На правах рукописи

Салищев Сергей Игоревич

СИНТЕЗ АЛГОРИТМОВ ОБРАБОТКИ СИГНАЛОВ С ОГРАНИЧЕНИЯМИ НА МИНИМАЛЬНЫЙ ПАРАЛЛЕЛИЗМ И ОБЪЁМ ПАМЯТИ

01.01.09 — дискретная математика
и математическая кибернетика

АВТОРЕФЕРАТ
диссертации на соискание учёной степени
кандидата физико-математических наук

Санкт-Петербург
2016

Работа выполнена в Санкт-Петербургском государственном университете

Научный руководитель:	доктор физико-математических наук, БАРАБАНОВ Андрей Евгеньевия
Официальные оппоненты:	КОЛЕСОВ Николай Викторович доктор технических наук, профессор, ГНЦ РФ АО «Концерн «ЦНИИ «Электроприбор», главный научный сотрудник
	ЛОБАНОВ Игорь Сергеевич кандидат физико-математических наук, Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, доцент
Ведущая организация:	Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»

Защита состоится 01 марта 2017 г. в 16 часов на заседании диссертационного совета Д 212.232.29 на базе Санкт-Петербургского государственного университета по адресу: 199178, Санкт-Петербург, 10 линия В.О., д. 33/35, ауд. 74.

С диссертацией можно ознакомиться в Научной библиотеке им. М. Горького Санкт-Петербургского государственного университета по адресу: 199034, Санкт-Петербург, Университетская наб., 7/9 и на сайте https://disser.spbu.ru/files/disser2/disser/4Ph9sRhxtR.pdf

Автореферат разослан "____"_____2016 года.

Ученый секретарь
диссертационного совета Д 212.232.29
доктор физико-математических наук,
профессор

Нежинский В.М.

Общая характеристика работы

Актуальность темы. Энергоэффективность - одна из основных характеристик полупроводниковых беспроводных устройств, определяющая время работы устройства от батареи и его тепловой режим. Стандартные модели энергопотребления учитывают только активную мощность, в которой оценка затрат энергии пропорциональна сложности в элементарных операциях. Для схем с литографическими нормами 45 нм и менее на первый план выходят потери энергии в результате токов утечки, которые не зависят от вычислительной сложности. Они складываются из размера памяти и количества параллельных вычислительных элементов. Оптимальный параллелизм для минимизации энергопотребления рассматривался Ву и Ли [Woo, 2008] для многоядерных суперскалярных процессоров. В отличие от рассматриваемой нами задачи, процессоры работают на высокой частоте и при высоком напряжении питания, что позволяет не учитывать энергопотребление памяти. Одним из базовых блоков для алгоритмов цифровой обработки сигналов является вычисление элементарных функций $\ln, \exp, \sin, \cos, \sqrt{x}, 1/x$ и т.п. Для аппаратных блоков вычислений с точностью 22 бита и выше обычно используется кусочно-полиномиальная аппроксимация. Проблемой энергоэффективной аппаратной реализации метода является оптимальный баланс между размером таблиц и степенью многочлена. Для элементарных функций таблицы являются избыточными и могут быть сокращены, используя гладкость функций. В работе [Strollo, 2011] предложен метод сокращения таблиц на 40% без существенного увеличения вычислений за счет использования двузвенного гладкого сплайна, однако вопрос точности решается эмпирически, при помощи тестирования на всех допустимых данных, что делает метод в описанном виде неприменимым для высоких точностей. Из более сложных алгоритмов наиболее часто используется Быстрое Преобразование Фурье. Существующие алгоритмы БПФ на специализированной аппаратуре можно разделить по асимптотике времени вычислений $\mathcal{O}(1)$, $\mathcal{O}(\ln n)$, $\mathcal{O}(n)$, $\mathcal{O}(n \ln n)$. При требовании переиспользования ресурсов и программно управляемой длины преобразования часто наиболее энергоэффективными оказываются аппаратные блоки БПФ с временем работы $\mathcal{O}(n \ln n)$ и памятью с произвольным доступом. К таким алгоритмам относится поточное БПФ [Johnson, 1992], которое может быть реализовано на реконфигурируемом вычислительном устройстве антимашине Хартенштейна [Hartenstein, 1991] Разработка поточных алгоритмов БПФ с ограничением на доступ к памяти может быть сведена к задаче составления расписания для синхронного графа потока данных [Parhi, Messerschmitt, 1991]. Алгоритм Джонсона применим только для

чистых оснований и для смешанного основания без параллелизма бабочек. Модификация алгоритма Джонсона, предложенная [Jo, Sunwoo, 2005], специализирована только для оснований 2/4. Во многих задачах более эффективными являются самосортирующие алгоритмы БПФ, такие, как алгоритм Джонсона-Буруса и Темплтона [Hegland, 1994], которые сформулированы только для однобанковой памяти. Также требуется модификация под наиболее эффективную архитектуру памяти, предоставляемую библиотекой компонентов, например, однопортовую память. Такие адаптации для алгоритмов БПФ без копирования пока не рассматривались в литературе. Другим часто используемым классом алгоритмов является LU факторизация тёплицевых и обратных к тёплицевым матриц. Задачи этого типа и большой размерности возникают в акустических задачах эхо- и шумоподавления и разделения источников сигнала. Для решения задачи факторизации обычно используются алгоритмы Левинсона и Шура, имеющие асимптотику сложности $\mathcal{O}(n^2)$, где n- длина вектора автокорреляций. Для задач большой размерности может использоваться быстрый алгоритм Шура, предложенный [Ammar, Gragg, 1987; Воеводин, Тыртышников, 1987] со сложностью $\mathcal{O}(n \ln^2 n)$, который рассматривается как пример энергоэффективной реализации сложных алгоритмов с БПФ.

Целью данной работы является разработка алгоритмов для минимизации сложности расчетов, характерных для статистической обработки сигналов. Сложность измеряется энергоэффективностью. Она зависит, в частности, от длины таблиц при расчете стандартных элементарных функций, от доступа к памяти в реализации БПФ, от самосортировки БПФ, от параллелизма в быстром алгоритме обращения тёплицевых матриц.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Сформулировать функционал сложности и другие требования, предъявляемые к алгоритмам для их эффективной реализации в аппаратуре;
2. Разработать эффективные целочисленные алгоритмы вычисления элементарных функций с заданной точностью и минимальной длиной таблицы;
3. Разработать расписание для реализации графа потока данных для БПФ на многобанковой памяти;
4. Исследовать оптимальный параллелелизм и размер памяти быстрого алгоритма Шура.

Основные положения, выносимые на защиту:

1. Метод качественной оценки мощности и выбора оптимального параллелизма для энергоэффективных специализированных КМОП вычислительных блоков (Лемма 1, Лемма 2).

2. Метод вычисления элементарных функций при помощи почти гладкого четырехзвенного квазисплайна и оценка точности полиномиальной аппроксимации с коэффициентами с фиксированной точкой, ограниченной на равномерной сетке (Теорема 1).

3. Теорема о размещении данных БПФ в многобанковой памяти при вычислении по произвольным смешанным основаниям (Теорема 2).

4. Теорема о размещении данных и порядке вычисления самосортирующегося БПФ (Теорема 4).

5. Теорема о размещении данных и порядке вычисления БПФ для однопортовой памяти (Теорема 3).

6. Анализ энергоэффективности алгоритма LU факторизации вещественных тёплицевых матриц на сверточном акселераторе для задачи эхокомпенсации при помощи быстрого алгоритма Шура (Лемма 3, Лемма 4, Лемма 5, Лемма 6).

Научная новизна: Все представленные результаты являются новыми.

1. Разработана модель энергопотребления для малопотребляющей цифровой схемы, выполняющей известный вычислительный алгоритм. Решена задача выбора оптимального параллелизма в данной модели.

2. Задача минимизации энергопотребления при расчёте значений стандартных функций сведена к минимизации длины таблиц. Разработаны новые методы аппроксимации квазисплайнами с неравномерным табулированием, удобные для аппаратной реализации.

3. Доказана теорема о размещении данных БПФ в многобанковой памяти при вычислении по произвольным смешанным основаниям, что обеспечивающая максимальную скорость вычислений при заданном параллелизме. Получены явные формулы БПФ в виде произведений Кронекера по стадиям произвольных порядков.

4. Доказана теорема о самосортирующейся модификации БПФ в многобанковой памяти по смешанным основаниям, а также аналогичная теорема для вычислительного устройства с однопортовой памятью.

5. Для быстрого алгоритма Шура найден минимальный объём памяти, вычислена длина критического пути и проведена оценка оптимального параллелизма.

Практическая значимость диссертационной работы состоит в сокращении площади и энергопотребления рассмотренных элементов и повышении их универсальности, что вносит существенный вклад в улучшение энергоэффективности автономных беспроводных устройств, реализуемых на базе специализированных полупроводниковых логических схем.

Достоверность изложенных в работе результатов обеспечивается практической реализацией предложенных схем и алгоритмов в виде полупроводниковых схем. Практическая реализация включала разработку моделей на языке SystemC, автоматическую верификацию с помощью системы "Aegis for SystemC", логический синтез полупроводниковых схем уровня вентилей для акселераторов из моделей SystemC, логическое моделирование работы схем и синтез виртуальной топологии для малопотребляющего процесса производства полупроводниковых кристаллов с геометрическими нормами 22 нм.

Апробация работы. Основные результаты работы докладывались на: Международной конференции Общества Инженеров Акустиков (AES) (Россия, Санкт-Петербург, 2003), международной конференции по компьютерному анализу и моделированию (CDAM) (Беларусь, Минск, 2004), конференции молодых ученых "Гироскопия и Навигация" (Россия, Санкт-Петербург, 2004), семинаре кафедры теоретической кибернетики СПбГУ (Россия, Санкт-Петербург, 2015, 2016), семинарах лаборатории Intel Labs (2013 - 2015).

Личный вклад. Результаты, выносимые на защиту, получены автором самостоятельно.

Публикации. Основные результаты по теме диссертации изложены в 12 печатных публикациях [1–12], в том числе 4 [1–4] — в журналах, рекомендованных ВАК, 5 [5,6,9–11] — в тезисах докладов на международных конференциях на английском языке, из них 3 [9–11] индексируются Scopus, [12] заявка на патент США.

Работы [4, 5, 7–11] написаны в соавторстве. В работе [4] автору принадлежит постановка задачи, формулировка всех теорем и их доказательство, кроме доказательства теоремы 4. В работе [8, 9] автору принадлежит раздел, посвященный практическому опыту реализации. В [7] автору принадлежит постановка задачи, анализ существующих систем для выделения общих требований, раздел по использованию Java в системном программировании. В [10, 11] автору принадлежит постановка задачи и разработка алгоритма обнаружения ошибок синхронизации с помощью анализа достижимости. В работе [5] автор выполнял математическое моделирование.

Объем и структура работы. Полный объем диссертации — 209 страниц. Основной текст диссертации — 167 страниц. Диссертация состоит из введения, четырех глав и заключения с 9 рисунками, 14 таблицами и 5 приложениями. Список литературы содержит 85 наименований.

Содержание работы

Во **введении** обосновывается актуальность исследований в области разработки алгоритмов энергоэффективных вычислений.

Первая глава посвящена анализу различных факторов, влияющих на энергоэффективность и удобство применения вычислительных блоков при построении малопотребляющих специализированных полупроводниковых схем, а также анализу методов оптимизации энергопотребления. Также в главе рассматривается влияние на энергопотребление других факторов, таких, как архитектура памяти, представление числовых данных, архитектура процессора и операционной системы, использование языков управляемого выполнения (Java, C# и т.п.), моделей параллельных вычислений и средств автоматической верификации. Мощность вычислительного блока складывается из статической мощности неотключаемых от питания компонент и мощности компонент с управляемым питанием. К неотключаемым компонентам относятся память кода и память состояния, остальные компоненты можно отключать на время простоя. К отключаемым компонентам относятся вычислительные устройства, память промежуточных данных и постоянное запоминающее устройство. Пусть алгоритм выполняется на вычислительном блоке с p процессоров. Минимизируем потребляемую мощность по p.

Алгоритм характеризуется постоянными параметрами: B - вычислительная сложность алгоритма, в количестве элементарных операций в секунду; K - нераспараллеливаемая доля алгоритма.

Вычислительный блок характеризуется постоянными параметрами: N_0 - количество вентилей в неотключаемой части; N_1 - количество вентилей в отключаемой части без вычислительных компонент; $\hat{N}_2$ - количество вентилей в последовательной реализации вычислительных компонент блока; f - тактовая частота.

Количество вентилей в вычислительных компонентах блока определяется как $N_2 = p\hat{N}_2$. Тогда мощность вычислительного блока на участке линейного роста от частоты можно оценить как

$$P_{leak} = c_0(N_0 + S(N_1 + N_2))$$

$$P_d = c_1 f S(N_1 + N_2)$$

$$P = P_{leak} + P_d = c_0(N_0 + S(N_1 + N_2)) + c_1 f S(N_1 + N_2)$$

где P_{leak} - потери мощности от токов утечки, P_d - динамическая мощность, c_0, c_1 - постоянные коэффициенты, а $S = S(p) \leq 1$ - скважность или отношение времени вычислений к общему времени, включающему простой.

Функцию $S(p)$ заменяют на функцию ускорения от параллелизма $l(p)$, которую можно описать с помощью закона Амдала [Amdahl, 1967].

$$S(p) = \frac{B}{l(p)f}, \qquad l(p) = \frac{p}{1 + K(p-1)}.$$

$$P = c_0 N_0 + B(c_0/f + c_1)\frac{p\hat{N}_2 + N_1}{l(p)}, \qquad \arg\min_{f, f \leq f_0} P = f_0.$$

Лемма 1 *Мощность достигает минимума по p при значении*

$$p_0 = \arg\min_p P = \max\left(1, \sqrt{\frac{N_1(1-K)}{\hat{N}_2 K}}\right).$$

При этом минимальная мощность имеет значение

$$P_{\min} = c_0 N_0 + B(c_0/f_0 + c_1)\left((1-K)\hat{N}_2 + KN_1 + 2\sqrt{KN_1\hat{N}_2(1-K)}\right).$$

Функция мощности зависит от параллелизма p и размера задачи n:

$$P(p, n) = c_0 N_0(n) + B(n)(c_0/f + c_1)\frac{p\hat{N}_2 + N_1(n)}{l(p, n)},$$

где величины $N_0(n)$, $N_1(n)$ зависят от размера памяти, который обычно растет с ростом размера задачи. Величина $\hat{N}_2$ не зависит от n, поскольку алгоритм вычислений не меняется.

Используем закон Густафсона-Барсиса [Gustafson, 1988]:

$$l(p, n) = (1 - \alpha)p + \alpha.$$

Лемма 2 *Пусть размер памяти $N_0(n)$, $N_1(n)$ и сложность алгоритма $B(n)$ растут линейно по размеру задачи. Тогда $P(1, n) = O(n^2)$, а минимальная мощность $P_{\min}(n) = O(n)$ при $n \to \infty$.*

Во **второй главе** изучаются способы расчёта элементарных функций при ограничениях на общий объём памяти вычислительного блока. Задача сводится к целочисленному программированию и оптимальной равномерной аппроксимации многозвенными квазисплайнами.

Функция f на отрезке Δ разбивается на сегменты, и внутри каждого сегмента аппроксимируется полиномом. В памяти хранятся параметры, для расчёта значений полиномов по всем сегментам, сведённые в таблицу. Требуется минимизировать длину таблицы в битах при равномерном ограничении на точность расчёта значений функции.

Пусть длины всех сегментов полиномиальной интерполяции равны 1. Тогда границы сегментов полиномиальной интерполяции есть целые числа, $x_i = i$, $0 \le i < 2^{k_0}$, где 2^{k_0} — количество звеньев почти гладкого квазисплайна. Квазисплайн будем обозначать $\{p_i(x)\}$, где $p_i(x)$ - полином i-го звена. Его коэффициенты $p_{ij} \in Q_{l_m}$, где Q_{l_m} — множество двоичных дробей с дробной частью из l_m цифр.

В пределах каждого звена выберем сетку отсчётов $\bar{x}_j = 2^{-k_2} j$ при $0 \le j < 2^{-k_2}$ с шагом $\delta = 2^{-k_2}$.

На промежутке $I_0 = [-1, 1]$ выберем N интерполяционных узлов и построим по ним фундаментальные полиномы $l_k(x) = \omega(x)/((x - x_k)\omega'(x_k))$. Числом Лебега назовём

$$\lambda_{N,\nu} = \sup_{x \in [-1,1]} \sum_{k=1}^{N} |l_k^{(\nu)}(x)|.$$

Теорема 1 *Пусть $N > 2$, $|f^{(N)}(x)| \le M$ на $[0, 2^{k_0}]$ и $\varepsilon_m > 0$ — заданная точность аппроксимации. Определим*

$$\bar{\varepsilon} = \left(\varepsilon_m - \frac{\delta^2 M}{8(N-2)!}\right)\left(1 + \frac{\delta^2 \lambda_{N,2}(I_0)}{2}\right)^{-1}.$$

Пусть существует решение следующей смешанной целочисленной задачи линейного программирования относительно коэффициентов почти гладкого квазисплайна $\{p_i(x)\}$ на отрезке $[0, 2^{k_0}]$ при ограничении на представление $p_{ij} \in Q_{l_m}$:

$$\begin{cases} \varepsilon = \sum_{\nu=0}^{N} \sum_{i=1}^{2^{k_0}-1} \alpha_{i,\nu} |d_{i,\nu}| \\ \varepsilon_i \le \bar{\varepsilon} \\ \varepsilon^* = \min_{p_i, i \in [0..2^{k_0}-1]} \varepsilon \end{cases}$$

где

$$\varepsilon_i = \sup_{0 \le j < 2^{k_2}} |f_i(x_i + \bar{x}_j) - p_i(x_i + \bar{x}_j)|, \qquad 0 \le i < 2^{k_0},$$

$$d_{i,\nu} = p_i^{(\nu)}(x_i) - p_{i-1}^{(\nu)}(x_i), \qquad 0 < i < 2^{k_0}.$$

Тогда квазисплайн $\{p_i(x)\}$ обеспечивает заданную равномерную точность аппроксимации функции f:

$$\sup_{x \in [0, 2^{k_0}]} |f(x) - p(x)| \le \varepsilon_m,$$

а количество ненулевых бит для коэффициентов в промежуточных узлах квазисплайна L не превосходит

$$L \leq \sum_{i=0}^{2^{k_0}-1} \sum_{\nu=0}^{N-1} \max(0, 2 + \lceil l_m + \log_2 |d_{i,\nu}| \rceil),$$

где l_m - минимальная допустимая длина дробной части коэффициента полиномиальной аппроксимации в одном сегменте.

Были построены таблицы для квадратичной аппроксимации функций $\sin, \ln, 1/x, \sqrt{x}$ в интервале точностей 24-32 бит. Уменьшение размера таблиц для двузвенного квазисплайна составляет более 40%, что соответствует результатам Стролло. Для четырехзвенного квазисплайна уменьшение размера таблиц — более 60%, а уменьшение энергопотребления — до 2 раз, что превосходит известные результаты.

В **третьей главе** изучаются комбинаторные задачи размещения данных в многобанковой памяти при реализации быстрого преобразования Фурье (БПФ). Они формулируются в терминах антимашины в классификации Хартенштейна [Hartenstein, 1991] как задачи составления расписаний для гомогенного синхронного графа потока данных.

Предположим, что алгоритм БПФ последовательно выполняет бабочки порядков $n_0, n_1, \ldots, n_K$. Длина входного массива есть $N = \prod_{i=0}^{K} n_i$. Память разделена на R банков памяти, и число R есть наименьшее общее кратное оснований $(n_k)_{k=0}^{K}$. Предполагается, что на каждом такте процессор может вычислить любое количество бабочек с записью результата в тех же ячейках памяти, из которых были считаны исходные данные для выполнения бабочек. Однако на каждом такте разрешено лишь одно обращение к каждому банку памяти. Конфликтом доступа к памяти называется ситуация одновременного обращения к двум или более адресам одного банка памяти.

Требуется найти распределение входных данных по банкам $m(x)$, при котором на каждом такте отсутствует конфликт доступа к памяти. Требуется указать также для каждой стадии k БПФ все наборы одновременно выполняемых бабочек.

Наибольший общий делитель произвольного набора натуральных чисел $(m_j)_{j=0}^{J}$ обозначим $\mu((m_j)_{j=0}^{J})$, а наименьшее кратное набора — $\nu((m_j)_{j=0}^{J})$.

Теорема 2 *Пусть в алгоритме БПФ последовательно выполняются стадии по основаниям $n_0, n_1, \ldots, n_K$. Если алгоритм БПФ начинается с младших разрядов, то определим $\alpha = (n_K, \ldots, n_0)$ и цифровое представление номера n компоненты входного массива $p = n_\alpha = (p_K, \ldots, p_0)$.*

Если алгоритм БПФ начинается со старших разрядов, то определим $\alpha = (n_0, \ldots, n_K)$ *и цифровое представление номера* n *компоненты входного массива* $p = n_\alpha = (p_0, \ldots, p_K)$.

Входной массив размерности $N = \prod_{k=0}^{K} n_k$ *записывается в* R *банков данных, где* $R = \nu(n_0, \ldots, n_K)$ *— наименьшее общее кратное оснований бабочек.*

Для каждого $k = 0, 1, \ldots, K$ *введём обозначения*

$$M_k = \nu((n_i)_{i=0}^{k}), \quad d_k = \frac{M_k}{M_{k-1}}, \qquad s_{i,k} = \frac{M_i}{\mu(M_i, n_k)}, \quad v_{i,k} = \frac{s_{i,k}}{s_{i-1,k}}, \quad 0 \le i < k,$$

с доопределением $M_{-1} = s_{-1,k} = 1$.

Определим номер банка $m(n)$*, в который помещается элемент входного массива с номером* n *при* $n = 0, 1, \ldots, N-1$*, по формуле*

$$m(n) = \sum_{k=0}^{K} p_k q_k \mod R,$$

где $p = n_\alpha$ *и* $q_k = R/n_k$*. На стадии* k *при* $0 \le k \le K$ *одновременно выполняются все бабочки с номерами* ℓ*, в цифровом представлении которых совпадают целые части* $\lfloor p_j / v_{j,k} \rfloor$ *при* $0 \le j < k$ *и целые части* $\lfloor p_j / d_j \rfloor$ *при* $k < j \le K$*. Тогда при выполнении алгоритма БПФ не возникает конфликтов памяти, и на каждом такте задействованы все* R *банков памяти.*

Модифицируем архитектуру акселератора вычисления БПФ для использования 1rw памяти, в которой чтение и запись в один банк памяти не могут производиться за один такт. Для этого задействуем $2R$ банков памяти и потребуем, чтобы запись и чтение осуществлялись в непересекающиеся множества банков.

Теорема 3 *Пусть длина конвейера нечетна,* $p \mod 2 = 1$*. Рассмотрим следующий порядок обхода и функцию распределения индексов по банкам:*

$$\begin{aligned}
T_i^2(k) &= \begin{cases} t(k), & i = 0, \\ k, & i > 0, \end{cases} \\
t(k^0) &= \left[\ldots, \left(\sum_{i=2}^{n-1} k_i + \bar{k}_1 + \bar{r}\bar{\bar{k}}_1 \right) \mod R \right], \\
m_2(x) &= \left(2 \left(\sum_{i=1}^{n-1} x_i + q x_0 \right) - (x_0 \mod 2) \right) \mod 2R.
\end{aligned}$$

Такой выбор порядка обхода и функции распределения по банкам обеспечивает отсутствие конфликтов для архитектуры потокового БПФ акселератора с $2R$ банками 1rw памяти при выполнении одной бабочки за такт.

Инверсия индексов S_α требует дополнительного прохода по памяти. Цель самосортирующегося алгоритма состоит в замене начальной перестановки на постепенные частные перестановки на каждой стадии, осуществляемые при минимальной длине конвейера. Введём оператор перестановки начальной и конечной цифр в индексе произвольного вектора. Пусть вектор имеет длину N и число N делится на k^2, где $k \geq 1$. Тогда матрица перестановки Y_k^N определяется уравнениями

$$Y_k^N(e_{i,k} \otimes e_{\ell,N/k^2} \otimes e_{j,k}) = e_{j,k} \otimes e_{\ell,N/k^2} \otimes e_{i,k}, \qquad 0 \leq i,j < k, \quad 0 \leq \ell < N/k^2.$$

Алгоритм БПФ в частотной области описывается формулой $\mathcal{F}_N = S_\alpha E_{n-1,\alpha} E_{n-2,\alpha} \cdots E_{0,\alpha}$, где $E_{k,\alpha}$ — преобразование k-й стадии, S_α — перестановка по инверсии мультииндекса α. Пусть $N = rR^{n-1}$, где $R = rq$. Определим матрицы перестановок

$$L_k^n(e_{i,k} \otimes e_{j,m}) = e_{j,m} \otimes e_{i,k}, \qquad 0 \leq i < k, \quad 0 \leq j < m$$

$$Q_k = \begin{cases} I, & k = 0, \\ I_{rR^{k-1}} \otimes L_r^R \otimes I_{R^{n-k-1}}, & 1 \leq k \leq \lfloor \frac{n-1}{2} \rfloor, \\ I_{R^{n/2-1}} \otimes L_r^{rR} \otimes I_{R^{n/2-1}}, & k = \frac{n}{2}, \\ I_{R^{n-k-1}} \otimes [(I_{rR^{2k-n}} \otimes L_r^R) Y_R^{rR^{2k+1-n}}] \otimes I_{R^{n-k-1}}, & \lceil \frac{n+1}{2} \rceil \leq k \leq n-1. \end{cases}$$

$$\mathcal{F}_N = G_{n-1} \cdot \ldots \cdot G_0, \quad G_k = Q_k E_{k,\alpha}.$$

Последовательность выполнения бабочек на каждой стадии определяет необходимый объём внутренней памяти и длину конвейера. Последовательность выполнения бабочек в самосортирующемся алгоритме должна быть изменена для избежания конфликтов при обращении к памяти.

Пусть $\alpha = (\alpha_{n-1}, \ldots, \alpha_1, \alpha_0) = (R, \ldots, R, r)$ — мультииндекс, указывающий на последовательность выполнения стадий БПФ. Цифровая форма номера k есть $s = (k_0, k_1, \ldots, k_{n-1})$, где

$$k = s^{\alpha^\star} = k_{n-1} + R(k_{n-2} + \ldots + R(k_1 + Rk_0)\ldots),$$

и $0 \leq k_0 < r$ и $0 \leq k_i < R$ при $1 \leq i \leq n-1$. Разобьём каждую компоненту: $k_i = \overline{k}_i q + \underline{k}_i$, где $0 \leq \overline{k}_i < r$, $0 \leq \underline{k}_i < q = R/r$.

Пусть $i \geq (n+1)/2$ — номер стадии БПФ. При цифровом представлении номеров компонент массива

$$p = (\overline{k}_0, \underline{k}_1, \overline{k}_1, \ldots, \underline{k}_{n-1}, \overline{k}_{n-1})$$

в системе счисления, порождённой мультииндексом $\gamma = (r, q, r, \ldots, q, r)$, на вход каждой бабочки на стадии i подаётся набор отсчётов, у номеров которых все компоненты $\underline{k}_j$ и $\overline{k}_j$ одинаковые, кроме компонент $(\underline{k}_i, \overline{k}_i)$, которые пробегают все возможные значения. Поэтому естественным номером бабочки является $k = (p^i)^{\gamma_0}$, где $\gamma_0 = (r, q, r, \ldots, q, r)$ короче вектора γ на две компоненты и

$$p^i = (\overline{k}_0, \underline{k}_1, \overline{k}_1, \ldots, \underline{k}_{i-1}, \overline{k}_{i-1}, \underline{k}_{i+1}, \overline{k}_{i+1}, \ldots, \underline{k}_{n-1}, \overline{k}_{n-1}).$$

Определим операцию перестановки этой пары в конец мультииндекса p^i номера бабочки $k = (p^i)^{\gamma_0}$:

$$U_i(p^i) = (\overline{k}_0, \underline{k}_1, \ldots, \underline{k}_{n-1-i}, \overline{k}_{n-i}, \ldots, \underline{k}_{i-1}, \overline{k}_{i-1}, \underline{k}_{i+1}, \overline{k}_{i+1}, \ldots, \underline{k}_{n-1}, \overline{k}_{n-1}, \underline{k}_{n-i}, \overline{k}_{n-1-i})$$

при $i \geq (n+1)/2$.

Теорема 4 *Пусть $n > 2$ и функция распределения входного вектора по банкам данных $m(n)$ определяется теоремой 2. Порядок обхода бабочек на i-й стадии БПФ зададим номером такта $T_i(k)$ реализации бабочки с номером $k = (p^i)^{\gamma_0}$:*

$$T_i(k) = \begin{cases} \left\lfloor \frac{k}{q} \right\rfloor, & i = 0, \\ k, & 1 \leq i \leq \lfloor \frac{n-1}{2} \rfloor, \\ (U_i p^i)^{\gamma_0}, & \lfloor \frac{n+1}{2} \rfloor \leq i \leq n-1. \end{cases}$$

Предположим, что длина конвейера удовлетворяет условию

$$p \geq R - 1.$$

Тогда такой выбор порядка обхода и функции распределения по банкам обеспечивает отсутствие конфликтов при работе самосортирующего алгоритма для архитектуры потокового БПФ акселератора с R банками 1r1w памяти при выполнении одной бабочки размера R или q бабочек размера r за такт.

В **четвертой главе** рассмотрено аппаратное ускорение решения уравнений Юла–Уокера на основе быстрого алгоритма Шура с использованием аппаратного блока вычисления БПФ на примере системы эхоподавления.

В данной работе используется модель эхоподавления при помощи внутренней отрицательной обратной связи, которая сводится к решению в реальном времени системы уравнений Юла-Уокера $Th = c$, где T - тёплицева, положительно определенная матрица. Длина вектора h — 10^4 и более.

Для обращения матрицы T используется быстрый алгоритм Шура [Ammar, Gragg, 1987] для факторизации T^{-1} на основе БПФ.

Лемма 3 *Для реализации быстрого алгоритма Шура длины $n = 2^p$ достаточно $M(n) = 4n$ ячеек памяти, вещественных или комплексных в зависимости от типа входных данных.*

Будем рассматривать реализацию алгоритма Шура для акселератора БПФ в форме антимашины Хартенштейна.

Лемма 4 *Количество операций чтения $T_1(2^m)$ в рассматриваемой реализации быстрого алгоритма Шура удовлетворяет неравенствам $T_1(2^m)^- \leq T_1(2^m) \leq T_1(2^m)^+$, где*

$$\begin{aligned} T_1(2^m)^- &= 1.25 \cdot 2^m m^2 + 7.25 \cdot 2^m, \\ T_1(2^m)^+ &= 1.25 \cdot 2^m m^2 + 9.75 \cdot 2^m. \end{aligned}$$

Лемма 5 *Количество операций чтения $T_{max}(2^m)$ на критическом пути в реализации быстрого алгоритма Шура удовлетворяет неравенствам $T_{max}(2^m)^- \leq T_{max}(2^m) \leq T_{max}(2^m)^+$, где*

$$\begin{aligned} T_{max}(2^m)^- &= 13 \cdot 2^m - 2m - 4, \\ T_{max}(2^m)^+ &= 17 \cdot 2^m - 2m - 4. \end{aligned}$$

Лемма 6 *Пусть $n = 2^m$ и конвейер имеет длину l. Тогда время исполнения быстрого алгоритма Шура $T_{2k}(n)$ для вещественных данных на 2^k процессорах удовлетворяет неравенству $T^-_{2k}(n) \leq T_{2k}(n) \leq T^+_{2k}(n)$, где*

$$\begin{aligned} T^-_{2k}(n) &= 2^{m-k}(13 \cdot 2^k - 1.5k + 1.25m^2 - 1.25k^2 - 7.75) \\ &\quad +(l-1)(13 \cdot 2^m - 2m - 4), \\ T^+_{2k}(n) &= 2^{m-k}(17 \cdot 2^k - 1.5k + 1.25m^2 - 1.25k^2 - 2.75) \\ &\quad +(l-1)(17 \cdot 2^m - 2m - 4). \end{aligned}$$

В качестве примера был проведен анализ оптимального параллелизма и типа памяти акселератора БПФ при вычислении адаптивного линейного фильтра эхоподавления на 4096 отсчетов в реальном времени.

Были получены оценки мощности с помощью закона Амдала и прямой оценки времени работы при данном параллелизме p. Обе оценки хорошо согласуются друг с другом. Минимальное значение мощности достигается при

использовании однопортовой памяти и $p = 4$, что приводит к уменьшению потребляемой мощности на 27%.

В **заключении** приведены основные результаты работы, которые заключаются в следующем:

1. Разработан метод качественной оценки мощности и выбора оптимального параллелизма для энергоэффективных специализированных КМОП вычислительных блоков для параллельных вычислений.
2. Разработан метод вычисления элементарных функций при помощи почти гладкого четырехзвенного квазисплайна и оценка точности полиномиальной аппроксимации с коэффициентами с фиксированной точкой, ограниченной на равномерной сетке.
3. Доказана теорема о размещении данных БПФ в многобанковой памяти при вычислении по произвольным смешанным основаниям.
4. Доказана теорема о размещении данных и порядке вычисления самосортирующегося БПФ.
5. Доказана теорема о размещении данных и порядке вычисления БПФ для однопортовой памяти.
6. Проведен анализ энергоэффективности алгоритма LU факторизации вещественных тёплицевых матриц на сверточном акселераторе для задачи эхокомпенсации при помощи быстрого алгоритма Шура.

Публикации автора по теме диссертации

1. Салищев С.И. **Вычислительные аспекты компенсации акустического эха // Гироскопия и навигация**. 2005. № 1. с. 90.
2. Салищев С.И. **Быстрый алгоритм Шура в задаче подавления акустического эха // Вестник молодых ученых. Серия: прикладная математика и механика**. 2005. Т. 3. С. 77–87.
3. Салищев С.И. **Кусочно-полиномиальная аппроксимация с сокращенными таблицами и гарантированной точностью // Компьютерные инструменты в образовании**. 2012. № 5. С. 3–10.
4. Салищев С.И. Шеин Р.Е. **Новые алгоритмы для конвейерного вычисления БПФ по смешанному основанию без копирования на многобанковой памяти с произвольным доступом // Компьютерные инструменты в образовании**. 2013. № 2. С. 18–30.

5. Echo Compensation by Equalizer with Precise Spectrum Estimation / S. I. Salischev, A. E. Barabanov, K. M. Putyakov et al. // Audio Engineering Society Conference: 21st International Conference: Architectural Acoustics and Sound Reinforcement. 2002. Jun. URL: http://www.aes.org/e-lib/browse.cfm?elib=11191.

6. Salischev S. Computational aspects of real-time acoustic echo cancellation // 7th international conference: Computer data analysis and modeling. Vol. 2. 2004. P. 146–149.

7. Салищев С.И. Ушаков Д.С. Использование языков и сред управляемого исполнения для системного программирования // Системное программирование. 2009. Т. 4. С. 198–216.

8. The Moxie JVM experience. Technical Report TRCS-08-01: Tech. Rep.: / S. I. Salishev, S. M. Blackburn, M. Danilov et al.: Australian National University, Department of Computer Science, 2008. Jan.

9. Demystifying Magic: High-level Low-level Programming / S. I. Salishev, D. Frampton, S. M. Blackburn et al. // Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments. VEE '09. New York, NY, USA: ACM, 2009. P. 81–90. URL: http://doi.acm.org/10.1145/1508293.1508305.

10. Static analysis method for deadlock detection in SystemC designs / S. Salishev, M. Moiseev, A. Zakharov et al. // System on Chip (SoC), 2011 International Symposium on. 2011. Oct. P. 42–47.

11. Salishev S., Glukhikh M., Moiseev M. A Static Analysis Approach for Verification of Synchronization Correctness of SystemC Designs // Proceedings of the 2013 Euromicro Conference on Digital System Design. DSD '13. Washington, DC, USA: IEEE Computer Society, 2013. P. 89–96. URL: http://dx.doi.org/10.1109/DSD.2013.17.

12. Salishev S. Continuous-flow conflict-free mixed-radix fast fourier transform in multi-bank memory. 2014. jul. WO Patent App. PCT/IB2013/000,446. URL: http://google.com/patents/WO2014108718A1.