий набор средств реализации параллелизма уровня команд, а также состоят из множества вычислительных ядер. Однако, в основе традиционной архитектуры лежит последовательный вычислительный процесс, что приводит к высокой аппаратной избыточности механизмов реализации параллелизма. Таким образом, существует потребность в разработке новых нетрадиционных архитектур вычислительных систем.

Наиболее перспективной нетрадиционной архитектурой является потоковая архитектура, в которой поток данных имеет приоритет над потоком команд и является инициатором вычислений. Вычислительные системы, функционирующие под управлением потока данных, теоретически, могут обеспечить большую производительность параллельных вычислений по сравнению с фон-Неймановскими. Тем не менее, зарубежные исследователи потоковой архитектуры столкнулись с целым рядом проблем, которые не позволили построить эффективные вычислительные системы на ее основе[1]. Одной из возможных областей применения потоковых архитектур является цифровая обработка сигналов (ЦОС). Принципы потоковой архитектуры и требования со стороны алгоритмов ЦОС хорошо сочетаются друг с другом в приложениях, для которых характерна высокая степень внутреннего параллелизма[2].

Основной группой российских исследователей потоковой модели вычислений был коллектив ИПИ РАН под руководством академика В.С. Бурцева (И.К. Хайлов, М.В. Твердохлебов и др.). В настоящее время этот коллектив в ИППМ РАН под руководством академика А.Л. Стемпковского продолжает исследования в рамках этой проблематики и, в частности, занимается разработкой параллельной потоковой вычислительной системы «Буран». ППВС «Буран» является системой массового параллелизма и реализует мелкозернистый параллелизм. В тоже время, основной сдерживающей причиной применения потоковых архитектур для решения задач ЦОС является стоимостной фактор, который делает нецелесообразным непосредственное использование технических решений из области потоковых систем массового параллелизма.

Для реализации параллельных вычислений ограниченной размерности в области ЦОС была предложена потоковая архитектура на основе рекуррентно-потоковой модели вычислений[3]. Она была названа многоядерной потоковой рекуррентной архитектурой (МПРА), а ее прототип – гибридной двухуровневой архитектурой рекуррентного обработчика сигналов (ГАРОС) с традиционным процессором на верхнем (управляющем) уровне и рекуррентным операционным устройством (РОУ) на нижнем уровне. Реализация принципов рекуррентно-потоковой модели вычислений позволяет сократить число стадий обработки инструкции и существенно снизить избыточность тегированных данных. Рекуррентная архитектура реализует мелкозернистый параллелизм на уровне операндов

---

[1] Arvind and D.E. Culler, Data flow architectures, Ann. Review in Computer Science, 1 (1986), P. 225–253.

[2] Iiro Hartimo. DFSP: A Data Flow Signal Processor. IEEE Transactions on Computers, v. C-35, N 1, January 1986. P. 23–33.

[3] Ю.А. Степченков, В.С. Петрухин, А.В. Филин Рекуррентное операционное устройство для процессоров обработки сигналов // Системы и средства информатики. Вып. 11. М.: Наука, 2001. – С. 283–315.

и крупнозернистый параллелизм на уровне программ.

Результаты апробации исходного прототипа МПРА показали, что архитектура имеет высокий потенциал эффективности, но ее существующая структура и набор функциональных возможностей не удовлетворяют требованиям производительности для задач ЦОС реального времени. Поэтому построение высокоэффективной потоковой рекуррентной архитектуры для решения задач ЦОС реального времени, а также разработка элементов методологии программирования и отладки ГАРОС (как самой архитектуры, так и специализированного программного обеспечения), являются необходимыми и актуальными задачами.

Объект исследования: рекуррентно-потоковая модель вычислений и архитектура на ее основе.

Предмет исследования: структурная организация и алгоритмы функционирования ключевых компонент новой архитектуры, а также методы и средства ее программирования и отладки.

Цель диссертационной работы состоит в разработке теоретических, алгоритмических, программных и модельных решений для создания прототипа устройства рекуррентного обработчика сигналов, который обладает требуемым уровнем производительности для решения задач ЦОС реального времени.

Для достижения поставленной цели необходимо:

1) Разработать теоретические основы программируемости ГАРОС, которые включают в себя: методики и алгоритмы реализации различных этапов разработки и отладки ПО; программную и аппаратную поведенческие модели архитектуры для проведения испытаний; набор средств аппаратно-программного моделирования и отладки архитектуры.

2) Разработать структурные элементы архитектуры, методы и алгоритмы их функционирования, которые позволят: эффективно реализовать поддержку параллелизма на различных уровнях; минимизировать избыточность тегированных данных; достичь требуемого уровня производительности для задач ЦОС реального времени.

3) Осуществить испытания всех разработанных средств путем реализации демонстрационной задачи распознавания изолированных слов и комплекта типовых алгоритмов ЦОС для подтверждения эффективности полученных результатов работы путем сравнения с современным высокопроизводительным сигнальным процессором.

Методы исследования. В работе используются методы системного анализа, теории алгоритмов, теории языков программирования, имитационное моделирование на программном и аппаратном уровнях, методы разработки и тестирования Test-Driven Development для программных средств и Assertion-Based Design для аппаратных средств.

Основные положения, выносимые на защиту:

1) методы и алгоритмы организации вычислительного процесса позволяющие достичь требуемого уровня производительности прототипа архитектуры для задач ЦОС реального времени;

2) структурные элементы архитектуры, а также методы и средства аппаратной поддержки алгоритма быстрого преобразования Фурье (БПФ) позволяющие минимизировать избыточность тегированных данных, которая является ключевой проблемой потоковых систем, и использовать дорогие элементы памяти рациональным образом;

3) элементы методологии программирования и отладки ГАРОС, позволяющие организовать полноценный и эффективный процесс разработки и отладки ПО для прототипа архитектуры, охватывающие большую часть этапов жизненного цикла ПО и позволяющие существенно сократить непроизводительные затраты разработчиков путем частичной автоматизации процессов верификации и валидации.

Научная новизна. Все результаты диссертационной работы являются новыми.

1) Впервые предложены методы и алгоритмы организации суперскалярных вычислений на уровне микроархитектуры вычислительных ядер ГАРОС, учитывающие специфику представления тегированных самодостаточных данных и процесса рекуррентной развертки. Разработанные решения позволяют использовать вычислительные ядра в двух, трех, а в некоторых случаях и в четырех задачном режиме.

2) Впервые предложены методы и алгоритмы реализации аппаратной поддержки алгоритма БПФ, которые учитывают специфику хранения и обработки тегированных самодостаточных данных и суперскалярность вычислительных ядер ГАРОС. Разработанные решения позволяют использовать вычислительные ядра в четырех задачном режиме на протяжении всего процесса вычисления БПФ, а также обеспечивают рациональное использование памяти самодостаточных данных путем максимального снижения их избыточности.

3) Впервые разработаны элементы методологии программирования и отладки ГАРОС, включающие в себя: методики и алгоритмы, комплект моделей и инструментов аппаратно-программного моделирования. Разработанный набор инструментов позволил успешно реализовать задачу распознавания изолированных слов и синтезировать ПЛИС прототип ГАРОС, который позволяет решать эту задачу в реальном времени.

Соответствие паспорту специальности. Диссертационное исследование соответствует следующим пунктам паспорта специальности 2.3.2 – «Вычислительные системы и их элементы»:

– Разработка научных методов и алгоритмов организации арифметической, логической, символьной и специальной обработки данных, хранения и ввода-вывода информации;

– Разработка научных методов и алгоритмов организации параллельной и распределенной обработки информации, многопроцессорных, многомашинных и специальных вычислительных систем.

Теоретическая и практическая значимость. Разработаны методы и алгоритмы организации вычислений в рамках МПРА, которые позволили создать эффективный прототип ГАРОС для решения задач ЦОС в реальном времени. Разработаны элементы методологии программирования и отладки ГАРОС, которые позволили организовать полноценный и эффективный процесс программирования и отладки полученного прототипа. Разработанный набор инструментов может быть использован в качестве основы для дальнейшего развития рекуррентных архитектур. Создан ПЛИС прототип ГАРОС, позволяющий эффективно решать задачу распознавания изолированных слов, а также ряда других типовых задач в области ЦОС (различные виды цифровых фильтров, частотный анализ при помощи алгоритма БПФ, алгоритм Витерби кодирования сигнала, поиск минимума/максимума и др.). Полученные результаты могут служить основой для разработки других вычислительных устройств на отечественной элементной и архитектурной базе.

Достоверность результатов обеспечивается:

– корректностью применения выбранных моделей, методов и алгоритмов для разработки, программирования и отладки архитектур вычислительных систем и их прототипов;

– корректностью исходных данных демонстрационной задачи распознавания изолированных слов, ограничений и допущений при проведении моделирования;

– согласованностью результатов программного моделирования, аппаратного моделирования и натурного эксперимента с ПЛИС прототипом и исходными данными демонстрационной задачи.

Апробация работы. Основные результаты диссертации докладывались и обсуждались на следующих международных конференциях и симпозиумах:

1) IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus (Санкт-Петербург, 2016-2018 гг.; Москва, 2019-2020 гг.; Санкт-Петербург, 2021-2023 гг.);

2) Проблемы разработки перспективных микро- и наноэлектронных систем, МЭС2 (Москва, Зеленоград, 2016-2022 гг.);

3) IEEE East-West Design & Test Symposium, EWDTS (Ереван, 2016 г.; Батуми, 2019 г.; Батуми, 2021 г.);

4) International Conference Engineering Technologies and Computer Science, EnT (Москва, 2020-2021 гг.).

Результаты диссертационной работы реализованы в виде комплекта моделей, элементов методологии программирования и отладки ГАРОС, а также комплекта специализированного ПО для проведения испытаний в ходе выполнения НИР «Информационные, управляющие и телекоммуникационные системы» в рамках государственного задания № 0063-2019-0010 по направлению «Концептуальные и методологические основы создания семейства потоковых самосинхронных процессоров и средств поддержки их проектирования». ПЛИС прототип был синтезирован в ходе выполнения гранта РНФ № 19-11-00334 «Инновационная архитектура самосинхронных цифровых сигнальных процессоров, управляемых потоком данных». Программный комплекс моделирования и отладки «ПК ПОТОК» был разработан в рамках проекта № 075-15-2020-799 «Методы построения и моделирования сложных систем на основе интеллектуальных и суперкомпьютерных технологий, направленные на преодоление больших вызовов».

Личный вклад. Все выносимые на защиту результаты получены лично автором. В работах [3, 8, 13, 23, 26] автором разработаны методы снижения избыточности тегированных данных, методы и средства аппаратной поддержки алгоритма БПФ. В работах [4, 8, 9, 13, 14, 17, 18, 20, 21, 22, 26] автором предложены структурные элементы прототипа архитектуры и механизмы их функционирования. В работах [1, 2, 5, 6, 7, 10, 11, 15, 16, 19, 24, 25] автору принадлежат: методы и средства моделирования и отладки прототипа архитектуры, имитационная модель ГАРОС и отдельные функциональные блоки аппаратной модели ГАРОС, элементы методологии программирования и отладки ГАРОС (в составе методик, алгоритмов, технологии, архитектуры программного комплекса «ПК ПОТОК» и его основных компонент), результаты испытаний, верификации и валидации прототипа ГАРОС. Остальные работы, опубликованные в соавторстве, выполнялись при непосредственном участии автора.

Публикации. Результаты диссертации изложены в 36 печатных работах, 24 из которых изданы в журналах, рекомендованных ВАК. Получено 11 свидетельств о регистрации программ для ЭВМ и 3 патента Российской Федерации на изобретение.

Структура и объем диссертации. Диссертация состоит из введения, четырех глав, заключения, списка литературы и приложения. Общий объем диссертации – 200 стр., в том числе 156 стр. основного текста, одно приложение, 35 иллюстраций и 16 таблиц в основном тексте. Список литературы состоит из 116 наименований.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** кратко изложено основное содержание диссертационной работы, цель которого – дать общее представление о решаемой проблеме, задачах, их научной новизне, теоретической и практической значимости, методах исследования и апробации результатов.

В **первой главе** исследуются основные особенности рекуррентной архитектуры и проблемы реализации ее прототипа в качестве высокопроизводительного цифрового сигнального процессора (ЦСП).

Зарубежные и отечественные исследования в области архитектур ЦСП показывают, что одним из наиболее эффективных методов повышения производительности, помимо увеличения числа вычислительных ядер, является совершенствование их микроархитектуры. В частности, наилучшие показатели достигаются при достижении оптимального уровня параллелизма уровня инструкций (instruction level parallelism – ILP). Большинство современных ЦСП от ведущих производителей имеют характеристику ILP равную 4 или 8. Исследования в этой области показали, что реализация слишком высокого уровня ILP также не имеет смысла ввиду чрезмерного усложнения аппаратных решений и значительного снижения энергоэффективности. Методами реализации ILP являются: конвейерная обработка инструкций; предсказание переходов; спекулятивные вычисления; суперскалярная микроархитектура; внеочередное исполнение инструкций; переименование регистров; VLIW-архитектуры; SIMD-инструкции. В рамках диссертационного исследования соискатель сосредоточил усилия над разработкой методов конвейеризации и суперскалярности.

Работы в области разработки потоковой модели вычислений выделяют такие преимущества данной модели, как: отсутствие счетчика команд и глобально обновляемой памяти; «естественное извлечение параллелизма» за счет использования принципов функционального программирования; естественная поддержка механизма внеочередного исполнения инструкций. Именно поэтому, потоковая модель вычислений имеет столь высокий потенциал эффективной реализации параллелизма. Однако, исследователи выделяют и целый ряд нерешенных (или решенных не в полной мере) вопросов: (1) избыточная организация памяти токенов (тегированных данных); (2) построение структур данных; (3) балансировка вычислительной нагрузки; (4) конвейеризация последовательных вычислений; (5) сложность реализации высокопроизводительной микроархитектуры; (6) обработка константных данных; (7) организация циклических вычислений. Те решения данных вопросов, которые были предложены, оказались недостаточно эффективными.

Также в главе вводится основная терминология и рассматриваются ключевые принципы новой рекуррентно-потоковой модели вычислений, разработанной Ю.А. Степченковым, В.С. Петрухиным, А.В. Филиным, и многоядерной потоковой рекуррентной архитектуры, которая является одной из возможных реализаций данной модели и предназначена для организации параллельных вычислений ограниченной размерности. Основным методом организации вычислительного процесса в рекуррентно-потоковой модели является графодинамический метод, который сводится к построению рекуррентно-динамических графов. Вычислительный процесс, организованный таким способом, называется рекуррентно-динамическим соответственно.

Суть метода заключается в применении некоторого функционального оператора F к текущему поколению графа $g(i)$, в результате которого формируется граф $g(i+1)$, и так далее. Цепочка таких преобразований формирует последовательность графов, которая называется графовой траекторией. Следовательно, графовая траектория – это исполняемая программа, алгоритм которой может быть представлен в виде последовательности графов $\{g(0), g(1), g(2), \ldots\}$ и рекуррентно сворачивается в начальный граф $g(0)$. Граф $g(0)$ кодирует начальное условие рекуррентной саморазвертки и называется рекуррентной цепочкой. Совокупность рекуррентных цепочек является исполняемой программой, называемой капсулой. Обобщая графодинамический метод до уровня задачи, получим представление решаемой задачи в виде совокупности капсул – капсульный стиль программирования.

Каждая капсула является потоковой структурой данных, аналогичной I-структуре. Следовательно, капсула является предлагаемым решением вопроса (2) в МПРА. Реализуется графодинамический метод за счет двух основных принципов: самодостаточность и рекуррентность.

Принцип самодостаточности заключается в логическом и физическом объединении элемента данных и соответствующей ему рекуррентной цепочки в элемент самодостаточных данных (ЭСД), называемый также операндом. Множество операндов образуют единый поток данных, в отличие от классической потоковой модели, которая подразумевает наличие отдельного потока данных и отдельного потока инструкций. Данный подход позволяет сократить до теоретически возможного минимума количество этапов обработки инструкций, за счет устранения необходимости постоянной выборки и дешифрации инструкций. Таким образом, принцип самодостаточности эффективно решает вопрос (4).

Принцип рекуррентности заключается в построении и аппаратной реализации такого оператора *F*, который обеспечит сжатие (рекуррентную свертку) потокового графа алгоритма в граф *g(0)* и распаковку (рекуррентную саморазвертку) графовой траектории с помощью специального устройства Преобразователя тегов (ПТ). В зависимости от того, насколько «удачно» будет построен оператор *F*, объем исполняемой программы может быть сжат вплоть до предельного значения в один операнд. В рамках МПРА реализован универсальный оператор, обеспечивающий развертку рекуррентной цепочки в графовую траекторию длины 3. Таким образом, принцип рекуррентности эффективно решает вопрос (1) и частично вопрос (3) за счет уменьшения общего числа инструкций. В дополнение к сжатию инструкций в МПРА реализуется механизм сжатия и упаковки нескольких элементов данных в ЭСД. Это позволяет использовать одну и ту же рекуррентную цепочку для некоторой последовательности обрабатываемых данных исходя из особенности выполняемого алгоритма. В области ЦОС число алгоритмов конечно, знание их особенностей позволяет применять различные методы сжатия данных, что невозможно для произвольного алгоритма. Комбинирование методов сжатия данных и сжатия инструкций в МПРА достигается еще большая эффективность решения вопроса (1).

В соответствии с выбранными способами организации памяти и управления вычислительным процессом, существующие архитектуры вычислительных систем могут быть классифицированы на три принципиальных класса: управляемые статическим потоком инструкций (Control-Flow/Static–CF/S); управляемые статическим потоком данных (Data-Flow/Static –DF/S); управляемые динамическим потоком данных (Data-Flow/Dynamic–DF/D). На рисунке 1 дано сравнение данных классов с точки зрения требуемых затрат памяти для хранения программы и данных, на рисунке 2 – сравнение по организации конвейера обработки инструкции.

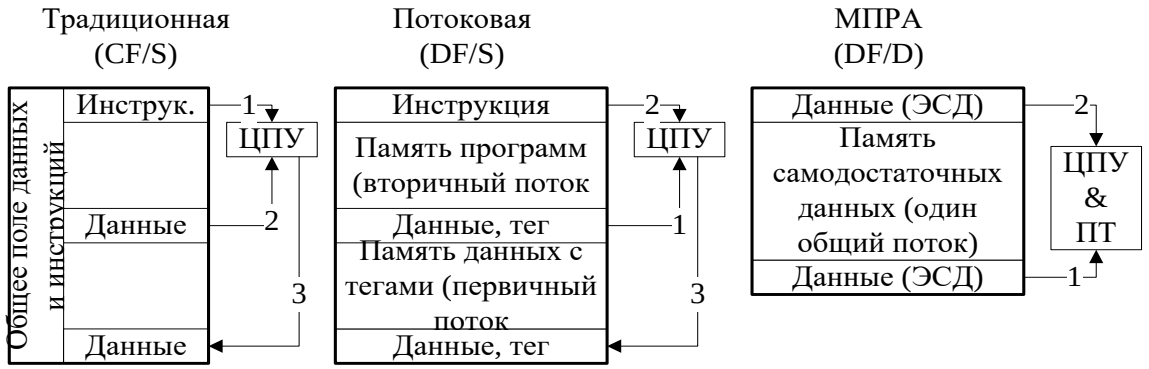

Рисунок 1 – Организация памяти в сравниваемых классах

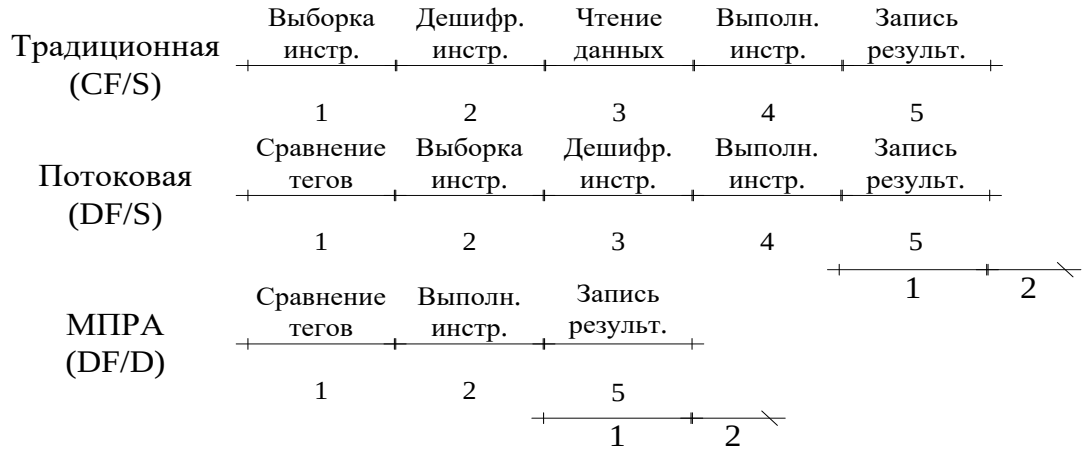

Рисунок 2 – Выполнение инструкций в сравниваемых классах

Для решения вопросов (6) и (7) в МПРА введены различные типы памятей констант, а также механизмы организации переходов и поддержки итеративных вычислений. Но не предлагается решений для вопроса (5). Кроме того, капсульный стиль программирования и модель программирования ГАРОС реализуют уникальный способ представления программ, для которого отсутствуют методы и подходы реализации, что делает разработку капсул крайне сложной задачей.

На основе результатов исследования и анализа осуществляется постановка целей и задач диссертационного исследования.

Вторая глава посвящена основным результатам разработки отдельных компонент ГАРОС и механизмов их функционирования, которые позволили достичь требуемого уровня производительности для задач ЦОС реального времени.

Для реализации высокопроизводительной микроархитектуры автором была существенно переработана структура вычислительных ядер ГАРОС (Вычислителей): значительно расширен регистровый файл путем добавления 5 регистров; добавлены новые соединения между функциональными блоками Вычислителей, что позволило расширить возможности организации их суперскалярного функционирования. Механизмы функционирования новых элементов были специфицированы и формализованы в виде алгоритмов. В соответствии с результатами анализа и разработанной структурой была доработана система команд.

Микроархитектура Вычислителя содержит 4 блока, которые могут функционировать одновременно: АЛУ, АУ40, Умножитель и Блок сдвига/округления. Вычислитель теоретически может выполнять до четырех инструкций одновременно и, следовательно, является четырех задачным. Однако входной интерфейс Вычислителя позволяет получить на входе только два входных данных и 2 инструкции. Данное ограничение не удалось преодолеть в рамках данного исследования, поэтому Вычислитель рассматривается как двух задачный, хотя в отдельных схемах может выполнять 3 или 4 инструкции.

Основной метод организации суперскалярных вычислений заключается в передаче на L- и R- входы Вычислителя двух различных кодов операций, которые должны быть совместимы между собой. Под совместимостью понимается отсутствие конфликтов доступа к регистрам и исполнительным блокам (например, операции MULи MAC несовместимы между собой, т.к. требуют для исполнения умножитель).

Вторым методом организации суперскалярных вычислений является поддержка двух схем вычислительного процесса. Первая схема (или первый тип суперскалярных вычислений) является типовой и заключается в одновременном исполнении двух различных инструкций. Результат одной из них помещается в один из регистров [A], [B] или [C], а результат второй – отправляется на E-шину в качестве выходного результата. Вторая схема (или второй тип суперскалярных вычислений) заключается в последовательном исполнении

двух инструкций. При этом результат первой инструкции используется в качестве входного данного для второй инструкции.

Методы специфицированы и формализованы в виде алгоритмов. На рисунке 3 приведена разработанная структура Вычислителя.

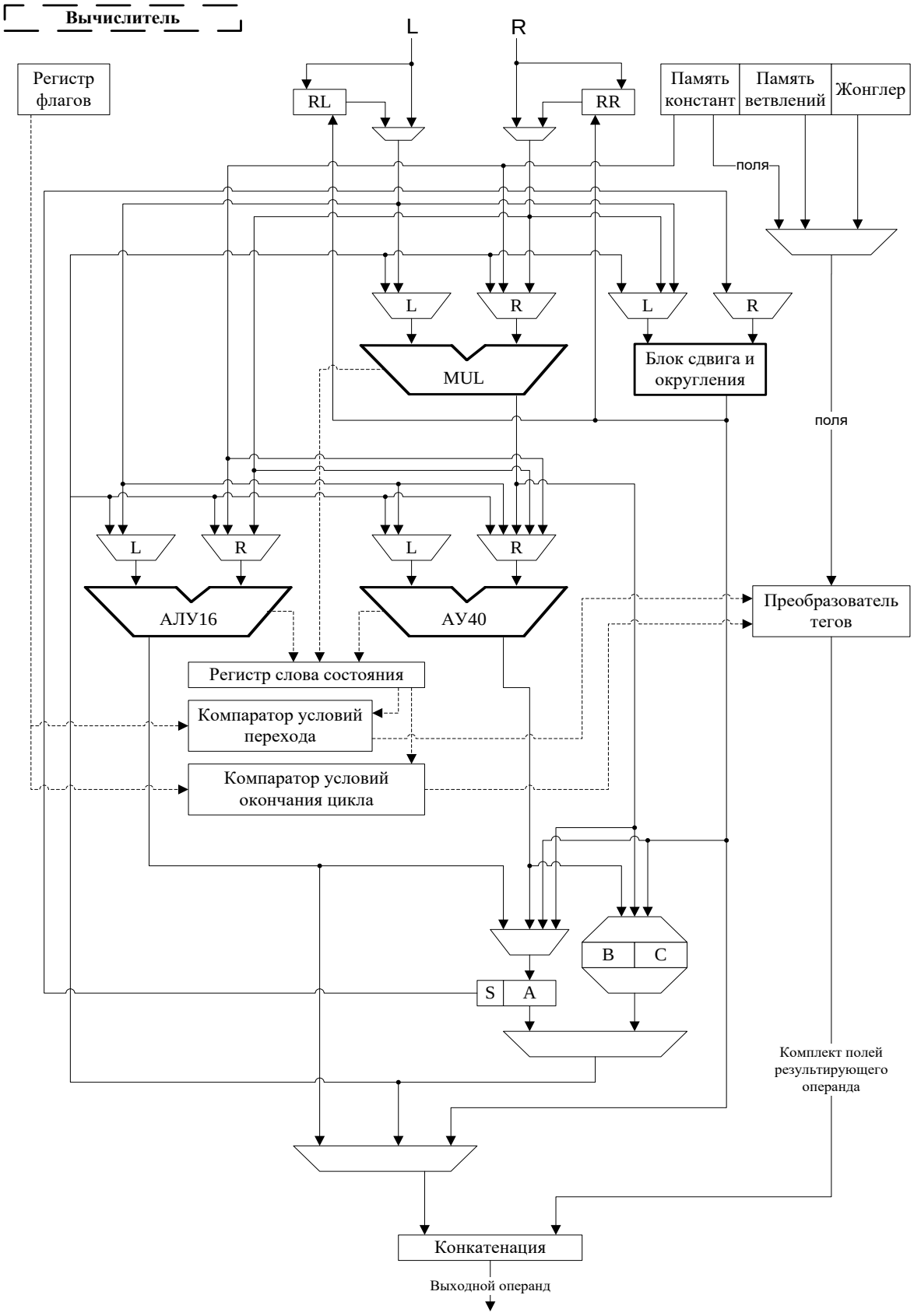

Рисунок 3 – Структура Вычислителя

Наиболее значимым результатом является разработка усовершенствованных средств аппаратной поддержки вычисления алгоритма БПФ, которая затронула все ключевые компоненты архитектуры. Современные высокопроизводительные ЦСП предоставляют средства аппаратной поддержки БПФ. Разработанные средства поддерживают алгоритм БПФ по основанию 2 с прореживанием по времени.

Первым ключевым элементом аппаратной поддержки БПФ является инструкция «Butterfly». На рисунке 4 представлена схема четырех-стадийной инструкции.

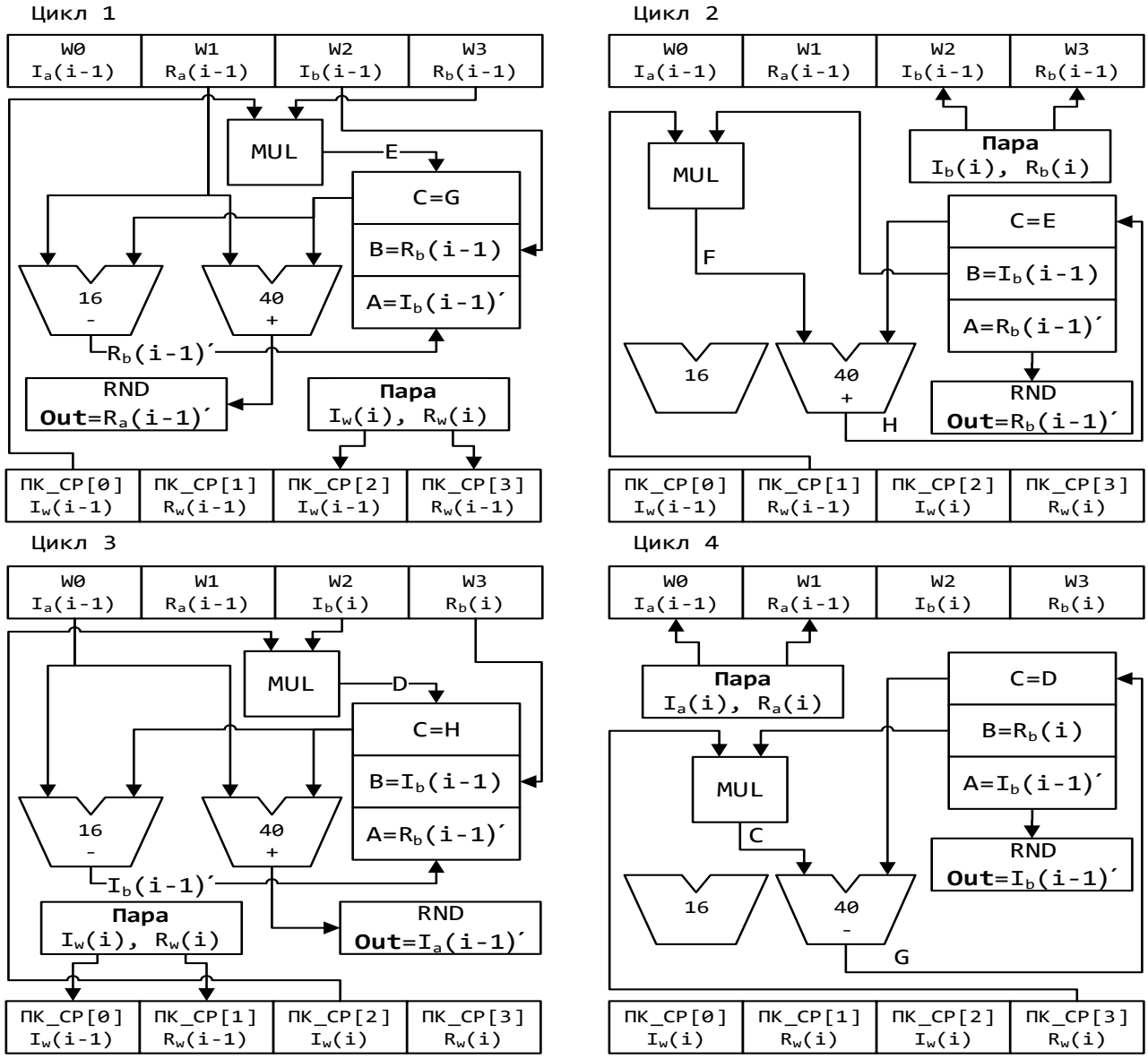

Рисунок 4 – Схема инструкции «Butterfly»

Данная инструкция на аппаратном уровне интерпретируется вычислительными блоками, как готовый четырех-стадийный сценарий функционирования. Для каждой стадии выполнения инструкции определена своя схема вычислений. Данный подход аналогичен эвристическому алгоритму, с той лишь разницей, что схема вычислений определена заранее, а не планируется компилятором или планировщиком задач.

Вторым ключевым элементом является дополнительный блок памяти, размещенный на уровне Буферной памяти ГАРОС. Данный блок включает в себя разделы для хранения действительных и мнимых частей поворотных коэффициентов, а также разделы для действительных и мнимых частей отсчетов. Блоки памяти отсчетов используются для хранения входных, промежуточных и выходных отсчетов (in-place реализация). На рисунке 5 представлена структура дополнительного блока памяти и элементов его управления.

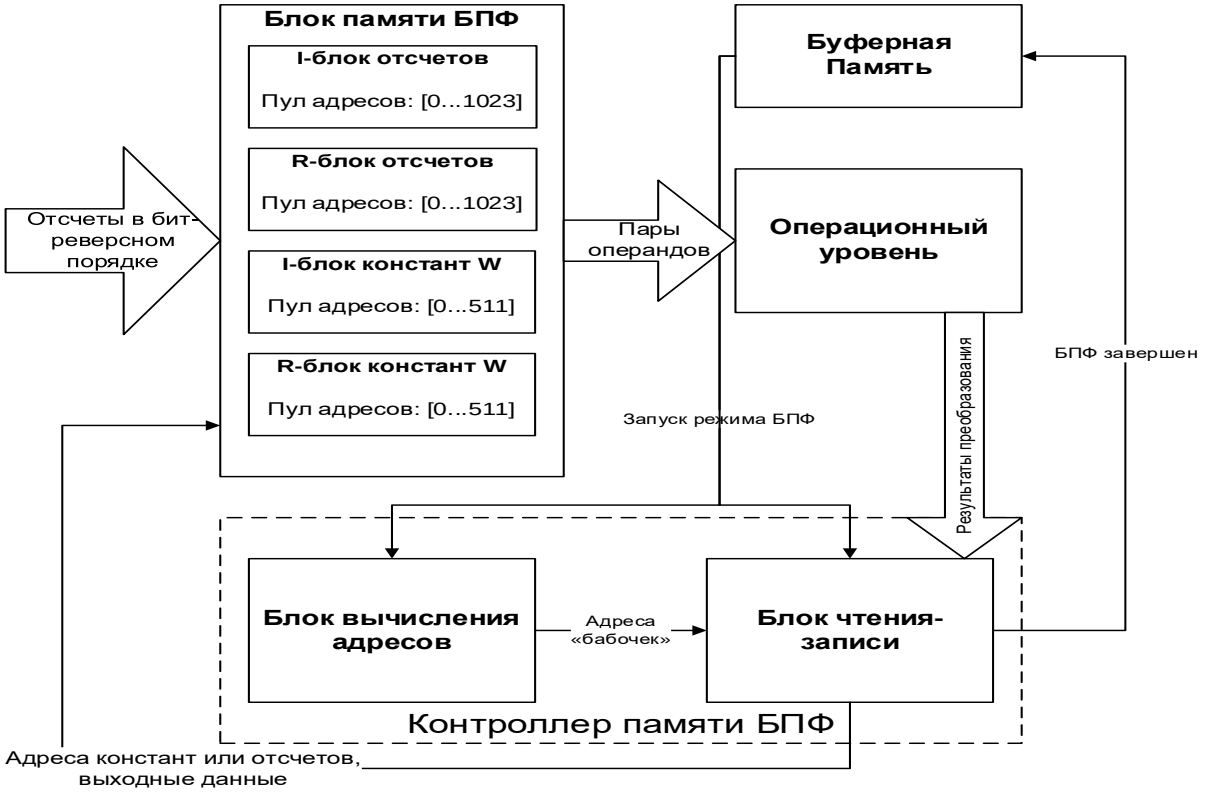

Рисунок 5 – Архитектура средств поддержки БПФ

Усовершенствованные средства поддержки БПФ имеют на 20% ниже аппаратные издержки и вычисляют типовой 256-точечный БПФ по основанию 2 на 12% быстрее исходной аппаратной реализации. В таблице 1 представлены результаты синтеза ПЛИС прототипа ГАРОС, реализующего разработанные средства поддержки БПФ.

Таблица 1 – Результаты синтеза средств поддержки БПФ

Имя узла в иерархии компиляции		**ALMs**	**ALUTs**	**Registers (bits)**	**Memory (bits)**	**M10Ks Blocks**	**DSP Blocks**	**Описание**
ros:rou	Initial	72765,2	84524	43996	147148	52	4	РОУ целиком
	Updated	61917,2	79973	28298	229068	62	4	
buffer_storage:c_BM		18308,9	24454	10989	141900	32	0	Буферная память
		15426,9	21568	11131	223820	42	0	
rou:c_rou		54396	60061	32887	5248	20	4	ROU без учета БП
		46431,2	58394	17042	5248	20	4	
compdev:c_compdev_0		4423,5	5671	576	0	0	1	Одно ядро Вычислителя
		5203,4	6957	614	0	0	1	
distributor:c_distributor		16808,3	21669	4481	0	0	0	Распределитель
		16431,4	21483	4481	0	0	0	
Относительно предыдущей версии		-15%	-5%	-36%	+56%	+19%	0%	Увеличение/снижение издержек

В **третьей главе** приводятся основные результаты разработки элементов методологии программирования и отладки ГАРОС.

Реализация методов и алгоритмов, представленных в главе 2, привела к тому, что существующий формат операндов потерял свою актуальность. Автором был проведен анализ несоответствий, на основе результатов которого был существенно переработан формат операндов. Изменилась разрядность операндных слов и теговых полей, кодировки теговых полей, компоновки различных типов операндов. В результате было специфицировано 35 типов операндов и более 100 различных теговых полей.

Основным теоретическим результатом данной работы является решение проблем программируемости МПРА. В диссертационной работе предлагаются элементы методологии программирования (см. рисунок 6), применение которых позволяет перейти от математической постановки к внутреннему представлению самодостаточных данных. Этот внутренний формат называется капсулой, а стиль программирования – парадигмой капсульного программирования. Ключевым этапом данной методологии является Этап III. Набор методик и алгоритмов, предложенных для реализации данного этапа, позволил организовать эффективный итеративный процесс разработки капсул в рамках парадигмы капсульного программирования. Интеграция данного процесса в ход разработки ГАРОС позволил довести ее до успешного синтеза ПЛИС прототипа.

Рисунок 6 – Структура рекуррентно-потоковой методологии программирования

На основе спецификации ГАРОС и ее новых разработанных элементов, представленных в главе 2, автором была разработана программная имитационная модель и система имитационного моделирования. Все механизмы, представленные в спецификации ГАРОС, были формализованы в виде алгоритмов и реализованы в программной модели. По завершению ее отладки данные алгоритмы были использованы в качестве основы для описания на языке VHDL структурных элементов аппаратной модели ГАРОС и их поведения. Под руководством Ю.Г. Дьяченко автор принимал активное участие в разработке отдельных компонент аппаратной VHDL-модели. Для данной модели автор также разработал подсистему аппаратного моделирования

В рамках подэтапа III-2 методологии программирования описывается методика капсульного программирования. Согласно данной методике, процесс разработки капсулы требует построение потокового графа алгоритма и последующее распределение его узлов по доступным ресурсам архитектуры. Автоматизация данного распределения, по сути, означает разработку компилятора и требует отдельного исследования. Поэтому данный шаг выполняется вручную. Для упрощения ручного распределения ресурсов была разработана графовая нотация представления вычислительного процесса, названная граф-капсулой.

По мере развития средств аппаратно-программного моделирования, а также накопления опыта разработки ПО для МПРА, были аккумулированы эвристические правила эффективного распределения ресурсов РОУ в типовых ситуациях программирования тех или

иных алгоритмов. Для данных типовых ситуаций были построены шаблоны фрагментов капсул, обеспечивающих это эффективное распределение. Кроме того, средства моделирования стали предоставлять максимально полную информацию об использовании ресурсов РОУ, что позволило разработать программу автоматизированного построения граф-капсул.

Для реализации Этапа IV методологии был организован процесс разработки и тестирования ГАРОС, схема которого представлена на рисунке 7. Данный процесс был реализован автором в рамках подсистемы автоматизированной верификации и валидации.

Совокупность всех разработанных программно-аппаратных средств была интегрирована в программный комплекс моделирования и отладки потоковой рекуррентной архитектуры «ПК ПОТОК.

Четвертая глава посвящена результатам испытаний всех предложенных методов, алгоритмов и механизмов развития прототипа ГАРОС в области ЦОС. Испытания были проведены как на модельном уровне, так и на уровне синтезированного ПЛИС прототипа.

Наиболее важным этапом всего исследования являлась разработка комплекта капсул демонстрационной задачи. Именно эти капсулы в совокупности использовались для анализа производительности и отладки ГАРОС как целого, а также для доказательства корректности его функционирования. Решенная задача распознавания изолированных слов использовалась в качестве эталонной. Поэтому так важно было разработать и апробировать рекуррентно-потоковую методологию программирования. В результате применения данной методологии демонстрационная задача была декомпозирована на ряд подзадач, каждая из которых была распределена для решения как на управляющем уровне, так и на РОУ.

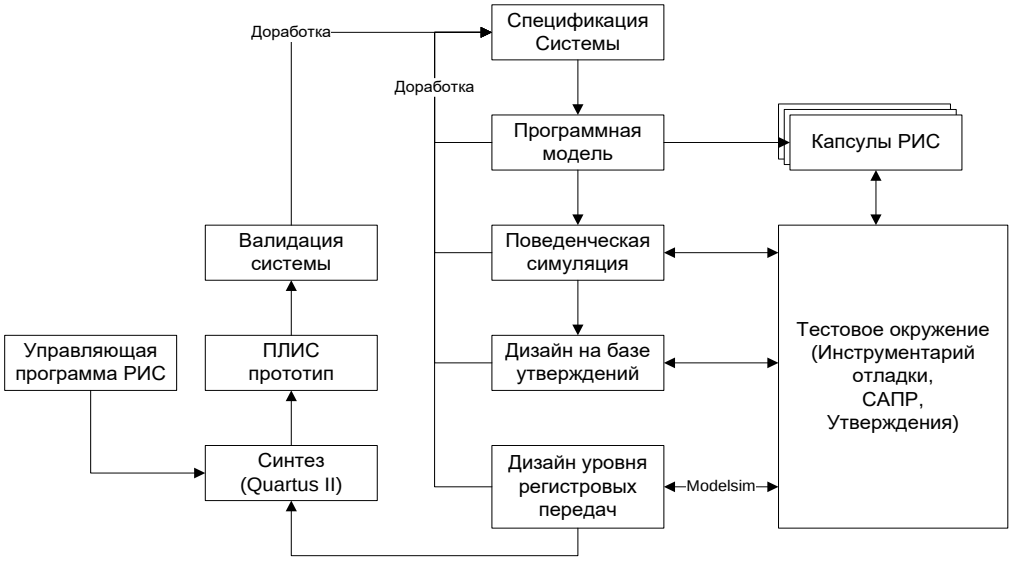

Рисунок 7 – Схема процесса разработки и тестирования ГАРОС

Используя методику капсульного программирования, каждая задача РОУ была реализована в виде капсулы на различных этапах разработки ГАРОС. Для всех капсул были осуществлены модельные испытания и произведены измерения производительности. Например, в результате реализации методов и алгоритмов, предложенных в главе 2, был получен более чем двукратный рост скорости вычисления капсулы алгоритма полосовой фильтрации. В таблице 2 приводятся результаты реализации капсул алгоритмов распознавания в сравнении с dsPIC30F реализацией. Для вычислений использовались следующие типы памяти констант (ПК): память констант секционная (ПК_С), память констант секционная регистровая (ПК_СР), память констант секционная подгружаемая (ПК_СП).

Таблица 2 – Результаты реализации алгоритмов распознавания

Название алгоритма	Объем капсулы (в операндах)	Использованная ПК	Число циклов dsPIC30F	Число циклов РОУ	Коэф. ускорения
БПФ2_256	155	256 констант	~19000	1040	~18,2
Баттеруорт(две секции)	89	ПК_СР	1360	437	3,11
Полосовой фильтр (две полосы)	93	ПК_СР	1428*2	437	6,5
Натуральный логарифм (NaturLog1_x4)	58	-	196	119	1,65
RASTA фильтр + Экспоненцирование	105	ПК_С (6 кон-т)	~1000	164	6,1 (для 5 процессоров)
Натуральный логарифм (NaturLog0_x1)	25	-	36	15	2,4
Косинусное ИДПФ	26	ПК_С (38 кон-т)	1100 (без) / 484 (с ПК)	51	21,6 (без)/9,5(с ПК)
Рекурсия Дурбина-Скурра	297	-	~800	124	~6,5
PLP параметры	63	-	144	30	4,8
Delta-расширение	48	-	91	27	3,4
Евклидово расстояние	89 (edac)	ПК_СП (16 кон-т)	5632	1285 (edac)	4,4
Витерби (для текущего N=61)	124	ПК_СП (11 кон-т)	5408*4	5648	3,83

Также в главе представлены результаты синтеза ПЛИС прототипа с использованием Cyclone V GT Development Kit, согласно которым проект занял 49% ALM-блоков, 92% RAM-блоков, 71% общего объема памяти в битах, 9% пинов и 2% DSP-блоков. Основная цель создания данной версии макетного образца заключалась в проверке целостности и универсальности архитектуры в области ЦОС, а также – в верификации поведенческой модели прототипа на реальной аппаратуре. Задачи оптимизации основных аппаратных блоков макетного образца были менее приоритетны. В качестве управляющего процессора был выбран фон-неймановский процессор общего назначения NIOSII, что позволило упростить как отладку макетного образца, так и реализацию демонстрационной задачи.

Ограничения текущей реализации макетного образца позволили достичь уровня тактовой частоты NIOSII в 100 МГц, а РОУ – 12,5 МГц. Таким образом, продолжительность 1 цикла работы ROU составляет 8 циклов работы NIOSII. Данное соотношение использовалось для синхронизации УУ и РОУ, а также при оценке уровня производительности макетного образца. Полученные в ходе разработки и испытаний капсулы были также использованы в ходе испытания ПЛИС прототипа на полном наборе распознавания 100 произнесений. Результаты последовательного распознавания 100 слов библиотеки представлены в таблице 3.

Таблица 3 – Результаты апробации ПЛИС прототипа

Название капсулы	ЦикловNIOSIIна обработку 100 слов			Коэффициент ускорения	
	“Чистый” NIOSII	*Синхронный* ГАРОС*	*Асинхронный* ГАРОС*	*Синхронный* ГАРОС*	*Асинхронный* ГАРОС*
Баттеруорт (две секции)	2378	2809	1457	0.85	1.63
Полосовой фильтр (две полосы)	42298	11520	11520	3.67	3.67
Натуральный логарифм(NaturLog1_x4)	1601	2303	2102	0.70	0.76
RASTA фильтр + Экспоненцирование	1743	3349	716	0.52	2.43
Косинусное ИДПФ	570	274	115	2.08	4.96
Натуральный логарифм (NaturLog0_x1)	47	161	58	0.29	0.81
PLP параметры	151	269	79	0.56	1.91
VectQuant_ED	20841	7018	3247	2.97	6.42
Delta-расширение	165	513	100	0.32	1.65
Евклидово расстояние	13335	10815	4576	1.23	2.91
Другое	1592	1670	-	0.95	-
Итого	85799	40701	23970	2.11	3.58

** Синхронный ГАРОС – реализует синхронную дисциплину взаимодействия управляющего и операционного уровней архитектуры. Асинхронный ГАРОС – реализуется асинхронную дисциплину взаимодействия, соответственно*

Таким образом, модельные испытания показали высокие показатели производительности усовершенствованной версии прототипа ГАРОС, продемонстрировав коэффициент ускорения ~x4.5 по сравнению с эталонной одноядерной реализацией.

Результаты аппаратных испытаний на уровне ПЛИС прототипа с использованием синхронной дисциплины взаимодействия оказались ниже ожидаемых. Поэтому была осуществлена реализация асинхронной дисциплины, для которой коэффициент ускорения составил ~x3.6, что оказалось достаточно близким к модельным результатам.

Чтобы показать потенциал ГАРОС в области ЦОС, недостаточно было решить только демонстрационную задачу. Необходимо было продемонстрировать результаты его апробации на представительной выборке алгоритмов ЦОС. В качестве такой выборки был рассмотрен комплект алгоритмических ядер BDTI Benchmark Kernels для оценки производительности ЦСП. Алгоритмическими ядрами называются функции, которые представляют собой основные строительные блоки большинства задач обработки сигналов. Эти ядра являются наиболее ресурсоемкими частями задач ЦОС. Подразумевается, что эти ядра высоко оптимизированы и разрабатываются вручную для каждого конкретного ЦСП. В качестве эталона для реализации и сравнения была использована публичная документация ЦСП семейства TMS320C55x компании Texas Instruments.

Результаты бенчмаркинга ГАРОС позволяют сделать вывод, что архитектура является жизнеспособной и обладает хорошим потенциалом производительности в классе задач ЦОС. Большинство алгоритмических ядер показали идентичный уровень производительности, как и продукт компании Texas Instruments, лидирующей на рынке ЦСП.

В таблице 4 приведены результаты предварительной оценки производительности ГАРОС на алгоритмических ядрах по сравнению с C55x.

Таблица 4 – Предварительные оценки производительности ГАРОС

Алгоритм	**Производительность в циклах**		
	C55x	***ГАРОС***	
Block FIR 1 MAC	C[a]: nx * (2 + nh) / O[b]: 25	C: nx * (1 + nh) / O: 12	
Block FIR 2 MAC	C: nx/2 * (6 + nh) / O: 25	C: nx/2 * (1 + nh) / O: 12	
Single Sample FIR 1 MAC	C: (1 + nh) / O: 44	C: (1 + nh) / O:15	
Single Sample FIR 2 MAC	C: (1 + nh/2) / O: 24	***Real-time***	***Quasi real-time***
		C: (3 + nh/2) / O: 15	C: (1 + nh/2) / O: 15
Complex Block FIR	C: nx * [8 + 2*(nh-2)] / O: 51	C: nx * (1 + 2*nh) / O: 14	
LMS Adaptive FIR	C: nx * (5 + 2*nh) / O: 26	C: nx * (4 + 3*nh) / O: 12	
IIR Double Precision Form II	C: nx * (7 + 31*nbiq[c]) / O: 77	C: nx * (4 + 21 * nbiq) / O: 12	
IIR 4 Coefficient Form II	C: nx * (2 + 3*nbiq) / O: 44	C: nx * (2 + 5*nbiq) / O: 12	
IIR 5 Coefficient Form II	C: nx * (5 + 5*nbiq) / O: 60	C: nx * (2 + 6*nbiq) / O: 12	
IIR 5 Coefficient Form I	C: nx * (5 + 8*nbiq) / O: 68	C: nx * (1 + 7*nbiq) / O: 12	
Vector Dot Product	C: nx + 1 / O: 44	C: nx + 1 / O: 15	
Vector Add	C: nx * 3 / O: 23	C: nx * 2 / O: 14	
Vector Maximum	C: nx * 3 / O: 8	C: nx * 5 / O: 12	
Viterbi Decoder[d]	C: Frame * 34 / O: 121	C: Frame * 109 / O: n/a	
256-Point In-Place FFT	C: 5 * N/2 * $\log_2$N	C: 4 * N/2 * $\log_2$N	

[a)]Core - количество циклов, необходимое чтобы выполнить алгоритмическое ядро;[b)]Overhead - накладные расходы, не зависящие от параметров алгоритма;[c)]Количество биквадратных уравнений; [d)]GSM Half Rate Convolutional Encoder used (R=1/2, K=5, Frame=189)

Заключение посвящено основным итоги выполненного исследования. Здесь также определены перспективные направления исследований для развития МПРА и ее прототипа ГАРОС:

– Разработка семейства специализированных устройств преобразования тегов, что в перспективе позволит удлинить цепочки рекуррентных преобразований и добиться большей степени сжатия тегированных данных.

– Разработка способов и механизмов преобразования тегов, которые позволят повысить степень наполнения конвейера ГАРОС инструкциями и его производительность на последовательных фрагментах алгоритма.

– Разработка механизма объединения капсул.

– Разработка дополнительных элементов методологии программирования и

отладки (грамматика капсульного языка программирования, грамматика языка описания капсул высокого уровня, методы трансляции языка высокого уровня в капсулы).

– Самосинхронное исполнение прототипа ГАРОС на аппаратном уровне.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Yu.I. Shikunov, D.V. Khilko, Yu.A. Stepchenkov. Hardware and Software Modelling and Testing of Non-Conventional Data-Flow Architecture // 2016 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) St. Petersburg, Russia, February 02-03, 2016. – IEEE, P. 360 – 364. DOI: 10.1109/EIConRusNW.2016.7448187.

2. Yu.A. Stepchenkov, D.V. Khilko, Yu.G. Diachenko, Yu.I. Shikunov and D.I. Shikunov. Testing of Software and hardware testing of dataflow recurrent digital signal processor // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2016), Yerevan, October, 14–17, 2016. P. 168 – 171. DOI: 10.1109/EWDTS.2016.7807672.

3. Yu.I. Shikunov, Yu.A. Stepchenkov, D.V. Khilko, D.I. Shikunov. Data redundancy problems in data-flow computing and solutions implemented on the recurrent architecture // 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) St. Petersburg, Russia, 1–3 Feb., 2017. – IEEE, P. 335 – 338.

4. Yu.I. Shikunov, Yu.A. Stepchenkov, D.V. Khilko. Recurrent mechanism developments in the data-flow computer architecture //2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) Moscow, Russia, 29 Jan.–1 Feb., 2018. – IEEE, P. 1413 – 1418. DOI: 10.1109/EIConRus.2018.8317362.

5. Yu.I. Shikunov, Yu.A. Stepchenkov, D.V. Khilko, G.A. Orlov. Graph-capsule construction toolset for data-flow computer architecture // 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) Moscow, Russia, 29 Jan. –1 Feb., 2018. – IEEE, P. 1419 – 1423. DOI: 10.1109/EIConRus.2018.8317363.

6. Yu.A. Stepchenkov, N.V. Morozov, D.V. Khilko, Yu.I. Shikunov, G.A. Orlov. Hybrid multi-core recurrent architecture approbation on FPGA // 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) Moscow, Russia, January 28–31, 2019. – IEEE, P. 1705 – 1708. DOI: 10.1109/EIConRus.2019.8657140.

7. D.V. Khilko, Yu.A. Stepchenkov, Yu.I. Shikunov and G.A. Orlov. Modeling and debugging tools development for recurrent architecture // 2019 IEEE EAST-WEST DESIGN & TEST SYMPOSIUM Batumi, Georgia, September 13 – 16, 2019. P. 1 – 5. DOI: 10.1109/EWDTS.2019.8884412.

8. Yu.A. Stepchenkov, D.V. Khilko, Yu.I. Shikunov, G.A. Orlov. Iterator component development for data redundancy solution in data-flow architecture // 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) Moscow, Russia, January 27–30, 2020. – IEEE, P. 1869 – 1872.

9. Yu.A. Stepchenkov, D.V. Khilko, Yu.I. Shikunov, G.A. Orlov. Testing and optimization of Recurrent Signal Processor // 2020 International Conference Engineering Technologies and Computer Science (EnT 2020), Moscow, Russia 24–27 June 2020. P. 54 – 57.

10. Yu.A. Stepchenkov, D.V. Khilko, Yu.I. Shikunov, G.A. Orlov. DSP Filter Kernels Preliminary Benchmarking for Recurrent Data-flow Architecture // 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, Moscow, Russia, January 26-29, 2021. – IEEE, P. 2040 – 2044.

11. Yu.A. Stepchenkov, D.V. Khilko, Yu.I. Shikunov and G.A. Orlov. Design validation of recurrent signal processor FPGA prototype // Proceedings of IEEE East-West Design & Test Symposium (EWDTS 2021), Batumi, Georgia, September, 10–13, 2021, P. 157 – 161.

12. V.N. Zakharov, Yu.A. Stepchenkov, Yu.G. Diachenko and D.V. Khilko. Computing Dataflow Architectures: History and Implementation Perspectives // 2021 International Conference Engineering Technologies and Computer Science (EnT 2021), Moscow, Russia, 18–19 August 2021. P. 98 – 102. DOI: 10.1109/EnT52731.2021.00024

13. Yu.A. Stepchenkov, D.V. Khilko, Yu.I. Shikunov, G.A. Orlov. Optimizing Data-flow

Processor Architecture for Efficient Implementation of DSP Algorithms // 2022 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus) St. Petersburg, Moscow, Russia, January 25–28, 2022. – IEEE, P. 464 – 468.

14. Yu.A. Stepchenkov, D.V. Khilko, Yu.I. Shikunov, G.A. Orlov. Self-assembly phenomenon implementation problems in recurrent-dynamic architecture // 2023 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus) St. Petersburg, Moscow, Russia, January 24–27, 2023. – IEEE, P. 282 – 286.

15. Д.В. Хилько, Ю.А. Степченков. Теоретические аспекты разработки методологии программирования рекуррентной архитектуры // Системы и средства информатики, – М.: ТОРУС ПРЕСС, Т. 23, № 2, 2013 – С. 133 – 153.

16. Д.В. Хилько, Ю.А. Степченков, Ю.Г. Дьяченко, Ю.И. Шикунов, Н.В. Морозов. Аппаратно-программное моделирование и тестирование рекуррентного операционного устройства // Системы и средства информатики, – М.: ТОРУС ПРЕСС, Т. 25, № 4, 2015 – С. 78 – 90.

17. Ю.А. Степченков, Ю.Г. Дьяченко, Д.В. Хилько, В.С. Петрухин. Рекуррентная потоковая архитектура: особенности и проблемы реализации // Проблемы разработки перспективных микро- и наноэлектронных систем – 2016. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. М.: ИППМ РАН, 2016. Часть 2. С. 120 – 127.

18. Д.В. Хилько, Ю.А. Степченков, Д.И. Шикунов, Ю.И. Шикунов. Рекуррентная потоковая архитектура: технические аспекты реализации и результаты моделирования // Проблемы разработки перспективных микро- и наноэлектронных систем – 2016. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. М.: ИППМ РАН, 2016. Часть II. С. 128 – 135.

19. Д.В. Хилько, Ю.А. Степченков, Ю.И. Шикунов, Г.А. Орлов. Развитие средств капсульного программирования потоковой рекуррентной архитектуры // Проблемы разработки перспективных микро- и наноэлектронных систем – 2018. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. М.: ИППМ РАН, 2018. Часть III. С. 2 – 9.

20. Ю.А. Степченков, Н.В. Морозов, Ю.Г. Дьяченко, Д.В. Хилько, Д.Ю. Степченков. Развитие гибридной многоядерной рекуррентной архитектуры на ПЛИС// Системы и средства информатики, 2020. Т. 30. № 4. С. 95 – 101. DOI: 10.14357/08696527200409.

21. Ю.А. Степченков, Д.В. Хилько, Ю.И. Шикунов, Г.А. Орлов Специализированные преобразователи тегов для рекуррентного обработчика сигналов // Проблемы разработки перспективных микро- и наноэлектронных систем – 2020. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. М.: ИППМ РАН, 2020. Выпуск 2. С. 73 – 80.

22. Ю.А. Степченков, Н.В. Морозов, Ю.Г. Дьяченко, Д.В. Хилько. Аппаратная реализация рекуррентного обработчика сигналов // Системы и средства информатики, 2021. Т. 31. № 3. С. 113 – 122.DOI: 10.14357/08696527210310.

23. Д.В. Хилько, Ю.А. Степченков, Ю.И. Шикунов, Ю.Г. Дьяченко, Г.А. Орлов. Оптимизация аппаратной поддержки быстрого преобразования Фурье в рекуррентном сигнальном процессоре // Системы и средства информатики, 2021. Т. 31. № 4. С. 71 – 83.

24. Ю.Г. Дьяченко, Ю.А. Степченков, Н.В. Морозов, Д.В. Хилько, Д.Ю. Степченков, Ю.И. Шикунов Аппаратная верификация рекуррентного обработчика сигналов на ПЛИС // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2021. Выпуск 2. С. 77 – 82. DOI: 10.31114/2078-7707-2021-2-77-82.

25. Ю.А. Степченков, Н.В. Морозов, Ю.Г. Дьяченко, Д.В. Хилько, Д.Ю. Степченков, Ю. И. Шикунов. Аппаратная реализация алгоритмов цифровой обработки сигналов в рекуррентном потоковом процессоре на ПЛИС // М.: Известия вузов. Электроника. 2022. Том 27. № 3. С. 356-366. DOI: 10.24151/1561-5405-2022-27-3-356-366.

26. Д.В. Хилько. Реализация блочного КИХ-фильтра в потоковом рекуррентном сигнальном процессоре // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2022. Выпуск 4. С. 163 – 170. DOI: 10.31114/2078-7707-2022-4-163-170.

Патенты РФ на изобретение

1. Пат. 2 469 470 Российская Федерация, МПК H03K 3/00. Формирователь парафазного сигнала с высоким активным уровнем входа управления / Степченков Ю.А., Дьяченко Ю.Г., Шнейдер А.Ю., Прокофьев А.А., Хилько Д.В.; заявитель и патентообладатель ИПИ РАН. № 2011129014/08; заявл. 13.07.11; опубл. 10.12.2012, Бюл. № 34.

2. Пат. 2 693 319 Российская Федерация, МПК H03K 3/00. Самосинхронный динамический двухтактный D-триггер с единичным спейсером. Степченков Ю. А., Дьяченко Ю.Г., Хилько Д.В. Дьяченко Д.Ю., Степченков Д.Ю.; заявитель и патентообладатель ФИЦ ИУ РАН. № 2018141051; опубл. 02.07.2019, Бюл. № 19.

3. Пат. 2 725 780 Российская Федерация, МПК H03K 3/00. Сбоеустойчивый самосинхронный однотактный RS-триггер с нулевым спейсером. Степченков Ю.А., Дьяченко Ю.Г., Морозов Н.В., Орлов Г.А., Хилько Д.В.: заявитель и патентообладатель ФИЦ ИУ РАН. № 2019142820; опубл. 06.07.2020, Бюл. № 19.