

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»  
(МАИ)**



На правах рукописи

**КОНСТАНТИНОВ АЛЕКСАНДР АНДРЕЕВИЧ**

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА ИЗМЕРИТЕЛЬНО-ИНФОРМАЦИОННОГО  
И УПРАВЛЯЮЩЕГО КОМПЛЕКСА ДЛЯ ПОЛУНАТУРНОГО  
МОДЕЛИРОВАНИЯ ПОЛЕТА ЛЕТАТЕЛЬНОГО АППАРАТА**

Специальность

05.11.16 – Информационно-измерительные и управляющие системы  
(авиационная, ракетно-космическая техника и кораблестроение)

**ДИССЕРТАЦИЯ**

на соискание учёной степени кандидата технических наук

Научный руководитель –  
кандидат технических наук

**ЕГОРОВ АЛЕКСАНДР АЛЕКСАНДРОВИЧ**

Москва – 2014

## СОДЕРЖАНИЕ

Список сокращений и условных обозначений.....	4
Введение.....	5
Глава 1. Системный анализ метода электромеханического моделирования, как объекта автоматизации. Сравнительный обзор измерительно-информационных и управляющих комплексов для решения задач динамической аэроупругости и постановка задачи исследования.....	15
1.1. Системный анализ метода электромеханического моделирования, как объекта автоматизации.....	15
1.2. Сравнительный анализ измерительно-информационных и управляющих комплексов для наземных стендовых испытаний для решения задач динамической аэроупругости.....	19
1.3. Постановка задачи исследования.....	28
Глава 2. Исследование и модификация архитектуры ИИУК. Оценка и выбор архитектуры ИИУК для решения задач динамической аэроупругости в составе ИАВ.....	30
2.1. Исследование и модификация архитектуры ИИУК.....	30
2.1.1. Функциональная схема ИИУК.....	30
2.1.2. Общий вид структуры сети взаимосвязи ИИУК с АРМ.....	33
2.1.3. Структурная схема АБ ИИУК.....	49
2.1.4. Архитектура расширительных модулей для ИИУК.....	58
2.1.5. Архитектура вычислительного устройства.....	61
2.2. Оценка и выбор архитектуры ИИУК для решения задач динамической аэроупругости в составе ИАВ.....	67
Глава 3. Исследование и оценка цифровой реализации алгоритмов ИИУК на базе ПЛИС, на основе анализа аэродинамических уравнений. Разработка методики создания программного обеспечения для ИИУК с использованием ПЛИС. Оценка эффективности методики.....	71

3.1. Исследование и оценка цифровой реализации алгоритмов ИИУК на базе ПЛИС для ИАВ и оценка погрешностей преобразования. ....	71
3.2. Разработка методики создания программного обеспечения для ИИУК с использованием ПЛИС .....	74
3.3. Построение карт распределения ресурсов ПЛИС. ....	86
Глава 4. Реализация ИИУК для решения задач динамической аэроупругости. Оценка на соответствие требованиям к ИИУК. ....	120
4.1. Реализация ИИУК для решения задач динамической аэроупругости. ....	120
4.2. Оценка на соответствие требованиям к ИИУК. ....	131
ЗАКЛЮЧЕНИЕ .....	134
Список использованной литературы.....	137

## Список сокращений и условных обозначений

- DSP – digital signal processor (цифровой сигнальный процессор);
- RAID – redundant array of independent disks (избыточный массив независимых дисков);
- АБ – автономный блок;
- АДТ – аэродинамическая труба;
- АПК – аппаратно-программный комплекс;
- АЦП – аналого-цифровой преобразователь;
- БД – база данных;
- БУЗ – блок усилителей заряда;
- ДПМ – динамически подобная модель;
- ИАВ – имитатор аэродинамических воздействий;
- ИИУК – измерительно-информационный и управляющий комплекс;
- ЛА – летательный аппарат;
- МУ – модуль управления;
- ОС – операционная система;
- ОСРВ – операционная система реального времени;
- ПВС – потенциометр ввода сил;
- ПК – персональный компьютер
- ПЛИС – программируемая логическая интегральная схема;
- ПО – программное обеспечение;
- САУ – система автоматического управления;
- СДИ – система динамических испытаний;
- УИВК/ЧИ- управляющий измерительно-вычислительный комплекс частотных испытаний;
- ЦАП – цифро-аналоговый преобразователь;
- ЦОС – цифровая обработка сигналов;
- ЦСП – цифровой сигнальный процессор
- ЭММ – электромеханическое моделирование.

## Введение

### Актуальность проблемы:

Одной из важнейших задач в авиа и ракетостроении является исследование колебательных процессов конструкции летательного аппарата (ЛА) в полете. Для того, чтобы получить соответствующие результаты, проводятся наземные эксперименты с применением аэродинамических труб (АДТ). Такие исследования достаточно дороги, а технические средства для совершения таких измерений достаточно сложны. Аэродинамические трубы, в которых производится продувка динамически подобных моделей (ДПМ) и некоторых натуральных элементов конструкций имеют внушительные размеры, обычно несопоставимые с размерами продуваемой модели. Кроме того АДТ, в силу своих размеров и масс, очень инерционны, что не позволяет моментально остановить компрессорное колесо, а вместе с ним и воздушный поток, в случае возникновения аварийной ситуации, которая может привести к разрушению конструкции исследуемого объекта и повреждению элементов конструкции АДТ.

В первой половине 20 века был предложен метод электромеханического моделирования (ЭММ), суть которого сводилась к возбуждению колебаний точек исследуемого объекта, при помощи электро-механических силовозбудителей с определенной частотой и амплитудой, основной целью которого было, нахождение критических параметров колебаний, при которых возникало разрушение испытуемого образца, без проведения экспериментов над ним в АДТ.[23]

На сегодняшний день, на основе метода ЭММ разработана методика, позволяющая рассчитать значение воздействующих сил со стороны воздушного потока на ЛА, во время его движения при наземных испытаниях без использования АДТ. Исходными данными для решения задач, основанных на данной методике являются плотность набегающего потока, скоростной напор, число Маха и число Струхалья. [18, 20, 21,22, 23, 28]

В стендовых условиях обеспечивается значительное расширение режимов потока для модели, рассчитанной на испытания лишь для определенной аэродинамической трубы. Поскольку на стенде значения параметров виртуального воздушного потока и их сочетаний принципиально не ограничены, на одной модели возможны исследования любых режимов потока, включая недостижимые в АДТ. Средства ЭММ обеспечивают также воспроизведение других силовых воздействий, включая нелинейные характеристики конструкции. Электронное ограничение амплитуды колебаний предотвращает повреждение модели при вхождении в резонанс или малых величинах демпфирования. [37]

Таким образом появилась возможность создания нового типа измерительно-информационных и управляющих комплексов (ИИУК), в которых можно создавать колебания точек исследуемого объекта по рассчитанным характеристикам колебаний, в зависимости от вышеуказанных параметров.

Предыдущее поколение ИИУК для решения задач динамической аэроупругости было реализовано посредством больших вычислительных аналоговых машин. Постоянное развитие аэродинамических теорий, совместно с развитием техники, порождают все новые и новые требования к ИИУК, соблюдение которых необходимо для более качественного уточнения разрабатываемых моделей.

Исходя из выше сказанного, является актуальной задача исследования и разработки ИИУК, представляющего собой имитатор аэродинамических воздействий на натурную конструкцию ЛА.

#### Объект исследования:

Объектом исследования является аппаратно-программный комплекс (АПК) для ИАВ имитирующий воздействия аэродинамических сил на ЛА при проведении наземных стендовых испытаний.

### Предмет исследования:

Предметом исследования являются программное и аппаратное обеспечение информационно-измерительного и управляющего комплекса (ИИУК) для имитатора аэродинамических воздействий (ИАВ), позволяющие моделировать полетные условия ЛА при полунатурных стендовых испытаниях конструкций ЛА.

### Цель диссертационной работы

Целью диссертационной работы является исследование и разработка для ИАВ специализированного быстродействующего ИИУК, его методических, аппаратных и программных средств, обеспечивающих полунатурное моделирование аэродинамических воздействий во время проведения наземных испытаний конструкций ЛА с использованием метода электромеханического моделирования.

### Задачи диссертационной работы

Исходя из поставленной цели в процессе выполнения работы решаются следующие научно-технические задачи:

1. Исследование и модификация архитектуры ИИУК для ИАВ;
2. Исследование особенностей использования ПЛИС для оптимизации схем устройств управления ИИУК;
3. Реализация ПО для многоканальных ИИУК, работающих в режиме жесткого реального времени на основе анализа математического аппарата ИАВ;
4. Исследование и оценка цифровой реализации алгоритмов ИИУК на базе ПЛИС, на основе анализа аэродинамических уравнений в интегральном виде и формирование набора типовых операций для разрабатываемого алгоритма, реализуемого на ПЛИС;
5. Экспериментальная оценка характеристик реализованных аппаратной и программной частей ИИУК по разработанной методике.

## Основные методы исследования

Для решения поставленных задач в диссертационной работе используются методы цифровой обработки сигналов в режиме реального времени, в том числе метод перехода из аналоговой формы описания математического аппарата в цифровую, современные информационные технологии, ориентированные на параллельные вычисления.

## Новые научные результаты

Научная новизна полученных в диссертационной работе результатов заключается в следующем:

1. Разработана архитектура специализированного быстродействующего трехуровневого ИИУК, построенного на базе открытых международных модульных структурах РХИ, с использованием операционных систем жесткого реального времени и структур «АЦП-ПЛИС-ЦАП» для ИАВ. Разработанная архитектура обеспечивает выполнение сформулированных технических требований к быстродействию и точности, построена на базе технологий цифровой обработки сигналов в режиме реального времени с применением методов реализации параллельного выполнения алгоритмов сбора данных с датчиков (акселерометров), обработки и выдачи управляющих воздействий (по каналам).

2. Разработан алгоритм преобразования аэродинамических уравнений ИАВ из аналоговой формы в цифровую форму, с описанием в числах с фиксированной запятой (заданной разрядности), с выбором типовых арифметических операций для эффективной по времени реализации на ПЛИС.

3. Разработана математическая модель для оптимизации ресурсов ПЛИС, связывающая скорость выполнения программного кода, количество ресурсов, требуемых для его реализации и точности вычислений (целочисленный формат 8-64 разряда и числа с фиксированной запятой). Модель построена для типовых арифметических операций и позволяет при заданных скорости выполнения

программного кода и точности вычислений определить необходимые ресурсы ПЛИС.

4. Разработана методика проектирования программного обеспечения специализированного ИИУК, с использованием структур «АЦП-ПЛИС-ЦАП», обеспечивающая возможность управления выбором затрачиваемых ресурсов (емкость ПЛИС), времени исполнения кода ПЛИС (быстродействие) и точности вычислений (разрядность данных).

5. Разработан алгоритм применения созданной методики с целью обеспечения эффективности выбора технических параметров АПК на базе ПЛИС для реализации перспективных многоканальных измерительно-управляющих систем нового поколения.

#### Достоверность научных результатов

Достоверность разработанных архитектуры специализированного ИИУК, математических моделей и разработанных методик подтверждена согласованностью результатов математического моделирования и экспериментальных исследований специализированного ИИУК в составе лабораторного образца ИАВ.

#### Личный вклад автора

Личный вклад автора состоит в разработке и реализации архитектуры специализированного ИИУК и лабораторного образца ИАВ, в переводе аэродинамических уравнений в цифровое представление и оценка погрешностей преобразования, разработке цифрового представления аэродинамических уравнений в числах с фиксированной запятой, для реализации на ПЛИС, разработке методики создания ПО на ПЛИС, учитывающей требования по занимаемым ресурсам ПЛИС и быстродействию для специализированного ИИУК, формировании списка типовых математических операций для аэродинамических уравнений в формате чисел с фиксированной запятой, построение карт

распределения ресурсов для типовых операций и приведение их к полиномиальному описанию.

#### Научные положения, выносимые на защиту

1. Анализ и разработка архитектуры специализированного быстродействующего трехуровневого ИИУК, построенного на базе открытых международных модульных структур, с использованием операционных систем жесткого реального времени и структур «АЦП-ПЛИС-ЦАП» для ИАВ.
2. Алгоритм преобразования аэродинамических уравнений ИАВ из аналоговой формы в цифровую форму, с описанием в числах с фиксированной запятой (заданной разрядности), с выбором типовых арифметических операций для эффективной по времени реализации на ПЛИС.
3. Математическая модель для оптимизации ресурсов ПЛИС, связывающая скорость выполнения программного кода, количество ресурсов, требуемых для его реализации и точности вычислений (целочисленный формат 8-64 разряда и числа с фиксированной запятой).
4. Методика проектирования программного обеспечения специализированного ИИУК, с использованием структур «АЦП-ПЛИС-ЦАП», обеспечивающая возможность управления выбором затрачиваемых ресурсов (емкость ПЛИС), времени исполнения кода ПЛИС (быстродействие) и точности вычислений (разрядность данных).
5. Алгоритм применения разработанной методики с целью обеспечения эффективности выбора технических параметров АПК на базе ПЛИС для реализации перспективных многоканальных измерительно-управляющих систем нового поколения.

### Практическая ценность

Значение результатов диссертационной работы для практического применения заключается в том, что:

- Разработан опытный образец специализированного ИИУК на базе созданной методики проектирования математического аппарата на ПЛИС, который экспериментально подтвердил заданные технические характеристики, соответствующие современным требованиям по быстродействию, многоканальности измерений (сигналы от датчиков-акселерометров) и управления (сигналы управления на силовозбудители) - физические имитаторы аэродинамических воздействий) для ИАВ;
- Разработан лабораторный стенд ИАВ, в состав которого входит ИИУК, реализующий метод полунатурного моделирования (имитации) аэродинамических воздействий.

### Результаты диссертационной работы использовались:

Результаты диссертационной работы используются в предварительных наземных испытаниях в составе лаборатории ЭММ, обеспечивающей исследования при решении задач динамической аэроупругости ЦАГИ, для уточнения информации о характеристиках собственных частот элементов конструкции ЛА, а также в рамках НИОКР «Универсальной Интегрированной Автоматизированной Системы Измерения и Системы Управления» в ОАО НПО «Энергомаш».

### Апробация работы и публикации

Основные теоретические и практические результаты диссертации опубликованы в 2 научных трудах, из них по теме диссертации 2, среди которых 2 публикации [16,25] в ведущих рецензируемых изданиях, рекомендованных в действующем перечне ВАК. Основные результаты работы были представлены на ежегодной Международной конференции «Aerospace Testing & Industrial Control»

(г.Москва 2014г.) и на IX-ой Всероссийской научно-технической конференции «Проблемы совершенствования робототехнических и интеллектуальных систем летательных аппаратов» (г.Москва 2012г.)

### Структура и объем диссертации

Диссертация состоит из введения, 4 глав с выводами и заключения. Она изложена на 143 страницах машинописного текста, включает 51 рисунок, 27 таблиц и содержит список литературы из 68 наименований, среди которых 50 отечественных и 18 иностранных авторов.

### КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность темы диссертационной работы, определяются цели и задачи исследования, а также объект и предмет исследования. Формулируются основные положения, выносимые на защиту, проводятся основные научные и практические результаты, обосновывается их новизна. Приводится содержание и краткое описание глав диссертации.

Первая глава диссертационной работы посвящена анализу метода электромеханического моделирования (ЭММ), сравнительному обзору существующих средств для проведения наземных испытаний ЛА и их частей для решения задач динамической аэроупругости и динамического нагружения, а также ставятся задачи исследования.

Во второй главе предлагается модификация архитектуры ИИУК для реализации ИАВ. Производится оценка архитектуры ИИУК по быстродействию, многоканальности и точности. Показано, что наиболее перспективными высокопроизводительными вычислительными платформами являются те, которые обеспечивают поддержку параллельных вычислений. Для решения

современных задач моделирования, обработки информации в режиме реального времени и управления все чаще используют ПЛИС, поскольку в суммарной производительности они не уступают, а иногда даже и превосходят устройства, построенные на базе микроконтроллеров и многоядерных систем. Дополнительным преимуществом ПЛИС является их универсальность, отказоустойчивость и архитектура, обладающая естественным параллелизмом.

В третьей главе осуществляется преобразование аэродинамических уравнений к цифровому виду. Определяются погрешности, как следствие этого преобразования. Производится приведение типа данных аэродинамических уравнений к числам с фиксированной запятой, для реализации на ПЛИС. Предлагается методика разработки программного обеспечения для ПЛИС, реализующего математический аппарат. Проводится исследование ресурсоемкости и скорости вычисления для типовых математических операций преобразованных аэродинамических уравнений, исполняемых на ПЛИС. Проводится оценивание эффективности разработанной методики.

В четвертой главе представлена реализация ИАВ, в составе ЭММ, выполняющая решение аэродинамических уравнений, с учетом предлагаемой методики реализации алгоритма с учетом технических требований к ИИУК. Проведена оценка полученного решения на соответствие требованиям к ИИУК по быстродействию и точности.

#### Заключение:

В заключении отражены основные результаты работы, отражена практическая ценность полученных результатов.

Основные научные результаты заключены:

1. в проведении анализа и разработке трехуровневой архитектуры ИИУК, на базе модульной структуры РХІ, с использованием

операционных систем жесткого реального времени и структур «АЦП-ПЛИС-ЦАП»;

2. в разработанном алгоритме преобразования аэродинамических уравнений ИАВ в цифровую форму с описанием в числах с фиксированной запятой и выбором типовых арифметических операций для их эффективной реализации на ПЛИС;
3. в разработанной математической модели оптимизации ресурсов ПЛИС, связывающей скорость выполнения программного кода и количество ресурсов ПЛИС;
4. в разработанной методике проектирования ПО специализированного ИИУК с использованием структур «АЦП-ПЛИС-ЦАП»;
5. в разработанном ПО верхнего и нижнего уровня ИИУК на основе разработанной методики для решения задач динамической аэроупругости;
6. в разработке алгоритма применения созданной методики с целью обеспечения эффективности выбора технических параметров АПК на базе ПЛИС для реализации перспективных многоканальных измерительно-управляющих систем нового поколения;
7. в разработанном лабораторном образце специализированного ИИУК на базе технологий LabVIEW, подтверждающий экспериментально соответствие современным техническим требованиям для ИАВ;
8. в разработке и реализации лабораторного стенда полунатурного моделирования (имитации) аэродинамических воздействий, реализующий метод ЭММ и проведена экспериментальная оценка характеристик разработанного ИИУК в составе ИАВ с моделью конструкции ЛА.

# **Глава 1. Системный анализ метода электромеханического моделирования, как объекта автоматизации. Сравнительный обзор измерительно-информационных и управляющих комплексов для решения задач динамической аэроупругости и постановка задачи исследования.**

## **1.1. Системный анализ метода электромеханического моделирования, как объекта автоматизации.**

Метод электромеханического моделирования (ЭММ) сегодня является дополнительным средством исследования устойчивости аэроупругих колебаний. В первую очередь, он используется для моделирования силовых воздействий при наземных испытаниях, как воспроизведение аэродинамических силовых воздействий на ЛА или его ДПМ. Фактически, помимо аэродинамических сил, теми же средствами воспроизводятся и другие, например, силы инерции, демпфирования или гироскопические. Потребность в этом может возникнуть, в частности, для компенсации нежелательного влияния силовозбудителей на испытываемую конструкцию. Другой вариант – воспроизведение гироскопического момента и/или сил инерции в таком случае, как потеря лопатки двигателя.

Идея метода состоит в воспроизведении аэродинамических сил на механическую конструкцию в стендовых условиях, без реального воздушного потока. Для этого необходимо, в частности, заменить распределенные аэродинамические силы, действующие на упругую конструкцию при ее колебаниях в потоке, искусственными (эквивалентными) сосредоточенными силами с помощью электродинамических силовозбудителей.

Этот метод занимает промежуточное положение между полностью экспериментальными и полностью аналитическими методами. Фактически испытывается упругая конструкция – натуральный ЛА или его ДПМ. В то же время аэродинамические силы являются искусственными и должны вычисляться для всех «полетных» условий, возбуждающие силы должны распределяться между возбудителями и воспроизводиться специальным электронным блоком. В

качестве последнего многие годы использовалась аналоговая вычислительная техника. Установка для испытаний и основная часть оборудования являются таким же, как в стандартных наземных резонансных испытаниях.

Преимущество такого подхода очевидно – испытывается конструкция (и система управления) со всеми присущими ей особенностями, включая нелинейности, и не требуется разрабатывать математическую модель ЛА, которая неизбежно содержит определенные упрощения. Главный недостаток также очевиден – искусственные аэродинамические нагрузки не могут быть более точными, чем применяемые аэродинамические теории, как и в расчете. Количество силовозбудителей также всегда ограничено. Блок-схема такого эксперимента выглядит, как и для типовых испытаний с многоточечным возбуждением, однако здесь вместо генератора сигналов используются и преобразуются сигналы датчиков. Схема эксперимента с ЭММ (рис. 1.1) выглядит так же, как для типовых резонансных испытаний с многоточечным возбуждением, однако в данном случае для возбуждения колебаний конструкции используются сигналы датчиков, преобразованные вычислительным блоком.

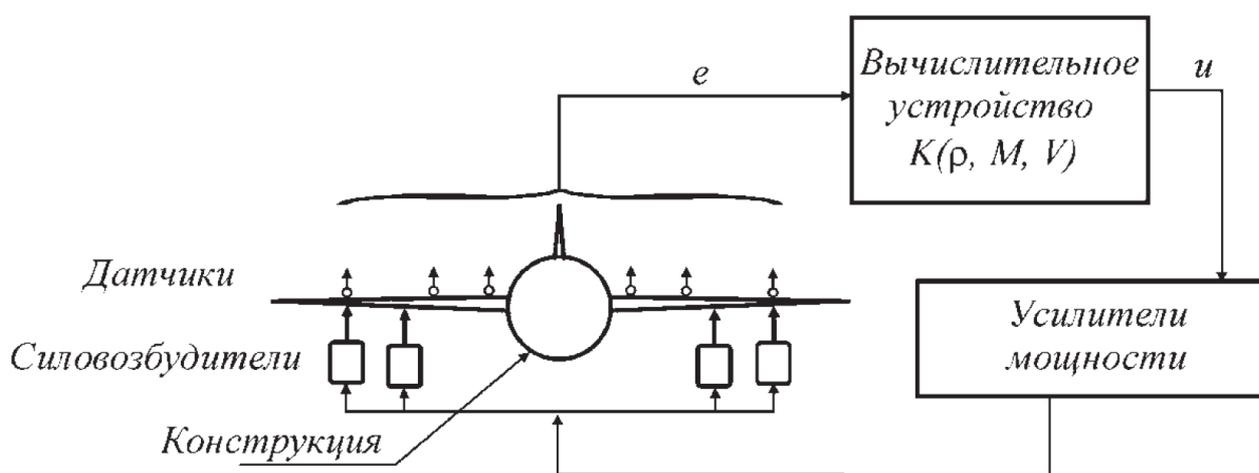


Рис.1.1. Схема испытаний методом ЭММ.



Рис.1.2. Место электромеханического моделирования при исследовании аэроупругости

Многие расчетные схемы приводят к линейной системе дифференциальных уравнений, описывающих малые колебания упругой конструкции в потоке:

$$M \ddot{y}(t) + H \dot{y}(t) + Gy(t) = f^A(t), \quad (1.1)$$

где  $M, H$  и  $G$  - матрицы инерции, демпфирования и жесткости соответственно,  $t$  - текущий момент времени,  $y$  - мгновенное значение вектора перемещений точек измерения конструкции;  $f^A$  - мгновенное значение вектора сосредоточенных аэродинамических сил, который может быть вычислен на основе любой аэродинамической теории и процедуры преобразования распределенных сил в ряд энергетически эквивалентных сосредоточенных сил. При вывешивании механической конструкции на стенде, значения вектора сосредоточенных аэродинамических сил становятся тождественно равными значениям вектора сил возбуждения [20, 23]:

$$f(t) = f^A(t), \quad (1.2)$$

где  $f$  - вектор сил возбуждения.

При использовании датчиков ускорения, дифференциальное уравнение (1.1) приводится к интегральной форме:

$$Ma(t) + \frac{H}{T} \int a(t)dt + \frac{G}{T^2} \iint a(t)dt^2 = f(t), \quad (1.3)$$

где  $a$  - мгновенное значение ускорения точек конструкции.

Вибродатчики, установленные на конструкции, преобразуют ускорения соответствующих точек в электрические сигналы  $e$ . Эти сигналы преобразуются вычислительным устройством в выходные напряжения  $u$ . После усиления мощности эти напряжения  $u$  преобразуются в силы  $f$  электродинамическими силовозбудителями. Соответствующие преобразования описываются уравнениями:

$$e = Ra ; u = Ke ; f = Qu, \quad (1.4)$$

где  $R$  и  $Q$  - диагональные матрицы калибровочных коэффициентов датчиков и силовозбудителей соответственно;  $K$  - матрица аэродинамических преобразований.

Данный метод оказался эффективным в ряде задач динамической аэроупругости, в том числе в случае, когда адекватная математическая модель конструкции (линейная или нелинейная, с САУ и без нее) отсутствует, неполна или ненадежна, и при условии, когда обычный эксперимент опасен или очень дорог. В настоящее время метод ЭММ наиболее доступен для относительно сложных динамических задач, требующих моделирования в наземных условиях, в т.ч и в модальных испытаниях. [39, 42]

Для исследования динамических процессов при решении задач динамической аэроупругости, к системам измерения и управления предъявляются серьезные требования по числу измерительных и управляющих каналов, а также быстрдействию и точности.

На основании современных работ в области аэродинамической аэроупругости [11, 22, 33, 48, 55], сформулированы технические требования к ИАВ:

1. 8 точек возбуждения;
2. 8 точек измерения;
3. расширяемость числа каналов;
4. фазовые искажения выходного сигнала не более 2 градусов;
5. частотный диапазон измеряемого сигнала от 1 до 300 Гц;
6. программное обеспечение, для решения аэродинамических уравнений, с вводом параметров с вводом значений скоростного потока, плотности и числа Маха.

### **1.2. Сравнительный анализ измерительно-информационных и управляющих комплексов для наземных стендовых испытаний для решения задач динамической аэроупругости.**

Для решения задач проведения модальных испытаний с многоточечным возбуждением, компанией Data Physics предлагается система ABACUS и построенную на нем систему SignalStar Matrix:

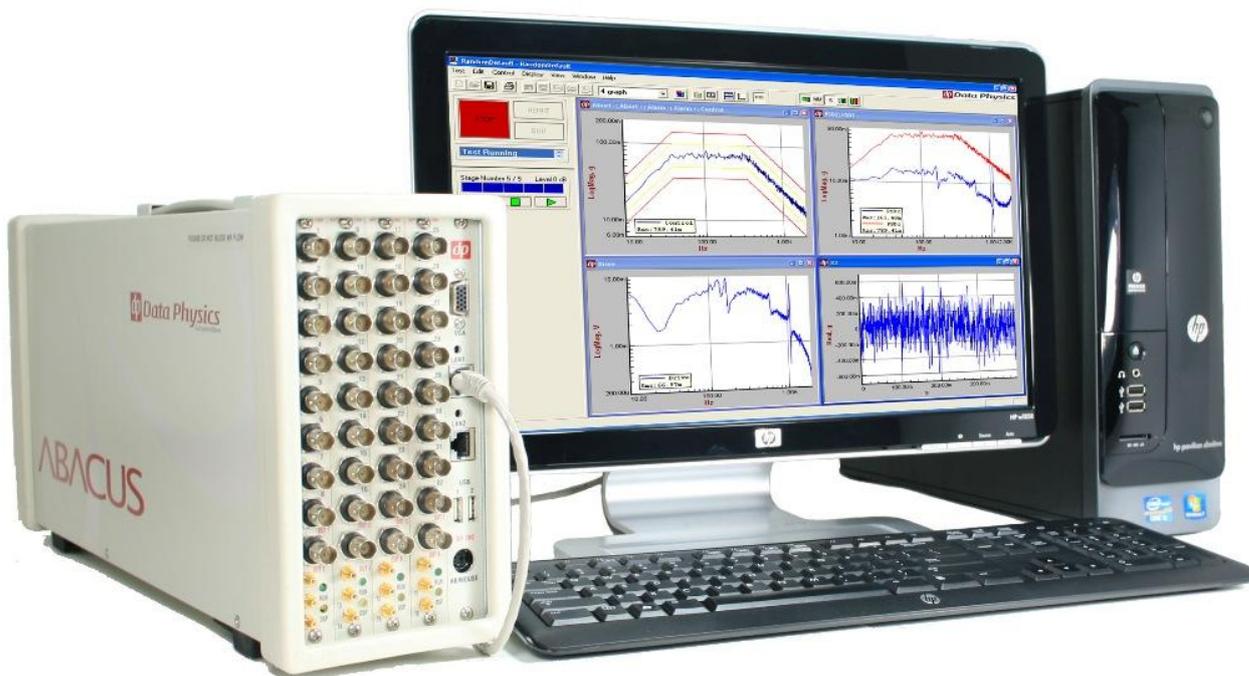


Рис.1.3. Внешний вид системы ABACUS.

ABACUS представляет собой расширяемую высокопроизводительную систему, выполненную в виде шасси со вставленными модулями, предназначенную для анализа сигналов. Расширение системы происходит установкой дополнительных шасси, связывающихся между собой посредством кабеля Ethernet.

Каждый модуль системы ABACUS имеет встроенный DSP-процессор, а каждый канал имеет собственный АЦП.

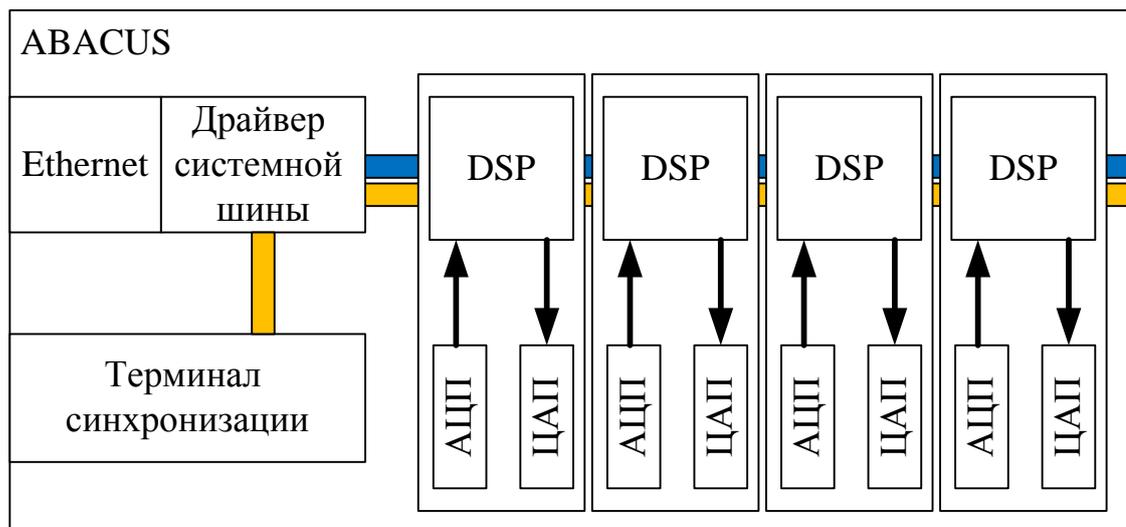


Рис.1.4. Архитектура системы ABACUS.

Ниже приведены основные технические характеристики системы ABACUS.

Таблица 1.1. Технические характеристики ABACUS

Число аналоговых измерительных каналов	от 4 до 32
Разрядность АЦП	24 бит
Тип АЦП	Дельта-сигма
Диапазон измерения	1/2/5/10В
Частота дискретизации	1..107.42 kSps (214.82 опционально)
Число аналоговых каналов управления	от 2 до 8
Разрядность ЦАП	24 бит
Диапазон изменения выходного сигнала	0.1/0.2/0.5/1/2/3/10В

Система SignalStar Matrix представляет собой связку систем ABACUS, работающих вместе по сети Ethernet. Таким образом можно расширить у готовой системы число каналов измерения (до 1024) и управления (до 16).

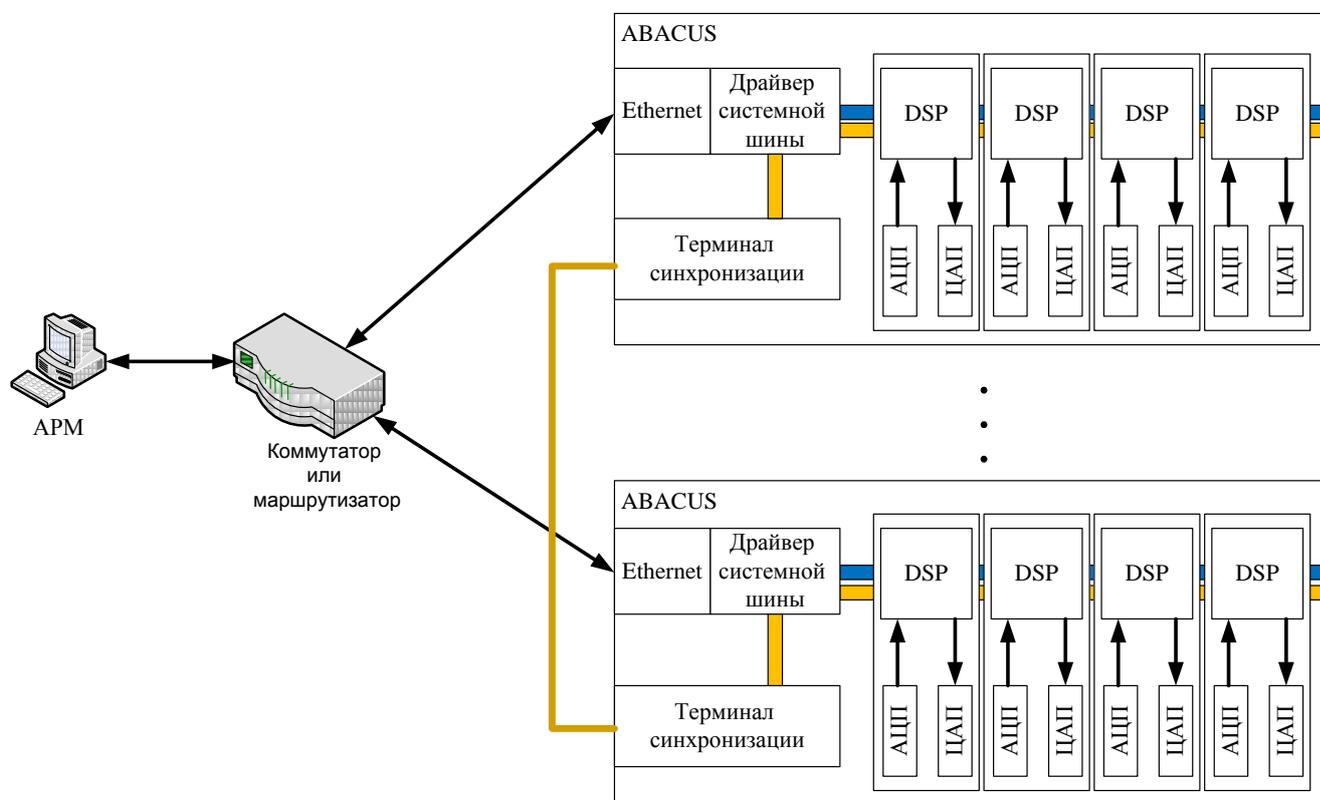


Рис.1.5. Архитектура системы SignalStar Matrix.

Программное обеспечение, поставляемое с SignalStar Matrix позволяет осуществлять анализ сигналов и управление вибростендами, однако, нет решений ИИУК для ИАВ в составе ЭММ ЛА. [35]

Испытания в области динамической аэроупругости активно проводит ЦАГИ. Так, в качестве стендового оборудования используется управляющий и измерительно-вычислительный комплекс частотных испытаний (УИВК/ЧИ) конструкций ЛА. Совместно со средствами возбуждения колебаний конструкций (вибровозбудителями с усилителями мощности) и средствами измерения колебаний конструкций (например, пьезоэлектрическими акселерометрами с усилителями заряда) интегрируются в систему для наземных динамических (частотных) испытаний конструкции (СДИ).

С помощью системы СДИ определяются характеристики собственных колебаний конструкций летательных аппаратов (собственные частоты, формы колебаний, коэффициенты демпфирования и обобщенные массы).

УИВК/ЧИ в этих системах выполняет следующие функции:

- Обеспечивает генерирование и управление амплитудой и фазой тестовых гармонических сигналов, которые поступают на входы соответствующих усилителей мощности, управляя параметрами прикладываемых к конструкции сил;
- Обеспечивает сбор и обработку сигналов с вибропреобразователей (например, с усилителей заряда в случае использования пьезоэлектрических акселерометров);
- На этапе проведения частотных испытаний комплекс обеспечивает визуализацию временных сигналов с датчиков вибраций и фазовых соотношений между приложенными силами и откликами конструкции (например, по фигурам Лиссажу), рассчитывает различные критерии выделения собственного тона, регистрирует временные процессы, амплитудно-фазовые частотные характеристики и формы колебаний конструкции;
- На этапе постобработки выполняет модальный анализ зарегистрированных данных с целью определения искомым характеристик собственных колебаний, оценивает достоверность полученных результатов и представляет результаты в графическом и табличном видах.

Структурная блок-схема УИВК/ЧИ представлена на рис.1.6. УИВК/ЧИ состоит из крейта LTC-45 с функциональными модулями, пульта управления, системного блока компьютера, монитора с клавиатурой, принтера, источника бесперебойного питания и блока электропитания блоков усилителя заряда (БУЗов).

При этом крейт, системный блок компьютера, блок электропитания и источник бесперебойного питания установлены в приборной

стойке.

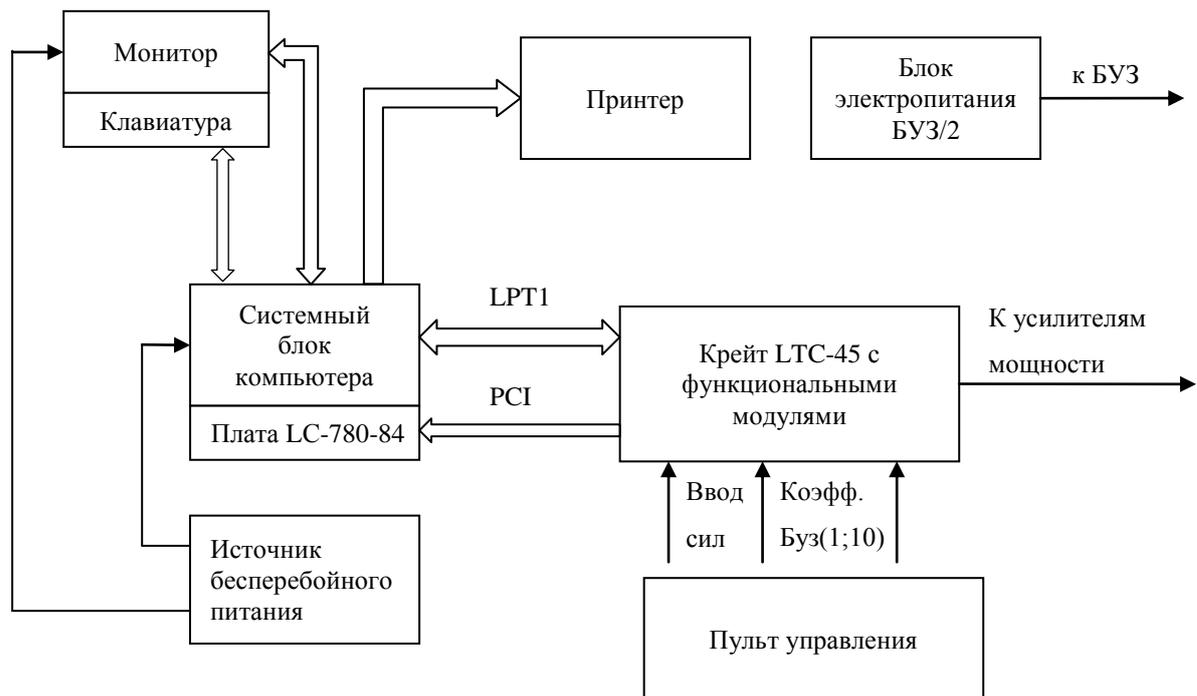


Рис.1.6. Структурная блок-схема УИВК/ЧИ

В состав базового комплекта УИВК/ЧИ входит: крейт (блок) LTC-45 в составе следующих функциональных модулей (ЗАО «Л-Кард»): крейт-контроллер LC-014, коммутатор-усилитель LC-101 (16шт.), АЦП/ЦАП LC-301, цифровой ввод/вывод LC-403 U, ЦАП LC-302, блок управления LC-Y, блок электропитания LTC-PW; системный блок промышленного компьютера с платой 32-х канального АЦП и 2-х канального ЦАП LC-780-84 (ЗАО «Л-Кард), установленной на шине РСІ, пульт ручного управления, монитор, принтер, источник бесперебойного питания.

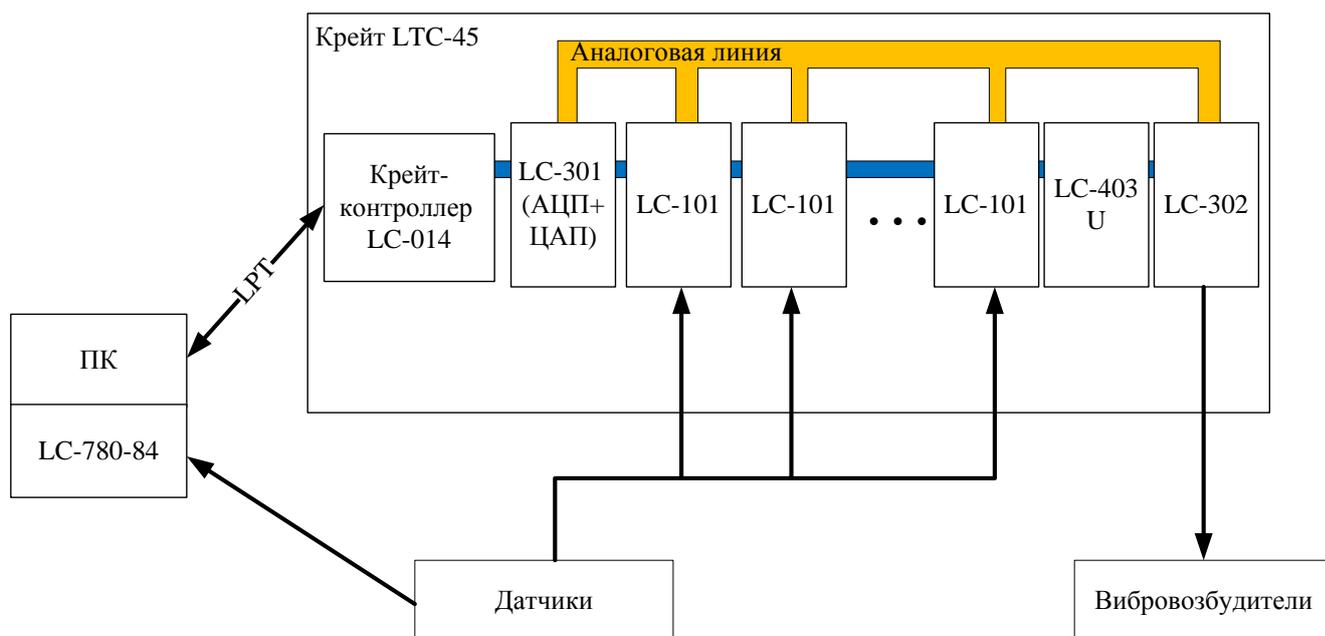


Рис.1.7. Архитектура УИВК/ЧИ.

С помощью программируемых ЦАП формируются 2 выходных аналоговых нормированных сигнала. Один из них имеет форму синуса ( $\sin$ ), а другой – косинуса ( $\cos$ ).

Выходные сигналы с ЦАП поступают на вход модуля управления (МУ), где суммируются. Фаза выходного сигнала МУ зависит от соотношения амплитуд сигналов  $\sin$  и  $\cos$ . Необходимое соотношение амплитуд задается с помощью модуля цифрового управления. Нормированный сигнал заданной фазы поступает на вход потенциометра ввода амплитуды сил – ПВС, а с выхода его подвижной «средней» точки сигнал поступает на вход модуля установки сил по каналам. Сигнал с выхода последнего подается на входы усилителей мощности, обеспечивающих питание вибровозбудителей.

Развиваемая вибровозбудителем сила пропорциональна току, протекающему через его подвижную катушку, который в свою очередь пропорционален сигналу управления усилителя мощности. Модуль установки амплитудных значений сил по каналам формирует сигналы управления на 8 усилителей мощности путем распараллеливания входного сигнала.

Сигналы с блоков усилителей заряда поступают на входы коммутатора. Одновременно на входы коммутаторов могут подключаться до 256 вибропреобразователей.

Регистрация сигналов от вибропреобразователей осуществляется по цепи: «коммутатор – модуль АЦП1 – порт LPT – память контроллера». При этом с выхода коммутатора сигналы с вибропреобразователей последовательными группами по 32 параллельных канала без промежуточной фильтрации поступают на вход многоканального АЦП1, который преобразует аналоговые сигналы в цифровые коды и загружает их в память компьютера.

Базовый комплект УИВК/ЧИ имеет следующие основные технические характеристики [4]:

- Число каналов управления возбуждением колебаний: 8;
- Число каналов сбора данных (параллельными группами по 32): 256;
- Форма сигнала управления: гармоническая;
- Диапазон частот сигналов управления и сбора, Гц: 0,1 – 1000;
- Максимальные амплитуды сигналов управления и сбора, В: 5;
- Приведенная погрешность задания сигнала по амплитуде, %: 0,5;
- Относительная погрешность задания сигнала по частоте: 0,001;
- Приведенная погрешность выделения синфазных и квадратурных составляющих входных сигналов, %: 2;
- Масса, кг: 45;

Кроме приведенных систем, в рамках программы развития Национального Исследовательского Университета (НИУ), создан стенд лаборатории ЭММ на каф. №602.

В состав данного стенда входят:

- Средства возбуждения колебаний (8 вибровозбудителей электродинамического типа в комплекте с усилителями мощности фирмы TMS, США);
- Средства измерения колебаний – акселерометры (24 шт.) и импендансные головки (8 шт.) фирмы PCB Piezoelectronics;
- Управляющий программно-аппаратный комплекс в составе персонального компьютера, измерительно-управляющей системы

реального времени CompactRIO (National Instruments, США) и стандартного программного обеспечения National Instruments.

Для управления испытаниями, реализующих режимы гармонического возбуждения испытываемой конструкции в заданном диапазоне частот, измерение и обработку сигналов датчиков, вычисление в реальном времени аэродинамических воздействий разработано программное обеспечение для системы CompactRIO.

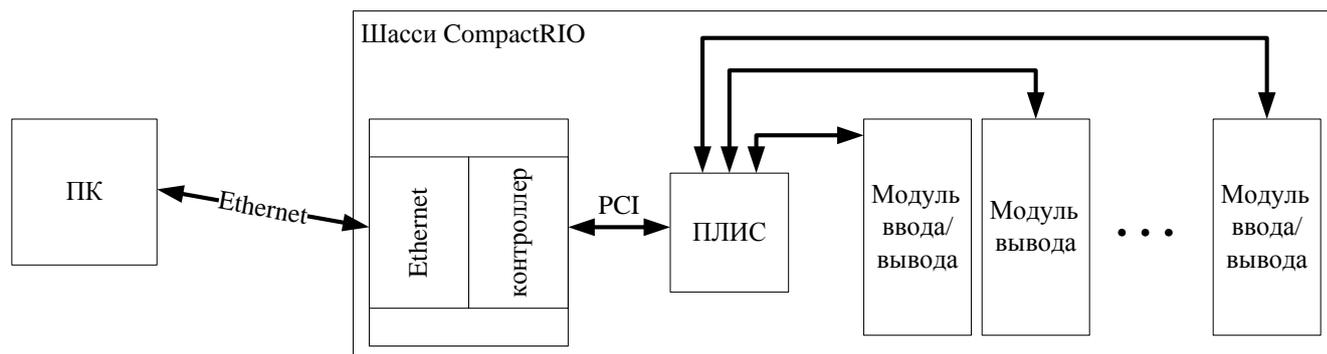


Рис.1.8. Архитектура ИИУК лаборатории ЭММ МАИ.

Это позволило проводить комплексные экспериментальные исследования аэроупругой устойчивости беспилотных ЛА, включая:

- проведение резонансных испытаний (определение частотных характеристик) упругой конструкции ЛА;
- исследования флаттера ЛА;
- определение характеристик системы «руль-привод» (амплитудных и частотных характеристик, запасов устойчивости, динамической жесткости);
- определение частотных характеристик системы «упругий ЛА-САУ»;
- расчетно-экспериментальное определение запасов устойчивости контура «упругий ЛА-САУ»;
- исследование характера возможных автоколебаний (частоты, предельные циклы, уровень вибрационных перегрузок). [46]

В ОАО ГосМКБ «Вымпел» был установлен специализированный измерительный комплекс НВ-16 (измерительная станция Н-2000),

представляющий собой основанный на ПК крейт (процессор Pentium III от 1000 МГц, ОЗУ 256 Мб) с возможностью установки до 16 специализированных модулей.

Основные параметры управляющей части:

- число каналов возбуждения - до 8, измерения – 16-32;
- частота дискретизации цифро-аналоговых преобразователей (ЦАП) 40 кГц (каждого канала), номинальная разрядность - 16 бит;
- диапазон частот возбуждения 2...500 Гц, при дискретности задания частоты 0,0001 Гц и стабильности 0,01%;
- амплитуда сигналов возбуждения 0...10В при дискретности задания 1 мВ с погрешностью амплитуды 0,1 %;
- диапазон частот измеряемых сигналов определяется частотой аналого-цифровых преобразователей (АЦП) и величиной приемлемой погрешности измерения. Частота дискретизации АЦП 20 кГц (каждого канала), номинальная разрядность АЦП - 16 бит.

Система возбуждения содержит:

- программные генераторы гармонических сигналов, управляющие модулями с ЦАП. Фазовые сдвиги выходных сигналов генераторов могут составлять 0,  $\pi$  а также  $\pm\pi/2$  относительно выбранного канала (или произвольное значение). Уровень этих сигналов устанавливается индивидуально для каждого канала по заданному амплитудному значению силы возбуждения, а также возможно одновременное изменение уровня всех каналов («общего уровня»);
- усилители мощности - генераторы тока, преобразующие выходные напряжения ЦАП в пропорциональный им ток силовых возбудителей, не зависящий от колебаний точек возбуждения и от изменения величины полного сопротивления нагрузки;
- электродинамические силовозбудители, у которых сила, приложенная к подвижной системе, пропорциональна току и не зависит от частоты и от

колебаний точки возбуждения в пределах рабочего перемещения. При использовании силовозбудителей с электромагнитом как источником магнитного поля, используются соответствующие блоки подмагничивания. Для возбуждения колебаний используются отечественные (АВДИ) или импортные (Prodera) силовозбудители, при этом разной мощности: для корпуса – на 200, 500 и 1000 Н, для органов управления – на 50 и 200 Н.

Система измерения включает в себя:

- датчики – используются преимущественно пьезоакселерометры и пьезодатчики силы (или комбинированные - импедансные головки). Перспективным является использование «интеллектуальных» датчиков (TEDS, с встроенным микрочипом);
- согласующую аппаратуру – «предусилители», а также средства фильтрации;
- программные модули: синхронного детектирования (вычисляющий в реальном времени значения амплитуд синфазного и квадратурного компонентов измеряемого сигнала для каждого канала), вычисления действующих значений, БПФ и другие.

Напряжения предусилителей поступают на входы модулей АЦП. Использование модулей с автономными АЦП в каждом канале принципиально исключает сдвиги по времени отдельных входных напряжений. [8]

### **1.3. Постановка задачи исследования**

Приведенный выше анализ показывает, что на данный момент существующие решения не соответствуют современным требованиям к ИИУК для решений задач динамической аэроупругости. В связи с этим, необходимо разработать ИИУК для ИАВ в составе ЭММ ЛА.

В соответствии с этим сформулированы следующие задачи, для выполнения сформулированных требований к ИИУК:

1. Сформулировать требования к современным информационно-измерительным и управляющим комплексам (таким как ИАВ) с повышенным требованием по быстродействию, многоканальности и точности;
2. Разработать архитектуру ИИУК для решения задач динамической аэроупругости;
3. Провести преобразование аэродинамических уравнений из аналоговой формы к дискретной;
4. Разработать методику реализации математического аппарата на ПЛИС для многоканальных систем измерения и управления с жестким требованием по точности и быстродействию;
5. Разработать алгоритм ПО, реализующий математический аппарат, на основе представленной методики;
6. Разработать ИАВ в составе ЭММ для решения задач динамической аэроупругости;
7. Оценить характеристики ИАВ на соответствие сформулированным требованиям.

#### **Выводы по первой главе.**

1. Проведенный анализ метода электромеханического моделирования показал, что для создания измерительно-информационного и управляющего комплекса (ИИУК) имитатора аэродинамических воздействий (ИАВ), необходима реализация аэродинамических уравнений (1.3) с высокоскоростной обработкой сигналов;
2. Сформулированы требования для измерительно-информационных и управляющих комплексов (ИИУК) для реализации ИАВ;
3. Представленный сравнительный анализ ИИУК показал, что существующие реализации не соответствуют сформулированным требованиям;
4. Приведена постановка задачи исследования.

## Глава 2. Исследование и модификация архитектуры ИИУК. Оценка и выбор архитектуры ИИУК для решения задач динамической аэроупругости в составе ИАВ.

### 2.1. Исследование и модификация архитектуры ИИУК.

#### 2.1.1. Функциональная схема ИИУК

В основе математического аппарата метода ЭММ лежит решение аэродинамических уравнений (1.3), основным источником информации для которых, являются, сигналы с датчиков с одной стороны, а также математическое описание измерений объекта и параметры моделируемого воздушного потока с другой. На Рис.2.1. изображена схема взаимодействия ИИУК с исследуемым объектом.

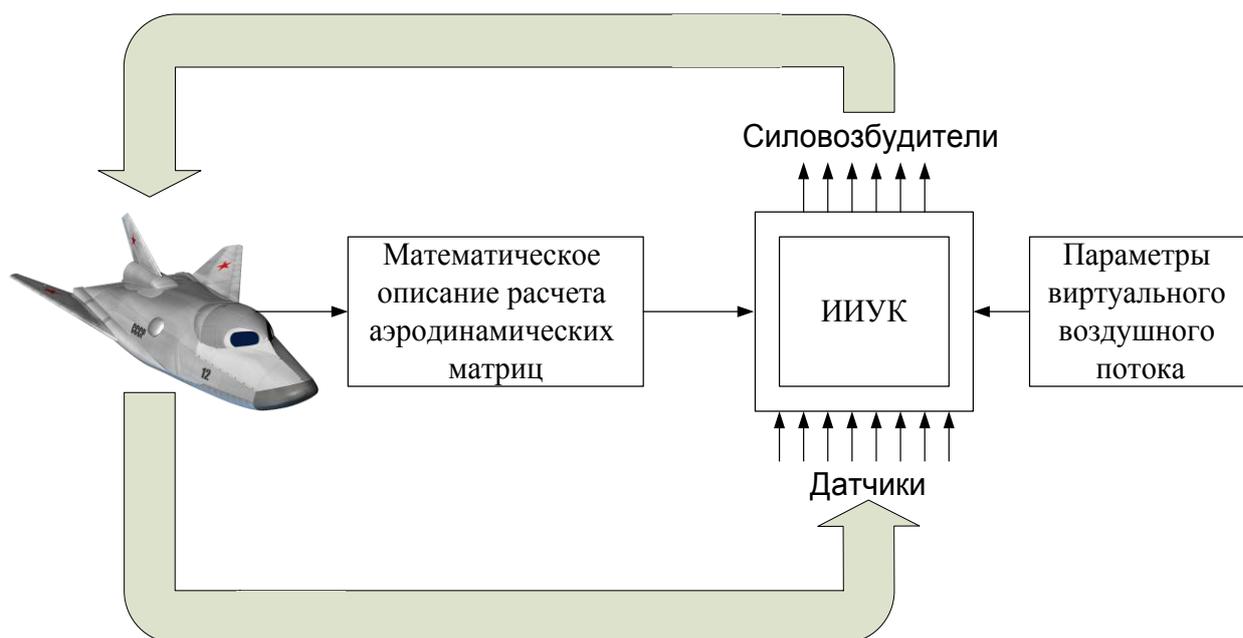


Рис.2.1. Схема взаимодействия ИИУК с исследуемым объектом

В современных стендовых испытаниях при решении задач динамической аэроупругости исследуются динамические процессы, анализ которых позволяет уточнить свойства ДПМ или натурного ЛА. Для того, чтобы максимально точно определить характеристики исследуемого объекта и упростить процедуру математического анализа, необходимо обеспечить синхронность поступающей

информации в систему. Кроме того, необходимо обеспечить безопасную и синхронную работу в режиме реального времени контура управления. Поэтому, для исследования динамических процессов ключевую роль играют системы синхронного измерения и управления, осуществляющие одновременный опрос измерительных и выдачи сигналов для управляющих каналов, с минимальным расхождением по времени. Величина предельного расхождения опроса измерительных и управляющих каналов задается для каждой системы, стенда и решаемой задачи индивидуально. Кроме задач измерения и управления, необходимо, в режиме реального времени производить различного рода вычисления, что накладывает серьезные ограничения и высокие требования к быстродействию системы измерения и управления.

Поскольку любой процесс испытания подразумевает взаимодействие оператора с ИИУК, последнему требуется человеко-машинный интерфейс, с выводом информации в графическом и цифровом, понятном оператору виде.

Для организации графического пользовательского интерфейса обычно используются вычислительные системы на базе персональных компьютеров (ПК) с операционной системой (ОС) Windows, Linux, MacOS. Все вышеперечисленные операционные системы, с организацией графического интерфейса, не являются ОС реального времени. Время выполнения кода в таких ОС зависит от многих факторов, которые могут приводить к непредсказуемым задержкам. К таким факторам относятся:

1. Параллельная работа сервисных программ, таких как экранная заставка, антивирусные программы и т.д.;
2. Обслуживание ОС прерываний от периферийных устройств, таких как USB порт, клавиатура, мышь и т.д. [63]

Операционные системы реального времени (ОСРВ) предназначены для обеспечения интерфейса к ресурсам критических по времени систем.

В качестве основного требования к ОСРВ выдвигается требование обеспечения предсказуемости или детерминированности поведения системы в

наихудших внешних условиях, что резко отличается от требований к производительности и быстродействию универсальных ОС.

Принято различать системы мягкого (soft) и жесткого (hard) реального времени. В системах жесткого реального времени неспособность обеспечить реакцию на какие-либо события в заданное время ведет к отказам и невозможности выполнения поставленной задачи. Системами мягкого реального времени называются системы, не попадающие под определение "жесткие", т.к. в литературе четкого определения для них пока нет. Системы мягкого реального времени могут не успевать решать задачу, но это не приводит к отказу системы в целом. В системах реального времени необходимо введение некоторого директивного срока, до истечения которого задача должна обязательно (для систем мягкого реального времени – желательно) выполняться. Этот директивный срок используется планировщиком задач как для назначения приоритета задачи при ее запуске, так и при выборе задачи на выполнение. [5]

Для того, чтобы обеспечить должный уровень быстродействия, и при этом, иметь возможность использовать интуитивно-понятный и дружелюбный графический интерфейс, необходимо систему разделить функционально на два уровня, по выполняемым задачам:

- 1) Автоматизированное рабочее место (АРМ) оператора (ПК, обеспечивающий отображение процесса испытания в графическом виде, ввод данных во время испытания, а также подготовку и настройку ИИУК до испытания);
- 2) Автономный блок (АБ) ИИУК (обеспечивает получение данных от датчиков, производит пересчет в выходные сигналы для силовозбудителей, обеспечивает синхронность исполнения по всем каналам и работу в режиме реального времени).

Для связи АРМ оператора и автономного блока (АБ) ИИУК используется высокоскоростной канал передачи данных. Кроме того, в данной конфигурации становится возможным разместить АБ ИИУК в непосредственной близости от датчиков, минимизировав протяженность проводов и их массу, которая может

оказывать негативное влияние на испытание, объект испытаний, а также обеспечить лучшую помехозащищенность. При этом АРМ оператора может быть на значительном удалении от АБ ИИУК.

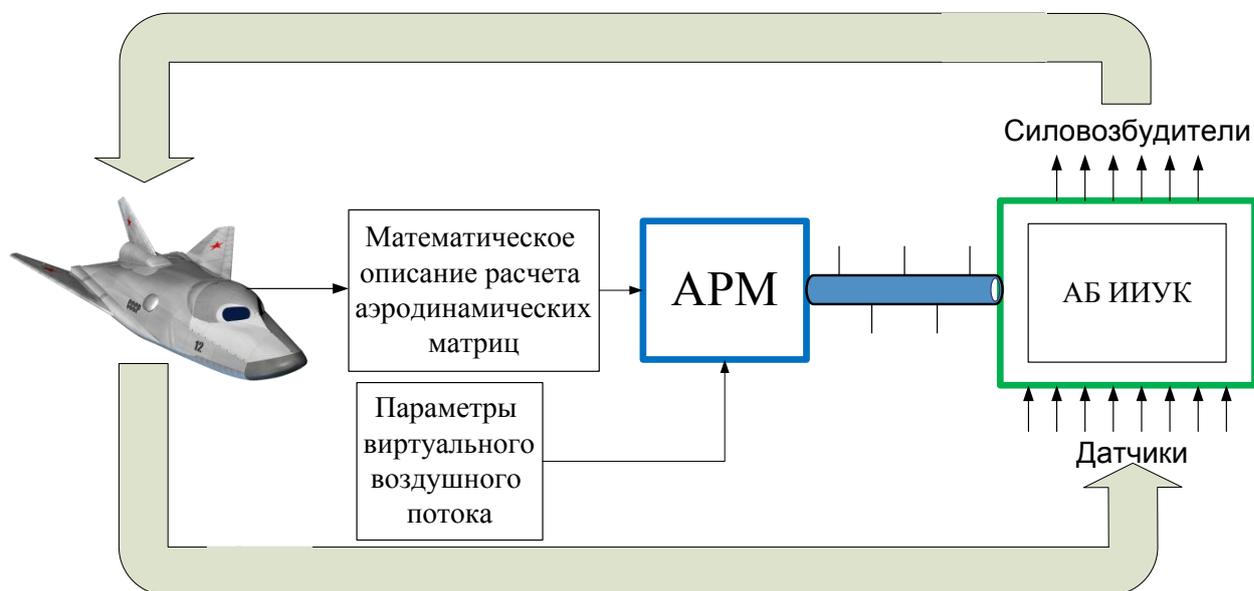


Рис.2.2. Схема функционального разделения уровней ИИУК

Для определения типов каналов связи АРМ с АБ ИИУК, необходимо учесть тенденции к использованию многопользовательского и распределенного доступа, при проведении испытаний, а также хранение истории испытаний и их результатов.

Для этого необходимо разработать общий вид структуры сети.

### **2.1.2. Общий вид структуры сети взаимосвязи ИИУК с АРМ**

Чтобы обеспечить многопользовательский режим доступа, необходим коммутационный узел, соединяющий всех клиентов - АРМ с АБ ИИУК, а также с сервером базы данных (БД) и файловым сервером.

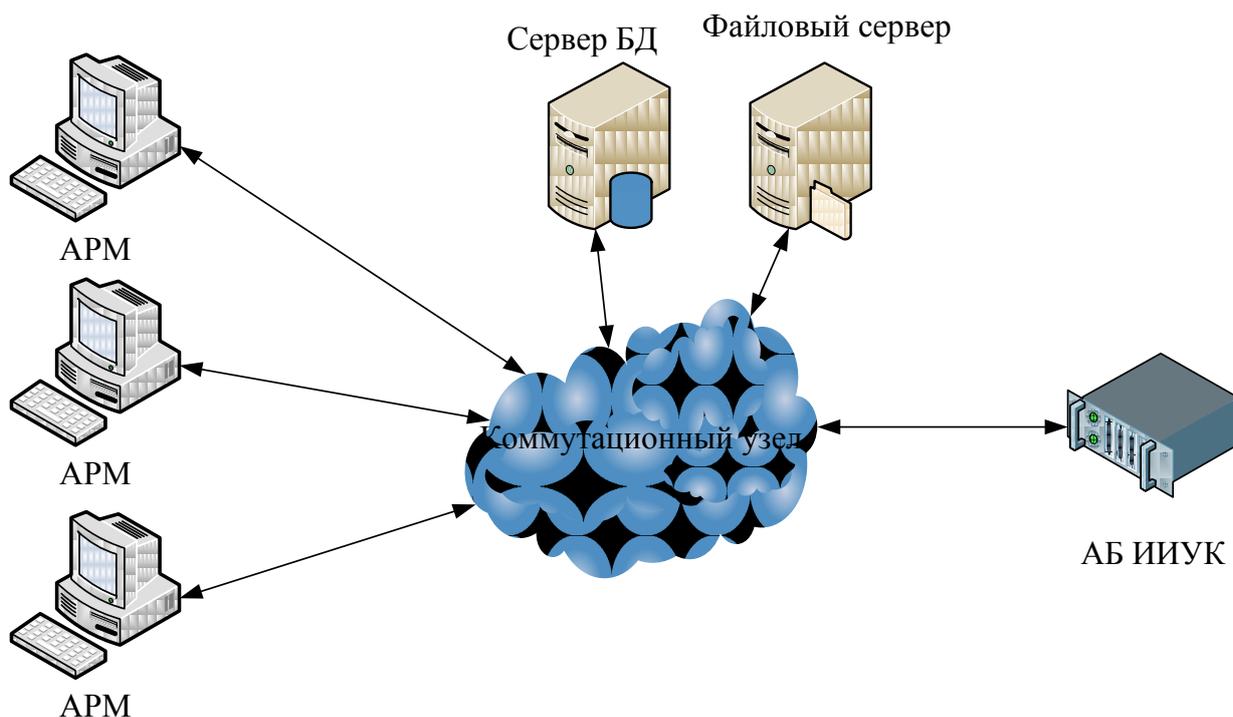


Рис.2.3. Многопользовательский режим связи АРМ с АБ ИИУК.

В качестве коммутационного узла могут выступать: коммутатор, маршрутизатор, сетевой сервер и др. Выбор конкретного решения зависит от выбранного канала связи и целевой топологии «сети».

Файловый сервер отвечает за хранение файлов. Сервер БД хранит всю информацию о средствах измерения и управления, а также другую информацию, необходимую для конфигурирования и проведения испытания.

Наиболее быстрой линией передачи данных является проводная. Для проводной передачи данных наиболее распространенными и перспективными являются такие цифровые интерфейсы, как: Ethernet (IEEE 802.3) и FireWire (IEEE 1394).

### Ethernet

Физической средой для организации канала передачи данных в проводной сети LAN служат кабели, чаще всего витая пара или оптоволоконный кабель. Витая пара состоит из восьми проводов, образующих четыре витых пары медных проводов, при этом используются разъемы RJ-45 и гнезда. Максимальная длина кабеля при использовании витой пары составляет 100 м, в то же время при

использовании оптоволоконного кабеля его длина может составлять от 10 км до 70 км в зависимости от типа оптоволокна. В зависимости от типа витой пары или оптоволокна скорость передачи данных может варьироваться в диапазоне от 100 Мбит/с до 10 000 Мбит/с.

На практике рекомендуется строить сеть большей пропускной способности, чем требуется в данный момент. Для обеспечения возможности дальнейшего расширения сети желательно проектировать ее таким образом, чтобы в начальный момент времени использовать не более 30 % пропускной способности. В настоящее время все больше приложений работают с использованием сети, требуется все более и более высокая производительность сети. Сетевые коммутаторы (упоминаемые далее) после нескольких лет работы довольно легко усовершенствовать, кабели же обычно заменить значительно сложнее.

### Fast Ethernet

Fast Ethernet — это сеть Ethernet, предназначенная для передачи данных со скоростью 100 Мбит/с. Сеть может быть построена на основе витой пары или оптоволоконного кабеля. (До сих пор существуют и используются устаревшие сети Ethernet со скоростью передачи данных 10 Мбит/с, однако такие сети не обеспечивают достаточной ширины полосы пропускания для некоторых приложений сетевого видео.) Большинство подключенных к сети устройств, например ноутбуки или сетевые камеры, оснащены интерфейсом Ethernet 100BASE-TX/10BASE-T, часто называемым интерфейсом 10/100, который поддерживает как скорость передачи данных 10 Мбит/с, так и Fast Ethernet. Тип витой пары, поддерживающей протокол Fast Ethernet, называется Cat-5.

Таблица 2.1. Технические характеристики Fast Ethernet:

Стандарт	Год выхода стандарта	Тип	Скорость передачи (Mbps)	Максимальная длина сегмента в метрах	Тип кабеля
IEEE 802.3u	1995	100Base-FX	100	Одномод. – 2 км Многомод.– 400м	Опто- волокну
		100Base-T		100 м	UTP/STP cat 5
		100Base-T4		100 м	UTP/STP cat >= 3
		100Base-TX		100 м	UTP/STP cat 5
IEEE 802.12	1995	100Base-V G		100 м	UTP cat 3,5
IEEE 802.3y	1998	100Base-T2		100 м	UTP cat 3,5
TIA/EIA- 785	2001	100Base-SX		300 м	Опто- волокну
IEEE 802.3ah	2004	100Base- LX10		10 км	
IEEE 802.3ah	2004	100Base- BX10	10 км		

### Gigabit Ethernet

Технология Gigabit Ethernet, которую также можно реализовывать на основе витой пары или оптоволоконного кабеля, предназначена для передачи данных со скоростью 1000 Мбит/с (1 Гбит/с). Данная технология становится очень популярной. Ожидается, что Gigabit Ethernet вскоре заменит технологию Fast Ethernet и станет фактически стандартом. Кабель Cat-5e поддерживает передачу данных по технологии Gigabit Ethernet, в нем все четыре пары витых проводов используются для достижения больших скоростей передачи данных. Для сетевых видеосистем рекомендуется использовать кабель категории Cat-5e и более поздних. Большинство интерфейсов совместимы с Ethernet 10 и 100 Мбит/с и часто называются интерфейсами 10/100/1000.

Для передачи данных на большие расстояния можно использовать оптоволоконные кабели, например 1000BASE-SX (длиной до 550 м) или 1000BASE-LX (длиной до 550 метров с многомодовым стекловолокном и длиной до 5 000 метров с одномодовым стекловолокном).

Таблица 2.2. Технические характеристики Gigabit Ethernet (GbE)

Стандарт	Год выхода стандарта	Тип	Скорость передачи (Mbps)	Максимальная длина сегмента в метрах	Тип кабеля
IEEE 802.3z	1998	1000Base-CX	1000	25 м	UTP/STP cat 5,5
		1000Base-LX		Одномод. — 5 км Многомод. — 550 м	Опто-волокну
		1000Base-SX		550 м	
IEEE 802.3ab	1999	1000Base-T		100 м	UTP/STP cat 5,5e,6,7
TIA 854	2001	1000BASE-TX		100 м	UTP/STP cat 6,7
IEEE 802.3ah	2004	1000BASE-LX10		10 км	Опто-волокну
		1000BASE-BX10		10 км	
IEEE 802.3ap	2007	1000BASE-KX		1 м	для объединительной платы
non-standard	-	1000BASE-EX		40 км	Опто-волокну
non-standard	-	1000BASE-ZX		70 км	

### 10 Gigabit Ethernet

Технология 10 Gigabit Ethernet — это технология последнего поколения, позволяющая передавать данные на скорости 10 Гбит/с (10 000 Мбит/с), возможно использование оптоволоконного кабеля или витой пары. Для связи на расстоянии до 10 000 м можно использовать стандарты 10GBASE-LX4,

10GBASE-ER и 10GBASE-SR на основе оптоволоконного кабеля. При использовании витой пары необходим кабель очень высокого качества (Cat-6a или Cat-7). Стандарт Ethernet со скоростью передачи 10 Гбит/с в основном используется для магистральных соединений при работе с высокопроизводительными приложениями, требующими больших скоростей передачи данных. [26, 50]

Таблица 2.3. Технические характеристики Gigabit Ethernet (GbE)

Стандарт	Год выхода стандарта	Тип	Скорость передачи (Gbps)	Максимальная длина сегмента в метрах	Тип кабеля
IEEE 802.3ae	2003	10GBASE-SR	10	26-300 м	Опто-волокну
		10GBASE-LX4		Одномод. — 10 км Многомод. — 300 м	
		10GBASE-LR		10 км	
		10GBASE-ER		40 км	
		10GBASE-SW		26 м — 40 км	
		10GBASE-LW			
		10GBASE-EW			
IEEE 802.3ak	2004	10GBASE-CX4	10	15м	медный кабель CX4

(продолжение)

IEEE 802.3an	2006	10GBASE-T	10	100 м	UTP/STP cat 6, 6a, 7
IEEE 802.3aq	2006	10GBASE-LRM	10	220 м	Опто-волокно
IEEE 802.3ap	2007	10GBASE-KX4	10	1 м	для объединительной платы
IEEE 802.3ap	2007	10GBASE-KR	10	1 м	
IEEE 802.3av	2009	10GBASE-PR	10	20 км	Опто-волокно

#### Fire Wire (IEEE 1394)

IEEE 1394 - это стандарт на высокоскоростную последовательную шину, который также является частью стандарта Serial SCSI.

Стандарт поддерживает пропускную способность шины на уровнях 100, 200, 400 800 и 1600 Mbits/s. В зависимости от возможностей подключенных устройств одна пара устройств может обмениваться сигналами на одной скорости (например, 100 Mbits/s), в то время как другая на той же шине - на другой (например, 400 Mbits/s). Такие высокие показатели пропускной способности последовательной шины практически исключают необходимость использования параллельных шин.

Основные достоинства технологии:

- Тонкий кабель заменяет груду громоздких проводов;
- Простота в использовании - отсутствие терминаторов, идентификаторов устройств или предварительной установки, возможность переконфигурировать шину без выключения компьютера;
- Гибкая топология, основанная на равноправии устройств;
- Высокая скорость обмена, что дает возможность подключать к шине самое высокоскоростное оборудование;

- Низкая нагрузка на процессор;
- Возможность использовать самые разнообразные устройства, как внешние, так и внутренние, широкие возможности для интеграции РС с бытовой аудио- видеоэлектроникой.

Сеть может включать до 63 узлов, каждый из которых имеет свой 6-разрядный физический идентификационный номер. К каждому узлу можно подключить до 16 устройств. Если этого недостаточно, несколько сетей могут быть соединены между собой мостами (всего до 1023 шинных перемычек). При этом каждая шина идентифицируется отдельным 10-разрядным номером. Таким образом, 16-разрядный адрес позволяет иметь до 64449 узлов в системе, что дает 1031184 устройств.

Устройства могут подключаться к любому доступному порту (на каждом устройстве обычно 1 - 3 порта). Шина допускает "горячее" подключение - соединение или разъединение при включенном питании. Нет также необходимости в каких-либо адресных переключателях, поскольку отсутствуют электронные адреса. Каждый раз, когда узел добавляется или изымается из сети, топология шины автоматически переконфигурируется в соответствии с шинным протоколом. Однако есть несколько ограничений. Между любыми двумя узлами может существовать не больше 16 сетевых сегментов, а в результате соединения устройств не должны образовываться петли. К тому же для поддержки качества сигналов длина стандартного кабеля, соединяющего два узла, не должна превышать 4,5м. Однако если приобрести FireWire репитер (рис. 2.5), который усиливает сигнал сеть можно проложить на расстояние до 72-100 метров.

Интерфейс позволяет осуществлять два типа передачи данных: синхронный и асинхронный. При асинхронном методе получатель подтверждает получение данных, а синхронная передача гарантирует доставку данных в необходимом объеме.

Стандарт 1394 определяет не только протокол передачи данных, но и общую структуру шины. Древообразная структура всегда имеет корневое

устройство, от которого происходит ветвление к логическим узлам, находящимся в других физических устройствах.

Корневое устройство отвечает за определенные функции управления. Так, если это PC, он может содержать мост между шинами 1394 и PCI и выполнять некоторые дополнительные функции по управлению шиной. Корневое устройство определяется во время инициализации и, будучи однажды выбранным, остается таковым на все время подключения к шине.

Порт FireWire изначально основывался на архитектуре локальных сетей, и возможность соединения компьютеров заложена в него изначально. Главное достоинство такого подключения является его очень высокая скорость 400 Mb/s.

Если один из компьютеров подключён к ЛВС, то используя его в качестве роутера сеть FireWire можно объединить с обычной локальной сетью. [28, 45]

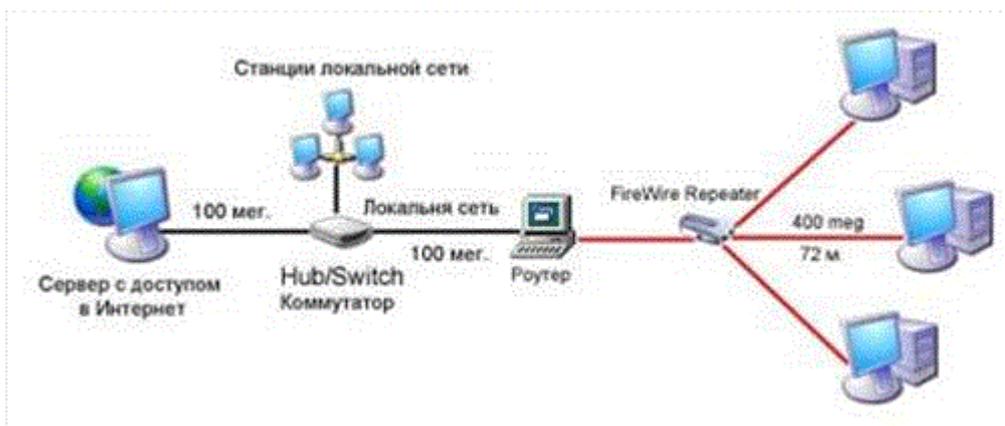


Рис.2.4. Сеть на основе FireWire.

## Структуры сети ИИУК для ИАВ

Одним из самых простых решений будет объединение всех устройств в одну локальную сеть через коммутатор или маршрутизатор.

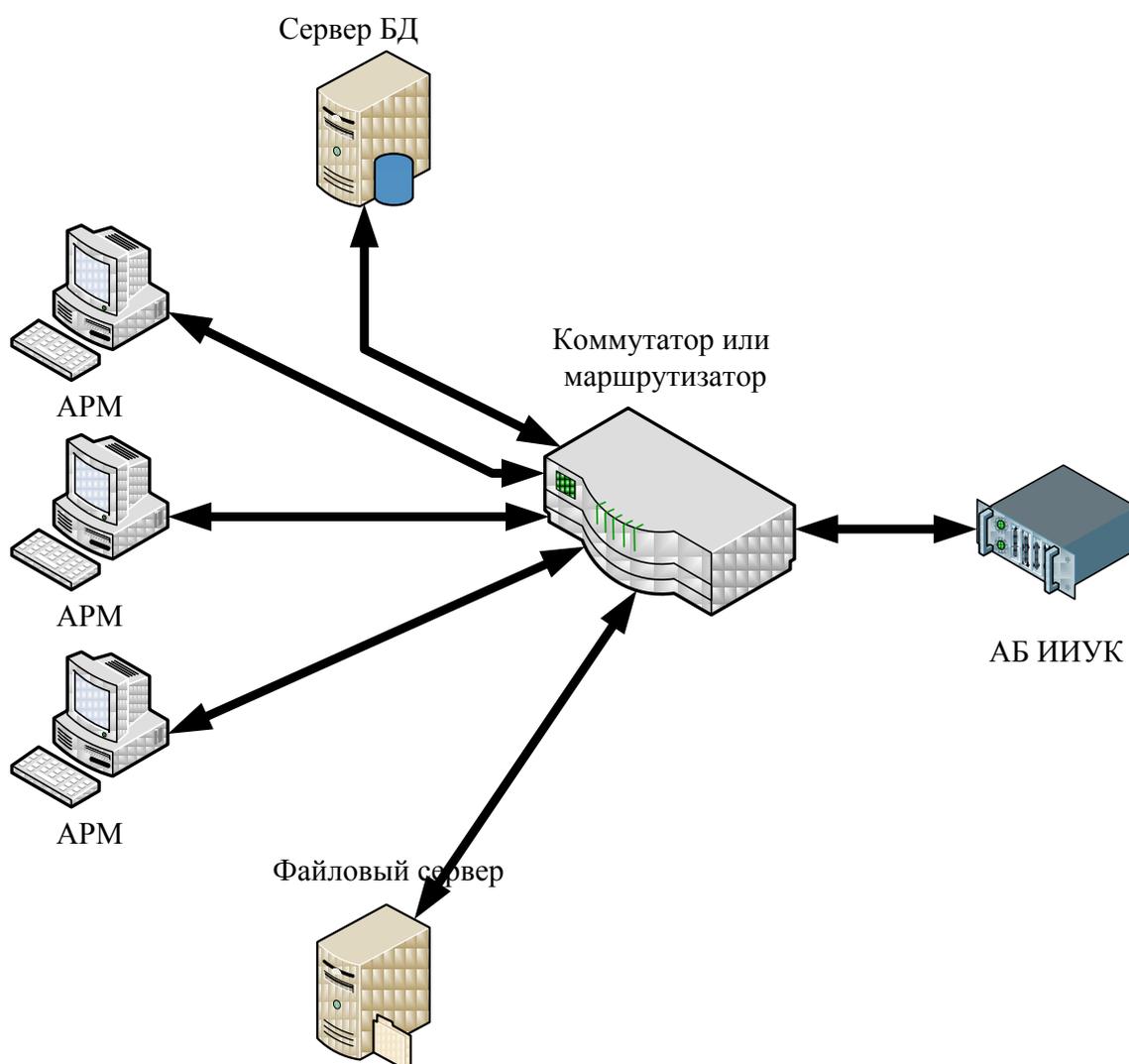


Рис.2.5. Структура сети с объединением устройств в одну локальную сеть через коммутатор.

Достоинством данного решения, зачастую является дешевизна и простота настройки. Однако, во время испытания, при передаче больших объемов телеметрии на определенные АРМы, коммутатор должен иметь возможность обрабатывать запросы от других АРМов к БД и файловому серверу, а также передавать данные обратно. В случае смешения потоков данных телеметрии и данных от/к БД или файловому серверу возможна потеря пакетов, что негативно сказывается на процессе проведения испытания. Рекомендуется данную структуру сети использовать при малой величине сети.

Для того, чтобы исключить данный недостаток, следует поставить высокопроизводительный центральный сервер, обеспечивающий маршрутизацию сетей.

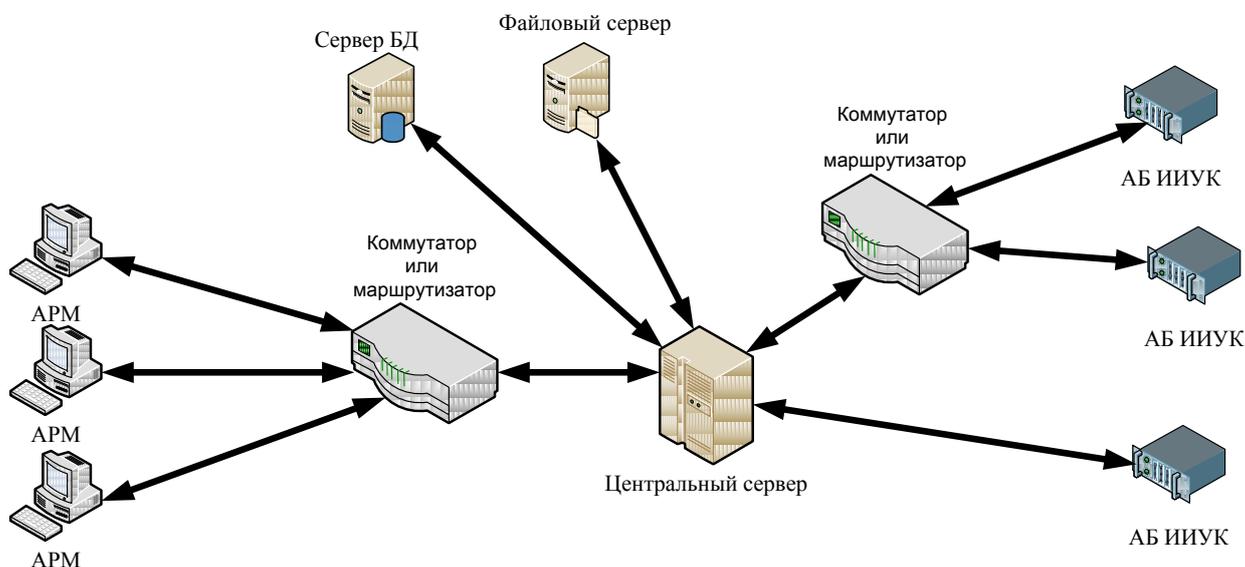


Рис.2.6. Структура сети для стендовых испытаний с центральным сервером.

Особенностью данного решения является возможность буферизации данных на центральном сервере, тем самым уменьшая возможность потери данных, при больших потоках телеметрии. Однако, данное решение полностью не убирает возможность перекреста потоков данных.

Действенным решением является разделение единой сети на две подсети. Одна используется для передачи данных в процессе испытаний, другая используется для организации обмена данными между БД и файловым сервером.

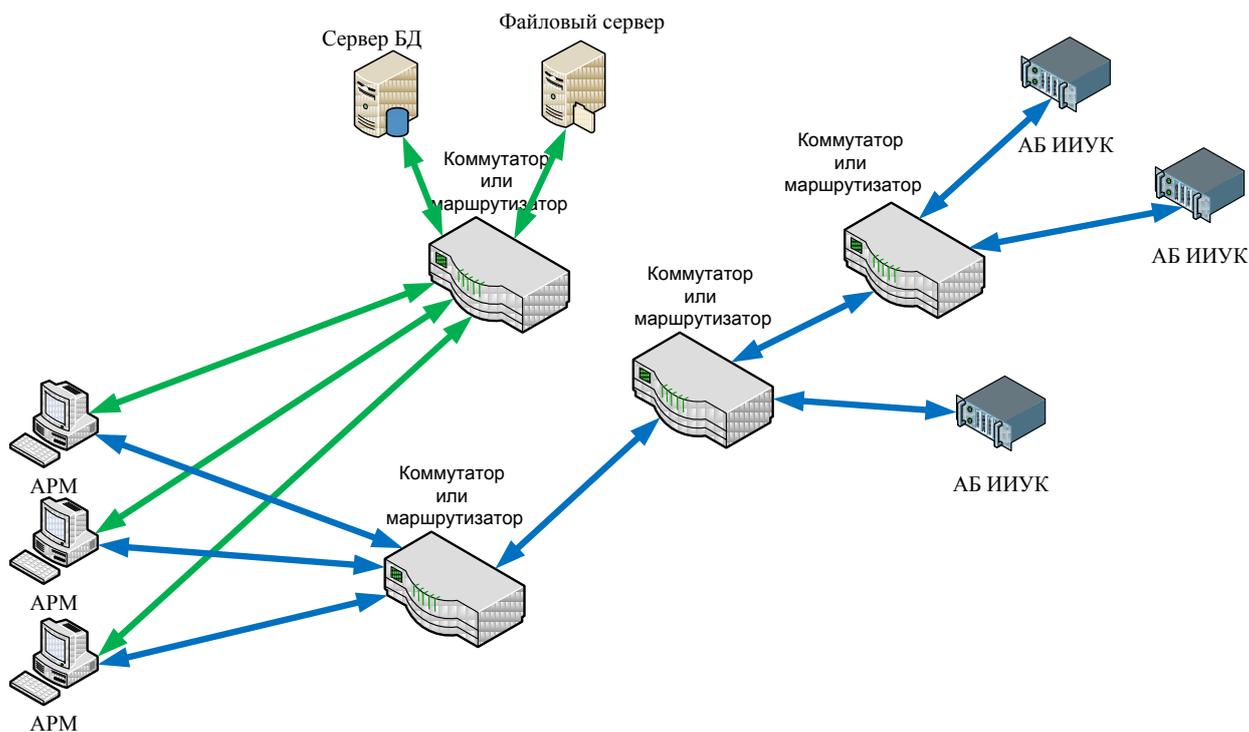


Рис.2.7. Структура сети для стендовых испытаний с разделенными подсетями БД и с узким местом в подсети стендовых данных ИИУК. Использование одного канала для передачи телеметрии от всех ИИУК. Цветами обозначены разные подсети.

Для возможности реализации данной архитектуры, необходимо обеспечить на каждом АРМ не менее двух сетевых карт, для разнесения подсетей. Данное решение лишено недостатков связанных с одновременной передачей данных во время испытаний и работой с файловым сервером и сервером БД. Однако, в результате возможности запуска одновременно нескольких стендов с большим количеством телеметрии, также возможен вариант переполнения сетевого потока данных в месте соединения всех потоков телеметрии. Такое решение может быть вполне приемлемо для одного-двух АБ ИИУК, работающих в одной подсети.

Самым эффективным способом ликвидации данного события является организация в одну подсеть ИИУК с АРМами, работающими с ним напрямую, оставляя второй подсети обмен данными с БД и файловым сервером.

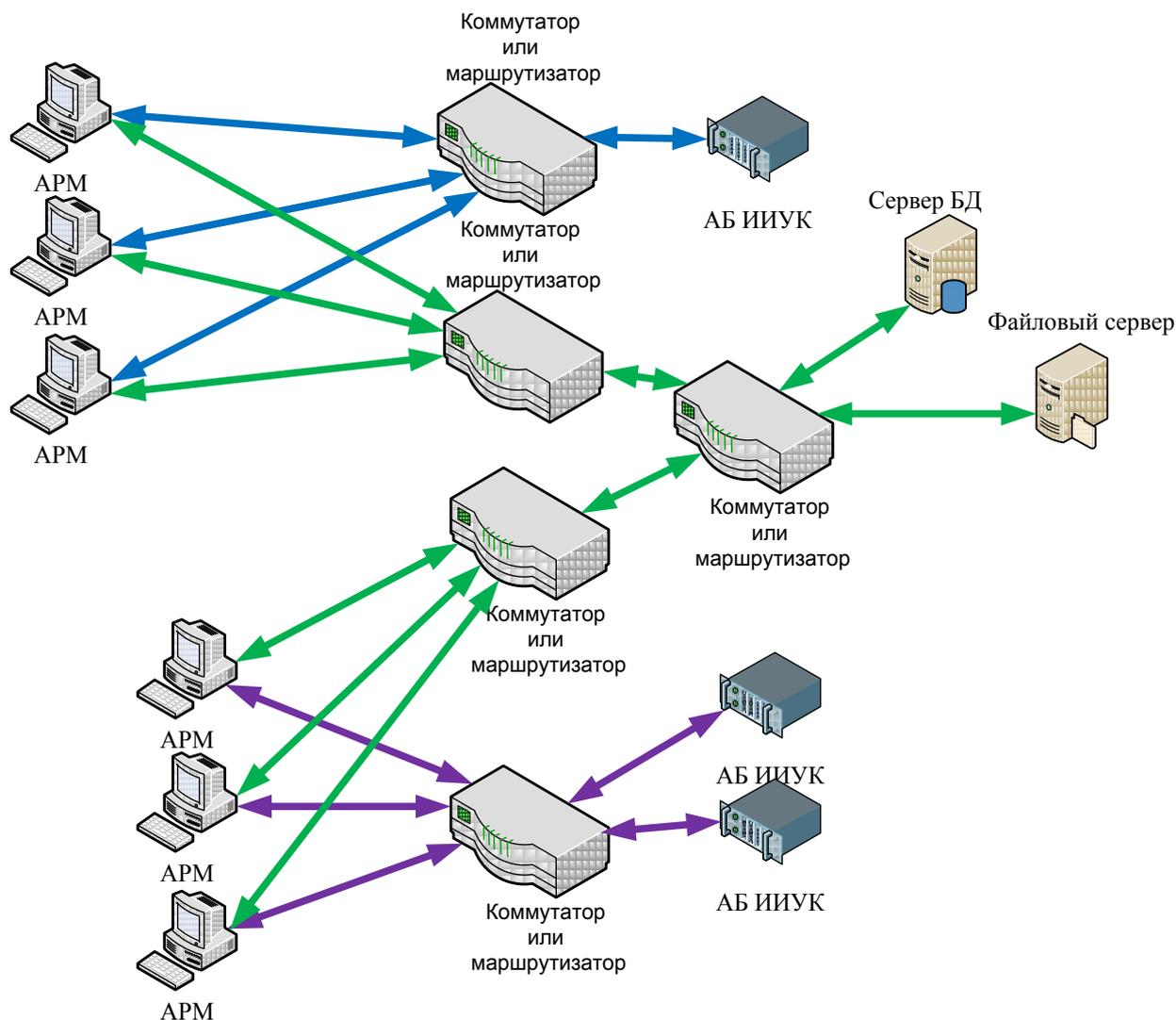


Рис.2.8. Структура сети для стендовых испытаний с разделенными подсетями БД и подсетями разделенных ИИУК. Цветами обозначены разные подсети.

Не рекомендуется объединять в одну подсеть АРМы и несколько ИИУК, если суммарный поток данных выше, чем может обработать коммутатор или маршрутизатор данной подсети. Как и в предыдущем случае, для возможности реализации данной сетевой архитектуры, необходимо обеспечить на каждом АРМ не менее двух сетевых карт.

В случае, если необходимо со всех ИИУК, в процессе испытания, сохранять всю телеметрию, необходима установка специализированного кластера большой емкости и скорости записи данных с установкой высокоскоростных RAID массивов (данное положение необходимо выполнить, если суммарная скорость

записи телеметрии превышает стандартную скорость возможной записи на носитель).

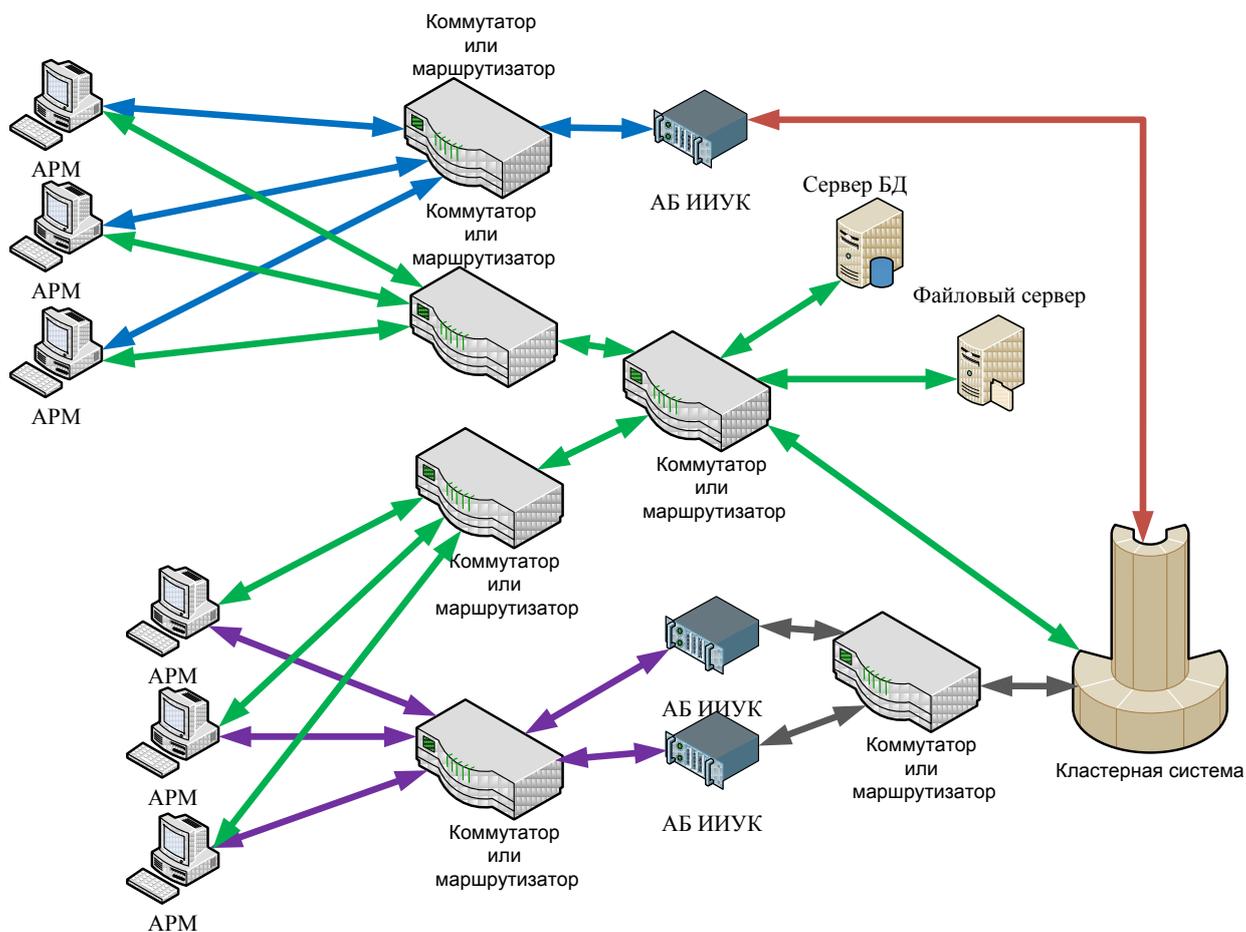


Рис.2.9. Структура сети для стендовых испытаний с разделенными подсетями БД и данных ИИУК и потоковой записью данных испытания. Цветами обозначены разные подсети и потоки данных.

Таблица 2.4. Рекомендуемые ограничения для типовых структур сети ИИУК.

	Суммарный поток данных по всей сети	Ориентиро- вочное число АБ ИИУК	Число АРМ	Стоимость сетевого оборудо- вания
Простая локальная сеть	До 1 Гб/с	Не более 5	Не более 5	Низкая
Сеть с центральным сервером	До 1 Гб/с	Не более 5	Не более 5	средняя
Сеть с разными подсетями – БД и ИИУК с узким местом	До 2 Гб/с	Не более 8	Не более 8	низкая
Сеть с разделенными подсетями БД и разными подсетями ИИУК	До n Гб/с (n – число независимых подсетей)	$\infty$	$\infty$	средняя
Сеть для записи поточной информации ИИУК	До n Гб/с (n – число независимых подсетей)	$\infty$	$\infty$	Очень высокая

Приведенные результаты показывают, что для создания ИИУК малого размера, для решения задач динамической аэроупругости в многопользовательском режиме, достаточно использовать сеть с разделенными подсетями – с выделенной сетью обмена данными между АРМ с сервером БД и файловым сервером, и выделенная сеть для обмена данными АРМ с ИИУК.

### **2.1.3. Структурная схема АБ ИИУК**

Основной задачей ИИУК является расчет силовых воздействий для многоточечного возбуждения на основании данных, поступающих из АРМ и датчиков. Для обеспечения многоточечного возбуждения, ИИУК должно быть многоканальным и расширяемым.

Для расширения (наращивания) системы и увеличения числа каналов измерения и управления, необходимо использовать соответствующие технологии и системные шины, связывающие расширяемые модули между собой.

Среди таких перспективных стандартов и спецификаций сегодня используются: PCIe, compactPCI, PXI и PXIe.

#### PCI Express

##### *Архитектура*

Архитектура PCI Express определяется слоями, как показано на рис.2.10. Для обеспечения совместимости с существующими приложениями и драйверами сохранена модель адресации PCI. Конфигурация PCI Express использует стандартный механизм PCI Plug-and-Play. Программный уровень генерирует запросы на чтение и запись, которые передаются уровнем транзакций устройствам ввода-вывода с использованием пакетно-ориентированного, с разделяемыми транзакциями (split-transaction) протокола. Уровень «Link» добавляет последовательный номер и CRC код (код контроля ошибок), что обеспечивает высоконадежный механизм передачи. Физический уровень состоит из двух каналов, которые реализованы, как передающая пара и принимающая пара. Начальная скорость в 2.5 Giga transfers/second/direction обеспечивает канал

связи с пропускной способностью в 200MB/s, что почти в 2 раза больше, чем у PCI.

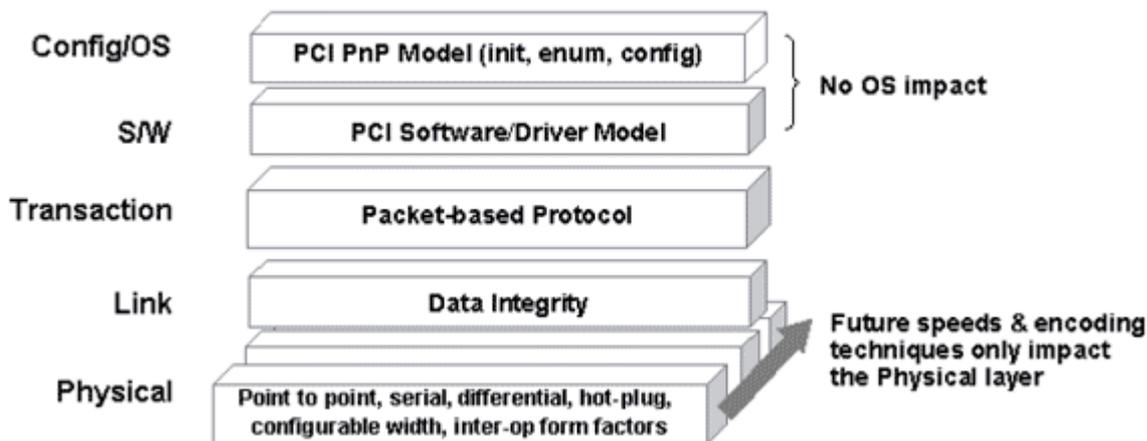


Рис.2.10. Многослойная архитектура PCI Express.

*«Физический» слой*

Основной канал PCI Express состоит из двух низковольтных, дифференциальных пар сигналов: передающая пара и принимающая пара (см. рис. 2.11.). Начальная частота в 2.5 Giga transfers/second/direction может быть увеличена до 10 Giga transfers/second/direction (это теоретический предел частоты для медного проводника).

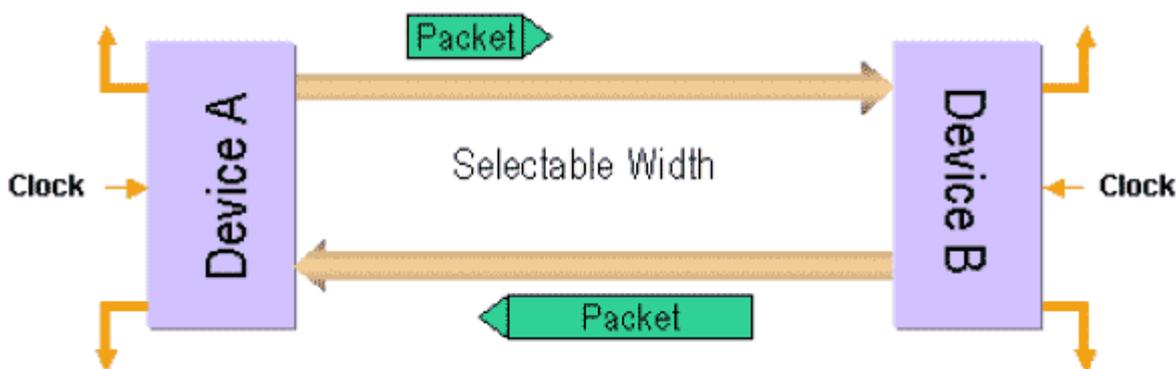


Рис.2.11. Схема дифференциального канала PCI Express

Пропускная способность PCI Express канала может быть линейно увеличена за счет добавления сигнальных пар. «Физический» слой поддерживает x1, x2, x4, x8, x12, x16 и x32 сигнальных пар в одном канале и распределяет байты данных внутри канала.

В процессе инициализации, каждый из PCI Express каналов автоматически устанавливает частоту и ширину канала в соответствии с возможностями агентов, находящихся на концах канала, при этом не требуется никакого программного обеспечения.

### *Слой связи («Link Layer»)*

Основное назначение слоя «Link» заключается в обеспечение правильной передачи пакета данных через канал PCI Express. Этот слой отвечает за целостность данных и добавляет к пакету данных порядковый номер и CRC код (код контроля ошибок), см. рис.2.12.

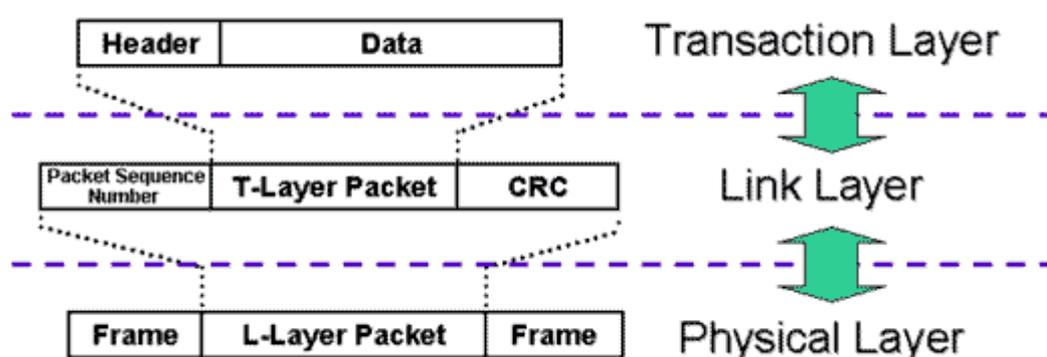


Рис.2.12. Слой связи «Link Layer».

Большинство пакетов инициируются слоем транзакций («Transaction Layer»), описанным в следующем разделе. Протокол передачи передает пакеты только в том случае, когда приемный буфер свободен, это позволяет избежать повторных передач данных и разгружает шину. Повторная передача поврежденных пакетов, так же обеспечивается Link слоем.

### *Слой транзакций («Transaction Layer»)*

Слой транзакций принимает запросы на чтение/запись от программного слоя и создает пакеты для передачи слою связи. Все запросы реализуются в виде отдельных транзакций. Некоторые из этих пакетов, требуют ответных пакетов, которые принимаются от слоя связи и проверяются на соответствие начальному запросу программного слоя. Каждый пакет имеет уникальный идентификатор,

который позволяет отправить ответный пакет правильному адресату. Формат пакетов поддерживает 32bit и 64bit адресацию.

Слой транзакций поддерживает четыре адресных пространства: три PCI (память, I/O и конфигурация) и пространство сообщений («Message Space»). В стандарте PCI 2.2 был предложен альтернативный способ распределения системных прерываний, называемый Message Signaled Interrupt (MSI). Спецификация PCI Express использует концепцию MSI в качестве основного метода распределения прерываний.

### *Программный слой («Software Layer»)*

Программная совместимость имеет важнейшее значение для третьего поколения шины ввода/вывода. Имеются два аспекта программной совместимости: инициализация и совместимость времени выполнения. PCI имеет отлаженную модель инициализации, с помощью которой, операционная система может обнаружить все имеющиеся дополнительные устройства и оптимальным образом распределить им системные ресурсы (память, прерывания и т.п.). Эта модель сохранена и в PCI Express, как следствие: изменений в операционной системе, для загрузки на PCI Express системах, не требуется. Кроме того, в PCI Express обеспечена поддержка старой (PCI) модели времени выполнения, таким образом, в прикладном ПО изменения так же не требуются. Новое ПО может использовать новые возможности PCI Express. [2]

## CompactPCI

Стандарт является прямым развитием шины PCI для применения в промышленных и особо ответственных приложениях.

Основные особенности, определяемые стандартом CompactPCI:

- Унифицированные размеры Eurocard в соответствии с IEEE 1101.1 (рис.2.13);
- Компактные соединители с шагом 2 мм (рис.2.14);
- Вертикальное расположение плат для наилучшего охлаждения;
- Надежная фиксация модулей в системном шасси;
- Высокая устойчивость к ударным и вибрационным воздействиям; металлическая передняя панель съемных модулей;
- Соединители для пользователя, расположенные как на передней, так и на задней стороне модуля;
- Применение стандартных шасси от разных изготовителей;
- Каскадное исполнение выводов питания для реализации режима «горячей» замены;
- Поддержка восьми слотов в базовой конфигурации (с расширением при использовании мостов).

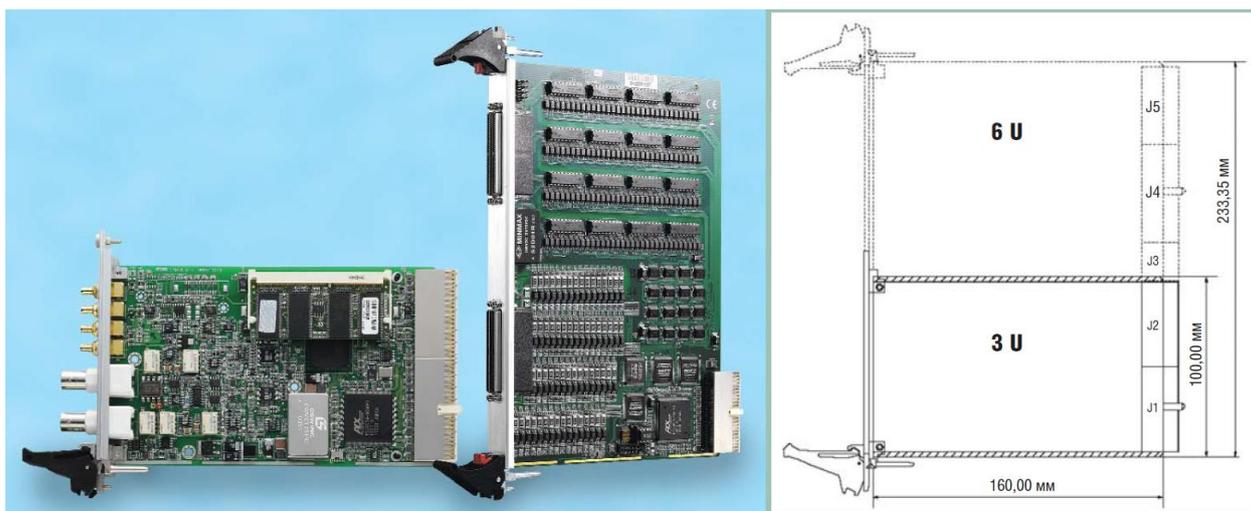


Рис.2.13. Внешний вид модулей CompactPCI и типовые размеры.

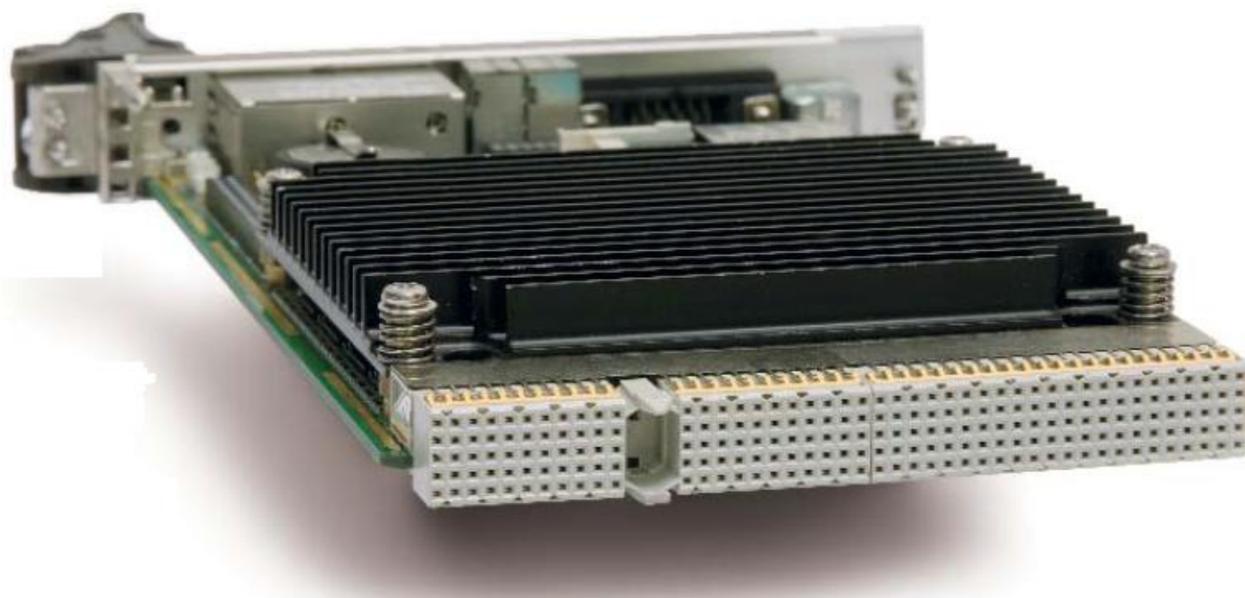


Рис.2.14. Разъем CompactPCI.

Относительно большое количество контактов служит для экранирования заземления, что позволяет исключить помех и обеспечить надежную передачу сигналов.

Разная длина контактов в системных разъемах CompactPCI обусловлена аппаратной реализацией режима Hot Swap («горячая» замена). «Длинные» контакты применяются для подключения сигналов питания, основная группа сигналов системной шины подключается контактами «нормальной» длины, а «укороченные» контакты служат для передачи специальных сигналов Hot Swap (при отсоединении модуля они первыми размыкаются и предупреждают систему о предстоящем его удалении.)

В отличие от шины PCI, где количество слотов может быть не более четырех, шина CompactPCI поддерживает до восьми слотов. Дальнейшее расширение количества слотов вплоть до 32 реализуется с помощью мостов PCI-PCI.

Каждый сегмент системы имеет один системный и семь периферийных слотов. Системный модуль занимает левый (системный) слот и обеспечивает арбитраж шины, формирование тактовых сигналов, прием и выдачу сигналов управления.

### Основные характеристики CompactPCI:

- Разрядность – 32/64 бит;
- Шины Адрес/Данные – мультиплексируемые;
- Тип шины – синхронная шина 33МГц или 66МГц. Частота шины зависит от того, способны ли все устройства, включая системное (мастер) работать на данной частоте. Если хотя бы одно из устройств не работает на частоте 66МГц – мастер выбирает частоту 33МГц;
- Пропускная способность в 32-разрядном варианте – 132МБ/с, в 64-разрядном варианте при частоте 66МГц – 528МБ/с. [12, 56]

### PXI

В основе архитектуры PXI лежит шина PCI, стандарт CompactPCI, а также дополнительные программно-аппаратные возможности, позволяющие создать на их базе практически любую автоматизированную вычислительную или контрольно-измерительную систему.

Пропускная способность системы – до 132 МБ/с для 32-разрядной шины PCI на частоте 33 МГц; для 64-разрядного варианта значение этого параметра составляет до 528МБ/с.

Спецификация PXI предусматривает дополнительные возможности синхронизации.

Синхронизация и тактирование в системах PXI реализуются применением:

- Наносекундной синхронизации модулей ввода/вывода на основе опорного тактового сигнала 10МГц, подаваемого на все модули;
- Контроллера и шины Star Trigger, управляющих передачей сигналов тактирования и синхронизации, реализующих подключение модулей с использованием топологии «звезда», учитывающих длину пути сигналов переключения для уменьшения задержки и для синхронизации моментов запуска различных приборов;

- Шины PXI Trigger, состоящей из восьми линий, сигналы которых подаются на все слоты в сегменте, с тем чтобы они могли взаимодействовать и аппаратным образом управлять друг другом;
- Локальной шины, служащей для передачи высокочастотных цифровых и аналоговых сигналов между соседними модулями.

Архитектура PXI позволяет использовать высокопроизводительные процессоры для задач, требующих сложного анализа или математической обработки данных. [56]

### PXI Express

В отличие от спецификации PXI, PXI Express имеет более высокий уровень производительности за счет того, что передача данных, сигналов запуска и тактирования происходит по дифференциальным цепям, построенным по принципу «точка-точка». Стандарт PXI Express определяет пропускную способность до 6ГБ/с и одиночного слота до 2ГБ/с.

Кроме этого PXI Express имеет более высокую точность и устойчивость к помехам благодаря тому, что в качестве основного синхронизирующего сигнала используется дифференциальный сигнал с частотой 100МГц.

PXI Express основан на Compact PCI Express и PXI, и в нем сохранились все основные присущие им характеристики.

Расширения, касающиеся инструментальных применений, и сохраненные в полном объеме функции PXI дополнены следующими:

1. Дифференциальными сигналами тактовой частоты 100МГц;
2. Сигналами синхронизации с изменяемой частотой;
3. Дифференциальными линиями запуска, организованными по принципу «точка-точка».

На рис.2.15. упрощенно показаны основные шины/сигналы синхронизации и запуска, используемые в PXI Express.

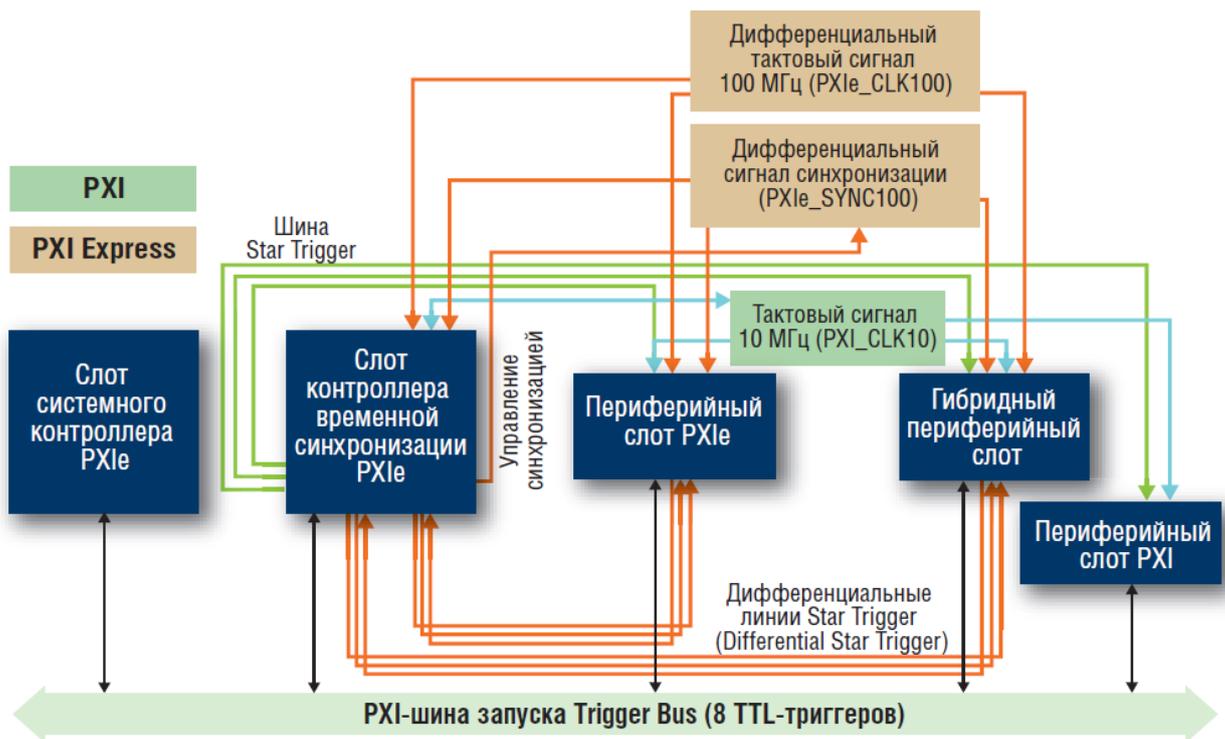


Рис.2.15. Шины синхронизации и запуска в PXI Express.

С точки зрения возможностей синхронизации, по сравнению с PXI здесь добавлены новые шины/сигналы:

1. Высокочастотный системный тактовый сигнал (PXIe\_CLK100) – дифференциальный низковольтный сигнал частотой 100МГц, подаваемый ко всем слотам;
2. Сигнал синхронизации (PXIe\_SYNC-100) – дифференциальный сигнал, направленный от объединительной панели к каждому модулю, синхронизирован с PXIe\_CLK100 и через каждые 10 тактов индицирует фазовые соотношения между основными тактовыми частотами 10 и 100 МГц;
3. Три пары дифференциальных линий (Differential Star Trigger) с топологией звезда, выполняющие функции, аналогичные шине PXI Star Trigger. [11]

Конструктивно, перечисленные системные шины выглядят в виде несущей платы с управляющим драйвером и набором разъемов для установки

измерительно-управляющих и вычислительных модулей, обычно размещаемой внутри промышленного шасси (крейта).

Для управления обменом данными между модулями и внешней сетью, устанавливается специализированная плата с установленным контроллером и предустановленной ОС. Такой контроллер обеспечивает обмен данными между внешней сетью и модулями, установленными в системную шину.

#### **2.1.4. Архитектура расширительных модулей для ИИУК.**

В качестве устройств ввода информации с датчиков, служат АЦП. Данные, получаемые с АЦП поступают в вычислитель, где происходит расчет значений силовых воздействий, и выдача управляющего воздействия. Управляющее воздействие выводится из системы посредством ЦАП, и затем, в аналоговой форме поступает на усилители силовозбудителей.

Для обеспечения работы данного контура необходимо выбрать схему передачи данных от АЦП к ЦАП.

Раздельное исполнение измерительного модуля, модуля управления и вычислителя.

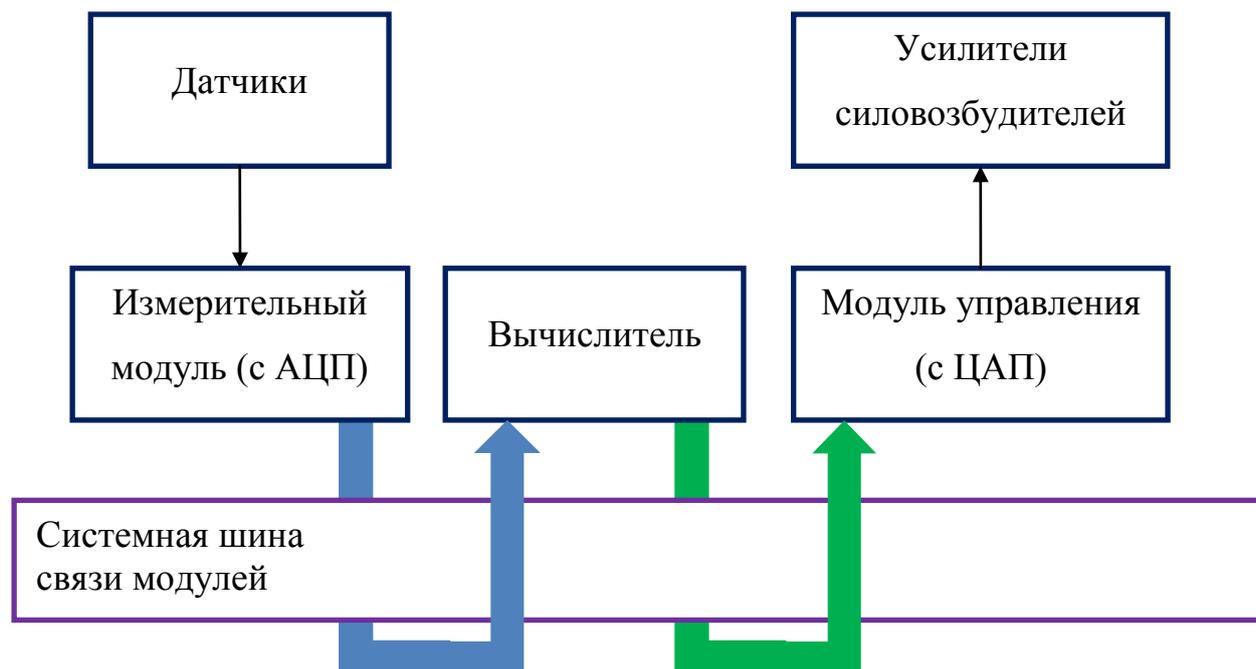


Рис.2.16. Архитектура с раздельным исполнением измерительной, вычислительной и управляющей частей.

Суммарная скорость получения информации на выходе ЦАП вычисляется по следующим формулам:

1. Для одноканальных АЦП и ЦАП:

$$t_{\Sigma} = (t_{ADC} + t_{sys}) * n + t_{pr}(m, n) + (t_{sys} + t_{DAC}) * m \quad (2.1)$$

2. Для многоканальных АЦП и ЦАП:

$$t_{\Sigma} = t_{ADC} + t_{sys} * (n + m) + t_{pr}(m, n) + t_{DAC} \quad (2.2)$$

где  $t_{\Sigma}$  - задержка выходного сигнала с момента поступления данных с датчиков;  $t_{ADC}$  - время преобразования аналогового сигнала в код;  $t_{sys}$  - время, необходимое для передачи данных по системной шине;  $t_{pr}(m, n)$  - время расчета значения силового воздействия (зависит от алгоритма и архитектуры вычислителя);  $t_{DAC}$  - время преобразования цифрового кода в аналоговый сигнал для усилителей силовозбудителей;  $n$  - число измерительных каналов;  $m$  - число каналов управления. Очевидно, что при увеличении количества модулей, и

соответственно измерительных и управляющих модулей, задержка выдачи управляющего воздействия увеличивается.

Аналогичная ситуация, при нахождении АЦП и ЦАП в рамках одного модуля, а расположение вычислителя, на другом модуле.

Архитектура сквозного потока данных «АЦП-Вычислитель-ЦАП»

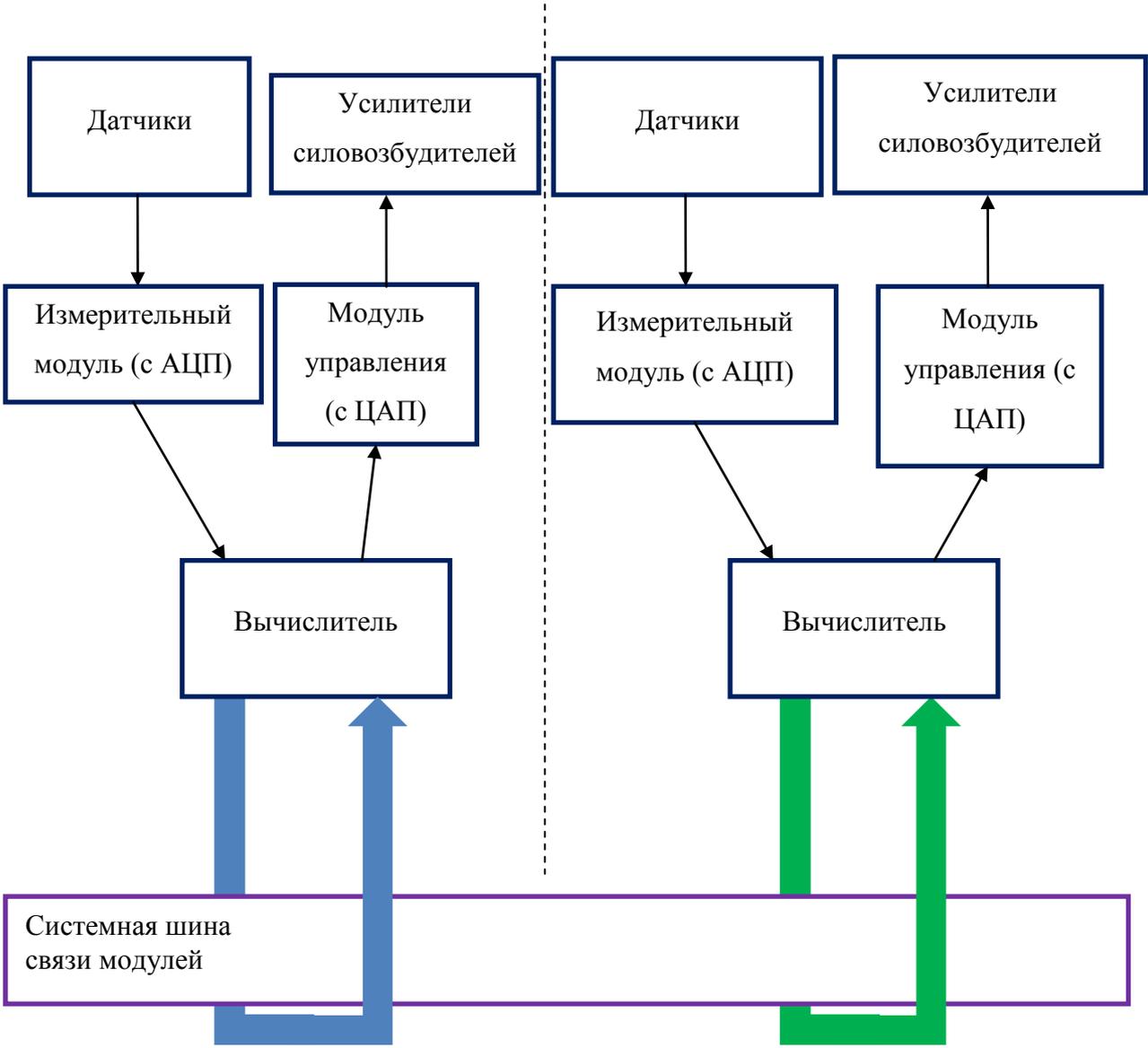


Рис.2.17. Архитектура с единым исполнением измерительной, вычислительной и управляющей частей.

Суммарная скорость получения информации на выходе ЦАП вычисляется по следующей формуле:

1. Для одноканальных АЦП и ЦАП:

$$t_{\Sigma} = t_{ADC} * n + t_{pr}(m, n) + t_{DAC} * m \quad (2.3)$$

2. Для многоканальных АЦП и ЦАП:

$$t_{\Sigma} = t_{ADC} + t_{pr}(m, n) + t_{DAC} \quad (2.4)$$

При увеличении количества однотипных модулей задержка, при расчете не изменяется, а при увеличении измерительных или управляющих каналов каждого модуля, изменяется.

Очевидно, наилучшим вариантом исполнения измерительно-управляющего модуля является архитектура «АЦП-Вычислитель-ЦАП»:



Рис.2.18. Архитектура «АЦП-Вычислитель-ЦАП»

## **2.1.5. Архитектура вычислительного устройства**

### **2.1.5.1. Микроконтроллеры**

Инженеры и ученые, занимающиеся разработкой встраиваемых систем, автоматических систем управления (АСУ), систем связи и устройств цифровой обработки сигналов (ЦОС) предъявляют дополнительные требования к данному классу вычислительных устройств, а именно:

1. Поддержка параллельных вычислений для повышения производительности микроконтроллерной системы за счет естественного параллелизма независимых задач, параллелизма независимых ветвей решаемой задачи или параллелизма обработки

совокупности однотипных данных, поступающих в систему одновременно;

2. Масштабируемость системы, построенной на базе предлагаемого производителем микроконтроллера для возможности увеличения вычислительной нагрузки системы путем добавления аналогичных устройств;
3. Высокая степень надежности электронных узлов для возможности внедрения микроконтроллерных систем в состав ответственных объектов, например, таких как системы управления на транспорте;
4. Расширенный диапазон напряжения питания для устройств, работающих от аккумуляторных батарей.

По данным аналитиков IC Insights, рынок 32-битных МК будет быстро расти благодаря увеличению спроса на более точные встроенные процессорные системы и к 2017г. Будет составлять 55% общемировых продаж, вытесняя тем самым 8 и 16-битные микроконтроллеры.

Несомненным лидером рынка 32-разрядных RISC –процессоров является компания ARM Holdings, которая разрабатывает и лицензирует технологию создания процессорных ядер для сторонних фирм, таких как Atmel, NXP, Texas Instruments и др., которые, в свою очередь, занимаются промышленным производством микросхем. Ярким примером высокопроизводительного вычислителя является ядро Cortex-A15 MCore, представленное компанией ARM в сентябре 2010 года. Архитектура этой модели предполагает возможность создания процессоров с одним, двумя, четырьмя и, возможно, 16 ядрами. Тактовая частота может варьироваться в пределах 1-5ГГц.

Альтернативным вариантом повышения производительности микроконтроллеров, является внедрение в них элементов других архитектур, таких как архитектура цифровых сигнальных процессоров (ЦСП).

Для повышения производительности устройств ЦОС разработчики нередко прибегают к использованию специализированных вычислителей таких, как цифровые сигнальные процессоры (Digital Signal Processor, DSP) и построенные

на их основе цифровые сигнальные контроллеры (Digital Signal Controller, DSC). ЦСП отличается от микроконтроллеров общего назначения урезанной периферией и архитектурой, модифицированной специально для решения задач ЦОС. Например, типовая операция ЦОС «умножение с накоплением» (MAC) на таких контроллерах обычно выполняется за один такт. [1]

#### 2.1.5.2. Многоядерные процессоры и системы

В процессе развития технологий производительность процессоров росла за счет совершенствования элементной базы и увеличения тактовой частоты процессора. Но в 2001 году ресурс на повышение тактовой частоты был исчерпан. Это связано как с ограничениями технологии производства микросхем, так и с тем фактом, что энергопотребление пропорционально четвертой степени тактовой частоты. Т.е. при увеличении частоты процессора, например в 2 раза, его тепловыделение увеличивается в 16 раз.

Для обеспечения дальнейшего роста производительности ведущие разработчики микропроцессоров переходят на производство многоядерных систем с новой архитектурой, позволяющей обеспечить параллельную обработку данных. Существует два вида параллельной обработки данных: конвейерность и параллельность. Идея конвейеризации заключается в разбиении выполняемой операции на последовательные этапы длительностью один такт и реализации каждого из них физическим блоком. Каждый такой блок, выполнив работу, передает результат вычислений следующему блоку и одновременно принимает новую порцию данных.

Идея физического распараллеливания заключается в аппаратном разделении вычислительных устройств таким образом, что каждый из них работает независимо от остальных. К сожалению, на практике, несмотря на простоту идеи физического параллелизма, установка нескольких вычислительных модулей вместо одного не гарантирует прямопропорционального роста вычислительной мощности системы. Возможность распараллеливания вычислений определяется не только архитектурой вычислительной системы, но и природой решаемой

задачи. Реальные вычислительные задачи не могут быть полностью параллельными. Всегда существуют операции, выполняемые последовательно, такие как операции по передаче данных.

В настоящее время не существует универсальных методов программирования многоядерных процессоров. Все методы программирования микропроцессоров зависят от прикладной ориентации: моделирование нелинейных систем, компьютерная графика, обработка сигналов. Каждый язык программирования поддерживает различные формы параллелизма, что увеличивает сложность решения прикладных задач.

Несмотря на вышеописанные недостатки, можно сказать, что появление многоядерных процессоров является качественным скачком на пути создания эффективных высокопроизводительных систем, обладающих существенно более высокими показателями производительность/стоимость по сравнению с существующими системами на базе супер ЭВМ и кластерных систем.

В настоящее время самыми известными многоядерными процессорами, поддерживающими параллельные вычисления, являются:

1. Графические процессоры (Graphic Processing Unit, GPU) компаний NVIDIA и AMD;
2. Центральные процессоры (Central Processing Unit, CPU) типа IBM CELL и Intel Core. [1]

#### 2.1.5.3. ПЛИС и реконфигурируемые цифровые вычислительные системы

Программируемые логические интегральные схемы (ПЛИС, FPGA) являются прямыми конкурентами микропроцессорных систем и ориентированы на использование в тех же областях науки и техники.

Современные ПЛИС содержат сотни тысяч универсальных логических блоков, из которых разработчик может создать любую специализированную вычислительную структуру, которая будет обладать наибольшей производительностью для конкретного алгоритма. Кроме того, ПЛИС позволяет организовать процесс вычисления параллельных операций и задач без

дополнительных временных затрат на пересылку данных по общим шинам, которые присущи многоядерным системам.

Система на кристалле, построенная на базе ПЛИС обладает глубоко оптимизированной внутренней структурой и в ней отсутствует чрезмерная избыточность, характерная для систем, спроектированных на основе универсальных компонентов.

В отличие от высокопроизводительных вычислителей других архитектур, реконфигурируемые системы позволяют адаптировать свою архитектуру к информационной структуре решаемой задачи или ее составных частей, вследствие чего обеспечивается высокая рабочая производительность близкая к пиковой (не ниже 60%), и появляется возможность получить линейный прирост производительности при наращивании количества аппаратных узлов системы.

Благодаря своей архитектуре и отказоустойчивости, в настоящее время ПЛИС используются не только в областях науки и техники, требующих использования высокопроизводительных вычислений, но и в сфере военных и космических технологий. [1]

#### 2.1.5.4. Процессор Multiclet

Мультиклеточное ядро – это группа идентичных процессорных блоков (2 и более), объединенных полносвязной однонаправленной коммутационной средой. Процессорный блок в мультиклеточной архитектуре называется клеткой.

Свойства мультиклеточной архитектуры:

- независимость от кол-ва клеток;
- динамическое распределение вычислительных ресурсов;
- все команды, готовые к выполнению, выполняются одновременно;
- снижение энергопотребления. Работает тогда, когда есть работа;
- масштабируемость, нет ограничений на кол-во клеток.

Кол-во одновременно выполняемых команд зависит от кол-ва клеток. Выбор оптимального числа клеток для каждой задачи — это задача, которая требует анализа. Создать универсальную систему все равно не получится,

поэтому можно ориентироваться на следующие данные: на общих задачах неплохо справляются 4 клетки, а с задачами обработки сигналов — 16 клеток, а для обработки видеоизображения их число может быть порядка сотен.

Особенностью мультিকлеточной архитектуры в том, что все, что в данный момент может быть выполнено – будет выполнено без каких-либо указаний со стороны программиста, основное условие – команда должна получить все данные для исполнения.

Стоит учитывать технологические ограничения: кол-во клеток ограничено, поэтому выполняться одновременно будет столько команд, сколько клеток в ядре.

В обычных процессорах, состоящих из одного или нескольких ядер, элементарной единицей исполнения является команда, выполняемая в определённом порядке. В мультиклеточном процессоре элементарной единицей исполнения является параграф, т.е. линейный участок, состоящий из неограниченного количества команд, после которых происходит переход на другой линейный участок с заданной меткой. Команды из этого участка будут выполняться параллельно там, где это возможно. Ещё одним отличием мультиклеточного процессора является передача результатов работы команд с помощью широковещательной рассылки, тогда как в много и одноядерных системах результаты передаются через память и регистры. Разумеется, в мультиклеточных процессорах имеется память и регистры для передачи данных между линейными участками, но внутри линейного участка основные операции можно провести без использования регистров и памяти. Данный факт даёт мультиклеточной архитектуре простоту реализации и уменьшение обращений к памяти, как следствие снижение энергопотребления. Кроме того выполнение одной и той же программы на нескольких клетках даёт возможности по реконфигурации (объединение клеток в task-группы) клеток, созданию отказоустойчивого процессора и масштабируемости. [27]

С учетом предъявляемых требований к стендовым системам, наиболее целесообразно, в качестве вычислителя использовать архитектуры, обеспечивающие параллельность исполнения операций. Наиболее эффективными, при решении задач построения быстродействующих измерительно-управляющих систем, является архитектуры ПЛИС, в силу аппаратной реализации алгоритма, и обеспечивающей естественный параллелизм исполнения кода.

## **2.2. Оценка и выбор архитектуры ИИУК для решения задач динамической аэроупругости в составе ИАВ.**

На основе анализа возможных структур и сформулированных технических требований к ИИУК для ИАВ, сформулированы следующие принципы построения ИИУК:

- Разделение ИИУК на АРМ и АБ ИИУК;
- В качестве АРМ выступает ПК с установленной ОС и специализированным ПО для проведения испытаний при решении задач динамической аэроупругости;
- Связь между АРМ и АБ ИИУК осуществляется посредством сети с разделенными подсетями – выделенная сеть для обмена с сервером БД и файловым сервером, и выделенная сеть для обмена данными с ИИУК;
- Для дальнейшего расширения и масштабирования системы требуется исполнение АБ ИИУК в виде промышленного шасси с установкой измерительно-управляющих модулей, работающих в режиме жесткого реального времени (~10-15 мкс на дискрету);
- Для обеспечения межмодульной синхронизации, необходимо использовать современные системные шины PXI или PXI Express;
- Для обеспечения взаимодействия АРМ и АБ ИИУК по сети, необходима установка процессорной платы в шасси с соответствующими интерфейсами;

- Для обеспечения максимального быстродействия и надежности, архитектура измерительно-управляющего модуля должна быть вида «АЦП – ПЛИС – ЦАП».
- В рамках одного измерительно-вычислительно и управляющего модуля необходимо использовать наборы одноканальных АЦП и ЦАП;

На рис. 2.19 изображена архитектура для одного АБ ИИУК и нескольких АРМ.

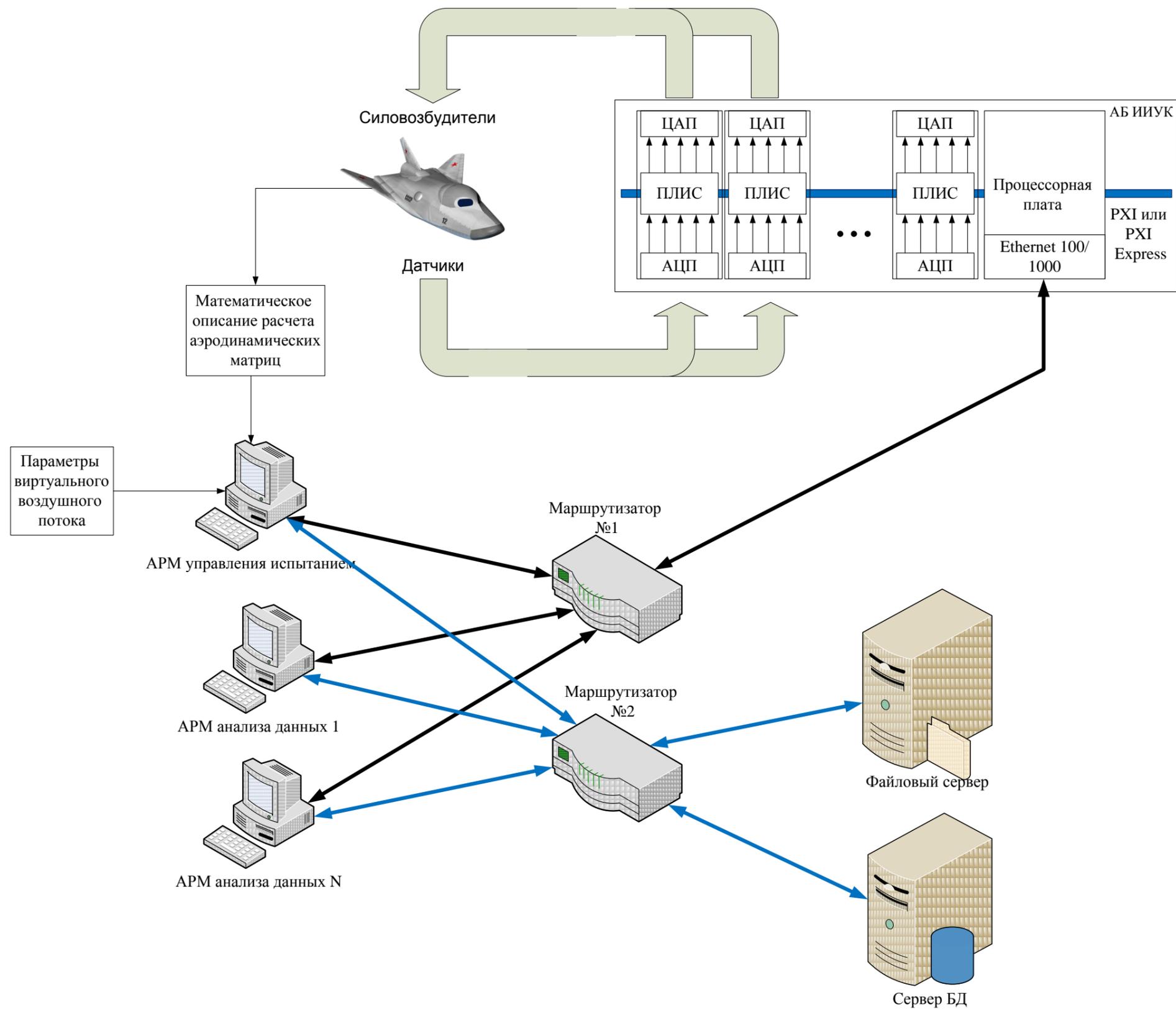


Рис.2.19. Модифицированная архитектура ИИУК.

### **Выводы по второй главе:**

1. В результате проведенного анализа структур измерительно-информационных и управляющих комплексов (ИИУК) сформулированы основные принципы построения модифицированных ИИУК;
2. Проведено функциональное разделение ИИУК;
3. Обоснована сетевая структура связи автономного блока (АБ) ИИУК с автоматизированным рабочим местом (АРМ);
4. Обоснован выбор стандарта PXI и PXI Express для расширяемого ИИУК с учетом требований к быстродействию и синхронизации;
5. Обоснована архитектура «АЦП-ПЛИС-ЦАП» расширительного модуля АБ ИИУК;
6. Обоснован выбор ПЛИС в качестве вычислителя для расширительного модуля АБ ИИУК, на основании сформулированных требований к ИИУК.

**Глава 3. Исследование и оценка цифровой реализации алгоритмов ИИУК на базе ПЛИС, на основе анализа аэродинамических уравнений. Разработка методики создания программного обеспечения для ИИУК с использованием ПЛИС. Оценка эффективности методики.**

### **3.1. Исследование и оценка цифровой реализации алгоритмов ИИУК на базе ПЛИС для ИАВ и оценка погрешностей преобразования.**

Исходная форма уравнения (1.3) представляет собой описание в аналоговом виде. Для реализации уравнение в цифровых вычислительных системах, необходимо его привести к цифровой форме посредством дискретизации.

Для дискретной формы, по правилам линейной алгебры, элемент вектора выходного сигнала рассчитывается как:

$$u_i(t_c) = \sum_j^n g_{ij} \iint a_j(t_c) dt^2 + \sum_j^n h_{ij} \int a_j(t_c) dt + \sum_j^n m_{ij} a_j(t_c), \quad (3.1)$$

где  $t_c$  - текущий момент времени,  $g_{ij}$  - элемент матрицы жесткости  $G$ ,  $h_{ij}$  - элемент матрицы демпфирования  $H$ ,  $m_{ij}$  - элемент матрицы инерции,  $a_j$  - ускорения точек исследуемого объекта в единицах измеряемой физической величины,  $n$  - число точек измерения,  $i$  - номер канала измерения и возбуждения,  $j$  - номер столбца матриц  $G$  и  $H$ .

Для ИИУК, предназначенного для решения задач динамической аэроупругости, в качестве датчиков используются акселерометры.

Значение ускорения в точке рассчитывается:

$$a_i(t_c) = \frac{f(e_i(t_c) - e_i^0) + f(e_i(t_{c-1}) - e_i^0)}{2}, \quad (3.2)$$

где  $i$  - номер датчика,  $e_i(t_c)$  - мгновенное значение ускорения,  $a_i$  - текущее значение ускорения в точке,  $e_i^0$  - цифровое значение уставки (смещение нуля),  $f$  - функция полосового фильтра,  $t_{c-1}$  - предыдущий момент выборки.

Параметры фильтра выбираются на основании технических требований при реализации ИИУК.

Значение скорости в точке рассчитывается:

$$v_i(t_c) = \frac{f(\dot{s}_i(t_c))}{\Delta t}, \quad (3.3)$$

где  $\Delta t = t_c - t_{c-1}$  - период выборки (постоянная интегрирования),  $\dot{s}_i(t_c)$  - промежуточная сумма, пропорциональная ускорению точек конструкции в текущий момент времени,  $v_i(t_c)$  - скорость точки конструкции в текущий момент времени.

$$\dot{s}_i(t_c) = \dot{s}_i(t_{c-1}) + a_i(t_c) \quad (3.4)$$

Значение перемещения в точке рассматривается:

$$y_i(t_c) = \frac{f(s_i(t_c))}{\Delta t}, \quad (3.5)$$

где  $y_i$  - значение перемещения в точке,  $s_i$  - промежуточная сумма, пропорциональная скорости точек конструкции в текущий момент времени.

$$s_i(t_c) = s_i(t_{c-1}) + f(\dot{s}_i(t_{c-1})) + \frac{a_i(t_c)}{2}, \quad (3.6)$$

Учитывая, что данные в цифровую систему поступают через АЦП, а выводятся из ЦАП, необходимо предусмотреть соответствующие операции масштабирования.

Численное представление сигнала при прохождении через АЦП:

$$e_i^*(t_c) = \left[ \frac{2^{N_{ADCi}} \cdot U_i(t_c)}{A_i} \right], \quad (3.7)$$

где  $e_i^*$  - численное представление исходного сигнала датчика после оцифровки в текущий момент времени,  $A_i$  - размах диапазона измерения,  $U_i(t_c)$  - текущее значение измеренного сигнала,  $N_{ADCi}$  - разрядность канала АЦП, квадратные скобки – округление до ближайшего целого.

Численное значение ускорения в точке:

$$a_i^*(t_c) = \left[ \frac{f(e_i^*(t_c) - e_i^{*0}) + f^*(e_i^*(t_{c-1}) - e_i^{*0})}{2} \right], \quad (3.8)$$

где  $a_i^*$  - численное значение ускорения в точке,  $e_i^{*0}$  - цифровое значение уставки (смещение нуля),  $f$  - функция фильтрации, квадратные скобки – округление до ближайшего целого.

Численное значение скорости в точке:

$$v_i^*(t_c) = \left[ \frac{f(\dot{s}_i^*(t_c))}{\Delta t} \right], \quad (3.9)$$

где  $v_i^*$  - цифровое значение скорости в точке,  $\dot{s}_i^*(t_c)$  - промежуточная сумма, пропорциональная ускорению точек конструкции в текущий момент времени.

$$\dot{s}_i^*(t_c) = \dot{s}_i^*(t_{c-1}) + a_i^*(t_c) \quad (3.10)$$

Численное значение перемещения в точке:

$$y_i^*(t_c) = \left[ \frac{f(s_i^*(t_c))}{\Delta t} \right], \quad (3.11)$$

где  $y_i^*$  - цифровое значение перемещения в точке,  $s_i^*(t_c)$  - промежуточная сумма, пропорциональная скорости точек конструкции в текущий момент времени.

$$s_i^*(t_c) = s_i^*(t_{c-1}) + f(\dot{s}_i^*(t_{c-1})) + \frac{a_i^*(t_c)}{2}, \quad (3.12)$$

Численное значение выходного сигнала ЦАП:

$$u_i^*(t_c) = \left[ \sum_j^n g_{ij} y_j^*(t_c) + \sum_j^n h_{ij} v_j^*(t_c) + \sum_j^n m_{ij} a_j^*(t_c) \right] \quad (3.13)$$

Восстановленное значение выходного сигнала ЦАП для силового воздействия:

$$R_i(t_c) = u_i^*(t_c) \cdot Q_i, \quad (3.14)$$

где  $R_i$  - восстановленное значение сигнала ЦАП,  $Q_i$  - масштабный коэффициент ЦАП.

$$Q_i = \frac{B_i}{2^{N_{DAC_i}}}, \quad (3.15)$$

где  $B_i$  - размах выходного диапазона изменения сигнала ЦАП,  $N_{DAC_i}$  - число разрядов ЦАП.

### **3.2. Разработка методики создания программного обеспечения для ИИУК с использованием ПЛИС**

Перед тем, как приступить к реализации алгоритма на ПЛИС необходимо проанализировать математические операторы и функции, лежащие в его основе, а также учесть аппаратные ограничения, которые накладывает архитектура ПЛИС.

#### **Ограничения, накладываемые архитектурой ПЛИС:**

- Все векторы должны быть заранее определенной длины (фиксированной);
- Все векторы должны быть одномерными;
- ПЛИС не поддерживает вычисления с плавающей запятой на аппаратном уровне;
- Алгоритм ограничен емкостью ПЛИС.

#### **Методика реализации программы для ПЛИС:**

- 1) Разбивка формул на отдельные типовые математические операции;
- 2) Разделение параметров на быстро и медленно меняющиеся;
- 3) Перенос расчета медленно меняющихся параметров на хост;
- 4) Приведение формул к целочисленной форме или в формате с фиксированной запятой, с заменой операции деления бинарным сдвигом или обратным умножением;

- 5) Предварительный выбор способа реализации каждой математической операции на ПЛИС;
- 6) Программное распараллеливание независимых операций (данное положение важно для синхронного исполнения параллельных веток);
- 7) Замена элементов лицевой панели константами на блок-диаграмме (если константа не представляет собой меняющийся параметр);
- 8) Свертка связанных констант и медленно меняющихся параметров в одно значение;
- 9) Выбор типа данных для каждой операции;
- 10) Предварительная разработка программы ПЛИС;
- 11) Отладка программы в режиме моделирования;
- 12) Предварительный расчет скорости работы алгоритма;
- 13) Предварительная оценка аппаратной емкости алгоритма;
- 14) Компиляция программы ПЛИС и запуск на целевом устройстве;
- 15) Проверка программы ПЛИС на соответствие требованиям;
- 16) При необходимости и возможности – корректировка программы ПЛИС
  - a. Увеличение производительности программы – замена отдельных участков кода на более быстрые;
  - b. Корректировка по емкости участка кода – замена отдельных участков кода на менее ресурсоемкие.

Ниже рассматривается каждый пункт методики более подробно.

### 3.2.1. Разбивка формул на отдельные типовые математические операции.

Разбивка формулы на составляющие необходима, для составления перечня математических операций, которые необходимо реализовать. При этом используются различные математические преобразования и теории для упрощения выражений, рассматриваются различные формы записи и соответственно программные реализации математических операций.

### 3.2.2. Разделение параметров на быстро и медленно меняющиеся

Разделим все параметры, которые участвуют в расчетах и рассчитываются сами или опрашиваются в режиме реального времени, на быстро меняющиеся и медленно меняющиеся.

Разделение осуществляется для каждого аппаратного модуля (вычислителя).

Быстро меняющимися параметрами назовем те, которые меняются в темпе поступающей информации и жестко привязаны к временным отсечкам вычислителя (аппаратного узла, контроллера, процессора), и в обязательном порядке должны быть реализованы в рамках этого модуля (назовем такой вычислитель «целевым устройством»).

Медленно меняющимися параметрами назовем те, которые могут не иметь жесткой привязки к временным отсечкам вычислителя, и расчет которых осуществляется со скоростью значительно меньшей, чем быстро меняющихся.

### 3.2.3. Перенос расчета медленно меняющихся параметров на хост

Быстро меняющиеся параметры зачастую участвуют в контуре управления и контуре системы автоматической защиты, в связи с чем требуется безотказность работы. Безотказность работы достигается разгрузкой «ответственного вычислителя» от лишних, для него, расчетов.

Такая разгрузка возможна, если архитектура комплекса разделена функционально и аппаратно на несколько уровней (например древовидная архитектура). При этом медленно меняющиеся параметры и операции над ними переносятся на уровень выше (назовем его «хост»), а результаты передаются на целевое устройство посредством каналов передачи данных.

Осуществлять обмен данными с ПЛИС можно несколькими способами. Наиболее часто встречающиеся:

- 1) Программный опрос (поллинг);
- 2) Обмен данными по прерываниям;
- 3) Канал FIFO DMA. [3]

#### 3.2.4. Приведение формул к целочисленной форме или в формате с фиксированной запятой, с заменой операции деления бинарным сдвигом или обратным умножением

Для того, чтобы заменить операцию деления сдвигом или операцией умножения, необходимо использовать масштабный коэффициент, учитывающий это замещение. Данный масштабный коэффициент удобно учитывать при расчете медленно меняющихся параметров, которые взаимодействуют с быстроменяющимися параметрами, вычисленным за счет бинарного сдвига.

Значение измеряемой физической величины внутри ПЛИС описывается в виде:

$$e_i^{*N_{ADC}}(t_c) = \left[ \frac{U_i(t_c) \cdot 2^{N_{ADCi}}}{A_i} \right], \quad (3.16)$$

где  $i$  - номер канала АЦП,  $e_i^*$  - численное представление исходного сигнала датчика после оцифровки в текущий момент времени,  $A_i$  - размах диапазона измерения,  $U_i$  - текущее значение измеренного сигнала,  $N_{ADCi}$  - разрядность канала АЦП.

Значение ускорения в точке:

$$a_i^*(t_c) = \left( f(e_i^*(t_c) - e_i^{*0}) + f^*(e_i^*(t_{c-1}) - e_i^{*0}) \right) \cdot 2^{-1}, \quad (3.17)$$

где  $a_i^*$  - численное значение ускорения в точке,  $e_i^{*0}$  - цифровое значение уставки (смещение нуля),  $f$  - функция фильтрации.

Значение скорости в точке рассчитывается:

$$v_i^*(t_c) = \left[ f\left( \dot{s}_i^*(t_c) \cdot 2^{-\left(n_i^h - N_{ADCi} + r\right)} \right) \right], \quad (3.18)$$

где  $v_i^*$  - численное значение скорости в точке,  $\dot{s}_i^*(t_c)$  - промежуточная сумма, пропорциональная ускорению точек конструкции в текущий момент времени,  $n_i^h$  - количество значащих разрядов интеграторов скорости,  $r$  - параметр, описывающий тип данных АЦП ( $r=0$  - беззнаковый тип данных,  $r=1$  - знаковый тип данных).

$$\dot{s}_i^*(t_c) = \dot{s}_i^*(t_{c-1}) + a_i^*(t_c), \quad (3.19)$$

Значение перемещения в точке:

$$y_i^*(t_c) = \left[ f\left( s_i^*(t_c) \cdot 2^{-\left(n_i^g - N_{ADCi} + r\right)} \right) \right] \quad (3.20)$$

где  $y_i^*$  - численное значение перемещения в точке,  $s_i^*(t_c)$  - промежуточная сумма, пропорциональная перемещению точек конструкции в текущий момент времени,  $n_i^g$  - количество значащих разрядов интеграторов перемещения.

$$s_i^*(t_c) = s_i^*(t_{c-1}) + \dot{s}_i^*(t_{c-1}) + a_i^*(t_c) \cdot 2^{-1} \quad (3.21)$$

Цифровое значение силового воздействия:

$$u_i^*(t_c) = \left[ \left( \sum_j^n g_{ij} y_j^*(t_c) + \sum_j^n h_{ij} v_j^*(t_c) + \sum_j^n m_{ij} a_j^*(t_c) \right) \cdot 2^{-\left(W_i - N_{DACi} + r\right)} \right], \quad (3.22)$$

где  $W$  - разрядность расширенной разрядной сетки результата,  $N_{DACi}$  - разрядность  $i$ -ого канала ЦАП.

$$W_i = n_i^k + \log_2 n + 1, \quad (3.23)$$

где  $n_i^k$  - разрядность блока перемножения вектора строки на вектор-столбец.

Восстановленное значение выходного сигнала ЦАП для силового воздействия:

$$R_i(t_c) = u_i^*(t_c) \cdot Q_i, \quad (3.24)$$

где  $R_i$  - восстановленное значение сигнала ЦАП,  $Q_i$  - масштабный коэффициент ЦАП.

$$Q_i = \frac{B_i}{2^{N_{DAC_i}}}, \quad (3.25)$$

где  $B_i$  - размах выходного диапазона изменения сигнала ЦАП,  $N_{DAC_i}$  - число разрядов ЦАП.

### 3.2.5. Предварительный выбор способа реализации каждой математической операции на ПЛИС

Выбираются библиотеки LabVIEW и их функции, посредством которых реализуется каждая математическая операция. Для одинаковых операций изначально рассматривается один вариант реализации.

Для отобранных библиотек и их функций, можно построить карты распределения ресурсов и определить скорости работы каждой операции (произведя ряд компиляций простейших программ с их использованием с различными конфигурациями). Полученные данные интерполируются полиномами малых степеней - одномерными для скалярных величин, и двумерными для операций с векторами, для последующей оценки ресурсоемкости операции с произвольной разрядностью и глубиной.

### 3.2.6. Программное распараллеливание независимых операций

Независимыми операциями называются те, результаты и входные сигналы которых не зависят друг от друга. Такие операции можно выполнить параллельно, уменьшив время исполнения кода.

Узлы и функции с различным быстродействием целесообразно размещать в разных параллельных циклах, тогда медленно исполняемые функции не будут тормозить выполнение быстрых.

При наличии длинных цепочек последовательно связанных узлов, эффективно применять конвейеризацию алгоритма, а также применять Single Cycle Timed Loop (однотактовые циклы).

Также целесообразно, выносить в отдельный цикл обмен данными между вычислителями, а получаемые и принимаемые данные передавать в математический аппарат посредством областей памяти, расположенных на ПЛИС. Однако, следует учесть, что компоненты программы, которые совместно используются в различных ветвях алгоритма, являются разделяемыми. Если не предпринять специальных мер, то не исключено, что один и тот же ресурс в один и тот же момент времени может быть затребован различными параллельно исполняемыми циклами – при этом возникает подмена или даже потеря данных до того, как они будут обработаны. Одновременное обращение к ресурсам может повлиять на детерминизм исполнения задач, послужить причиной некорректного выполнения операций.

Для достижения максимально возможного детерминизма и повышения быстродействия следует избегать использования разделяемых ресурсов, однако на практике в приложениях ПЛИС редко это удается.

Наиболее часто в качестве разделяемых ресурсов выступают как компоненты ПЛИС и модуля ввода-вывода, так и компоненты программ:

- каналы ввода-вывода;
- блоки памяти, используемые просто для хранения данных – Memory или для быстрого обмена данными – FIFO (first input – first output);

- локальные переменные и нереентерабельные subVI (подпрограммы и функции).

Каждый из перечисленных выше типов разделяемых ресурсов может обслуживаться с использованием различных режимов арбитража:

- 1) Always Arbitrate – постоянно управлять доступом к ресурсу. Реализуется на аппаратном уровне как циклическое обслуживание запросов. Повторный запрос от некоторого источника не будет удовлетворен, пока не будут обслужены все источники, запросившие ресурс;
- 2) Arbitrate if Multiple Accessors Only – этот вариант, установленный по умолчанию, позволяет включить схему арбитража, если предполагается, что возможно более одного запроса на захват ресурса. Выбор такого способа арбитража экономит ресурсы ПЛИС и повышает скорость обработки данных;
- 3) Never Arbitrate – отключение схемы арбитража. Самый экономный и быстрый способ организации работы с общими ресурсами.

В тех случаях, когда нужны нестандартные решения, исключения конфликтов добиваются программными средствами, используя функции библиотеки синхронизации. [3]

При необходимости синхронного ввода данных от каналов (синхронный опрос), функция опроса входов должна вызываться один раз на все каналы. При использовании для каждого канала своего экземпляра функции опроса каналов, происходит их поочередный вызов, что заметно уменьшает скорость работы с функциями ввода/вывода в рамках всего приложения.

### 3.2.7. Замена элементов лицевой панели константами на блок-диаграмме

При наличии элементов лицевой панели, которые хранят единственное значение, которое не может быть изменено в процессе работы программы, их заменяют константами на блок-диаграмме. Это связано с тем, что массивы

объектов лицевой панели, а также массивы данных расходуют большое количество логических элементов ПЛИС. В связи с этим не рекомендуется создавать на лицевой панели массивы индикаторов или управляющих элементов. Для хранения массивов данных лучше использовать блочную память ПЛИС – в объектах типа Memory или FIFO. [3]

### 3.2.8. Свертка связанных констант в одно значение

При наличии целой цепочки операций над константами (часто последовательно организованной), целесообразно ее заменить заранее вычисленным результатом. То же справедливо для медленно меняющихся параметров, передающих значение результата вычислений ПЛИС, с хоста.

### 3.2.9. Выбор типа данных для каждой операции

При выборе типов данных для каждой операции необходимо учитывать диапазон изменения входных и выходных параметров. Так, для числа в диапазоне от 0 до 3 целесообразно выбрать тип данных fix-point формата fxp<+2,2>, вместо типовых I16, U16. Если же исходные данные представлены в форматах, требующих большего числа элементов памяти, то на соответствующем этапе обработки целесообразно выполнить преобразование типов данных.

В некоторых случаях, для повышения производительности рекомендуется объединять данные малой разрядности в слова. [3]

Рекомендуется избегать сложных и громоздких кластеров, данные которых, при выполнении функций, частично не задействуются. Учитывая, что трассировка на ПЛИС кластера представляет собой связанный пучок сигнальных линий, незадействованные данные кластера попросту тратят ресурсы. Гораздо эффективнее производить поэлементную трассировку. Данная операция требует от программиста больше времени, однако положительно сказывается на объеме затраченных ресурсов.

### 3.2.10. Предварительная разработка программы ПЛИС

Осуществляется разработка программы, реализующей математический аппарат с учетом пп.1-9, для ПЛИС.

### 3.2.11. Отладка программы в режиме моделирования

В данном пункте осуществляется отладка алгоритма ПЛИС, и устранение всех явных и неявных ошибок, возникающих во время программирования, адаптации математического аппарата для архитектуры ПЛИС в режиме моделирования. При этом следует учесть, что во время моделирования, не идет симуляция работы ПЛИС. Т.е. параллельные операции будут выполняться независимо, если на процессоре достаточно свободных потоков для осуществления параллелизма. Иначе, процедуры будут выполняться стандартным образом – последовательно. Для наиболее точного определения места возникающих ошибок следует использовать режим анимации, для наблюдения непосредственного следования потоков данных, и анализу получающихся на каждом шаге математических операций значений.

### 3.2.12. Предварительный расчет скорости работы алгоритма

Предварительный расчет скорости работы алгоритма производится анализом графа соединений математических операций (анализ маршрутов потоков данных, на основе предварительно разработанной программы в LabVIEW), где вершины графа – математические операции, а значения дуг, исходящих из этих вершин – время работы математической операции

Алгоритм нахождения времени выполнения алгоритма описывается следующей рекуррентной формулой [32]:

$$d(1) = 0,$$

$$d(j) = \max_{(i,j)} (d(i) + c(i, j)), \quad (3.10)$$

где  $d(j)$  - время выполнения алгоритма от начальной точки до текущей математической операции,  $d(i)$  - время выполнения алгоритма до вершины  $i$ ,  $c(i, j)$  - время выполнения алгоритма от вершины  $i$  до вершины  $j$ ,  $j = 2, \dots, n + 1$ ,  $n$  - число математических операций.

### 3.2.13. Предварительная оценка аппаратной емкости алгоритма

Для предварительной оценки емкости алгоритма необходимо суммировать емкость всех математических операций. Математические операции, которые расположены в цикле, необходимо учитывать единожды.

Данное оценивание необходимо, при определении емкости ПЛИС, требующейся для реализации данного алгоритма.

### 3.2.14. Компиляция программы ПЛИС и запуск на целевом устройстве

Для генерации исполнительного файла ПЛИС необходимо настроить параметры компилятора Xilinx. В LabVIEW, для этого, необходимо создать в дереве проектов файл спецификации к программе ПЛИС. В качестве основного критерия оптимизации компиляции доступен один из вариантов – по быстродействию или сложности структуры ПЛИС. По умолчанию выбран вариант оптимизации по быстродействию.

Таким образом, улучшение производительности может привести к неэкономному расходованию ресурсов ПЛИС, но вышеперечисленные приемы проектирования позволяют улучшить качество программного кода по обоим критериям. [3]

После создания исполнительного файла ПЛИС, его необходимо загрузить на целевое устройство посредством контроллера реального времени или ПК.

### 3.2.15. Проверка программы ПЛИС на соответствие требованиям

Для проверки программы ПЛИС на соответствие требованиям, необходимо разработать методику проверки на соответствие требованиям. В каждом конкретном случае подобная методика – уникальна, однако есть несколько типовых параметров, которые необходимо оценивать достаточно часто:

- 1) Скорость выполнения программы расчета;
- 2) Фазовый сдвиг между входным и выходным сигналами в заданном частотном диапазоне;
- 3) Амплитудные искажения сигнала;
- 4) Емкость алгоритма по затрачиваемым ресурсам ПЛИС.

### 3.2.16. Корректировка программы ПЛИС

- a) Увеличение производительности программы
  - a. Уменьшение длины путей прохождения данных в блок-диаграмме;
  - b. Компоновка внутри параллельных циклов функций, не замедляющих выполнение друг друга;
  - c. Использование реентерабельных функций и узлов;
  - d. Упаковка данных для пересылок между циклами и между программой ПЛИС программой хоста;
  - e. Использование синхросигналов повышенной частоты для структур Timed Loop.
- b) Уменьшение ресурсоемкости программы
  - a. Аккуратное применение ресурсоемких функций и возможная их замена;
  - b. Минимизация массивов и объектов лицевой панели;
  - c. Использование структур циклов Single Cycle Timed Loop;
  - d. Использование циклов для последовательного выполнения однотипных операций.

Применение методики при разработке ИИУК позволило увеличить быстродействие измерительно-вычислительного и управляющего контура ПО ПЛИС с 36,9кГц до 157,5 кГц, что соответствует увеличению в 4,2 раза.

Использование ресурсов ПЛИС:

1. Slices: уменьшилось на 4,9% (с 15120 до 14376);
2. Registers: уменьшилось на 5,6% (с 48236 до 45531);
3. LUTs: уменьшилось на 7,3% (с 44713 до 41435);
4. DSP: увеличилось на 14,5% (с 48 до 55);
5. RAM: увеличилось с 0 до 17 ячеек

### **3.3. Построение карт распределения ресурсов ПЛИС.**

Для оценки ресурсоемкости каждой математической операции и функциональной части, необходимо построить карту распределения ресурсов. На основе данной информации выводится аналитическая зависимость в виде одно- или двумерного полинома, в представлении набора коэффициентов для расчета.

При анализе преобразованных аэродинамических уравнений, видно, что типовыми операциями являются следующие:

1. Умножение вектора-строки на вектор-столбец;
2. Бинарный сдвиг;
3. Сложение векторов;
4. Сложение двух чисел;
5. Умножение двух чисел.

Рассмотрим, в качестве примера типовую операцию «умножение вектора-строки на вектор-столбец».

**Использование операции умножения и суммирования, согласно правилам линейной алгебры, для нахождения результата.**

Пример программы, реализующей данную операцию для восьмиэлементного массива:

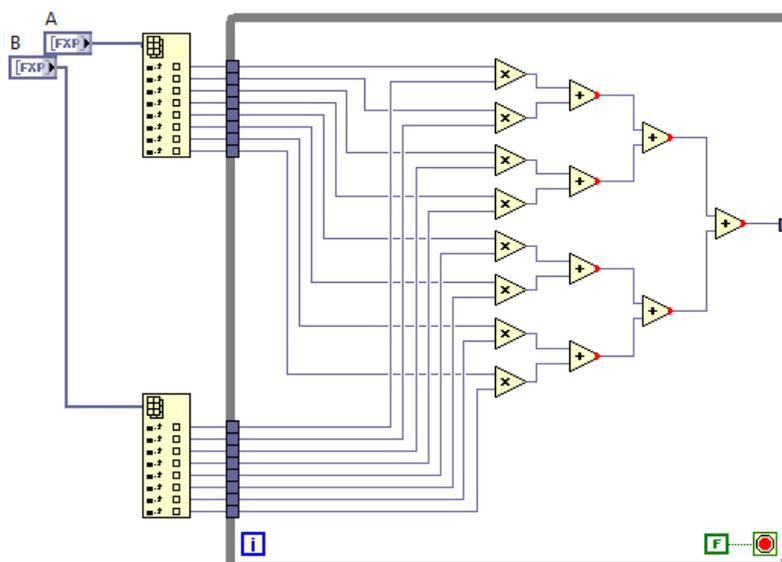


Рис. 3.1. Программа поэлементного произведения векторов, библиотека Numeric

Эксперименты показали, что скорость выполнения поэлементного произведения векторов - один такт ПЛИС

**Компиляция прошивки ПЛИС с рекомендуемыми параметрами:**

- цель оптимизации синтеза: Скорость выполнения;
- качество оптимизации процесса синтеза: Стандартное (среднее);
- качество разметки: Высокое;
- качество трассировки: Стандартное (среднее)

В результате компиляции, выяснено, что скорость исполнения алгоритма не зависит от разрядности или глубины вектора, при свободном размещении математической операции на кристалле и равно одному такту синхросигнала ПЛИС

Таблица 3.1. Число задействованных элементов Slice для библиотеки функций Numeric.

		Число элементов вектора						
		2	3	4	5	6	7	8
Разрядность, бит	1	334	326	337	335	377	352	369
	2	390	373	381	342	382	363	345
	3	328	390	356	392	366	383	352
	4	395	390	403	397	349	371	399
	5	350	372	345	367	413	403	420
	6	377	352	350	417	433	412	435
	7	355	394	400	415	450	429	431
	8	391	387	379	444	433	417	445
	9	344	403	404	424	455	447	467
	10	341	396	440	447	411	429	465
	11	345	418	450	452	475	447	426
	12	383	437	408	451	465	487	485
	13	388	470	422	468	469	464	510
	14	388	418	440	473	446	482	474
	15	406	434	449	439	477	500	521
	16	369	413	414	458	486	453	530
	17	414	442	452	490	476	463	535
	18	416	415	422	473	474	482	507
	19	456	449	432	469	494	503	489
	20	424	442	455	465	496	503	533
	21	420	378	396	445	453	461	521
	22	429	459	454	479	533	531	588
	23	410	462	467	502	534	541	520
	24	400	451	438	502	494	532	532
	25	423	457	451	484	486	519	568

(продолжение)

	26	447	430	518	483	494	516	534
	27	417	440	469	494	524	572	565
	28	446	452	474	530	528	552	570
	29	419	447	449	500	518	555	572
	30	460	478	483	510	515	550	580
	31	439	453	481	507	525	582	585
	32	450	450	506	488	518	555	589

Таблица 3.2. Число задействованных регистров (Registers)

		Число элементов вектора						
		2	3	4	5	6	7	8
Разрядность, бит	1	502	508	514	520	610	532	618
	2	602	610	618	550	634	574	586
	3	526	622	562	646	598	670	634
	4	618	634	650	666	634	658	714
	5	550	646	610	640	706	733	743
	6	634	598	634	706	767	747	791
	7	574	670	658	733	779	793	807
	8	650	682	682	775	791	775	823
	9	598	694	735	753	803	821	871
	10	610	670	775	795	783	835	887
	11	622	729	783	805	859	849	871
	12	682	767	759	815	871	895	919
	13	690	773	799	857	883	877	968
	14	698	747	807	867	863	924	952
	15	706	785	815	845	907	970	1000
	16	682	759	791	887	919	952	1016
	17	731	797	863	897	964	966	1064
	18	735	771	839	907	944	1012	1048

(продолжение)

19	771	809	847	885	988	1026	1064
20	775	815	887	928	968	1040	1112
21	745	738	795	853	878	935	1024
22	783	859	903	980	1056	1068	1176
23	755	833	911	990	1068	1146	1160
24	759	871	887	1000	1048	1160	1176
25	763	877	928	1010	960	1142	1256
26	799	851	968	1020	1072	1156	1240
27	773	860	980	1035	1122	1241	1264
28	807	895	984	1072	1160	1216	1304
29	779	869	960	1050	1140	1230	1353
30	815	907	968	1060	1152	1276	1369
31	819	913	976	1070	1164	1290	1385
32	823	887	1016	1080	1176	1272	1401

Таблица 3.3. Число задействованных ячеек LUTs для библиотеки функций Numeric :

		Число элементов вектора						
		2	3	4	5	6	7	8
Разрядность, бит	1	504	507	509	508	605	515	610
	2	607	616	608	523	615	530	537
	3	512	611	526	625	538	625	549
	4	623	610	627	628	552	558	631
	5	524	613	544	552	642	676	676
	6	624	540	553	642	741	676	750
	7	531	626	559	676	743	748	753
	8	627	636	568	759	745	676	745
	9	539	638	676	676	750	739	783
	10	543	564	743	752	676	710	780
	11	548	676	741	757	781	709	709
	12	635	752	676	745	788	779	788
	13	633	738	746	792	785	709	821
	14	636	676	745	774	709	745	745
	15	632	747	740	710	780	808	805
	16	568	676	676	786	784	744	814
	17	676	746	783	781	815	745	852
	18	676	676	709	774	745	821	777
	19	743	749	709	709	818	776	776
	20	745	747	776	744	744	776	847
	21	690	623	660	699	703	741	849
	22	743	776	772	811	841	776	884
	23	676	710	769	804	847	877	809
	24	676	776	710	813	776	877	776
	25	677	772	744	815	813	808	900

(продолжение)

	26	757	709	810	776	776	808	809
	27	676	709	813	776	808	901	808
	28	743	784	810	843	876	808	899
	29	676	710	744	777	808	809	933
	30	746	774	744	776	809	899	933
	31	743	786	745	776	809	899	933
	32	759	710	811	777	777	808	933

Таблица 3.4. Общее число задействованных элементов ПЛИС для библиотеки функций Numeric

		Число элементов вектора						
		2	3	4	5	6	7	8
Разрядность, бит	1	1340	1341	1360	1363	1592	1399	1597
	2	1599	1599	1607	1415	1631	1467	1468
	3	1366	1623	1444	1663	1502	1678	1535
	4	1645	1634	1680	1691	1535	1587	1744
	5	1424	1631	1499	1559	1761	1812	1839
	6	1635	1490	1537	1765	1941	1835	1976
	7	1460	1690	1617	1824	1972	1970	1991
	8	1671	1705	1629	1978	1969	1868	2013
	9	1481	1735	1815	1853	2008	2007	2121
	10	1494	1630	1958	1994	1870	1974	2132
	11	1515	1823	1974	2014	2115	2005	2006
	12	1693	1956	1843	2011	2124	2161	2192
	13	1711	1981	1967	2117	2137	2050	2299
	14	1722	1841	1992	2114	2018	2151	2171
	15	1744	1966	2004	1994	2164	2278	2326
	16	1637	1848	1881	2131	2189	2149	2360

(продолжение)

17	1821	1985	2098	2168	2255	2174	2451
18	1827	1862	1970	2154	2163	2315	2332
19	1970	2007	1988	2063	2300	2305	2329
20	1944	2004	2118	2137	2208	2319	2492
21	1855	1739	1851	1997	2034	2137	2394
22	1955	2094	2129	2270	2430	2375	2648
23	1841	2005	2147	2296	2449	2564	2489
24	1835	2098	2035	2315	2318	2569	2484
25	1863	2106	2123	2309	2259	2469	2724
26	2003	1990	2296	2279	2342	2480	2583
27	1866	2009	2282	2305	2454	2714	2637
28	1996	2131	2268	2445	2564	2576	2773
29	1874	2026	2153	2327	2466	2594	2858
30	2021	2159	2195	2346	2476	2725	2882
31	2001	2152	2202	2353	2498	2771	2903
32	2032	2047	2333	2345	2471	2635	2923

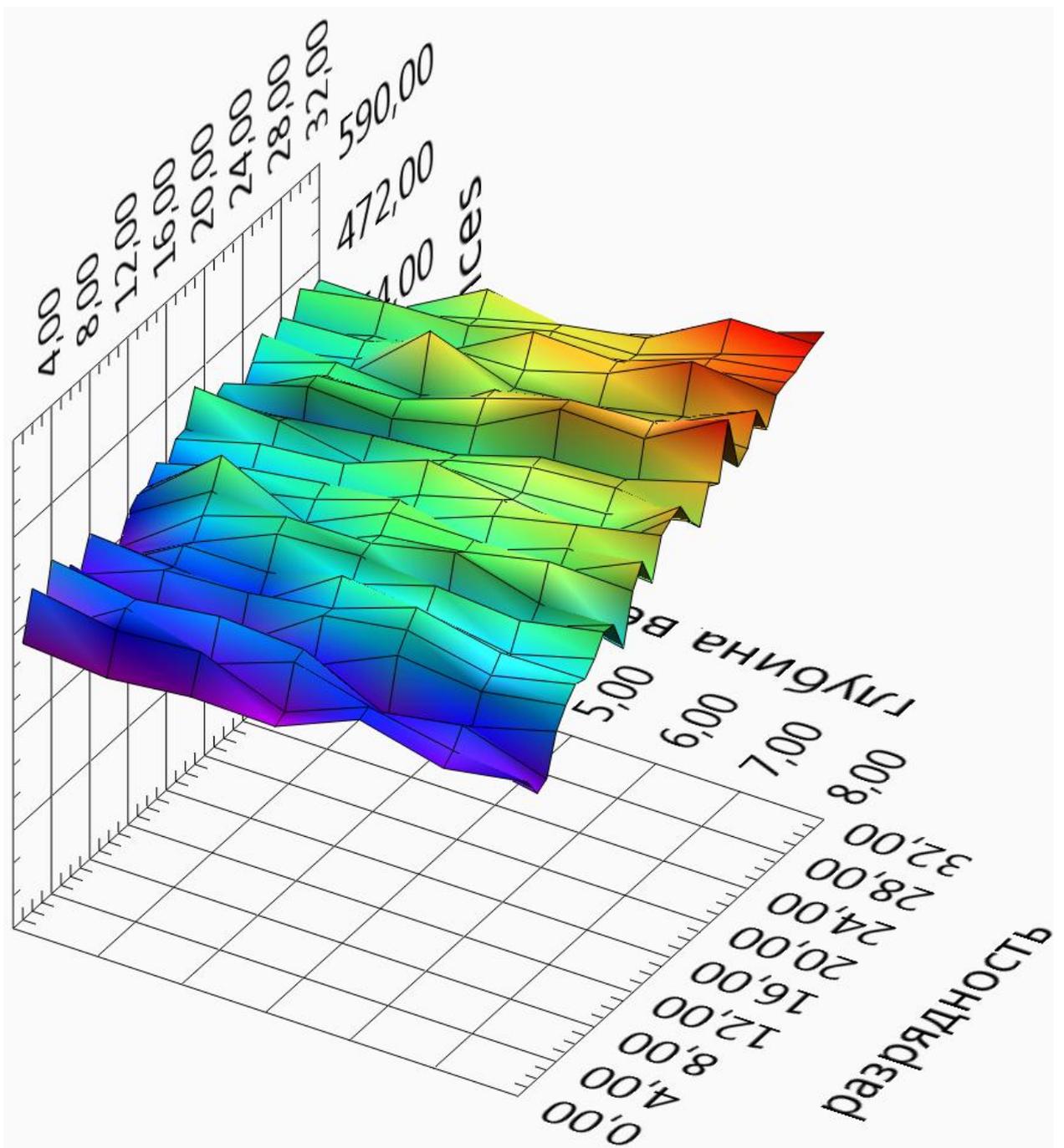


Рис. 3.2. Карта затраченных элементов Slices по полным экспериментальным данным для библиотеки Numeric, рекомендуемые настройки компиляции.

Поверхность, полученная на рис.3.2 описывается полиномом с матричными коэффициентами:

Таблица 3.5. Матричное значение коэффициентов полинома (часть 1)

		Индекс коэффициента В			
		0	1	2	3
Индекс коэффициента А	0	3,616E+3	-5,485E+3	3,604E+3	-1,188E+3
	1	-1,611E+3	2,473E+3	-1,500E+3	461,453E+0
	2	102,566E+0	-157,355E+0	95,671E+0	-29,518E+0
	3	-1,767E+0	2,740E+0	-1,685E+0	525,538E-3

Таблица 3.6. Матричное значение коэффициентов полинома (часть 2)

		Индекс коэффициента В		
		4	5	6
Индекс коэффициента А	0	207,597E+0	-18,344E+0	643,924E-3
	1	-76,087E+0	6,398E+0	-215,326E-3
	2	4,880E+0	-411,329E-3	13,876E-3
	3	-87,742E-3	7,463E-3	-253,801E-6

Полиномиальное восстановление поверхности имеет относительную ошибку по поверхности 2,465%.

Для восстановления поверхности из полиномов, следует произвести следующие вычисления:

$$f(u, v) = \sum_{i=0}^n A_i(u) \cdot v^i, \quad (3.11)$$

где  $f$  - восстановленное значение в точке пересечения  $u$  и  $v$ ,  $u$  - глубина массива,  $v$  - разрядность элемента массива,  $A(u)$  - коэффициент, описываемый полиномиальной зависимостью от глубины массива  $u$ ,  $n$  - степень полинома, описывающего зависимость восстановленного значения от разрядности слова.

$$A_i(u) = \sum_{j=0}^m B_{ij} \cdot u^j, \quad (3.12)$$

где  $B$  - элемент матрицы коэффициентов,  $m$  - степень полинома, описывающего зависимость от глубины массива.

На рис.3.3 изображена восстановленная поверхность.

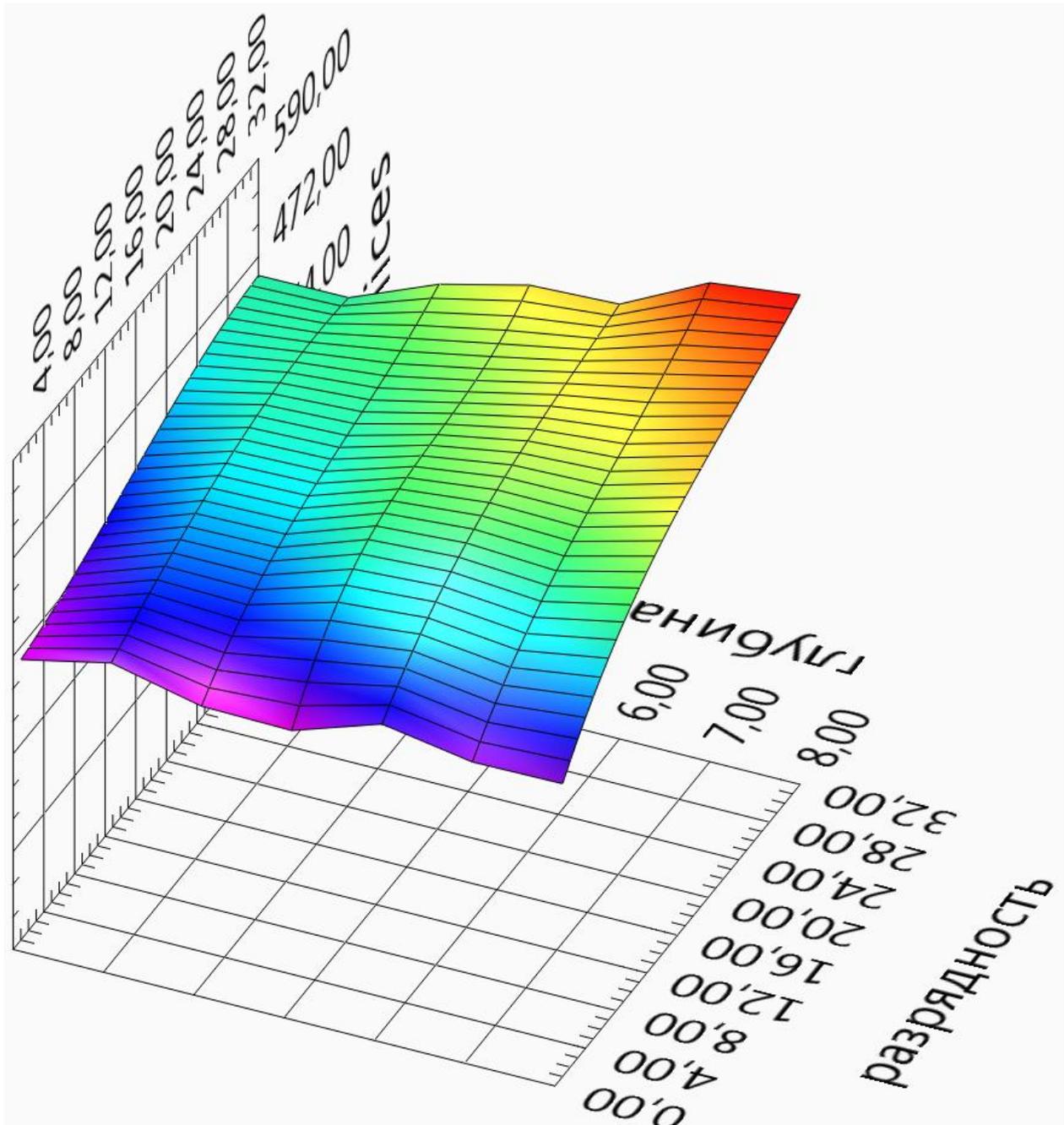


Рис. 3.3. Карта затраченных элементов Slices по полным экспериментальным данным (224 точки) для библиотеки Numeric, рекомендуемые настройки компиляции, восстановленная по полиномиальному преобразованию.

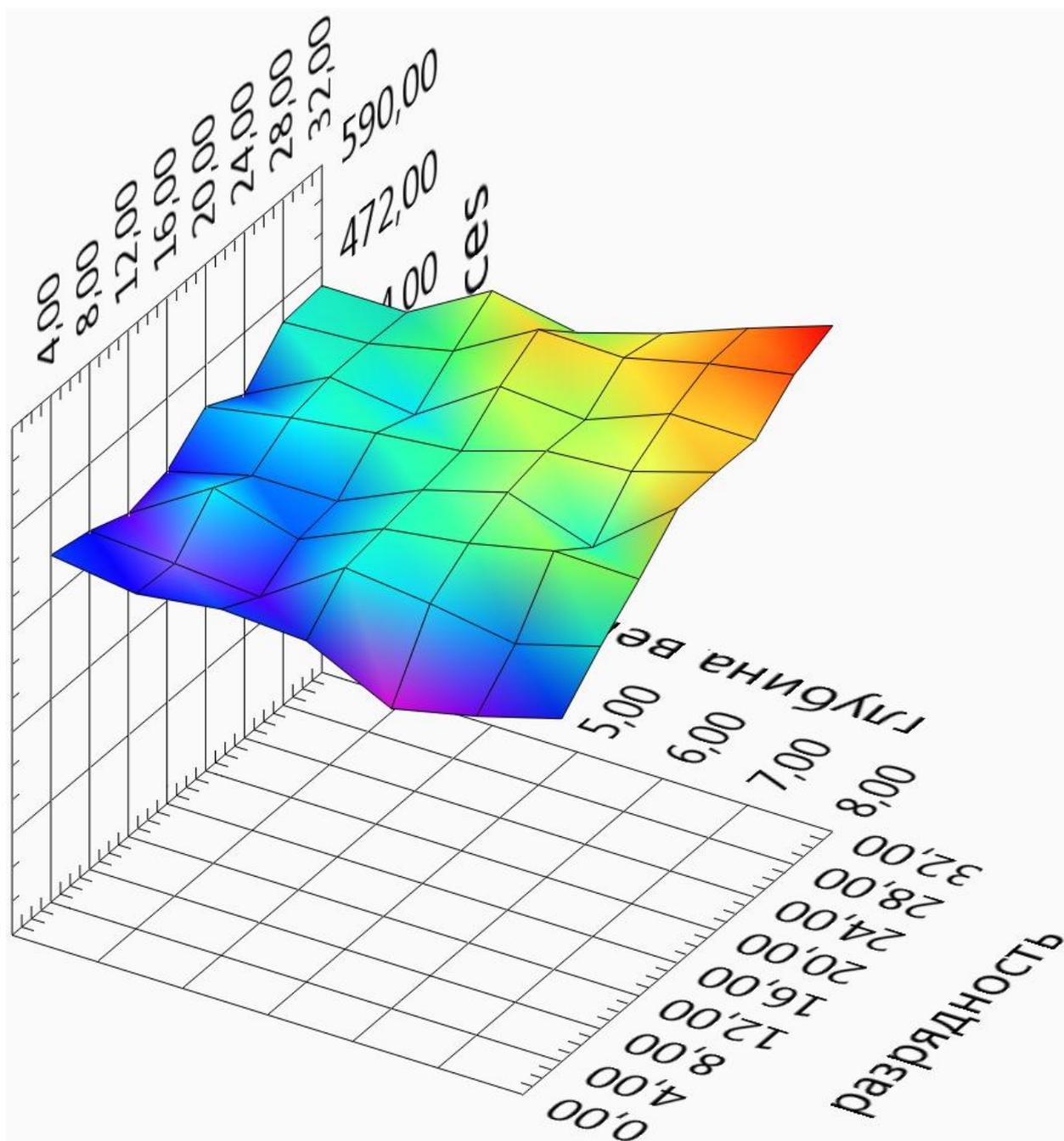


Рис. 3.4. Карта затраченных элементов Slices, по сокращенным в 4 раза экспериментальным данным (56 точек), для библиотеки Numeric, рекомендуемые настройки компиляции.

Поверхность, полученная на рис.3.4 описывается двумерным полиномом с матричными коэффициентами:

Таблица 3.7. Матричное значение коэффициентов полинома (часть 1)

		Индекс коэффициента В			
		0	1	2	3
Индекс коэффициента А	0	2,42E+02	4,95E+02	-4,44E+02	1,73E+02
	1	1,96E+01	-9,29E+01	9,16E+01	-3,81E+01
	2	-1,28E+01	2,18E+01	-1,45E+01	4,85E+00
	3	4,43E-01	-6,87E-01	4,23E-01	-1,33E-01

Таблица 3.8. Матричное значение коэффициентов полинома (часть 2)

		Индекс коэффициента В		
		4	5	6
Индекс коэффициента А	0	-3,39E+01	3,24E+00	-1,21E-01
	1	7,92E+00	-8,02E-01	3,14E-02
	2	-8,73E-01	8,02E-02	-2,94E-03
	3	2,26E-02	-1,98E-03	6,99E-05

Полиномиальное восстановление поверхности имеет относительную ошибку по поверхности 2,836% в сравнении с полной сеткой и 1,603% по сравнению с исходными данными по разреженной сетке.

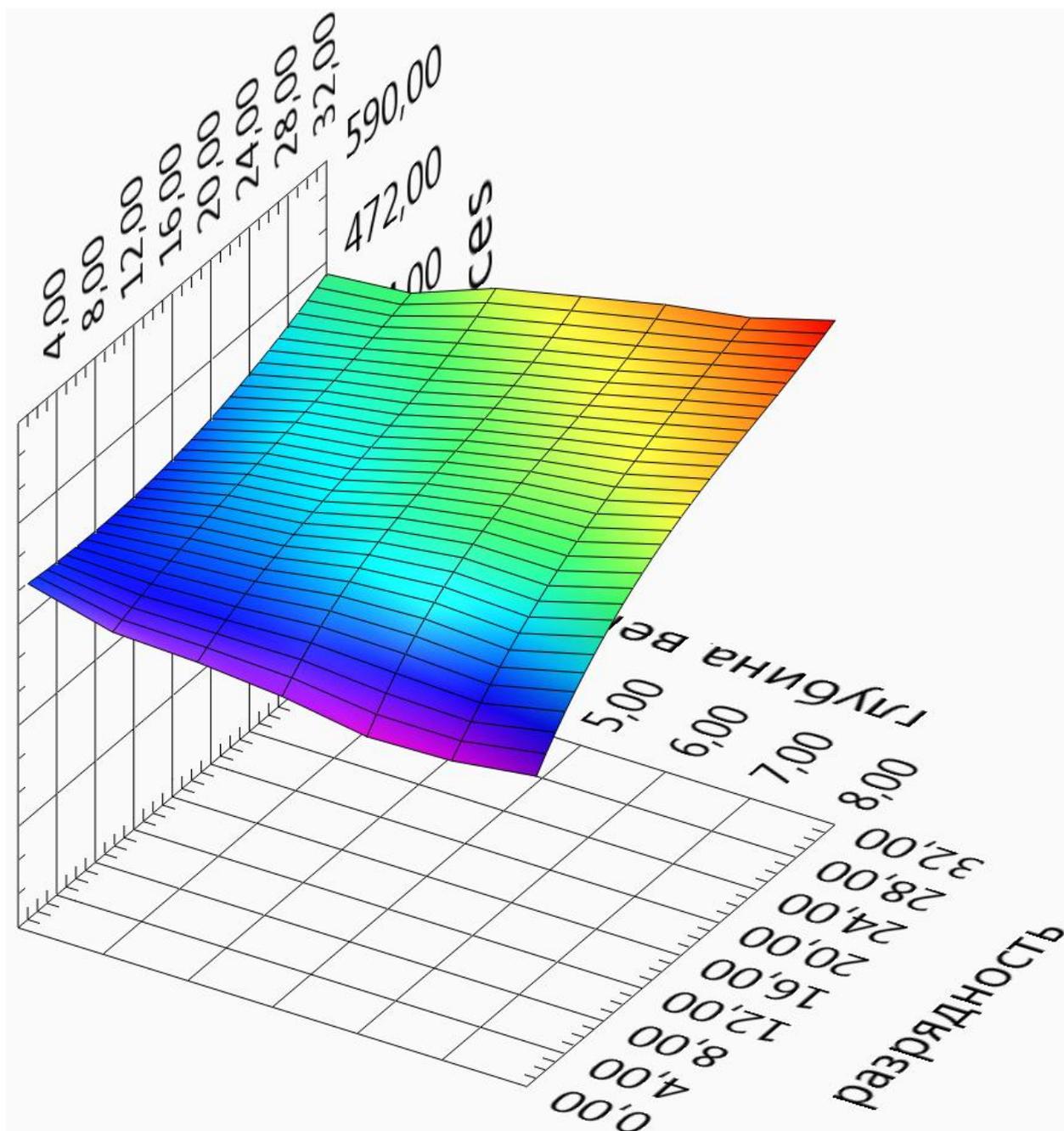


Рис. 3.5. Карта затраченных элементов Slices по сокращенным в 4 раза экспериментальным данным (56 точек) для библиотеки Numeric, рекомендуемые настройки компиляции, восстановленная по полиномиальному преобразованию.

Данные результаты показывают, что при построении карт распределения ресурсов, по разреженной сетке, не сильно увеличивает погрешность полиномиального описания поверхности.

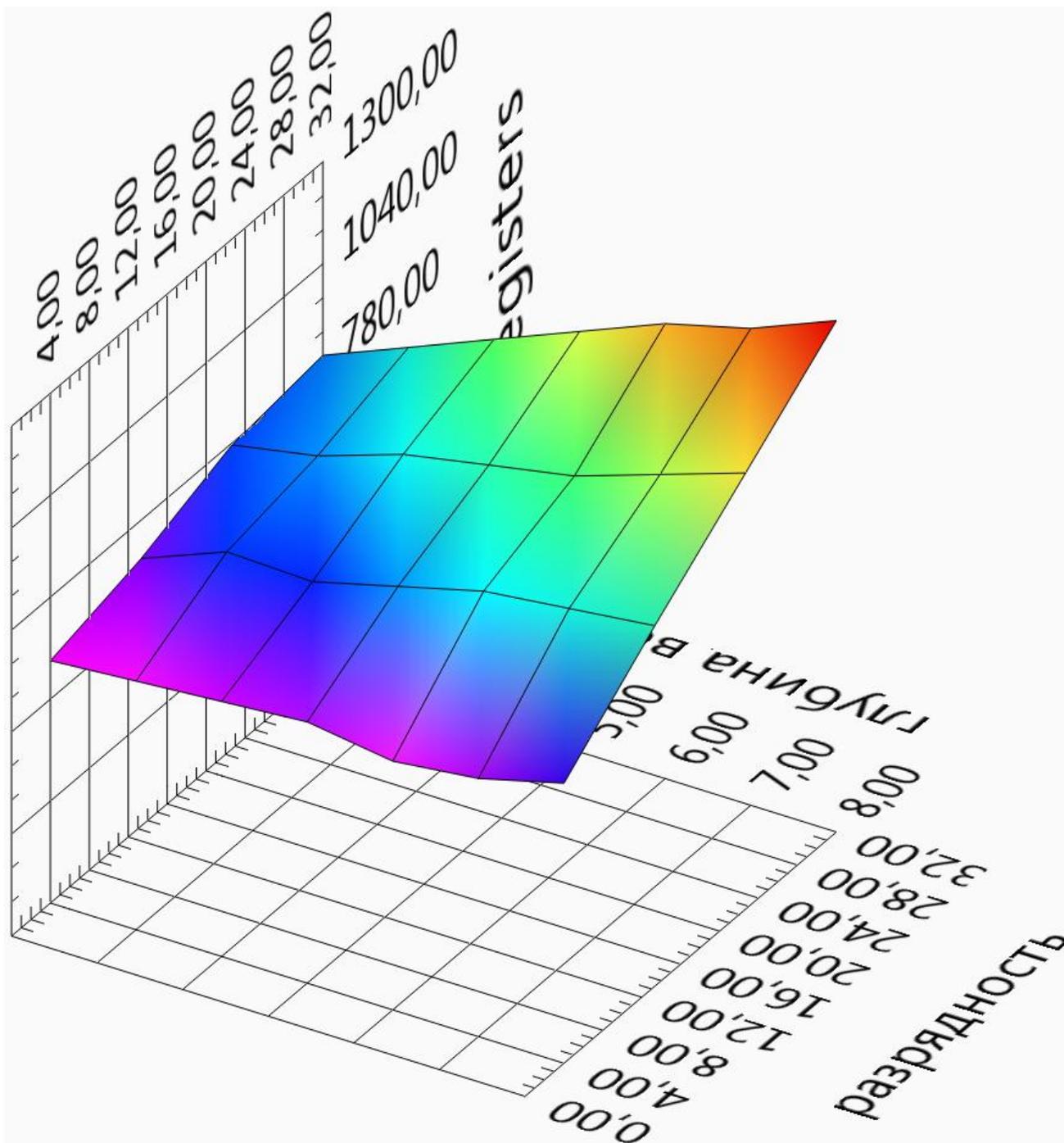


Рис. 3.6. Карта затраченных элементов Registers по сокращенным экспериментальным данным (28 точек) для библиотеки Numeric, рекомендуемые настройки компиляции.

Поверхность, полученная на рис.3.6 описывается двумерным полиномом с матричными коэффициентами:

Таблица 3.9. Матричное значение коэффициентов полинома (часть 1)

		Индекс коэффициента В			
		0	1	2	3
Индекс коэффициента А	0	1,14E+02	9,01E+02	-7,72E+02	3,07E+02
	1	-1,52E+02	1,23E+02	1,97E+01	-3,23E+01
	2	1,94E+01	-2,02E+01	5,20E+00	3,82E-01
	3	-4,68E-01	5,24E-01	-1,77E-01	1,50E-02

Таблица 3.10. Матричное значение коэффициентов полинома (часть 2)

		Индекс коэффициента В		
		4	5	6
Индекс коэффициента А	0	-6,12E+01	5,94E+00	-2,22E-01
	1	9,21E+00	-1,06E+00	4,37E-02
	2	-3,57E-01	5,17E-02	-2,36E-03
	3	3,41E-03	-7,22E-04	3,69E-05

Полиномиальное восстановление поверхности имеет относительную ошибку по поверхности 1,610% по сравнению с исходными данными по разреженной сетке.

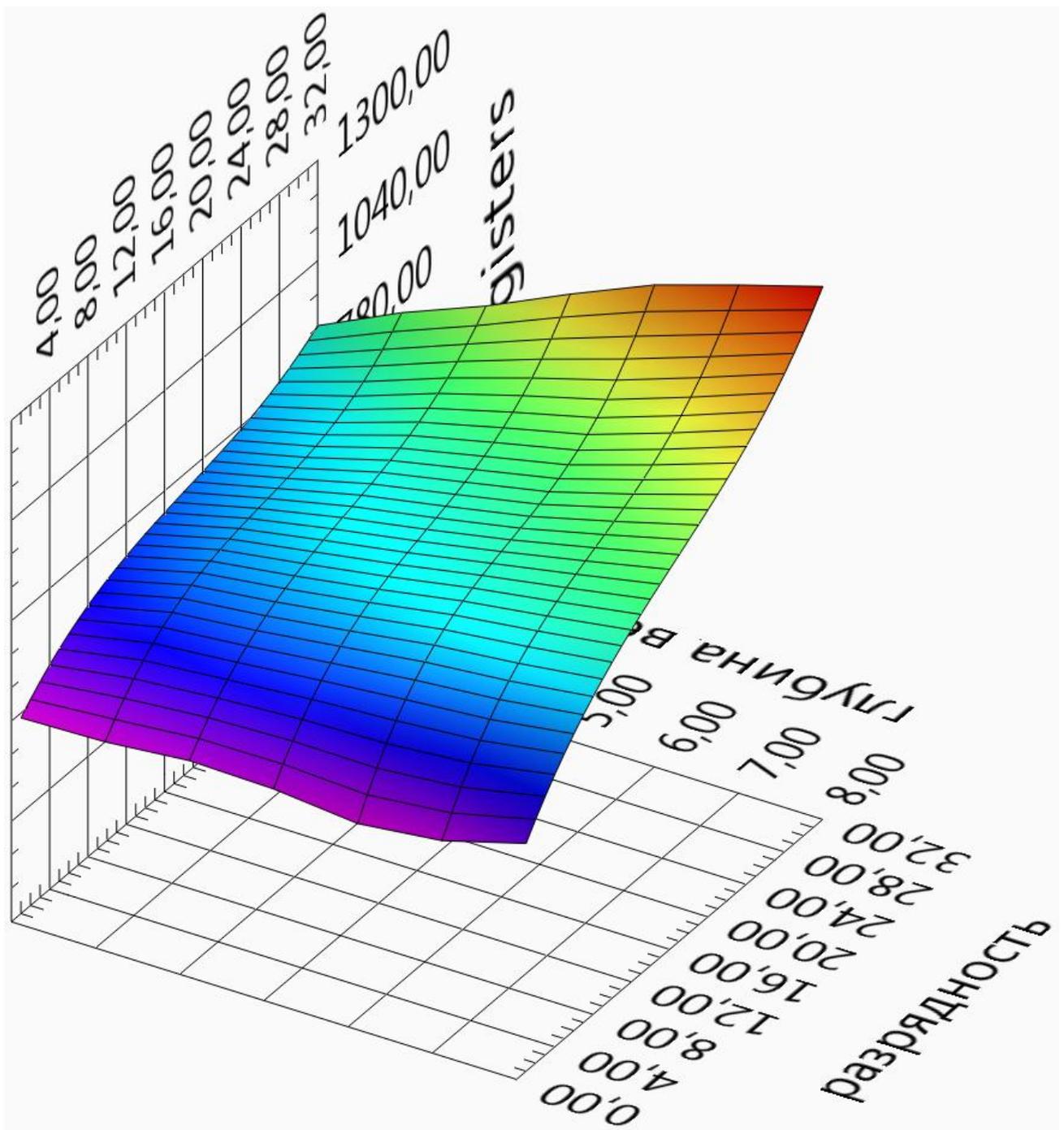


Рис. 3.7. Карта затраченных элементов Registers по сокращенным экспериментальным данным (28 точек) для библиотеки Numeric, рекомендуемые настройки компиляции, восстановленная по полиномиальному преобразованию.

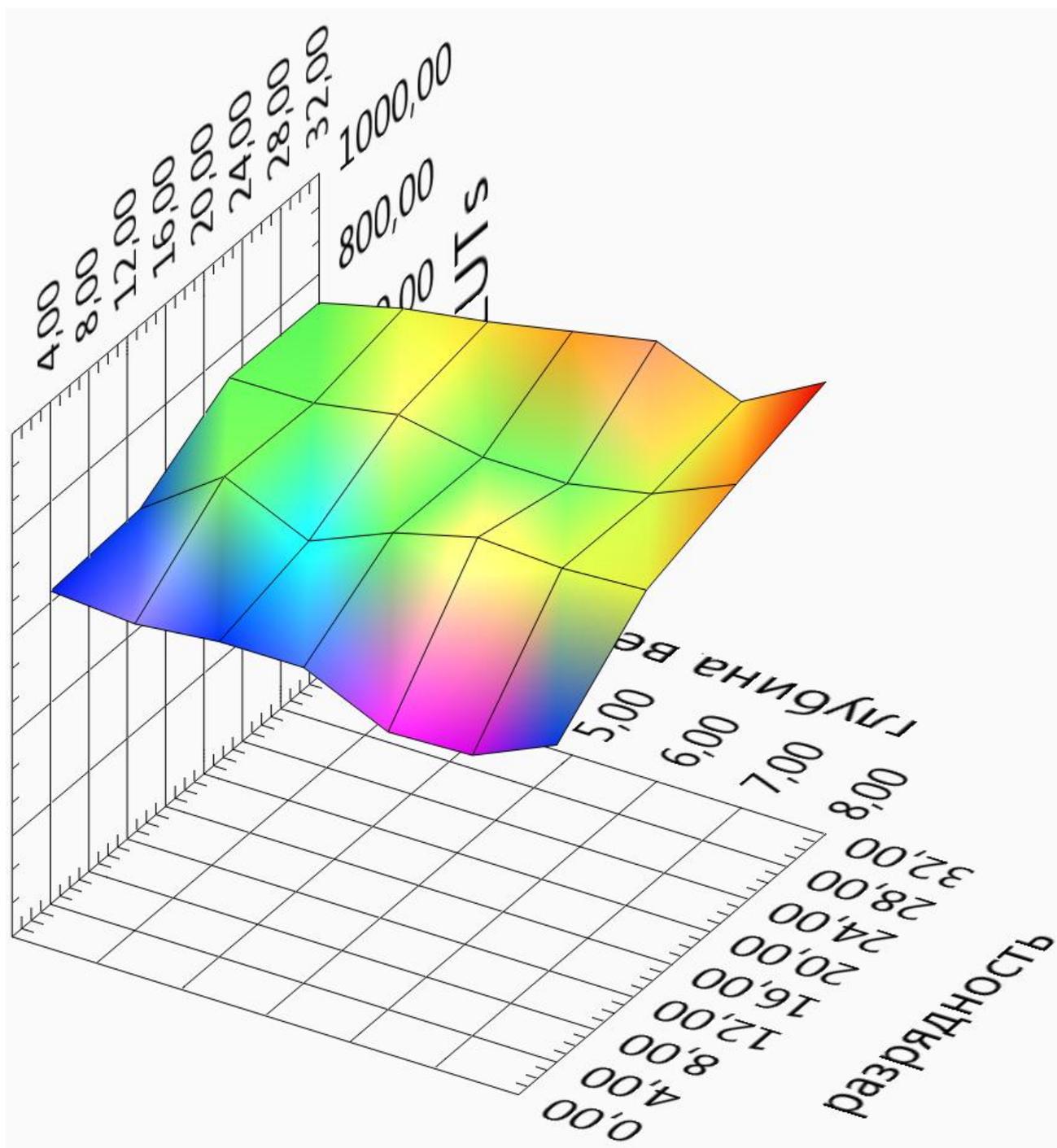


Рис. 3.8. Карта затраченных элементов LUTs по сокращенным экспериментальным данным (28 точек) для библиотеки Numeric, рекомендуемые настройки компиляции.

Поверхность, полученная на рис.3.8 описывается двумерным полиномом с матричными коэффициентами:

Таблица 3.11. Матричное значение коэффициентов полинома (часть 1)

		Индекс коэффициента В			
		0	1	2	3
Индекс коэффициента А	0	1,77E+03	-1,07E+03	1,61E+02	9,32E+01
	1	-1,14E+03	1,51E+03	-7,72E+02	1,97E+02
	2	9,67E+01	-1,32E+02	7,07E+01	-1,91E+01
	3	-2,00E+00	2,77E+00	-1,51E+00	4,16E-01

Таблица 3.12. Матричное значение коэффициентов полинома (часть 2)

		Индекс коэффициента В		
		4	5	6
Индекс коэффициента А	0	-3,77E+01	4,88E+00	-2,15E-01
	1	-2,63E+01	1,75E+00	-4,59E-02
	2	2,75E+00	-2,02E-01	5,95E-03
	3	-6,13E-02	4,61E-03	-1,40E-04

Полиномиальное восстановление поверхности имеет относительную ошибку по поверхности 1,677% по сравнению с исходными данными по разреженной сетке.

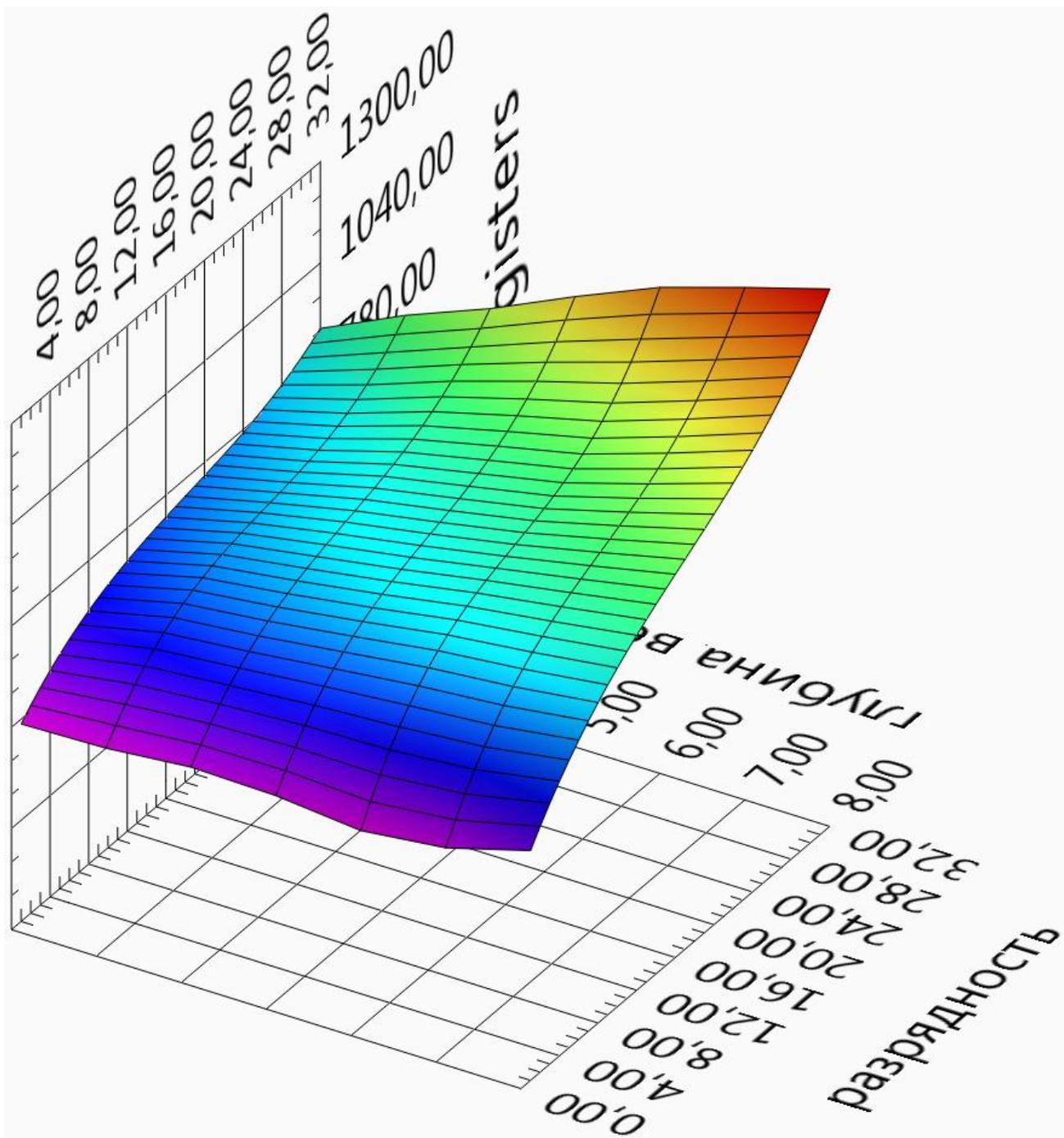


Рис. 3.9. Карта затраченных элементов Registers по сокращенным экспериментальным данным (28 точек) для библиотеки Numeric, рекомендуемые настройки компиляции, восстановленная по полиномиальному преобразованию.

### Компиляция прошивки ПЛИС с пользовательскими параметрами:

- цель оптимизации синтеза: Площадь кристалла;
- качество оптимизации процесса синтеза: Высокое;
- качество разметки: Высокое;
- качество трассировки: Высокое.

Таблица 3.13. Число задействованных элементов Slices

		Число элементов вектора						
		2	3	4	5	6	7	8
Разрядность, бит	4	358	358	358	358	358	358	358
	8	358	358	358	358	358	358	358
	12	365	365	365	365	365	365	365
	16	365	365	365	365	365	365	365
	20	340	340	340	340	340	340	340
	24	340	340	340	340	340	340	340
	28	348	348	348	348	348	348	348
	32	348	348	348	348	348	348	348

Таблица 3.14. Число задействованных Registers

		Число элементов вектора						
		2	3	4	5	6	7	8
Разрядность, бит	4	538	562	586	610	634	658	682
	8	586	634	682	743	759	775	791
	12	634	735	759	783	839	863	887
	16	682	759	791	855	887	952	984
	20	743	783	855	928	968	1040	1080
	24	759	839	887	968	1048	1128	1176
	28	775	863	952	1040	1128	1216	1272
	32	791	887	984	1080	1176	1272	1369

Таблица 3.1. Число задействованных LUTs

		Число элементов вектора						
		2	3	4	5	6	7	8
Разрядность, бит	4	486	491	494	498	503	506	510
	8	494	502	511	554	554	554	554
	12	502	554	554	554	587	587	588
	16	511	554	554	587	587	623	623
	20	555	554	588	623	623	655	655
	24	554	588	588	623	655	687	687
	28	554	587	623	655	687	719	719
	32	555	587	623	655	687	719	752

В результате испытаний, выяснено, что скорость исполнения алгоритма не зависит от разрядности или глубины вектора, при свободном размещении математической операции на кристалле и равно одному такту синхросигнала ПЛИС.

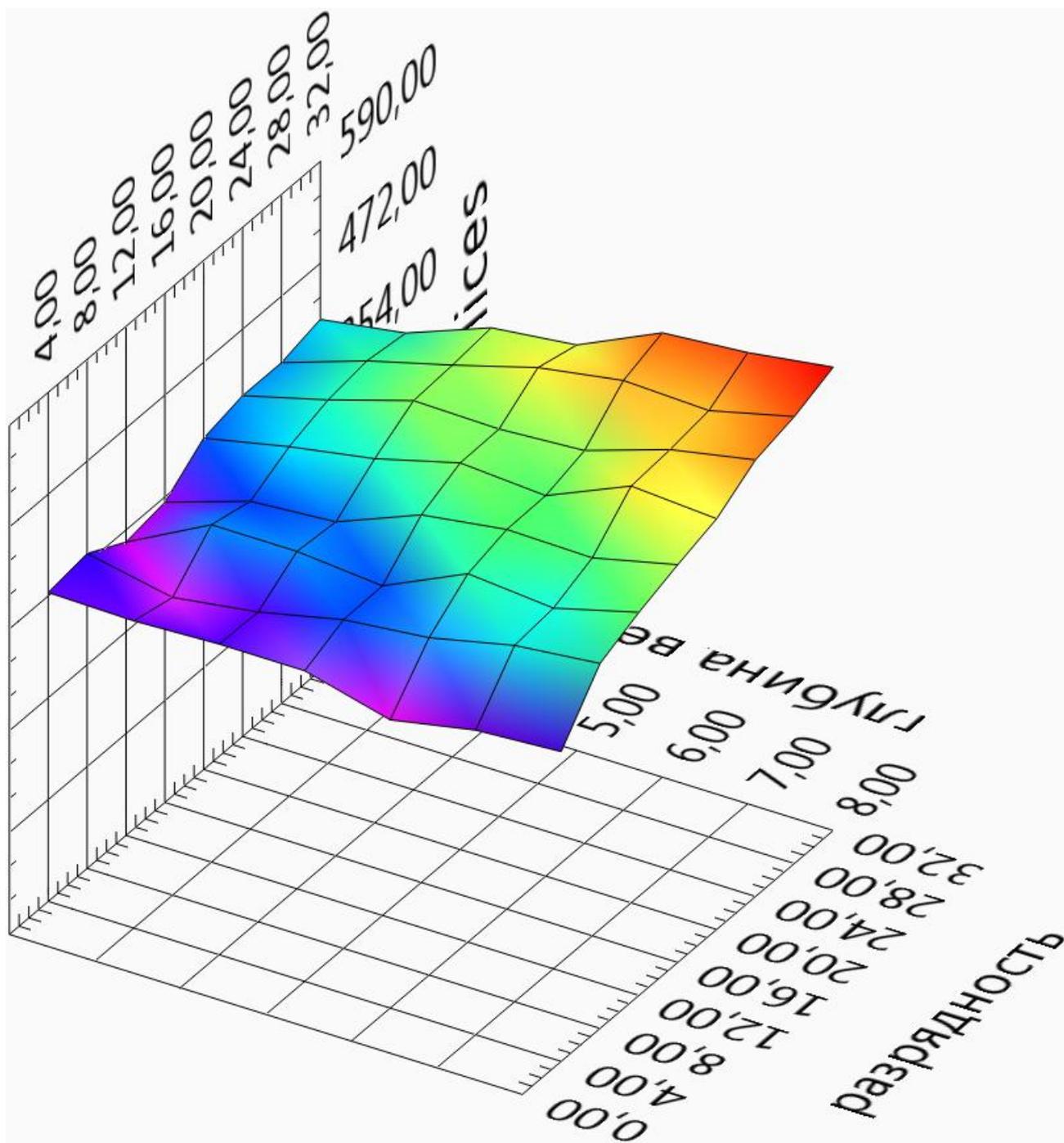


Рис. 3.10. Карта затраченных элементов Slices по экспериментальным данным для библиотеки Numeric, пользовательские настройки компиляции.

Поверхность, полученная на рис.3.10 описывается полиномом с матричными коэффициентами:

Таблица 3.15. Матричное значение коэффициентов полинома (часть 1)

		Индекс коэффициента В			
		0	1	2	3
Индекс коэффициента А	0	-9,27E+02	2,05E+03	-1,32E+03	4,33E+02
	1	2,93E+02	-4,78E+02	3,10E+02	-1,03E+02
	2	-2,09E+01	3,36E+01	-2,15E+01	7,08E+00
	3	4,32E-01	-6,85E-01	4,34E-01	-1,42E-01

Таблица 3.16. Матричное значение коэффициентов полинома (часть 2)

		Индекс коэффициента В		
		4	5	6
Индекс коэффициента А	0	-7,65E+01	6,85E+00	-2,43E-01
	1	1,82E+01	-1,64E+00	5,88E-02
	2	-1,26E+00	1,13E-01	-4,05E-03
	3	2,50E-02	-2,25E-03	8,03E-05

Полиномиальное восстановление поверхности имеет относительную ошибку по поверхности 1,191%.

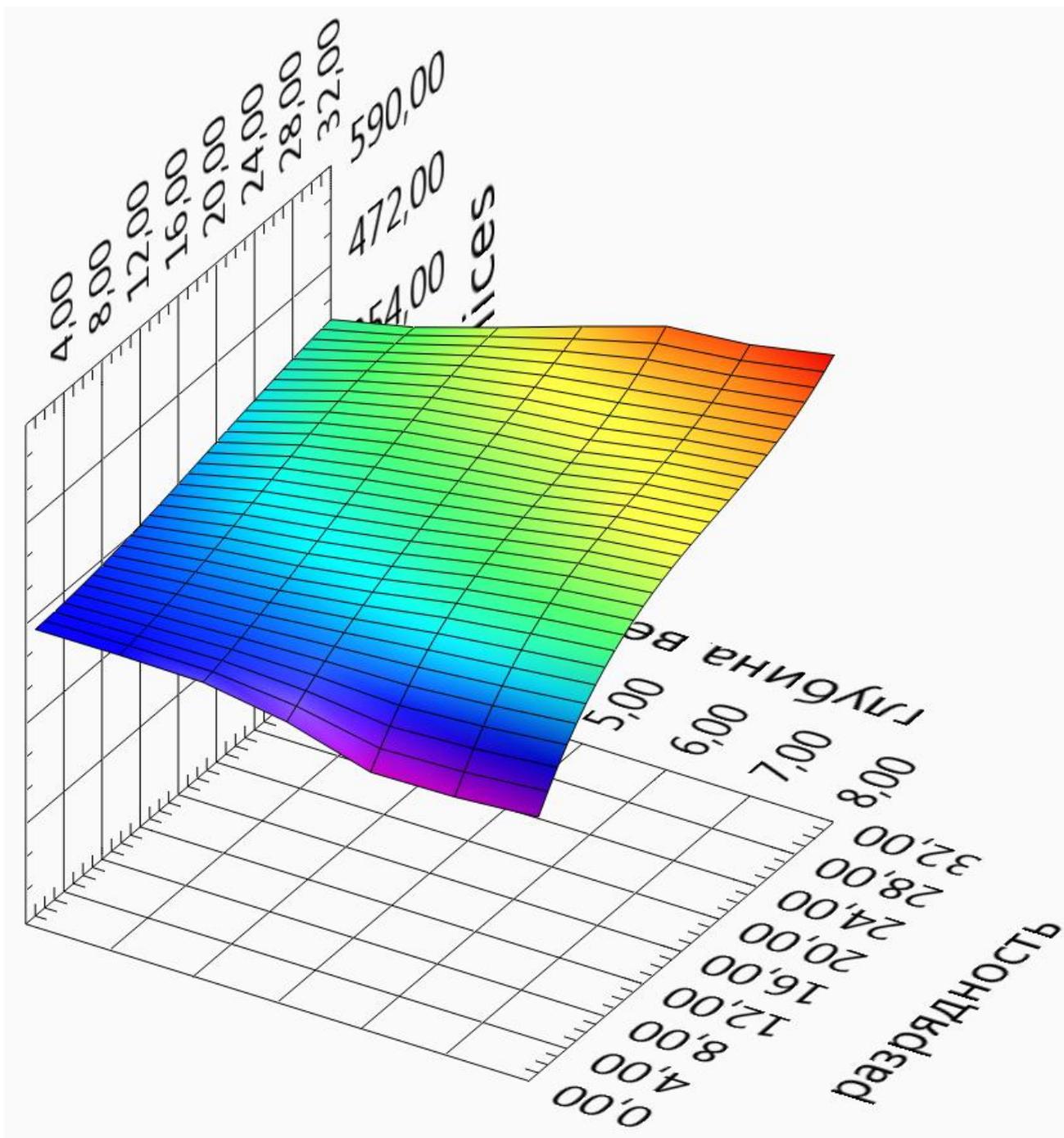


Рис. 3.11. Карта затраченных элементов Slices по экспериментальным данным для библиотеки Numeric, пользовательские настройки компиляции, восстановленная по полиномиальному преобразованию.

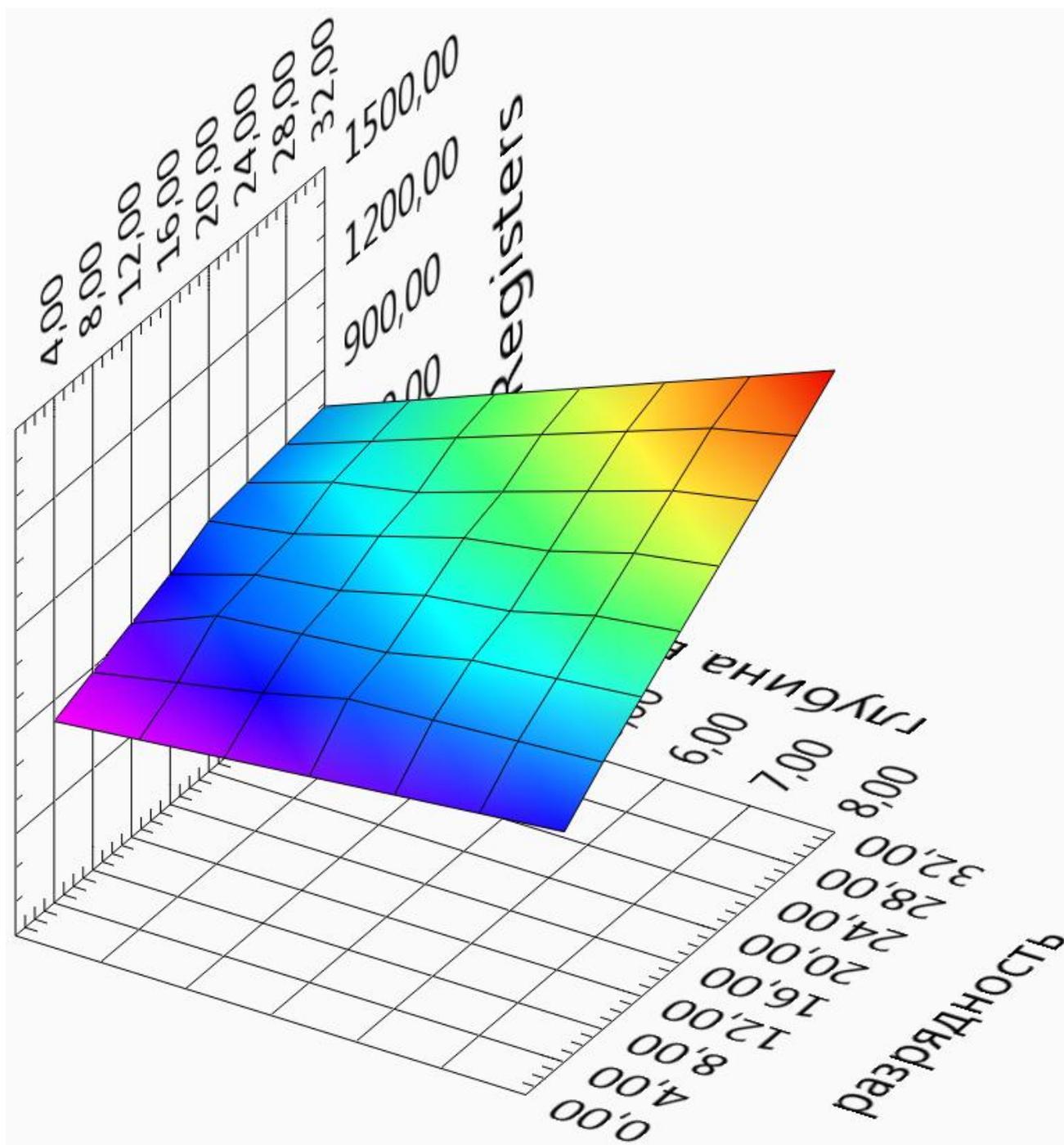


Рис. 3.12. Карта затраченных элементов Registers по экспериментальным данным для библиотеки Numeric, пользовательские настройки компиляции.

Поверхность, полученная на рис.3.12 описывается двумерным полиномом с матричными коэффициентами:

Таблица 3.17. Матричное значение коэффициентов полинома (часть 1)

		Индекс коэффициента В			
		0	1	2	3
Индекс коэффициента А	0	-8,00E+02	1,82E+03	-1,07E+03	3,20E+02
	1	4,55E+02	-6,81E+02	4,16E+02	-1,26E+02
	2	-3,73E+01	5,72E+01	-3,45E+01	1,04E+01
	3	7,92E-01	-1,22E+00	7,31E-01	-2,20E-01

Таблица 3.18. Матричное значение коэффициентов полинома (часть 2)

		Индекс коэффициента В		
		4	5	6
Индекс коэффициента А	0	-5,14E+01	4,25E+00	-1,42E-01
	1	2,04E+01	-1,70E+00	5,69E-02
	2	-1,68E+00	1,39E-01	-4,65E-03
	3	3,54E-02	-2,93E-03	9,75E-05

Полиномиальное восстановление поверхности имеет относительную ошибку по поверхности 0,66% по сравнению с исходными данными.

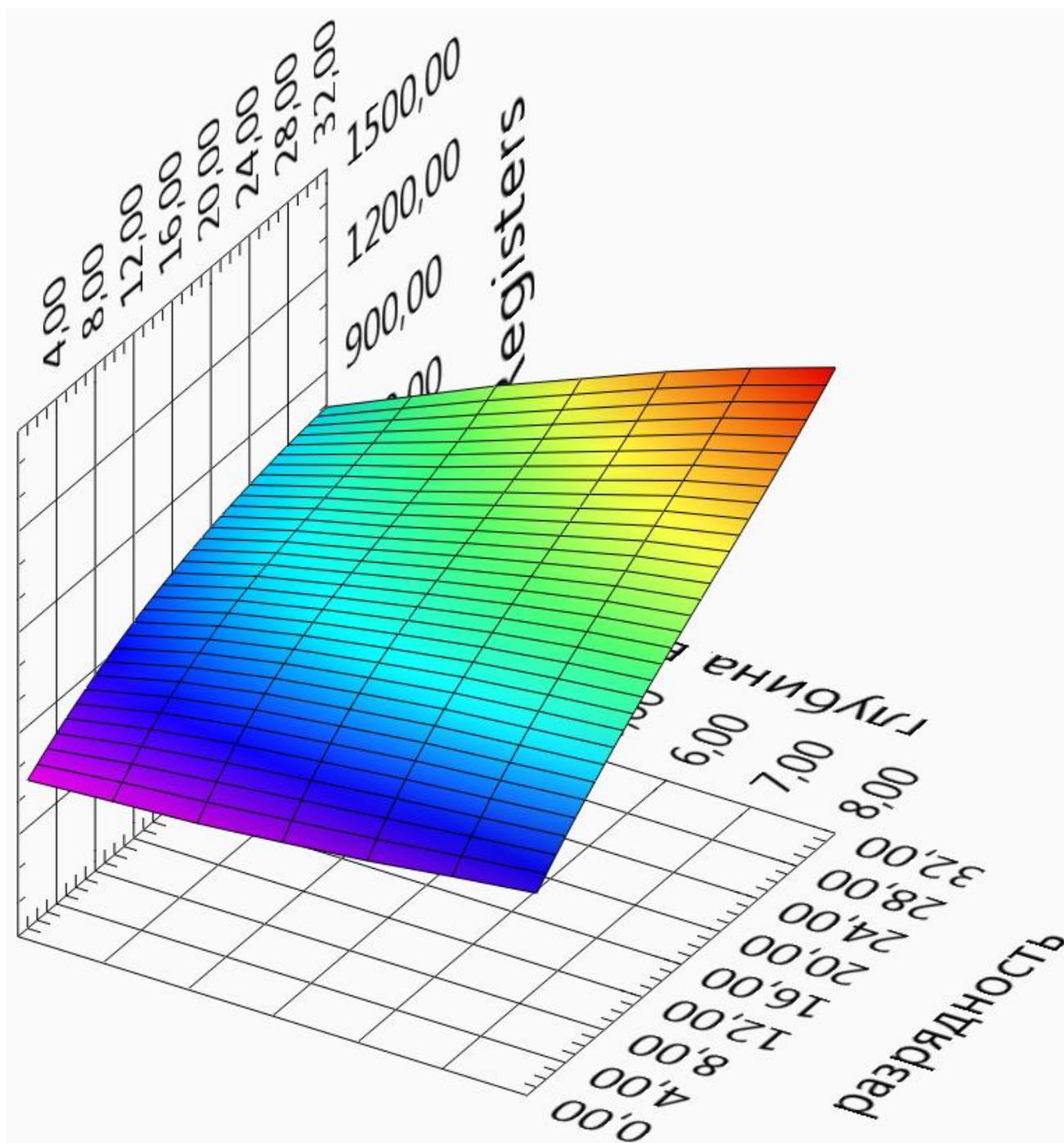


Рис. 3.13. Карта затраченных элементов Registers по экспериментальным данным для библиотеки Numeric, пользовательские настройки компиляции, восстановленная по полиномиальному преобразованию.

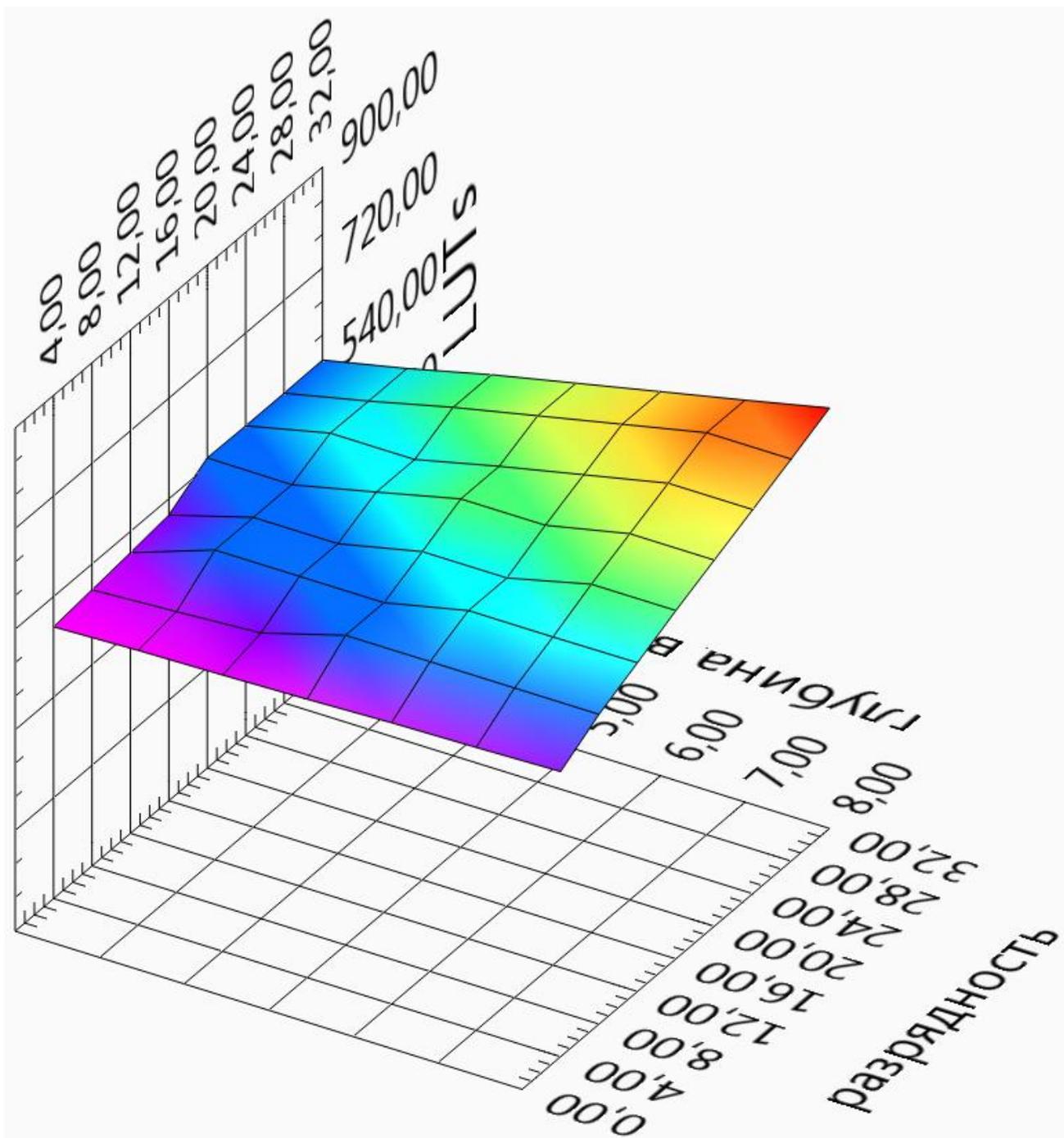


Рис. 3.14. Карта затраченных элементов LUTs по экспериментальным данным для библиотеки Numeric, пользовательские настройки компиляции.

Поверхность, полученная на рис.3.14 описывается двумерным полиномом с матричными коэффициентами:

Таблица 3.19. Матричное значение коэффициентов полинома (часть 1)

		Индекс коэффициента В			
		0	1	2	3
Индекс коэффициента А	0	-4,10E+02	1,24E+03	-7,03E+02	2,01E+02
	1	1,21E+02	-1,78E+02	1,05E+02	-3,07E+01
	2	-6,11E+00	9,25E+00	-5,35E+00	1,54E+00
	3	1,15E-01	-1,73E-01	9,77E-02	-2,76E-02

Таблица 3.20. Матричное значение коэффициентов полинома (часть 2)

		Индекс коэффициента В		
		4	5	6
Индекс коэффициента А	0	-3,12E+01	2,51E+00	-8,16E-02
	1	4,89E+00	-4,08E-01	1,39E-02
	2	-2,46E-01	2,07E-02	-7,24E-04
	3	4,33E-03	-3,64E-04	1,28E-05

Полиномиальное восстановление поверхности имеет относительную ошибку по поверхности 0,898% по сравнению с исходными данными.

На основании полученных данных о распределении ресурсов, можно сделать вывод, что при компиляции с пользовательскими настройками, описанными выше, распределение ресурсов более равномерное и менее емкое.

Аналогичным образом строятся карты распределения ресурсов для других вариантов реализации типовых математических операций.

Проведенные исследования показали, что описанный вариант наиболее эффективный для реализации операции умножения вектора-строки на вектор-столбец.

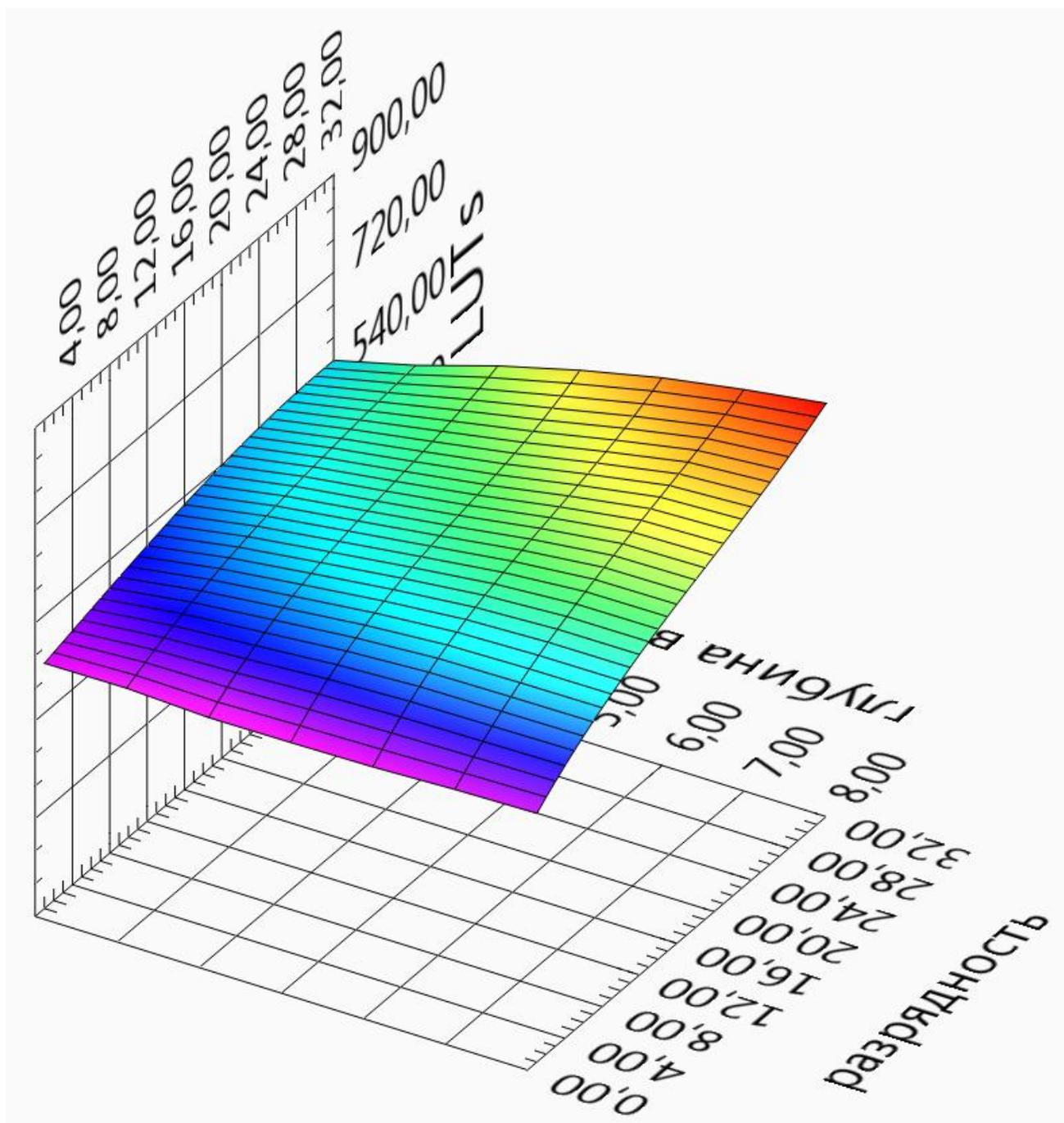


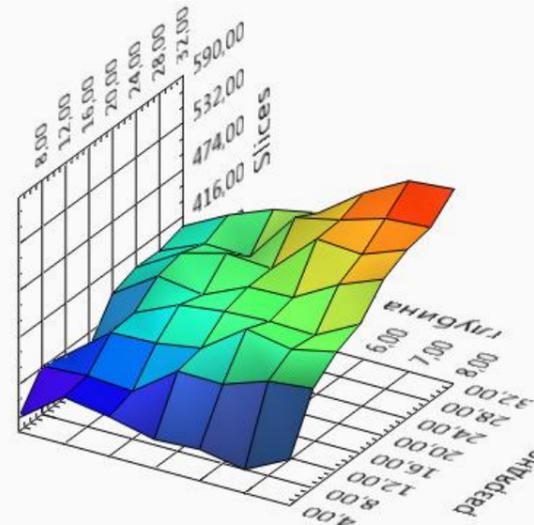
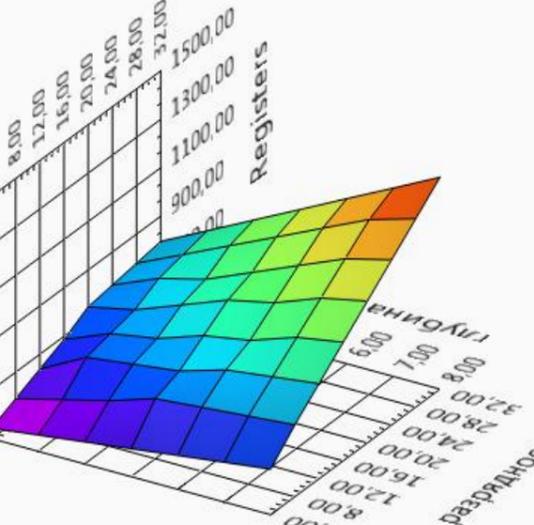
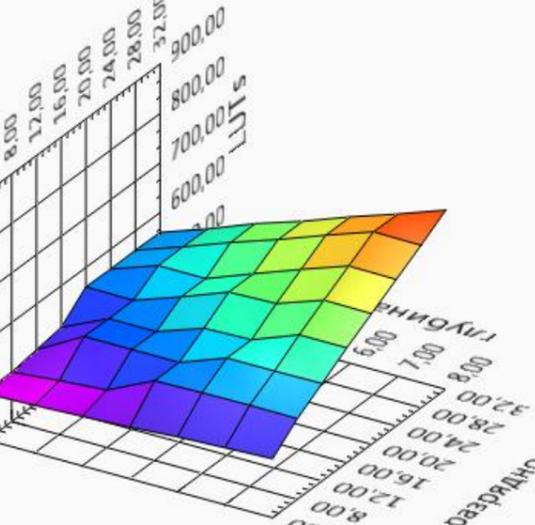
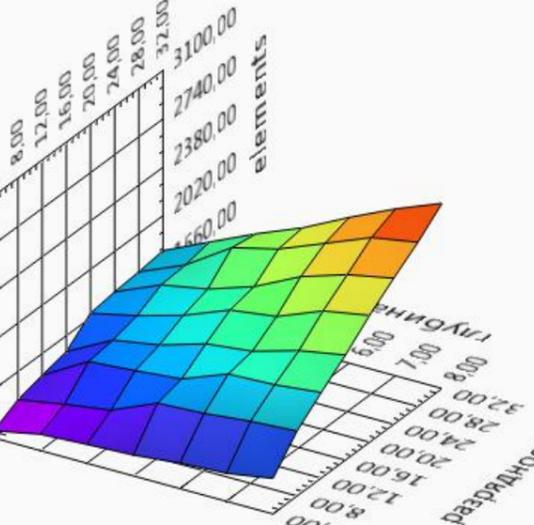
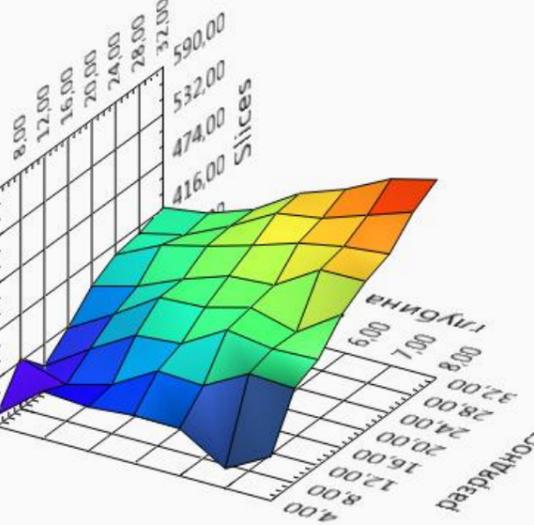
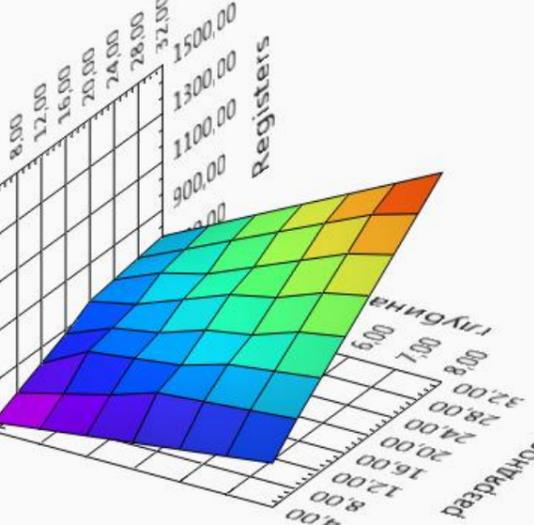
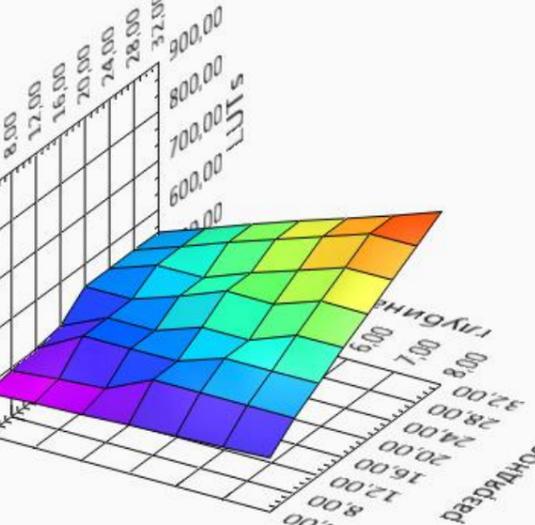
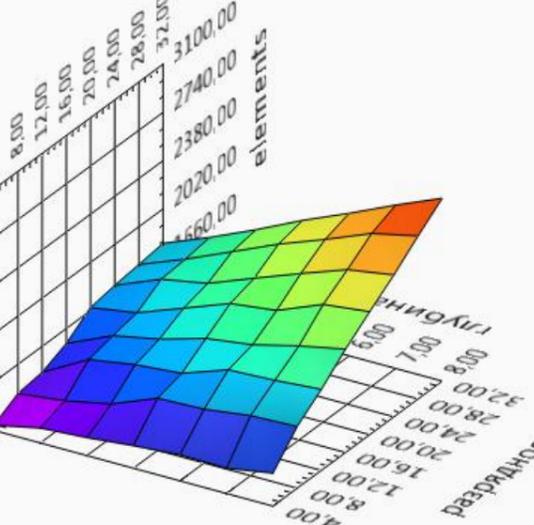
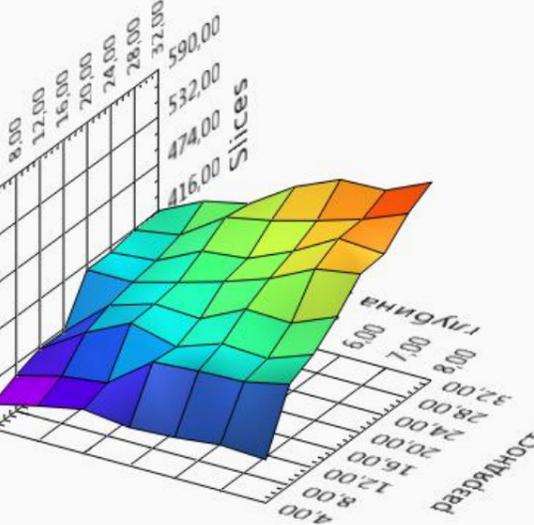
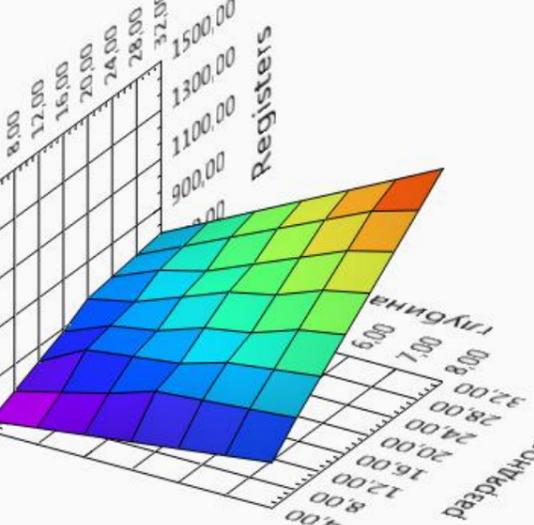
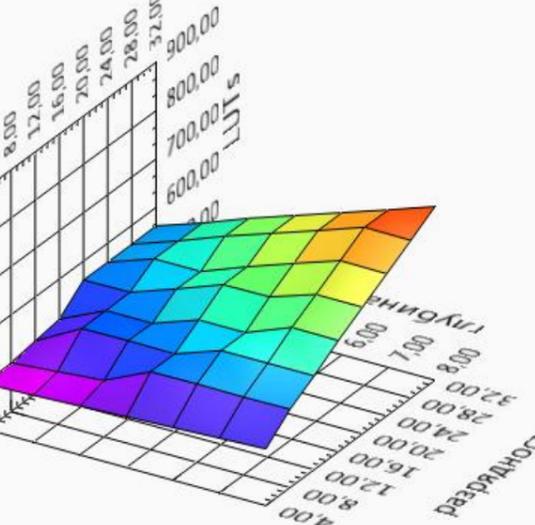
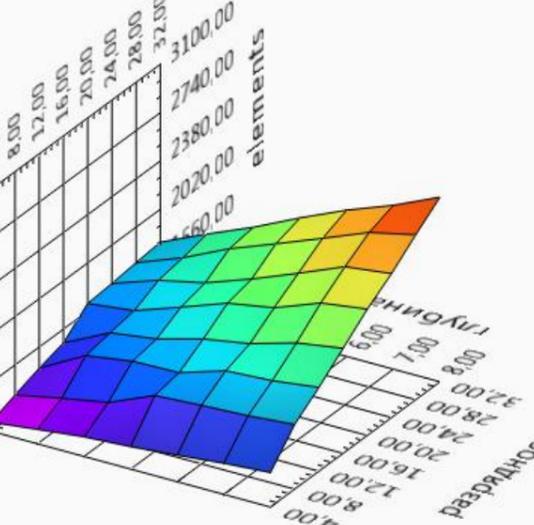
Рис. 3.15. Карта затраченных элементов LUTs по экспериментальным данным для библиотеки Numeric, пользовательские настройки компиляции, восстановленная по полиномиальному преобразованию.

На основании проведенных моделирований и испытаний, с использованием различных библиотек функций, получены следующие результаты

Таблица 3.21. Сравнительная таблица результатов компиляции, с рекомендуемыми настройками

Библиотека	Slices	Registers	LUTs	Elements	Время выполнения, такт
FPGA Math&Analysis					2
Numeric					1
FPGA Hight Throughput Math					2

Таблица 3.22. Сравнительная таблица результатов компиляции, с пользовательскими настройками

Библиотека	Slices	Registers	LUTs	Elements	Время выполнения, такт
FPGA Math&Analysis					2
Numeric					1
FPGA Hight Throughput Math					2

Из приведенных данных, видно, что наилучшим способом, для выполнения операции перемножения вектора-строки на вектор-столбец является библиотека Numeric с компиляцией прошивки ПЛИС с пользовательскими настройками.

Аналогичным образом, рассматриваются варианты реализации типовых операций различными библиотеками функций среды программирования, и находится наилучшее решение.

#### **Выводы по третьей главе:**

1. Проведено преобразование аэродинамических уравнений к дискретной форме записи в формате чисел с фиксированной запятой;
2. На основании анализа аэродинамического уравнения в дискретной форме записи выделены типовые математические операции;
3. Разработана методика реализации ПО ПЛИС для ИИУК в составе ИАВ;
4. Построены карты распределения ресурсов для типовой операции «умножение вектор-строки на вектор-столбец».
5. Проведена оценка ресурсов и рекомендован вариант реализации типовой операции «умножение вектор-строки на вектор-столбец».

## **Глава 4. Реализация ИИУК для решения задач динамической аэроупругости. Оценка на соответствие требованиям к ИИУК.**

### **4.1. Реализация ИИУК для решения задач динамической аэроупругости.**

На основании технического задания № СН1/719ТМ-462-МАИ, ЦАГИ выставлены требования на разработку ИИУК для решения задач динамической аэроупругости.

#### Требования по составу

- В состав входят 4 модуля преобразования аналоговых сигналов;
- Управление всеми каналами –одновременное;
- Изменение общих коэффициентов модулей: - ручное - пошаговое, изменением значений с помощью клавиатуры, джойстика или мыши, или – программное. Общие («основные») коэффициенты – безразмерные, их значение - от 0 до 1

#### Технические требования на модуль:

- Число входных основных преобразуемых напряжений – 4;
- Число входных дополнительных напряжений – 8;
- Число выходных напряжений – 4;
- Амплитуда входных и выходных аналоговых напряжений – 5 В;
- Разрядность АЦП и ЦАП каждого напряжения – 16 бит;
- Частота преобразования АЦП и ЦАП каждого напряжения – не менее  $100 \cdot 10^3$  1/с;
- Диапазон исследуемых частот - 1 - 300 Гц;
- Суммарная погрешность преобразования гармонических сигналов по модулю - 2%;
- Суммарная погрешность преобразования гармонических сигналов по фазе – 2 град;

- Источнк входных основных преобразуемых напряжений – выходы усилителей акселерометров

Процедура обработки данных, поступающих с акселерометра, для одного канала представлена на схеме ниже.

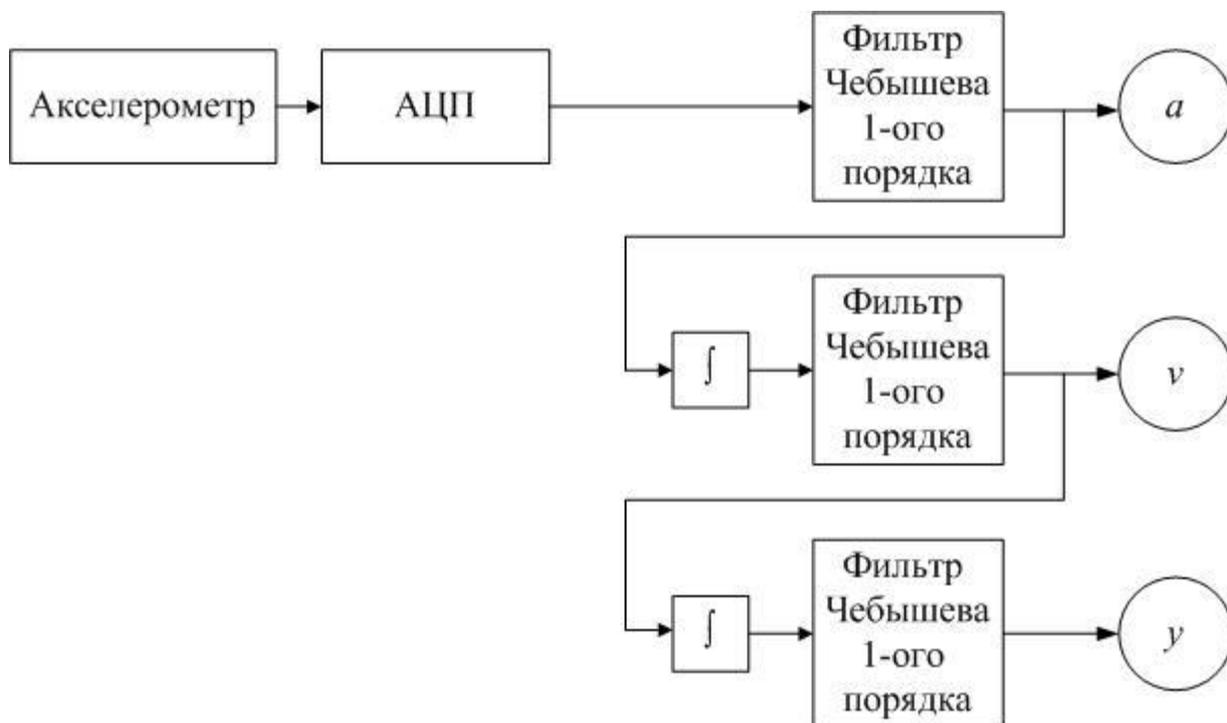


Рис. 4.1. Схема обработки данных для одного канала системы.

Фильтрацию сигналов необходимо производить вследствие накопления ошибки показаний акселерометров и интегралов.

На основании данных технических требований изготовлен макетный образец на базе оборудования National Instruments, а именно:

1. *Nl PXIe 8100 Atom D410 1.66 GHz Real Time SW*

Промышленный контроллер формата PXI Express осуществляет управление модулями ввода/вывода сигналов, установленными в шасси, по внутренним шинам обмена данными (PXI Express, 1 ГБ/сек) и синхронизации и работает под управлением ОСРВ.

- Шина PXI Express;
- Центральный процессор 1.66 ГГц Intel Atom D410;
- ОЗУ 512 МБ DDR2, 800 МГц;
- Размер жесткого диска 512 МБ;
- Наличие Ethernet 10/100/1000;
- Наличие 2 портов USB;
- Наличие порта RS-232;
- Наличие LPT порта;
- Пропускная способность шины данных PXI Express – 1 ГБ/сек.
- Размер – 2 слот в шасси PXI Express высотой 3U.

2. *Nl PXI-7854R Virtex-5 LX110 R Series Multifunction RIO Module (8 AI. 8 AO. 96 DIO) 750kS/sec for LabVIEW FPGA (2 модуля)*

Многофункциональный реконфигурируемый модуль ввода/вывода аналоговых и дискретных сигналов на базе ПЛИС формата PXI имеет возможность установки в шасси PXI и характеристики не хуже ниже перечисленных:

- Шина PXI;
- Количество входных аналоговых каналов – 8 с общим проводом или дифференциальных;
- Режим опроса аналоговых каналов – синхронный;
- Частота дискретизации АЦП – 750 кГц/канал;
- Разрешение АЦП – 16 бит;

- Входной диапазон сигналов –  $\pm 10$  В;
- Количество выходных аналоговых каналов – 8;
- Частота дискретизации ЦАП – 1 МГц;
- Разрешение ЦАП – 16 бит;
- Выходной диапазон сигналов –  $\pm 10$  В;
- Количество дискретных каналов ввода/вывода – 96;
- Уровень логических сигналов – TTL;
- Тактирование дискретных каналов – аппаратное до 40 МГц;
- Стабильность опорного генератора тактовой частоты – 100 ppm;
- Управление вводом/выводом – аппаратно-программное на базе ПЛИС;
- ПЛИС Xilinx Virtex-5 LX110;
- Поддержка режима обмена данными по каналам DMA;
- Встроенная память – 512 кБ;
- Возможность реализации собственных алгоритмов цифровой обработки сигналов, протоколов обмена данными, специализированных схем запуска, счетчиков/таймеров и т.п. на базе ПЛИС с временным разрешением 25 нс.
- Размер – 1 слот в шасси PXI высотой 3U;
- Совместимость с программным драйвером NI-RIO.

### 3. NI PXIe 1078. 9 Slot 3U PXI Express Chassis

Промышленный корпус формата “Евромеханика” для размещения 9 модулей ввода-вывода сигналов в формате PXI Express и/или PXI.

- 3 слота PXI Express высотой 3U;
- 5 гибридных слота типа PXI/PXI Express высотой 3U;
- Пропускная способность 1 ГГб. в сек.
- Наличие встроенной шины синхронизации модулей.
- Питание переменным током до 240 В, 50 Гц;

В итоге, ИИУК представляет собой трехуровневую систему (Рис.4.2), во главе которой находится блок ввода-вывода данных через человеко-машинный интерфейс, располагаемом на ПК.

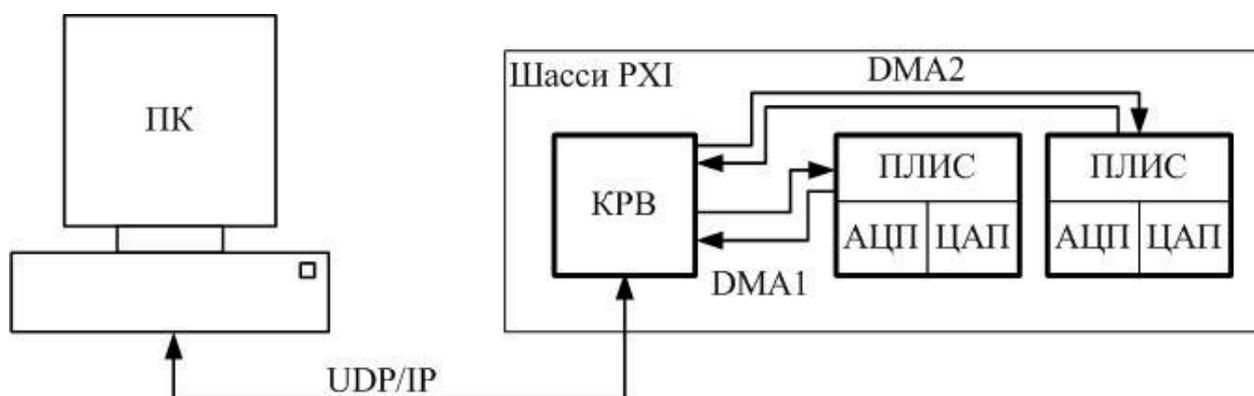


Рис. 4.2. Структурная схема включения оборудования в системе СБФАВ.

В самом низу иерархии системы располагаются модули ЦОС с вычислителями, реализованными на ПЛИС, и модулями АЦП/ЦАП. Промежуточным звеном системы является шасси с шиной PXI с установленным процессором, на котором работает операционная система реального времени (ОСРВ). Для простоты, процессор с установленной ОСРВ будем называть «контроллером реального времени» (КРВ).

### Трансляция данных

Передача данных «вверх» и «вниз» занимает очень важное место в системе, поскольку ИИУК должен отображать на графическом интерфейсе пользователя различные диаграммы состояния и графики, получаемые в процессе проведения эксперимента.

Разработка принципа передачи данных для данной системы достаточно тонкая задача, поскольку число передаваемых параметров высоко.

Сетевая передача данных делится на 2 участка:

- ПК ↔ КРВ;
- КРВ ↔ ПЛИС.

ПЛИС работает на частоте 40 МГц, а оцифровка и выдача сигнала идет с частотой 750кГц. Т.е. получение данных происходит в 53 раза медленнее.

Поэтому необходимо использовать в программе задержки по времени. Кроме того, несмотря на то, что КРВ работает с частотой более 1 ГГц, обработать всю информацию в режиме реального времени, по мере поступления данных от АЦП, не удастся. Поэтому необходимо использовать буфер DMA (англ.: Direct Memory Access; русск.: прямой доступ к памяти) в режиме FIFO (англ.: First input –First output; русск.: очередь). ПЛИС работает с DMA в одноканальном режиме.

При передаче с ПЛИС на КРВ необходимо передавать данные с всех каналов ( по 4 значения на канал – ускорение, скорость, перемещение, сигнал силового возбуждения). Все данные имеют разрядность 16 бит.

Без организации специализированного протокола обмена данными между устройствами, поток данных телеметрии (значения ускорений, скоростей, перемещений и численных значений силовых значений) с одного модуля составляет порядка 24,41 Мбит/сек, при отправке данных по мере их поступления, т.е. с частотой 100 кГц. Суммарно с четырех модулей, поток данных телеметрии составляет 97,66 Мбит/сек, без учета передаваемых данных о системе, данных об аэродинамических матрицах и т.п.

При работе ПЛИС, накопление данных идет быстрее, чем может обработать КРВ, поэтому необходимо организовывать передачу через буфер DMA выборкой с более низкой частотой.

Параллельно из КРВ на ПЛИС должны передаваться матрицы аэродинамических коэффициентов, характеризующие виртуальный воздушный поток. Частота передачи данных коэффициентов достаточно низка, по сравнению с передачей данных из ПЛИС в КРВ. В процессе проведения экспериментов было решено передавать данные из ПЛИС с частотой 10 кГц, что соответствует 30 точкам на период от сигнала частотой в 300 Гц.

Так как архитектура ПЛИС не позволяет организовать обращение к данным по указателям, как в архитектуре процессоров с последовательным исполнением кода (таких, как архитектура Фон-Неймана), то, наиболее эффективным является обращение к временным хранилищам данных, организованным в виде блоков памяти с настроенным типом данных.

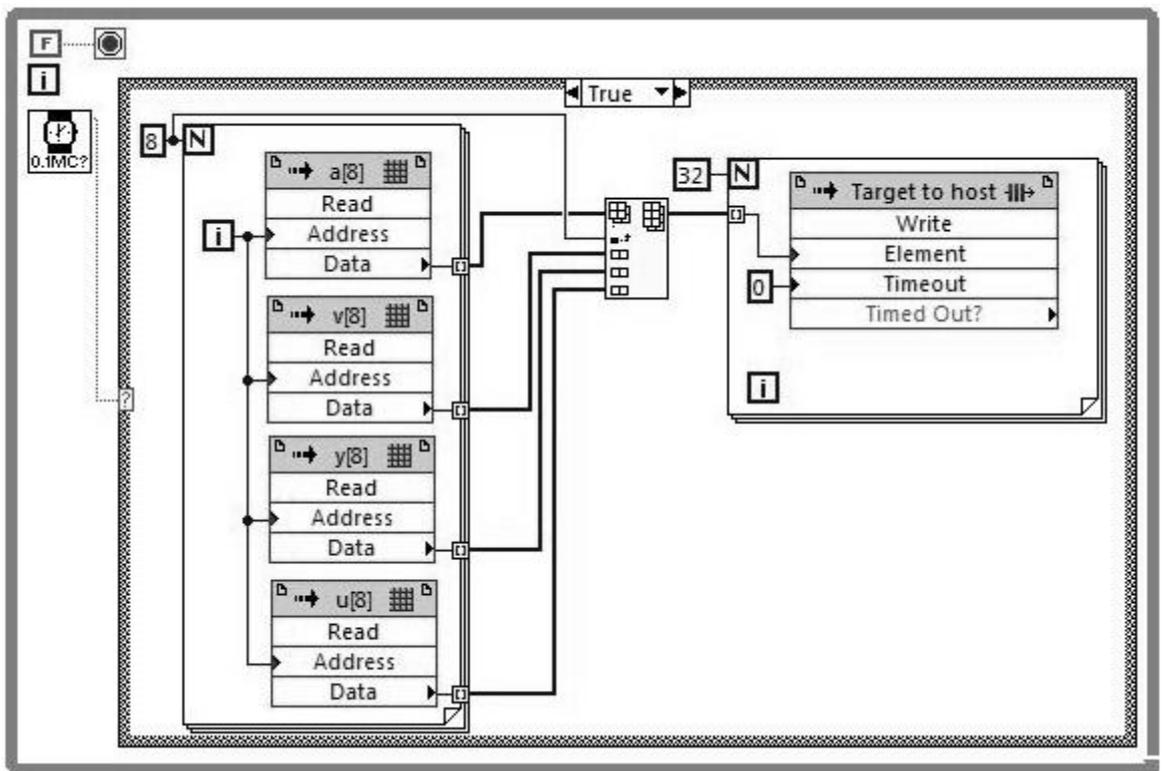


Рис. 4.3. Тестовая программа передачи данных с ПЛИС в КРВ.

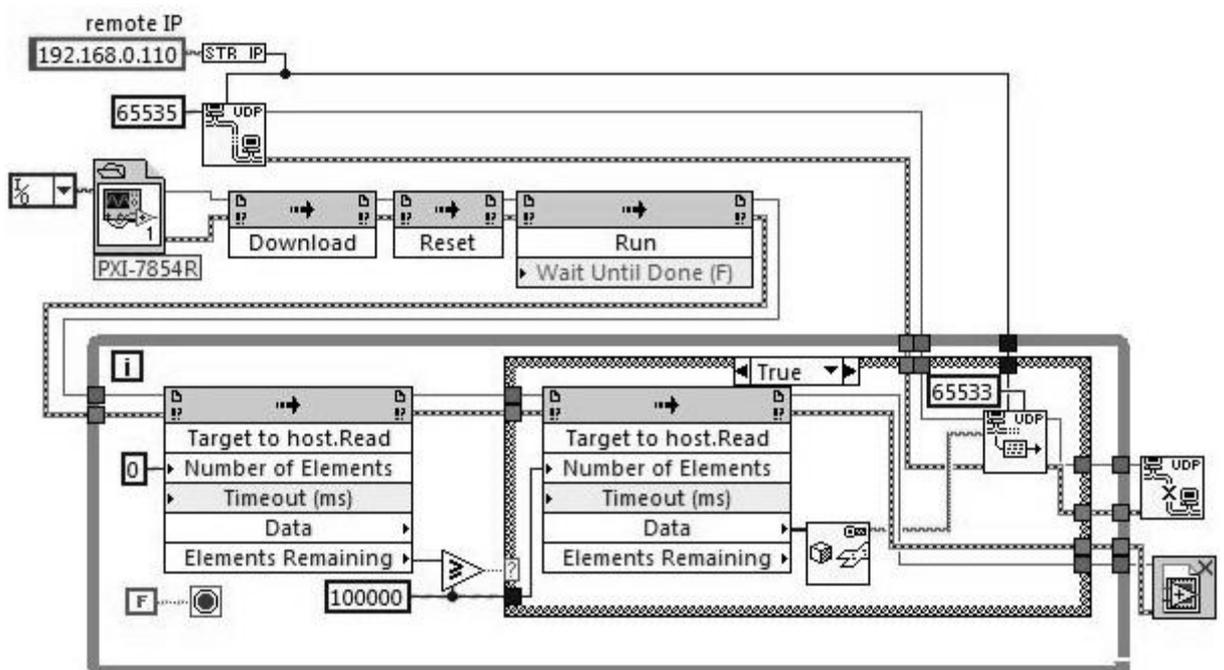


Рис. 4.4. Тестовая программа для КРВ – Прием данных из ПЛИС и ретрансляция их на ПК.

Таким образом реализован алгоритм обработки и передачи информации одновременно по всем каналам для ИИУК с использованием оборудования National Instruments. [16]

### **Разработаны следующие программные средства:**

- Программа управления специальным блоком и отображения измерительной информации (интерфейс оператора)
- Программа преобразования аналоговых сигналов с целью воспроизведения аэродинамических соотношений между комбинацией ускорений (сигналов датчиков) и мгновенных значений сил (входных напряжений усилителей мощности)

### **Программа управления специальным блоком и отображения измерительной информации (интерфейс оператора)**

Согласно требованиям технического задания № СН1/719ТМ-462-МАИ на разработку и изготовление ИИУК программа управления специальным блоком и отображения его состояния должна обеспечивать установку и изменение режима испытаний с помощью клавиатуры ПК или по заданной программе, отображение информации на экране монитора - табличное и графическое.

Ниже представлены примеры окон разработанного предлагаемого интерфейса оператора программы управления специальным блоком и отображения измерительной информации.



Дальнейшая оптимизация интерфейса оператора будет проводиться в последующих этапах проекта в процессе согласования с заказчиком.

**Программа преобразования аналоговых сигналов с целью воспроизведения аэродинамических соотношений между комбинацией ускорений (сигналов датчиков) и мгновенных значений сил (выходных напряжений усилителей мощности)**

Программа расчета выходных сигналов проводит преобразования по следующей формуле.

$$u_i = [(\sum_j^8 b_{ij}^* y_j^* + \sum_j^8 d_{ij}^* \dot{y}_j^* + c_i^* \ddot{y}_i^*) 2^{-15}] \quad (4.1)$$

где:

$u_i^*$  – сигналы для силовозбудителей

$b_{ij}^*$  – коэффициенты преобразования перемещений элементов конструкции в сигналы силовозбудителя

$d_{ij}^*$  – коэффициенты преобразования скоростей элементов конструкции в сигналы силовозбудителя

$c_i^*$  – коэффициенты преобразования ускорений элементов конструкции в сигналы силовозбудителя

$y_i^*$  – перемещения элементов конструкции

$\dot{y}_i^*$  – скорости элементов конструкции

$\ddot{y}_i^*$  – ускорения элементов конструкции

Квадратные скобки – округление до целых.

Все переменные реализованы в виде 16 разрядных регистров. Результатом умножений является 32 разрядная величина, с погрешностью вычислений  $2^{-16}$ , далее суммируются полученные 32-х разрядные значения. Общая погрешность вычислений будет  $17 \cdot 2^{-16}$  или менее чем  $2^{-11}$ . В результате отбрасывания 16-ти младших разрядов результата суммирования, накопленная погрешность

вычисления не попадает в итоговый результат и не вносит дополнительной вычислительной ошибки в амплитуду выходных сигналов.

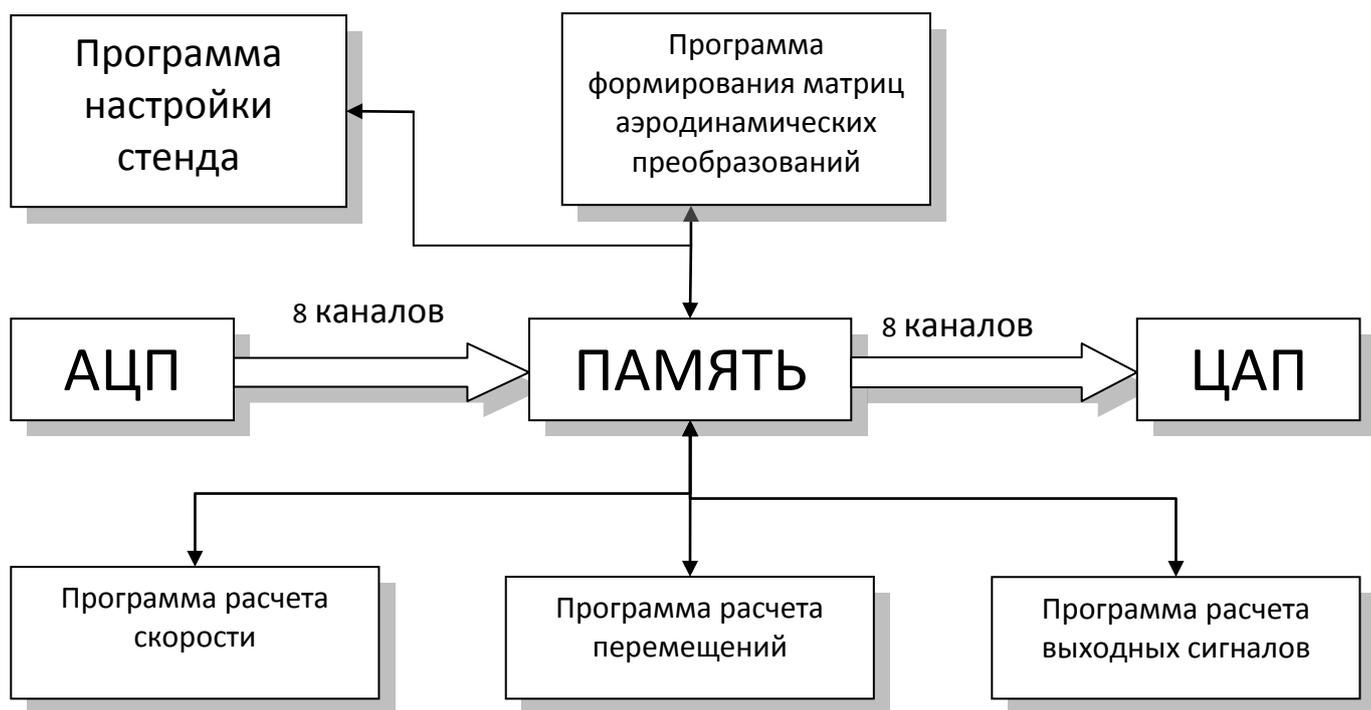


Рис. 4.7. Структурная схема программы преобразования аналоговых сигналов.

## 4.2. Оценка на соответствие требованиям к ИИУК.

Согласно техническому заданию № СН1/719ТМ-462-МАИ на разработку и изготовление специализированного блока формирования аэродинамических воздействий ИИУК предъявляются технические требования:

Число входных каналов (основных преобразуемых напряжений)	- 4 кан.
Число входных каналов (дополнительных напряжений)	- 8 кан.
Число выходных аналоговых каналов (напряжений)	- 4 кан.
Амплитуда входных и выходных аналоговых напряжений	- 5 В
Разрядность АЦП и ЦАП каждого напряжения	- 16 бит
Частота преобразования АЦП и ЦАП каждого напряжения	- $100 \cdot 10^6$ 1/с
Диапазон исследуемых частот	- 1 - 300 Гц.
Суммарная погрешность преобразования гармонических сигналов по модулю	- 2%
Суммарная погрешность преобразования гармонических сигналов по фазе	- 2 град

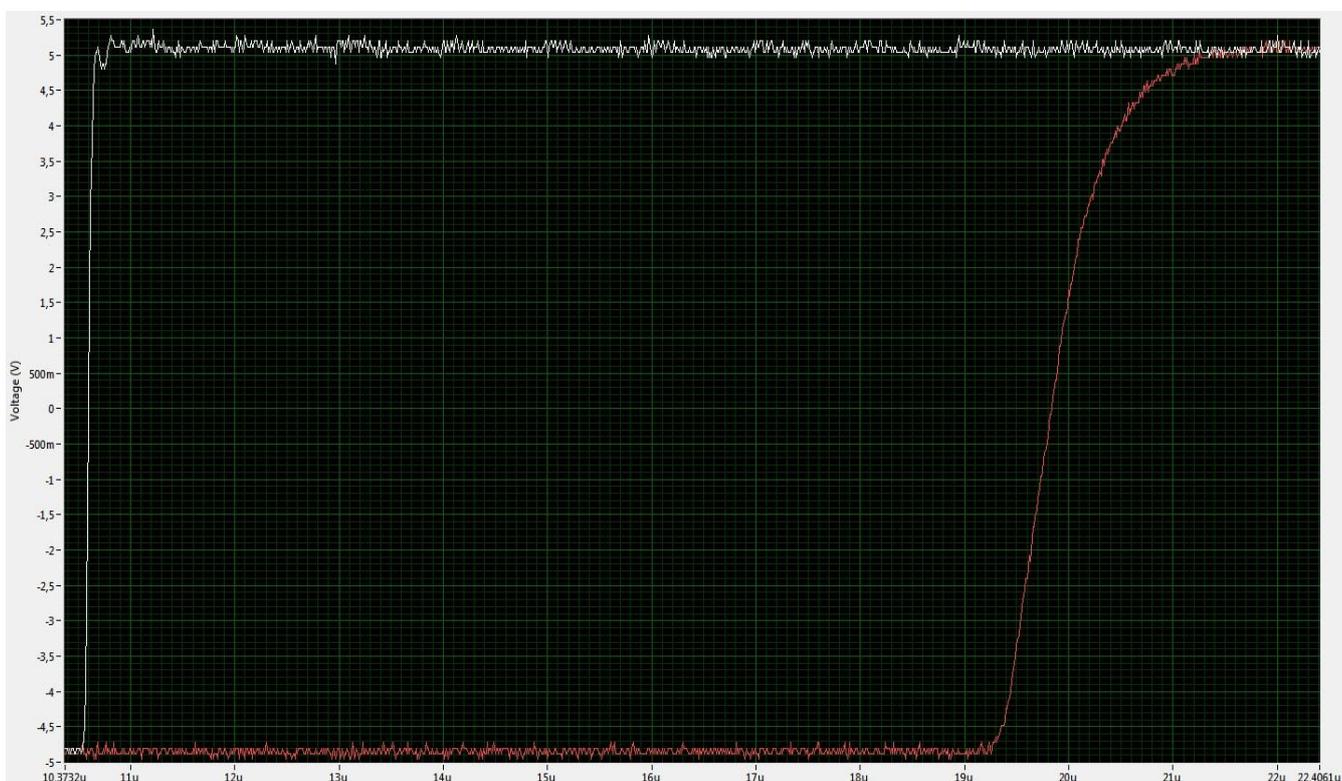


Рис. 4.8. Задержка системы при подаче ступенчатого воздействия.

Испытание №1: На входной канал модуля было подан ступенчатый сигнал для удобства визуального контроля суммарной задержки АЦП, программы формирования управляющего воздействия и ЦАП. Как видно по рис.3.22, задержка составила 11 микросекунд.

Исходя из максимальной частоты входного сигнала 300 Гц и максимальной фазовой задержки 2 градуса получаем что максимальная задержка при формировании управляющего воздействия равна  $1/(300*(360/2))$  или 18,519 микросекунд.

Полученная задержка укладывается в требуемую с запасом 7,5 микросекунд.

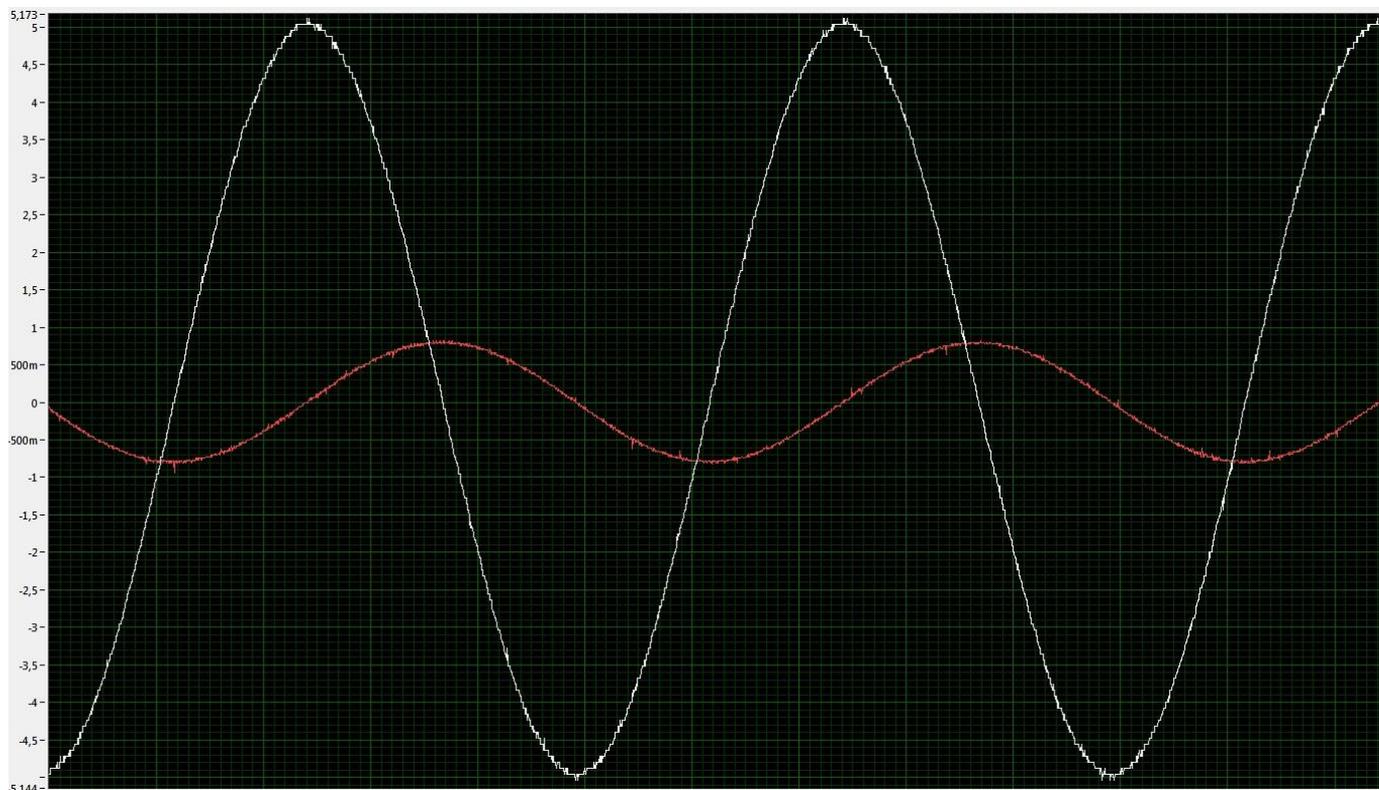


Рис. 4.9. Суммарная погрешность преобразования гармонических сигналов по модулю.

Испытание №2: На входной канал модуля было подан синусоидальный сигнал амплитудой 10В (белый график) частотой 2 герца. Красный график – интеграл входного сигнала.

$$\int A \sin(\omega t) dt = -\frac{A}{\omega} \cos(\omega t),$$
$$\omega = 2\pi f;$$
(4.2)

где  $A$  – амплитуда,  $f$  – частота.

Исходя из соотношения (1) получаем, что при частоте  $f$  равной 2 и амплитуде равной 10,  $\frac{A}{\omega}$  равно 795,774 мВ. Измеренное значение равно 790 мВ. Суммарная относительная погрешность преобразования составляет 0,725%.

### **Выводы по четвертой главе:**

1. Реализован ИИУК для ИАВ для решения задач динамической аэроупругости;
2. Проведена оценка ИИУК на соответствие техническим требованиям. Показано, что ИИУК для ИАВ соответствует требованиям по быстродействию, точности, а архитектура ИИУК позволяет его масштабировать.

## ЗАКЛЮЧЕНИЕ

### Основные результаты работы:

1. Проведен анализ и разработана архитектура специализированного быстродействующего трехуровневого ИИУК, построенного на базе открытых международных модульных структурах PXI, с использованием операционных систем жесткого реального времени и структур «АЦП-ПЛИС-ЦАП» для ИАВ, обеспечивающая реализацию параллельного выполнения алгоритмов сбора данных с датчиков (акселерометров), обработки и выдачи управляющих воздействий (по каналам).
2. Разработан алгоритм преобразования аэродинамических уравнений ИАВ из аналоговой формы в цифровую форму, с описанием в числах с фиксированной запятой (заданной разрядности), с выбором типовых арифметических операций для эффективной по времени реализации на ПЛИС.
3. Разработана математическая модель для оптимизации ресурсов ПЛИС, позволяющая связать скорость выполнения программного кода, количество ресурсов ПЛИС, требуемых для его реализации и точности вычислений (целочисленный формат 8-64 разряда и числа с фиксированной запятой). Математическая модель построена для выбранных типовых арифметических операций и позволяет при заданных скорости выполнения программного кода и точности вычислений определить необходимые ресурсы ПЛИС.
4. Разработана методика проектирования ПО специализированного ИИУК, с использованием структур «АЦП-ПЛИС-ЦАП», обеспечивающая возможность управления выбором затрачиваемых ресурсов (емкость ПЛИС), времени исполнения кода ПЛИС (быстродействие) и точности вычислений (разрядность данных).

5. Разработано и протестировано в составе ИАВ программного обеспечения верхнего и нижнего уровня специализированного ИИУК на основе разработанной методики для решения задач динамической аэроупругости.
6. Разработан алгоритм применения созданной методики с целью обеспечения эффективности выбора технических параметров АПК на базе ПЛИС для реализации перспективных многоканальных измерительно-управляющих систем нового поколения.
7. Разработан лабораторный образец специализированного ИИУК на базе технологий LabView, который позволил экспериментально подтвердить заданные технические характеристики, соответствующие современным требованиям по быстродействию, многоканальности измерений (сигналы от датчиков-акселерометров) и управления (сигналы управления на силовозбудители - физические имитаторы аэродинамических воздействий) для ИАВ.
8. Разработан и реализован лабораторный стенд полунатурного моделирования (имитации) аэродинамических воздействий, реализующий метод электромеханического моделирования в составе: усилители мощности, силовозбудители, датчики-акселерометры, установленные на модель крыла и специализированный ИИУК. Проведена экспериментальная оценка характеристик разработанного ИИУК в составе ИАВ с моделью конструкции ЛА.

#### Практическая ценность

Значение результатов диссертационной работы для практического применения заключается в следующем:

1. Модифицированная архитектура ИИУК может быть применена при построении многоканальных и высокоскоростных измерительных и управляющих систем;

2. Разработанная методика может применяться при оптимизации математического аппарата при программировании ПЛИС для ИИУК ИАВ;

#### Практическая реализация и внедрение результатов работы

Разработанный ИИУК для ИАВ используется при проведении стендовых полунатурных испытаний в ФГУП ЦАГИ при проведении наземных испытаний ЛА. Разработанная методика реализации программного обеспечения ПЛИС внедрена в учебный процесс кафедры Авиационные приборы и измерительно-вычислительные комплексы МАИ. Факт внедрения подтвержден актами, приложенными к диссертационной работе.

#### Результаты диссертационной работы использовались

Разработанная методика реализации программного обеспечения ПЛИС использовалась при проектировании Универсальной Интегрированной Автоматизированной Системы Измерения и Системы Управления (УИАСИ и СУ) в ОАО НПО Энергомаш для обеспечения стендовых испытаний ракетных двигателей и их агрегатов.

## Список использованной литературы

1. Андреев В.С. Математическое и программное обеспечение систем автоматизации проектирования цифровых систем обработки сигналов: дисс. канд. техн. Наук. – СПб. 2013;
2. Архитектура шины PCI Express [электронный ресурс]. // URL: [http://www.bestor.spb.ru/v3/Overs?o\\_id=1249](http://www.bestor.spb.ru/v3/Overs?o_id=1249) (дата обращения 09.10.2014);
3. Баран Е. Д. LabVIEW FPGA. Реконфигурируемые измерительные и управляющие системы // М.: Издательство ДМК Пресс, 2009г. – 447с.;
4. Белоусов Л.С., Логунов Б.А., Рослов Ю.А.. Управляющий и измерительно-вычислительный комплекс для частотных испытаний конструкций. // Труды Центрального Аэрогидродинамического института им. проф. Н.Е. Жуковского. - Вып. 2738 – М.: ЦАГИ, 2013, с.317-325;
5. Бурдонов И.Б., Косачев А.С., Пономаренко В.Н. Операционные системы реального времени [электронный ресурс]. // URL: [http://citforum.ru/operating\\_systems/rtos/1.shtml](http://citforum.ru/operating_systems/rtos/1.shtml). Препринт Института системного программирования РАН (дата посещения 08.10.2014);
6. Бутусов Д. Н., Андреев В.С. Модельное проектирование генератора гармонических сигналов на базе ПЛИС // «Известия СПбГЭТУ «ЛЭТИ» - 2011. - № 10. - С. 55-60;
7. Бутусов Д.Н. Автоматизация проектирования встраиваемых систем: дисс. канд. техн. наук – СПб. 2012;
8. Быков А.В., Парафесь С.Г., Смыслов В.И. Программно-аппаратный комплекс для проведения расчетно-экспериментальных исследований аэроупругой устойчивости летательных аппаратов. // Вестник МАИ, 2009, т.16, №5, стр. 56-63;
9. Быков А.В., Смыслов В.И. Задача о флаттере маневренных летательных аппаратов с учетом его колебаний в двух плоскостях // Ученые записки ЦАГИ. 2011. Т. XLII. №3. С.92-100;

10. Галюченко А.М., Кувшинов В.М. Особенности взаимодействия цифровой системы управления с упругими колебаниями конструкции самолета // Труды ЦАГИ. 1990. Вып. 2466. 39с;
11. Головастов А. PXI Express: замена игрока или пополнение в команде PXI [электронный ресурс]. // URL: [www.cta.ru/cms/f/447014.pdf](http://www.cta.ru/cms/f/447014.pdf) (дата обращения 09.10.2014);
12. Головастов А.. CompactPCI и PXI: не соревнуйся, а дополняя друг друга [электронный ресурс]. // URL: [www.cta.ru/cms/f/385018.pdf](http://www.cta.ru/cms/f/385018.pdf) (дата обращения 09.10.2014);
13. Дайперт Б. Обзор приборов программируемой логики // Электронные компоненты, 2005, №№ 2, 3;
14. Дмитриев С.Н., Хамидуллин Р.К. Коррекция матрицы демпфирования с использованием экспериментальных значений коэффициентов модального демпфирования // Инженерный журнал: Наука и инновации. 2013. - №3(15), с.1-12;
15. Дьяков И. А. Моделирование цифровых и микропроцессорных систем. Язык VHDL. // Учебное пособие. - Тамбов: Изд-во ТГТУ, 2001. - 68 с;
16. Егоров А.А., Константинов А.А., Бодунков Н.Е. Разработка алгоритмов и программного обеспечения специализированного блока формирования аэродинамических воздействий для электромеханического моделирования в задачах аэроупругости. // Приборы и системы. Управление. Контроль. Диагностика. 2014. №2, с.33-38;
17. Егоров А.А., Резник Ю.О. Объектно-ориентированный анализ и проектирование нового поколения интеллектуальных приборных комплексов для отработки аэрокосмических технологий. // Приборы и системы. Управление. Контроль. Диагностика. 2000. №8;
18. Жаров. Е.А., Смыслов В.И. Оборудование для наземных резонансных испытаний с многоточечным возбуждением // Авиационная промышленность. 1972. №9, с.100-103;

19. Каплун Д.. Метод октавной цифровой фильтрации с сокращением объема вычислений на ПЛИС [электронный ресурс]. // URL: [http://www.kite.ru/articles/plis/2008\\_09\\_131.php](http://www.kite.ru/articles/plis/2008_09_131.php) (дата обращения 15.10.2014);
20. Карклэ П.Г., Малютин В.А., Мамедов О.С. и другие. О современных методиках наземных испытаний самолетов в аэроупругости. // Труды Центрального Аэрогидродинамического института им. проф. Н.Е. Жуковского. - Вып. 2708 – М.: ЦАГИ, 2012. – 35с.;
21. Карклэ П.Г., Нарижный А.Г., Смыслов В.И. Стендовое исследование случайных колебаний самолета при электромеханическом моделировании аэродинамических сил на разных режимах полета // Ученые записки ЦАГИ. 1998. Т. XXIX, №1-2;
22. Карклэ П.Г., Пронин М.А., Смыслов В.И. Стенд для исследований флаттера упругой демонстрационной модели при воспроизведении аэродинамических сил // Труды ЦАГИ. 2012. Вып. 2706. С.1-12;
23. Карклэ П.Г., Смыслов В.И. Электромеханическое моделирование в задачах аэроупругости // Полет. – 2008. №10. С.25-31;
24. Комаров В.А. Синтез многопользовательских распределенных измерительно-управляющих систем: автореф. канд. техн. наук – Красноярск. 2009;
25. Константинов А.А. Методика реализации уравнений имитатора аэродинамических воздействий на программируемой логической интегральной схеме [электронный ресурс]. // Труды МАИ. 2014. №77. URL: <http://www.mai.ru/science/trudy/published.php>
26. Локальная сеть и Ethernet [электронный ресурс]. // URL: [http://www.axis.com/ru/products/video/about\\_networkvideo/ip\\_networks.htm](http://www.axis.com/ru/products/video/about_networkvideo/ip_networks.htm) (дата посещения 09.10.2014);
27. Мультиклеточный процессор – это что [электронный ресурс]. // URL: <http://habrahabr.ru/post/226773/> (дата обращения 12.10.2014);
28. Нарижный А.Г., Педора А.П., Смыслов В.И. Вибрационные испытания с воспроизведением воздействий потока при исследованиях аэроупругости на динамически подобных моделях // Уч. Записки ЦАГИ. Т. XXXII. № 1-2. 2001;

29. Особенности организации и использование FireWire [электронный ресурс]. // URL: [ats/C325729F00717F7B43257B0B00099104/Особенности организации и использование FireWire.doc.html](ats/C325729F00717F7B43257B0B00099104/Особенности_организации_и_использование_FireWire.doc.html) (дата обращения 09.10.2014);
30. Парафесь С.Г., Смыслов В.И. Методы и средства обеспечения аэроупругой устойчивости беспилотных летательных аппаратов // М.: Издательство МАИ, 2013. – 176с.;
31. Педора А.П., Смыслов В.И. Исследование аэроупругой устойчивости маневренных беспилотных летательных аппаратов // Тр. ЦАГИ. 2005, Вып. 2669.;
32. Писарук Н. Н. Кратчайшие пути в графах [электронный ресурс]. // URL: <http://pisaruk.narod.ru/slides/shortPath.pdf> (дата посещения 29.09.14);
33. Попович А. Применение технологии разработки «систем на кристалле» на основе ПЛИС // Компоненты и технологии, 2004, №4, с.114-116;
34. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. М.: Мир. 1978. 848с.;
35. Сайт компании Data Physics. [электронный ресурс]. // URL: [www.dataphysics.com](http://www.dataphysics.com) (дата посещения 06.10.2014);
36. Сайт компании National Instruments [электронный ресурс]. // URL: [www.ni.com](http://www.ni.com) (дата посещения 06.10.2014);
37. Смыслов В.И. К истории модальных испытаний ЛА в ЦАГИ. // Труды Центрального Аэрогидродинамического института им. проф. Н.Е. Жуковского. - Вып. 2738 – М.: ЦАГИ, 2013. С.245-255.;
38. Смыслов В.И. Решение задач динамической аэроупругости методами электромеханического моделирования. // Труды ЦАГИ. 1983. Вып. 2200. С.42-50;
39. Смыслов В.И. Четырехканальное оборудование для наземных резонансных испытаний // Авиационная промышленность. 1968. №4, с.94-97;
40. Смыслов В.И., Васильев К.И., Грузов Ю.А. Основные характеристики четырехканального оборудования для динамических испытаний АВДИ-1А // Сб. «Вибрационная техника». – М.: МДНТП, 1986, с.127 -130;

41. Смыслов В.И., Васильев К.И., Грузов Ю.А., Язвин В.М. Многоканальное оборудование для определения динамических характеристик механических конструкций. // Приборы и системы управления. 1979. №7.;
42. Смыслов В.И., Васильев К.И., Язвин В.М. Оборудование для многоточечного возбуждения колебаний при динамических испытаниях // Приборы и системы управления. 1976. №10.;
43. Смыслов В.И., Нарижный А.Г., Педора А.П. Моделирование трубных исследований флаттера при стендовых вибрационных испытаниях с воспроизведением аэродинамических сил. // ТВФ. 1995. Т.LXIX. №5-6. с.49-56.;
44. Смыслов В.И., Стрелков С.П. Электромеханическое моделирование флаттера // Аннотации докл. Второго всесоюз. Съезда по теоретической и прикладной механике. М.: Наука, 1964.;
45. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б., Гук И. И. Основы цифровой обработки сигналов: Курс лекций. СПб.: БХВ-Петербург, 2003. с.733.;
46. Стандарт IEEE 1394 (технология FireWire) [электронный ресурс]. // URL: <http://cs.usu.edu.ru/home/vitalik/computers/hardware/data/ieee.htm> (дата обращения 09.10.2014);
47. Стенд электромеханического моделирования аэродинамических сил для исследования аэроупругого поведения ЛА [электронный ресурс]. // URL: [http://www.mai.ru/science/equipment/index.php?SECTION\\_ID=1290&ELEMENT\\_ID=30767](http://www.mai.ru/science/equipment/index.php?SECTION_ID=1290&ELEMENT_ID=30767) (дата обращения: 07.10.2014);
48. Федосов В.П. Цифровая обработка звуковых и вибросигналов в LabVIEW (+DVD-ROM). – М.: ДМК Пресс. 2009. 1296с.;
49. Шагурин И., Шалтырев В., Волон А. «Большие» FPGA как элементная база для реализации систем на кристалле [электронный ресурс].//URL: <ftp://77.47.129.53/pub/journals/EK/2006/EK/PDF/05/05.083-88.pdf> (дата обращения 15.10.2014);

50. Шпаргалка по типам и стандартам Ethernet 802.3 [электронный ресурс]. // URL: <http://habrahabr.ru/post/208202/> (дата посещения 09.10.2014)
51. Antoniou A. Digital Signal Processing. Signals, Systems and Filters // A. Antoniou. - McGraw-Hill, 2005. - 965 p.;
52. Chen P.C. A Damping Perturbation Method for Flutter Solution: The g-Method // AIAA Journal, Vol. 38, No. 9, Sept. 2000. pp.1519-1524.;
53. Chen P.C. and Liu D.D. Unsteady Supersonic Computation of Arbitrary Wing-Body Configurations Including External Stores // Journal of Aircraft, Vol. 27, No. 2, Feb. 1990, pp. 108-116.;
54. Chen P.C., Gao, X.W., and Tang, L. Overset Field-Panel Method for Unsteady Transonic Aerodynamic Influence Coefficient Matrix Generation // AIAA Journal, Vol. 42, No. 9, September 2004.;
55. Chen P.C., Lee H.W., Liu D.D. Unsteady Subsonic; Aerodynamics for Bodies and Wings with External Stores Including Wake Effort // Journal of Aircraft, Vol. 30, No. 5, Sep.-Oct. 1993, pp. 618-628;
56. CompactPCI. Материалы из свободной энциклопедии [электронный ресурс]. // URL: <http://ru.wikipedia.org/wiki/CompactPCI> (дата обращения 09.10.2014)
57. Cory L. Clark. LabVIEW digital signal processing and digital communications. // McGraw-Hill, London, 2005, 205p.;
58. Crochier R. E. The Analysis of Linear Digital Circuits // R. E. Crochier, A. V. Oppenheim. Proceedings of IEEE. – 1975. – V. 63 (4). - P. 581 – 595.;
59. Duchon J. Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces. // Constructive Theory of Functions of Several Variables, edited by W. Schempp and K. Zeller, Springer, Oberwolfach, Germany, 1976, pp. 85-100;
60. Harder R.L., Desmarais R.N. Interpolation Using Surface Splines. // AIAA Journal, Vol. 9, No. 2, 1972, pp. 189-191.;
61. Jie Zeng, Dallas W. Kingsbury, Erich Ritz, Ping-Chih Chen и др. GVT-based ground flutter test without wind tunnel // 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2011, pp. 1-17.;

62. Karpel M. Extension to the Minimum-State Aeroelastic Modeling Method // AIAA Journal, Vol. 29, No. 11, 1991, pp. 2007-2009.;
63. LabVIEW Real-Time 8. Разработка приложений. // Издательство National Instruments, 2008, 433 стр.;
64. Ljung L. System Identification-Theory for the User // Prentice Hall, Upper Saddle River, NJ, 1999.;
65. Rodden, W.P., Giesing, J.P. and Kalman, T.P. New Method for Nonplanar Configurations. // AGARD Conference Proceedings, CP-80-71, Pt. II, No. 4, 1971.
66. Sanchez Pena RS, Sznaiar M. Robust Systems Theory and Applications // John Wiley & Sons, Inc., 1998. p.490.;
67. Van Overschee P., de Moor B Subspace. Identification for Linear Systems: Theory-Implementation- Applications. // Kluwer Academic Publishers, Boston / London / Dordrecht, 1996.;
68. Zhou K., Doyle J.C. Essentials of Robust Control // Prentice, Hall, 1998.