

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное
учреждение высшего образования
ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
Институт компьютерных технологий и информационной безопасности
Кафедра Интеллектуальных и многопроцессорных систем

На правах рукописи



БУРЯКОВ ДМИТРИЙ СЕРГЕЕВИЧ

**МЕТОДЫ И ПРОГРАММНЫЕ СРЕДСТВА ОБЕСПЕЧЕНИЯ
ИЗОХРОННОЙ ПЕРЕДАЧИ ДАННЫХ В КОМПЛЕКСАХ
ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ**

Специальность: 2.3.5 – Математическое и программное обеспечение
вычислительных систем, комплексов и компьютерных сетей, технические
науки

**ДИССЕРТАЦИЯ НА СОИСКАНИЕ УЧЕНОЙ СТЕПЕНИ
КАНДИДАТА ТЕХНИЧЕСКИХ НАУК**

Научный руководитель:
доктор технических наук,
профессор



И.И. Левин

Таганрог – 2025 г.

СПИСОК СОКРАЩЕНИЙ

ЦОС – цифровая обработка сигналов

ФАР – фазированная антенная решетка

РЛС – радиолокационная станция

CPU – central processing unit

ЦСП – цифровой сигнальный процессор

ASIC – application-specific integrated circuit

ПЛИС – программируемая логическая интегральная схема

ЭВМ – электронно-вычислительная машина

РВС – реконфигурируемая вычислительная система

РПВС - реконфигурируемая проблемно-ориентированная вычислительная система

OSI – open systems interconnection

CRC – cyclic redundancy check

PTP - precision time protocol

TCP - transmission control protocol

UDP - user datagram protocol

RTP - real-time transport protocol

MMCM - mixed-mode clock manager

HPBW - half power beam width

ОГЛАВЛЕНИЕ

| | |
|---|-----|
| ВВЕДЕНИЕ | 5 |
| 1. АНАЛИЗ МЕТОДОВ И СРЕДСТВ ОБЕСПЕЧЕНИЯ ИЗОХРОННОЙ ПЕРЕДАЧИ ДАННЫХ В КОМПЛЕКСАХ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ | 24 |
| 1.1 Сетевые технологии..... | 24 |
| 1.2 Методы и средства синхронизации информационных потоков в архитектуре dataflow (потока данных)..... | 41 |
| 1.3 Анализ элементной базы для обеспечения когерентной обработки данных в комплексах ЦОС..... | 51 |
| 1.4 Принципы обеспечения гарантоспособной изохронной передачи данных в комплексах ЦОС..... | 65 |
| 1.5 Выводы..... | 70 |
| 2 КОМБИНИРОВАННЫЙ МЕТОД ИЗОХРОНОЙ ПЕРЕДАЧИ ДАННЫХ . | 72 |
| 2.1 Комбинированный метод изохронной передачи данных | 72 |
| 2.2 Реализация комбинированного метода изохронной передачи данных . | 95 |
| 2.3 Анализ гарантоспособности реализации комбинированного метода . | 114 |
| 2.4 Выводы..... | 124 |
| 3. МОДЕРНИЗИРОВАННЫЙ МЕТОД ИЗОХРОННОЙ ПЕРЕДАЧИ ДАННЫХ | 127 |
| 3.1 Необходимость усовершенствования комбинированного метода изохронной передачи данных | 127 |
| 3.2 Модернизированный метод изохронной передачи данных..... | 132 |
| 3.3 Реализация модернизированного метода изохронной передачи данных | 142 |
| 3.4. Анализ гарантоспособности реализации модернизированного метода | 153 |

| | |
|--|-----|
| 3.5 Выводы..... | 156 |
| ЗАКЛЮЧЕНИЕ | 158 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 161 |
| ПРИЛОЖЕНИЕ 1 | 181 |
| ПРИЛОЖЕНИЕ 2 | 186 |

ВВЕДЕНИЕ

Актуальность темы исследования. Когерентная обработка данных предполагает согласованное преобразование информационных массивов от множества источников со скоростью, не меньшей, чем темп поступления информации.

В настоящее время когерентная обработка цифровых сигналов широко используется в различных областях электроники и радиотехники. Она применяется в таких важных направлениях, как телекоммуникации, радиосвязь, радиолокация и радиоастрономия. Одним из ключевых применений когерентной обработки являются системы с фазированными антенными решетками (ФАР) [1]. Фазированные антенные решетки, функционирующие совместно с высокопроизводительными вычислительными устройствами цифровой обработки сигналов (ЦОС), открывают новые возможности в области обработки радиосигналов [2], в частности, одновременное формирование десятков и даже сотен независимо управляемых диаграмм направленности антенны значительно повышает точность локализации объектов в пространстве и позволяет автоматически сопровождать ведомые группой лучей десятки воздушных и космических объектов [3]. Адаптивное формирование диаграмм направленности ФАР помогает снизить влияние помех, включая интерференцию от других источников излучения, что особенно актуально в условиях сильных электромагнитных полей, характерных для современных густонаселенных городских сред [4]. Следует отметить, что ФАР сохраняет работоспособность при частичной деградации антенной решетки: при выходе из строя отдельных антенных элементов и/или каналов связи система продолжает эффективно функционировать при несколько сниженных характеристиках [5].

Основными параметрами диаграммы направленности фазированной антенной решетки являются ширина основного лепестка диаграммы направленности, уровень боковых лепестков, а также их пространственное направление относительно полотна антенной решетки [6]. Данные параметры

обеспечиваются за счёт необходимого фазового распределения по апертуре ФАР [7].

Формирование диаграммы направленности производится путем когерентного, то есть синфазного суммирования сигналов от всех без исключения антенных элементов ФАР [8]. Следует отметить, что рассинхронизация информационных потоков приведёт к обработке операндов, полученных в разное время, что недопустимо, так как возникнут фазовые ошибки, способные ухудшить диаграмму направленности или даже сделать её формирование невозможным.

В этой связи когерентная обработка выдвигает жесткие требования к системе передачи данных комплекса цифровой обработки сигналов: должно быть гарантировано поступление операндов, относящихся к одному моменту физического времени, от множества аналого-цифровых преобразователей сигналов с антенных элементов в модули цифровой обработки, формирующие диаграмму направленности ФАР в пределах одного такта. Такую передачу данных в дальнейшем будем называть изохронной.

Изохронная передача данных требует скоростных и помехозащищенных каналов передачи информации, а также программных средств, обеспечивающих синхронизацию процессов оцифровки, обработки и передачи данных. В небольших комплексах ЦОС с ФАР, состоящих из сотен антенных элементов, обеспечение изохронной передачи данных не представляет особой сложности. Однако современные радиолокационные станции (РЛС), осуществляющие мониторинг обширного воздушного и космического пространства, имеют в своем составе фазированную антенную решетку, состоящую из десятков тысяч антенных элементов, при этом протяженность линий связи для передачи данных от блоков оцифровки в блоки ЦОС может достигать сотен метров [9]. Для подобных систем традиционные методы синхронизации оказываются недостаточно эффективными: они или требуют значительных аппаратных/временных затрат

или не обеспечивают надлежащего уровня надежности (гарантоспособности) передачи множества информационных потоков за один временной такт.

Кроме того, для больших радиолокационных комплексов с ФАР возникают новые задачи, связанные, в частности, с необходимостью отслеживания космического мусора на низкой околоземной орбите, количество которого растет с каждым годом [10]. Для решения подобной проблемы требуются более совершенные РЛС с еще большим количеством антенных элементов для увеличения разрешающей способности станции [11].

Очевидно, что с ростом числа каналов передачи данных и увеличения протяженности линий связи в комплексах ЦОС с ФАР значительно возрастает вероятность возникновения ошибок в каналах передачи данных, включая нарушения целостности данных и значительные временные рассогласования. Хотя данная проблема может быть тривиально решена за счет дублирования или, точнее, троирования каналов [12] (дублирование только обнаружит ошибку в канале, но не исправит ее), но такой подход сопряжен с существенными аппаратными затратами, что является неприемлемым. В современных РЛС невозможно реализовать в три раза больше информационных каналов для троирования в пространстве, также невозможно трехкратно увеличить жесткий временной интервал передачи и обработки информационных массивов (троирования во времени), поскольку ограничения по латентности сигнала делают такой подход недопустимо расточительным.

В связи с этим актуальной задачей становится обеспечение надежной передачи данных на имеющихся информационных каналах РЛС, что можно реализовать с помощью специальных инструментальных программных средств обеспечения гарантоспособной передачи данных.

Гарантоспособность определяется как свойство системы (программного комплекса) предоставлять основные требуемые услуги, которым можно оправданно доверять [13]. Понятие гарантоспособности было введено французским учёным Жаном-Клодом Лапри в 1985 году в работе "Dependable Computing and Fault Tolerance: Concepts and Terminology" (Гарантоспособные

вычисления и отказоустойчивость: Концепции и терминология) [13]. Жан-Клод Лапри, один из ведущих ученых в области отказоустойчивых вычислений, разработал эту концепцию для формализации требований к гарантоспособным вычислительным системам.

Его идеи продолжили развивать такие ученые, как А. Авиженис [14], Б. Рэнделл [15], Т. Андерсон [16], которые существенно расширили рамки первоначальной концепции Лапри, дополнили теорию гарантоспособности методами предотвращения и маскирования ошибок, ввели понятие атрибутов гарантоспособности, разработали атрибутивную модель оценки гарантоспособности.

Их идеи легли в основу множества стандартов для критически важных систем. В авионике используется стандарт DO-178C [17], который регламентирует разработку программного обеспечения для самолётов. Стандарт IEC 61508 [18] детально регламентирует подходы к разработке, верификации и валидации программного обеспечения, используемого в гарантоспособных системах. Основная цель - минимизировать вероятность отказов и обеспечить предсказуемую работу программного обеспечения даже в условиях сбоя.

Отечественными учеными, занимающимися вопросами гарантоспособности программного обеспечения, являются такие ученые, как В.А. Харитонов [19], Р.Ю. Царев [20]. Вопросами гарантоспособности для компьютерных сетей занимались В.А. Игнатов [21], В.С. Харченко [22].

Однако методы гарантоспособности вычислительных систем и компьютерных сетей существенно отличаются от методов гарантоспособности комплексов ЦОС, которые характеризуются большим числом информационных каналов и жесткими требованиями к латентности данных.

Степень разработанности темы исследования. В рамках диссертационного исследования проведен анализ существующих методов и

средств обеспечения надежной (гарантоспособной) передачи данных в комплексах ЦОС.

Вопросами передачи данных в комплексах ЦОС занимались такие ученые, как В.К. Слока, П.А. Созинов, С.А. Топчиев, Е.А. Семерников, О.В. Катаев, И.И. Левин и другие [23, 24, 25, 26, 27, 28, 29]. Следует отметить, что в своих работах ученые рассматривали вычислительные средства комплексов ЦОС с небольшим числом информационных каналов (до 20000), работающих на частотах дискретизации менее 100 МГц.

Проведенный анализ показал, что стандартные сетевые протоколы не обеспечивают требуемую латентность для системы изохронной передачи данных в комплексах ЦОС, содержащих множество информационных каналов и передающих данные на высоких частотах. Основными ограничивающими факторами выступают механизмы повторных запросов, потенциальные нарушения последовательности доставки пакетов, а также значительный уровень джиттера, что в совокупности негативно влияет на временные характеристики передачи данных [32].

В процессе исследования был выполнен анализ элементной базы для средств обеспечения когерентной обработки данных. Вычислительные блоки для когерентной обработки данных в комплексах ЦОС могут быть реализованы на различной элементной базе: на универсальных и сигнальных процессорах, специализированных интегральных схемах (ASIC) или программируемых логических интегральных схемах (ПЛИС).

Анализ показал, что универсальные процессоры нецелесообразны для создания комплексов ЦОС с большим числом каналов из-за недостаточной вычислительной мощности, ограниченной поддержки параллелизма и сложности интеграции с аппаратными компонентами системы. Цифровые сигнальные процессоры, несмотря на их специализацию, не ориентированы на многоканальную обработку и не могут обеспечить высокую эффективность при построении комплексов ЦОС с большим количеством информационных каналов и высокой частотой передачи данных. Применение ASIC

экономически целесообразно только в условиях массового производства, тогда как для специализированных систем, таких как вычислительные комплексы для обработки данных фазированных антенных решеток, их использование приводит к существенному увеличению стоимости.

В ряде работ [30, 31] описано применение реконфигурируемых вычислительных систем (РВС) на основе ПЛИС в составе устройств, ориентированных на многоканальную высокопроизводительную цифровую обработку сигналов. ПЛИС обладают значительным аппаратным ресурсом для эффективной реализации алгоритмов цифровой обработки сигналов и поддерживают множество внешних интерфейсов, что позволяет организовать многопоточную когерентную обработку больших объемов данных.

Таким образом, можно утверждать, что в настоящее время отсутствуют методы и средства, обеспечивающие гарантированную передачу данных по множеству каналов с высокой частотой дискретизации в комплексах ЦОС, что представляет собой актуальную задачу для дальнейших исследований.

Целью работы является повышение гарантированности программных комплексов для изохронной передачи данных.

Объектом исследований является программное обеспечение реконфигурируемых вычислительных систем.

Предметом исследований являются методы создания программных инструментальных средств для обеспечения изохронной передачи данных в системах параллельной и распределенной обработки данных.

Научная задача, решаемая в диссертации, - разработка методов создания программных инструментальных средств для реконфигурируемых вычислительных систем, повышающих гарантированность программного комплекса изохронной передачи данных при большом числе каналов и высоких частотах дискретизации.

Для достижения поставленной цели и решения научной задачи необходимо:

- провести анализ существующих методов и средств для обеспечения гарантоспособной изохронной передачи данных при большом числе каналов;

- разработать принципы обеспечения изохронной передачи данных в комплексах ЦОС;

- разработать комбинированный метод изохронной передачи данных по каналам связи, обеспечивающих выравнивание потоков данных, а также анализ и парирование аварийных ситуаций;

- реализовать комбинированный метод в виде инструментальных программных средств для реконфигурируемых вычислительных систем;

- разработать модернизированный метод изохронной передачи данных комплекса когерентной цифровой обработки сигналов, обеспечивающий заданный уровень гарантоспособности при значительном увеличении количества каналов данных;

- реализовать модернизированный метод в виде инструментальных программных средств для реконфигурируемых вычислительных систем.

Методы исследований. В ходе исследований были использованы: методы цифровой обработки сигналов; методы теории графов; статистический анализ; методы структурной организации вычислений.

Экспериментальные исследования были проведены на ряде проблемно-ориентированных реконфигурируемых вычислительных систем, предназначенных для формирования диаграмм направленности фазированных антенных решеток и цифровой обработки сигналов.

Научная новизна диссертационной работы заключается в том, что в ней разработаны:

- комбинированный метод изохронной передачи данных, сочетающий использование единого машинного времени и опорной тактовой частоты отличающийся от известных введением служебных промежутков при передаче массивов операндов;

- алгоритм выбора опорного канала и формирования диапазона допустимого рассогласования задержек данных, отличающийся процедурой

назначения опорного канала, у которого задержка данных наиболее близка к математическому ожиданию задержек данных всех каналов;

- модернизированный метод изохронной передачи данных, отличающийся процедурами переключения опорных каналов, предварительной проверкой контрольных сумм и нумерацией массивов операндов;

- алгоритм переключения опорного канала без прерывания передачи данных, отличающийся процедурой автоматического назначения нового опорного канала с задержкой данных в канале, максимально приближенной к задержке данных текущего опорного канала в случае его аварийного отключения.

Теоретическая значимость научных результатов исследования заключается в развитии методов программирования реконфигурируемых вычислительных систем.

Автором показано, что комбинирование методов синхронизации с использованием единого машинного времени и опорной тактовой частоты обеспечивают необходимую точность для реализации когерентной обработки данных в комплексах ЦОС.

Автором показано, что при увеличении числа каналов в комплексе ЦОС значительно снижается гарантированность для традиционных средств обеспечения изохронной передачи данных.

Автором доказано, что совокупность различных методов обнаружения и парирования аварийных ситуаций обеспечивает заданный уровень гарантированности программных инструментальных средств для РВС при когерентной обработке информации от множества антенных элементов фазированной антенной решетки при высокой частоте дискретизации.

Полученные результаты могут быть использованы для дальнейшего совершенствования методов программирования реконфигурируемых вычислительных систем, ориентированных на многоканальную обработку потоков данных в режиме реального времени.

Практическая значимость научных результатов:

- разработанный автором комбинированный метод изохронной передачи данных позволяет повысить гарантированность прикладных программ РВС с низкими аппаратными и временными затратами, а также с фиксированными частотами дискретизации, в 3,8 раз по сравнению с традиционными методами;

- разработанный автором модернизированный метод изохронной передачи данных позволяет повысить гарантированность прикладных программ РВС для обеспечения изохронной передачи данных в 2,5 раз по сравнению с комбинированным методом;

- разработанный алгоритм переключения опорных каналов позволил сохранить корректное функционирование программы для обеспечения изохронной передачи данных при аварийном отключении текущего опорного канала без прерывания процесса передачи данных;

- создана инструментальная программа, реализующая комбинированный метод для ПЛИС Xilinx Virtex-7 XC7VX485T. (Свидетельство о государственной регистрации программы для ЭВМ №2025660549, РФ). Оценка гарантированности программы изохронной передачи данных показала повышение гарантированности в 3,85 раз по сравнению с реализацией базового метода, что обеспечивает заданный уровень гарантированности 0,95 на доверительном интервале шесть часов при 42100 антенных элементах (информационных каналах).

Использование результатов работы. Результаты диссертационного исследования использованы:

- при выполнении ряда работ в Научно-исследовательском центре супер-ЭВМ и нейрокомпьютеров (ООО «НИЦ СЭ и НК»), а именно: СЧ ОКР «Разработка цифровой диаграммообразующей системы фазированной антенной решетки приемного аппаратурно-антенного комплекса». Акт о внедрении от «29» мая 2025 г., утвержден техническим директором.

- в учебном процессе кафедры Интеллектуальных и многопроцессорных систем (ИМС) Института компьютерных технологий и информационной

безопасности (ИКТИБ) Южного федерального университета (ЮФУ), акт от 30 мая 2025 г., утвержден директором ИКТИБ ЮФУ. Материалы диссертации используются в лекционном курсе по дисциплине «Методы цифровой обработки сигналов, изображений и ситуаций» (тема № 5 «Спектральный анализ. Дискретное преобразование Фурье») и «ПЛИС-технологии и методы создания эффективных прикладных программ для РВС» (тема № 20 «Параллельные системы» модуля 1 «Основы работы с ПЛИС») для подготовки магистров направления подготовки 01.04.02 Прикладная математика и информатика (образовательная программа «Прикладная математика для высокопроизводительных вычислительных систем»).

Степень достоверности результатов, полученных соискателем, подтверждена корректностью и непротиворечивостью математических выкладок, результатами машинного моделирования и экспериментами на комплексах ЦОС для формирования диаграмм направленности от ФАР, а также внедрениями, подтвержденными соответствующими актами. Результаты диссертации докладывались и обсуждались на научно-технических конференциях, где соискатель выступал с докладами по данной проблематике и получил положительный отзыв научной общественности.

Апробация работы. Основные результаты, представленные в диссертации, докладывались и обсуждались на всероссийских и международных научно-технических конференциях:

- Всероссийской научно-технической конференции «Многопроцессорные вычислительные и управляющие системы» (МВУС-2022), посвященной 100-летию академика РАН А.В. Каляева (г. Таганрог, 27–30 июня 2022 года);

- XIX Ежегодной молодежной научной конференции «Достижения и перспективы научных исследований молодых ученых Юга России» (г. Таганрог, «20» апреля 2023 г.);

- I Всероссийской школе-семинаре Национального центра физики и математики для студентов, аспирантов, молодых ученых и специалистов

«Центр исследования архитектур суперкомпьютеров» (г. Саров, 21-25 августа 2023 г.);

- XVI Всероссийской мультikonференции по проблемам управления (МКПУ-2023) (г. Волгоград, 11-15 сентября 2023 г., секция «Управление в распределенных и сетевых системах» (УРСС-2023));

- XX Ежегодной молодежной научной конференции «Наука Юга России: достижения и перспективы» (г. Ростов-на-Дону, 15–26 апреля 2024 г.);

- XIX Международной конференции «Супервычисления и математическое моделирование» (г. Саров, 20–24 мая 2024 г.);

- XIV Всероссийском совещании по проблемам управления (г. Москва, 17-20 июня 2024 г.);

- XXI Всероссийской ежегодной молодежной научной конференции с международным участием «Наука Юга России: достижения и перспективы» (г. Таганрог, 22 апреля 2025 года).

География апробации и обсуждения работы позволяет сделать вывод о широкой информированности научной общественности о результатах диссертационных исследований.

Наиболее значительными публикациями по теме диссертации являются:

1. Левин, И.И. Система передачи данных на основе программируемых логических интегральных схем в устройствах цифровой обработки сигналов с фазированными антенными решетками / И.И. Левин, Е.А. Семерников, Д.С. Буряков // Вестник компьютерных и информационных технологий. – 2022. – Т. 19. – № 3(213). – С. 42-50. – DOI 10.14489/vkit.2022.03.pp.042-050. – EDN NOSXIW. **(ведущий рецензируемый журнал, входит в перечень ВАК категории К2).**

2. Левин, И.И. Некоторые методы синхронизации информационных потоков в системах цифровой обработки сигналов / И.И. Левин, Д.С. Буряков // Известия ЮФУ. Технические науки. – 2024. – № 5(241). – С. 243-254. – DOI

10.18522/2311-3103-2024-5-243-254. **(ведущий рецензируемый журнал, входит в перечень ВАК категории К2).**

3. **Свидетельство о государственной регистрации программ для ЭВМ № 2025660549, РФ.** Программа обработки изохронных потоков данных /И.И. Левин, Д.С. Буряков, М.Д. Чекина // Зарегистр. в Реестре программ для ЭВМ 16.04.2025 г. Правообладатель: ООО «НИЦ супер-ЭВМ и нейрокомпьютеров».

4. Буряков Д.С., Левин И.И. Подсистема межблочной синхронной передачи данных на основе ПЛИС в комплексах цифровой обработки сигналов. Молодой ученый. – 2022. – № 15 (410). – С. 14-20. – URL: <https://moluch.ru/archive/410/90408/>

5. Буряков, Д.С. Система передачи данных на основе ПЛИС в устройствах цифровой обработки сигналов с фазированными антенными решетками / Д.С. Буряков, И.И. Левин // Многопроцессорные вычислительные и управляющие системы: материалы Всероссийской научно-технической конференции (МВУС-2022), Таганрог, 27–30 июня 2022 года. – Ростов-на-Дону: Южный федеральный университет, 2022. – С. 138-142. – EDN RZUVNX.

6. Буряков Д.С., Подход к повышению пропускной способности линий связи в комплексах когерентной цифровой обработки сигналов (276 с.) // Материалы научных мероприятий: Всероссийской конференции с международным участием «Угрозы и риски на Юге России в условиях геополитического кризиса»: материалы докладов (г. Ростов-на-Дону, 15–18 марта, 26–29 апреля 2023 г.); XIX Ежегодной молодежной научной конференции «Достижения и перспективы научных исследований молодых ученых Юга России»: тезисы докладов (г. Ростов-на-Дону, 17–28 апреля 2023 г.) / [гл. ред. акад. Г.Г. Матишов]. – Ростов-на-Дону: Изд-во ЮНЦ РАН, 2023. – 352 с. – ISBN 978-5-4358-0251-1.

7. Буряков Д.С., Организация когерентной обработки сигналов в комплексах ЦОС (22 - 23 с.) // Всероссийская школа-семинар Национального центра физики и математики для студентов старших курсов, аспирантов,

молодых ученых и специалистов «Центр исследования архитектур суперкомпьютеров»: Тезисы – Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2023 – 63с., ил.

8. Буряков Д.С., Методы повышения пропускной способности линий связи в комплексах когерентной цифровой обработки сигналов (133 – 136 с.) // XVI Всероссийская мультиконференция по проблемам управления (МКПУ-2023): материалы мультиконференции (г. Волгоград, 11-15 сентября 2023 г.). В 4 т. Т. 2. Управление в распределенных и сетевых системах (УРСС-2023) / редкол.: И.А. Каляев, В.Г. Пешехонов, С.Ю. Желтов [и др.]. – Волгоград: ВолгГТУ, 2023. – 368 с. – ISBN 978-5-9948-4703-9, ISBN 978-5-9948-4705-3 (Т. 2).

9. Буряков Д.С. Методы повышения гарантоспособности систем передачи данных в комплексах ЦОС / Д.С. Буряков // XIV Всероссийское совещание по проблемам управления (ВСПУ-2024): сборник научных трудов, 17-20 июня 2024 г. – М.: ИПУ РАН, 2024. – С. 2513-2517.

10. Буряков Д.С. Методы согласованной цифровой обработки сигналов / Д.С. Буряков // Наука Юга России: достижения и перспективы: Тезисы докладов XX Всероссийской ежегодной молодежной научной конференции с международным участием, Ростов-на-Дону, 15–26 апреля 2024 года. – Ростов-на-Дону: Южный научный центр РАН, 2024. – С. 90. – ISBN 978-5-4358-0258-0.

11. Буряков Д.С. Обнаружение и устранение аварийных ситуаций в каналах передачи данных систем ЦОС // Супервычисления и математическое моделирование. Тезисы XIX Международной конференции (г. Саров, 20-24 мая 2024 г.). – Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2024. – С. 38-39 – ISBN 978-5-9515-0559-0.

12. Буряков, Д.С. Реализация комбинированного метода изохронной передачи данных на РВС / Д.С. Буряков // Наука Юга России: достижения и перспективы: Тезисы докладов XXI Всероссийской ежегодной молодежной научной конференции с международным участием, Ростов-на-Дону, 14–25

апреля 2025 года. – Ростов-на-Дону: Южный научный центр РАН, 2025. – С. 104. – ISBN 978-5-4358-0270-2.

Личный вклад автора. Все представленные в диссертации результаты получены автором лично.

Положения, выдвигаемые для защиты:

- при увеличении числа информационных каналов в комплексе ЦОС, функционирующих на высокой частоте дискретизации, для традиционных методов и существующих средств значительно снижается (ниже допустимого уровня) гарантоспособность программных средств обеспечения изохронной передачи данных;

- введение служебных промежутков при передаче массивов операндов для комбинированного метода изохронной передачи данных, сочетающего использование единого машинного времени и опорной тактовой частоты, позволяет повысить гарантоспособность программных средств обеспечения изохронной передачи данных;

- введение процедур переключения опорных каналов, предварительной проверки контрольных сумм и нумерации массивов операндов для модернизированного метода изохронной передачи данных позволяет дополнительно повысить гарантоспособность программных средств обеспечения изохронной передачи данных для систем ЦОС, содержащих до 106500 информационных каналов.

Результаты, выдвигаемые для защиты:

- комбинированный метод изохронной передачи данных, сочетающий использование единого машинного времени и опорной тактовой частоты отличающийся от известных введением служебных промежутков при передаче массивов операндов;

- алгоритм выбора опорного канала и формирования диапазона допустимого рассогласования задержек данных, отличающийся процедурой назначения опорного канала, у которого задержка данных наиболее близка к математическому ожиданию задержек данных всех каналов;

- модернизированный метод изохронной передачи данных, отличающийся процедурами переключения опорных каналов, предварительной проверкой контрольных сумм и нумерацией массивов операндов;

- алгоритм переключения опорного канала без прерывания передачи данных, отличающийся процедурой автоматического назначения нового опорного канала с задержкой данных в канале, максимально приближенной к задержке данных текущего опорного канала в случае его аварийного отключения.

Структура работы. Диссертационная работа состоит из введения, трех глав, заключения, списка использованных источников и двух приложений.

Во введении обоснована актуальность диссертационной работы, сформулирована цель и аргументирована научная новизна исследований, показана практическая значимость полученных результатов, представлены выносимые на защиту научные положения, а также приведено краткое содержание каждой из глав.

В первой главе проведен анализ существующих методов и средств обеспечения когерентной обработки данных в комплексах цифровой обработки сигналов. Проведен анализ стандартных сетевых технологий: протокол TCP, протокол UDP, семейство технологий Real-Time Ethernet, Real-Time Transport Protocol. Анализ показал, что традиционные протоколы сетевых технологий не обеспечивают изохронную передачу данных в таких комплексах вследствие ряда ограничений, к числу которых относятся: наличие механизмов повторных запросов, приводящих к недетерминированным задержкам, нарушение последовательности доставки пакетов данных, а также повышенный уровень джиттера, что в совокупности делает эти протоколы непригодными для задач обеспечения изохронной передачи данных.

Проведен анализ методов и средств синхронизации информационных потоков в архитектуре dataflow, который показал, что существующие решения для архитектуры dataflow для обеспечения изохронной передачи данных

требуют значительных аппаратных и временных затрат, что приводит к низкой эффективности комплексов ЦОС.

Рассмотрено применение универсальных и сигнальных процессоров, полностью заказных микросхем и программируемых логических интегральных схем в качестве элементной базы для построения систем когерентной обработки данных в комплексах ЦОС.

Доказано, что реконфигурируемые вычислительные системы на основе ПЛИС могут служить перспективным решением для построения комплексов когерентной цифровой обработки сигналов. Показано, что в настоящее время отсутствуют методы и средства для реконфигурируемых вычислительных систем на основе ПЛИС, обеспечивающие надежную (гарантоспособную) изохронную передачи данных для множества каналов.

Сформированы принципы обеспечения изохронной передачи данных в комплексах ЦОС.

Во второй главе доказано, что существующие методы изохронной передачи данных в комплексах цифровой обработки сигналов обладают рядом существенных недостатков. К числу таких недостатков относится значительное снижение пропускной способности, обусловленное необходимостью выделения части ширины канала для передачи временных меток. Кроме того, наблюдается увеличение аппаратных и временных затрат, связанное с реализацией сложных алгоритмов синхронизации данных между различными каналами. Также отмечается отсутствие механизмов динамической компенсации задержек между разными каналами при передаче данных.

Для устранения указанных недостатков предложен новый комбинированный метод изохронной передачи данных, сочетающий использование единого машинного времени и опорной тактовой частоты, отличающийся от известных введением служебных промежутков при передаче массивов операндов. Для введения служебных промежутков в информационный поток протокол комбинированного метода предполагает

передачу данных на несколько большей частоте относительно частоты поступления (оцифровки).

Предложено разместить в служебном промежутке заголовков массива операндов с временной меткой и по окончанию массива контрольную сумму для проверки целостности массива операндов. Выравнивание задержек данных между разными каналами предложено осуществить с помощью буферной памяти, запись в которую происходит асинхронно, по мере поступления массивов операндов от каждого канала, а чтение выполняется синхронно, обеспечивая выравнивание задержек.

Выделены наиболее вероятные аварийные ситуации, характерные для системы изохронной передачи данных: аварийное отключение канала; нарушение целостности передаваемой информации; превышение допустимого порога рассогласования задержек поступления массивов операндов. Предложены алгоритмы их обнаружения и парирования: процедура назначения опорного канала и формирования диапазона допустимого рассогласования; процедура проверки целостности данных с использованием контрольных сумм; процедура зануления данных в сбойных каналах.

Создана параллельно-конвейерная программа, реализующая комбинированный метод на базе ПЛИС Xilinx Virtex-7 XC7VX485T. Показано, что занимаемый аппаратный ресурс для системы изохронной передачи данных не превысил 5% от общего ресурса.

Выполнена оценка гарантоспособности программного комплекса обеспечения изохронной передачи данных на основе атрибутивной модели. Оценка показала повышение гарантоспособности в 3,85 раз по сравнению с реализацией базового метода, что обеспечивает заданный уровень гарантоспособности 0,95 на доверительном интервале шесть часов при 42100 антенных элементах (информационных каналах).

В третьей главе доказано, что перспективные задачи, решаемые радиолокационными станциями с фазированными антенными решетками, требуют повышения разрешающей способности, которое обеспечивается за

счет увеличения количества антенных элементов. Установлено, что рациональным значением количества элементов, при котором достигается значительный прирост разрешающей способности ФАР, является 90000.

Доказана необходимость модернизации комбинированного метода для сохранения заданного уровня гарантоспособности при увеличении количества каналов до 90000. Ключевым направлением совершенствования комбинированного метода является повышение эффективности процедур парирования ошибок, а также разработка новых механизмов обработки ранее не учитываемых типов ошибок. Для этого разработаны следующие алгоритмы: алгоритм переключения опорных каналов при отказе текущего опорного канала, обеспечивающий работоспособность системы изохронной передачи данных даже в случае аварийного отключения опорного канала; алгоритм предварительной проверки контрольных сумм массивов операндов перед их отправкой в блоки обработки, позволяющий увеличить долю успешно парируемых ошибок целостности данных; алгоритм обнаружения и устранения задержек, совпадающих с периодом следования массивов операндов, что дает возможность выявлять ранее не регистрируемые ошибки рассогласования, способные привести к значительному снижению гарантоспособности системы изохронной передачи данных.

Создана инструментальная программа, реализующая модернизированный метод изохронной передачи данных. Показано, что аппаратные затраты на реализацию данной программы не превышают 7% от общего ресурса ПЛИС Xilinx Virtex-7 XC7VX485T.

Проведен анализ гарантоспособности разработанной программы, реализующей модернизированный метод изохронной передачи данных. Оценка показала повышение гарантоспособности в 2,54 раза по сравнению с реализацией комбинированного метода, что обеспечивает заданный уровень гарантоспособности 0,95 на доверительном интервале шесть часов при 106500 антенных элементах (информационных каналах).

В заключении изложен основной научный результат диссертации и сформулированы теоретические и прикладные результаты, полученные в диссертационной работе.

В приложении приведены акты о внедрении результатов диссертации, представлены копии актов о внедрении результатов диссертации от ООО «НИЦ СЭ и НК» (г. Таганрог) и кафедры Интеллектуальных и многопроцессорных систем (ИМС) Института компьютерных технологий и информационной безопасности (ИКТИБ) «Южный федеральный университет» (ЮФУ), а также свидетельство о государственной регистрации программ для ЭВМ.

1. АНАЛИЗ МЕТОДОВ И СРЕДСТВ ОБЕСПЕЧЕНИЯ ИЗОХРОННОЙ ПЕРЕДАЧИ ДАННЫХ В КОМПЛЕКСАХ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

1.1 Сетевые технологии

Формально передача данных от множества распределенных узлов в комплексах цифровой обработки сигналов может осуществляться с помощью сетевых технологий. Современные сетевые технологии основываются на пакетной передаче данных, а правила их обработки определяются стандартами сетевых протоколов [32, 33].

Сетевые технологии базируются на фундаментальной модели функционирования компьютерных сетей, обладающей модульной архитектурой и известной как базовая эталонная модель взаимодействия открытых систем (англ. Open Systems Interconnection model, OSI) [34]. Данная модель была разработана Международной организацией по стандартизации (ISO) в 1984 году с целью унификации сетевых протоколов и стандартизации коммуникационных процессов, что обеспечило межплатформенную совместимость оборудования различных производителей.

Предложенная модель, представленная на рисунке 1.1, основывается на семиуровневой системе передачи данных между сетевыми узлами. Каждый уровень модели отвечает за определенные задачи в процессе подготовки или обработки данных, применяя соответствующие технологии обмена между уровнями и стандартные протоколы передачи данных. В направлении передачи данные последовательно обрабатываются на каждом уровне по нисходящей траектории, подвергаясь инкапсуляции, которая заключается в добавлении специфичных для каждого уровня служебных заголовков. На стороне приема осуществляется восходящая обработка, в ходе которой происходит декапсуляция - последовательное удаление служебных заголовков на соответствующих уровнях.

| Модель открытых систем OSI | | | | | |
|----------------------------|---|--|--|---|--|
| Уровень (layer) | Тип данных | Функции уровня | Особенность адресации | Примеры протоколов | |
| Уровни хоста (узла) | 7. Прикладной (application) | Данные (строки из байтов) | Доступ к сетевым службам | URL | HTTP(S), FTP(S), RPC, POP3 |
| | 6. Представительский (представления) (presentation) | | Представление (кодировка) и шифрование данных | | ASCII, EBCDIC |
| | 5. Сеансовый (session) | | Управление сеансом связи | | PAP |
| Уровни связи (сети) | 4. Транспортный (transport) | Сегменты (segment) / Дейтаграммы (datagram) | Прямая связь между конечными пунктами и надёжность | Порт | TCP, UDP, SCTP, PORTS |
| | 3. Сетевой (network) | Пакеты (packet) | Определение маршрута и логическая адресация | IP-адрес | IPv4, IPv6, IPsec, AppleTalk |
| | 2. Канальный (data link) | Биты (bit) / Кадры (frame) | Физическая адресация | MAC-адрес (физический адрес компьютера) | PPP, IEEE 802.22, Ethernet, DSL, ARP, L2TP, сетевая карта. |
| 1. Физический (physical) | Электрические сигналы, Биты (bit) | Работа со средой передачи, сигналами и двоичными данными | | | USB, кабель ("витая пара", коаксиальный, оптоволоконный), радиоканал |

Рисунок 1.1. Модель открытых систем OSI

Физический уровень модели OSI представляет собой фундаментальную основу передачи данных в сетях. Этот уровень занимается непосредственной передачей битовых потоков между устройствами через различные физические среды. На физическом уровне определяются ключевые характеристики канала передачи данных. Здесь происходит преобразование последовательностей логических нулей и единиц в соответствующие физические сигналы, которые впоследствии могут передаваться по медным проводам, оптическим волокнам или через радиоволны [35].

Важной функцией физического уровня является обеспечение синхронизации передачи. Для правильной интерпретации битов приемное устройство должно точно знать, когда начинается и заканчивается каждый бит. Это достигается различными методами, включая использование отдельного тактового сигнала или самосинхронизирующихся кодов, таких как манчестерское кодирование.

Физический уровень, несмотря на свою "прозрачность" для конечного пользователя, остается критически важным компонентом любой сети. Именно на физическом уровне определяются ключевые характеристики канала передачи данных.

Канальный уровень играет важную роль в организации надежной передачи данных между непосредственно соединенными узлами сети. Этот уровень выполняет функции, которые обеспечивают корректную передачу информации через физическую среду, добавляя к простой битовой передаче физического уровня элементы структурирования и контроля [36].

Основная задача канального уровня заключается в организации данных в формализованные блоки, называемые кадрами (фреймами) [37]. В отличие от физического уровня, который работает с неструктурированным потоком битов, канальный уровень добавляет к данным специальные заголовки и окончания, которые содержат служебную информацию. Эти кадры имеют четкую структуру, включающую адресацию, контрольные суммы и другую информацию, необходимую для корректной доставки.

Одной из ключевых функций канального уровня является физическая адресация с использованием MAC-адресов (Media Access Control) [38]. Каждое сетевое устройство имеет уникальный MAC-адрес, который используется для идентификации устройств в пределах одной локальной сети. Когда устройство отправляет данные, канальный уровень формирует кадр, указывая MAC-адрес отправителя и получателя. Это позволяет коммутаторам и другим устройствам правильно направлять трафик внутри сегмента сети.

Важным аспектом работы канального уровня является обнаружение и коррекция ошибок. Для этого используются различные механизмы контроля целостности данных, такие как CRC (Cyclic Redundancy Check) [39]. При формировании кадра вычисляется контрольная сумма, которая добавляется в конец фрейма. Принимающее устройство повторно вычисляет эту сумму и сравнивает ее с полученным значением. Если значения не совпадают, кадр считается поврежденным и может быть запрошен повторно.

Канальный уровень делится на два подуровня: LLC (Logical Link Control) и MAC (Media Access Control). Подуровень LLC обеспечивает интерфейс между сетевым уровнем и оборудованием, а также отвечает за

управление потоком и контроль ошибок. Подуровень MAC управляет доступом к физической среде и отвечает за адресацию [38, 40].

Канальный уровень присутствует почти во всех протоколах передачи данных, поскольку обеспечивает неструктурированный поток битов, поступающий от физического уровня необходимой разметкой для корректной передачи данных. Однако его функции могут выполняться вышележащими уровнями или вообще отсутствовать в простых протоколах.

Сетевой уровень является ключевым звеном в организации передачи данных в различных сетях (совокупности объединенных узлов). Этот уровень отвечает за логическую адресацию, маршрутизацию и доставку пакетов в сложных составных сетях, состоящих из множества промежуточных узлов [41]. В отличие от канального уровня, который обеспечивает связь между двумя непосредственно соединенными устройствами, сетевой уровень позволяет данным преодолевать границы отдельных сетевых сегментов.

Основной функцией сетевого уровня является определение оптимального пути для передачи данных от отправителя к получателю через потенциально сложную сетевую инфраструктуру [35]. Для его реализации сетевой уровень использует логические адреса, такие как IP-адреса в стеке TCP/IP, которые имеют иерархическую структуру и позволяют идентифицировать устройства независимо от их физического расположения [42].

Формат данных на сетевом уровне содержит заголовок с управляющей информацией, включая адреса отправителя и получателя, параметры маршрутизации и другую служебную информацию, а также собственно передаваемые данные. Сетевой уровень реализует процедуры фрагментации и повторной сборки пакетов. Когда пакет необходимо передать через сеть с меньшим максимальным размером блока данных (MTU) [43], чем размер исходного пакета, сетевой уровень делит его на фрагменты, которые затем собираются воедино на принимающей стороне.

Сетевой уровень тесно взаимодействует с транспортным уровнем, получая от него сегменты данных и инкапсулируя их в пакеты, а также с канальным уровнем, передавая ему пакеты для дальнейшей отправки. Именно на сетевом уровне работают такие важные сетевые устройства, как маршрутизаторы, которые принимают решения о пересылке пакетов на основе их логических адресов и таблиц маршрутизации. Сетевой уровень остается критически важным компонентом современных компьютерных сетей, обеспечивая возможность глобального соединения устройств и становясь основой для таких масштабных систем как Интернет.

Транспортный уровень выполняет функцию обеспечения передачи данных между конечными системами. Этот уровень служит своеобразным мостом между нижними уровнями, отвечающими за физическую передачу данных, и верхними уровнями, работающими с прикладными программами. Основная задача транспортного уровня - организовать доставку данных от отправителя к получателю с требуемым уровнем надежности, независимо от характеристик нижележащей сети [44]. Главной особенностью транспортного уровня является его способность предоставлять два принципиально разных типа сервиса для вышестоящих уровней. Первый тип - это надежная передача данных с установлением соединения, гарантирующая доставку всех пакетов в правильной последовательности без ошибок. Второй тип - ненадежная передача без установления соединения, которая не гарантирует доставку, но работает быстрее и требует меньше ресурсов [45].

Транспортный уровень отвечает за мультиплексирование и демультимплексирование соединений с помощью системы портов. Порты позволяют одному хосту поддерживать множество одновременных сетевых соединений с разными приложениями.

Важной функцией транспортного уровня является управление потоком данных, которое предотвращает ситуацию, когда быстрый отправитель перегружает медленного получателя. Это реализовано через механизм

скользящего окна, который динамически регулирует скорость передачи данных в зависимости от возможностей получателя и состояния сети.

Транспортный уровень тесно взаимодействует с сетевым уровнем, получая от него пакеты и формируя из них логические соединения, и с сеансовым уровнем, предоставляя ему надежные каналы для передачи данных.

Сеансовый уровень отвечает за организацию, поддержку и завершение диалога (сеанса) между взаимодействующими приложениями на разных устройствах. Этот уровень обеспечивает структурированное взаимодействие между конечными точками связи, позволяя приложениям синхронизировать обмен данными и управлять диалогом [46]. В отличие от транспортного уровня, который обеспечивает простое соединение, сеансовый уровень добавляет логику управления взаимодействием, включая контроль точек восстановления, управление очередностью операций и согласование параметров сеанса.

Основная задача сеансового уровня - предоставить механизмы для установления, поддержания и корректного завершения сеансов связи между приложениями. Например, при загрузке большого файла или проведении видеоконференции сеансовый уровень может установить контрольные точки, чтобы в случае разрыва соединения не пришлось начинать передачу данных с самого начала. Вместо этого сеанс может быть возобновлен с последней успешно переданной порции информации.

Сеансовый уровень также отвечает за разрешение конфликтов, которые могут возникнуть при параллельном доступе к ресурсам. В распределенных системах, где несколько пользователей или процессов могут одновременно пытаться изменить одни и те же данные, сеансовый уровень может предоставлять механизмы блокировки или управления доступом [47].

Сеансовый уровень тесно взаимодействует с представительным уровнем (который отвечает за представление данных) и транспортным уровнем (который обеспечивает фактическую передачу). Он добавляет к простой передаче данных логику управления диалогом, что особенно важно для

сложных распределенных приложений, требующих длительного взаимодействия между компонентами. В современных облачных и распределенных системах функции сеансового уровня приобретают новое значение, обеспечивая *согласованность данных и непрерывность сервиса при работе в условиях возможных разрывов соединения* или переключения между разными точками доступа. В комплексах цифровой обработки сигналов от фазированных антенных решёток строгая временная синхронизация может быть реализована через сеансовые механизмы с поддержкой временных меток, что критически важно для корректной обработки данных.

Представительский уровень выполняет роль конвертера между сетевыми устройствами с разными способами представления данных. Этот уровень обеспечивает условия, чтобы информация, отправляемая из прикладного уровня одной системы, могла быть корректно интерпретирована прикладным уровнем другой системы, независимо от различий в их внутренних форматах данных. Если приложение отправляет данные, то представительский уровень преобразует их внутреннее представление в стандартный сетевой формат. На принимающей стороне происходит обратное преобразование в формат, понятный локальному приложению [48].

Важнейшей задачей этого уровня является обеспечение безопасности передаваемых данных. Здесь реализуются механизмы шифрования и дешифрования информации с использованием таких протоколов, как SSL (Secure Sockets Layer) и TLS (Transport Layer Security) [49]. Представительский уровень также отвечает за компрессию данных, что позволяет значительно сократить объем передаваемой информации и повысить эффективность использования сетевых ресурсов.

В современных сетевых технологиях многие функции представительского уровня реализованы непосредственно в прикладных протоколах. Следует отметить, что принципы, заложенные на этом уровне в модели OSI, остаются фундаментальными для обеспечения совместимости в гетерогенных сетевых средах.

Представительский уровень тесно взаимодействует с прикладным уровнем, получая от него данные в формате, специфичном для конкретного приложения, и преобразуя их в универсальный сетевой формат. С другой стороны, он передает сеансовому уровню уже подготовленные данные, готовые к передаче по сети. Такая архитектура позволяет прикладным программам сосредоточиться на своей работе, не заботясь о проблемах совместимости и безопасности на сетевом уровне [50].

Для реализации системы изохронной передачи данных в комплексах цифровой обработки сигналов от фазированных антенных решёток представительский уровень не требуется, поскольку в таких комплексах форматы обмена информацией строго стандартизированы и заранее определены на этапе проектирования системы.

Прикладной уровень занимает высшую позицию в модели OSI и служит непосредственным интерфейсом между сетевой инфраструктурой и конечным пользователем. Этот уровень обеспечивает доступ сетевым службам для прикладного программного обеспечения, определяя протоколы и методы взаимодействия, которые используют приложения для обмена данными по сети [51]. В отличие от нижележащих уровней, которые решают технические задачи передачи информации, прикладной уровень работает с содержательной частью данных и предоставляет пользователь-ориентированные сервисы.

Основная функция прикладного уровня заключается в предоставлении конкретных сетевых услуг, таких как доступ к веб-страницам, передача файлов, электронная почта или удаленный доступ к системам. Каждый тип сервиса использует специализированные протоколы, определяющие форматы сообщений, последовательности взаимодействия и правила обработки данных для конкретных прикладных задач.

Важной характеристикой прикладного уровня является его ориентация на решение пользовательских задач. Например, при открытии веб-страницы браузер (как прикладное приложение) использует протокол HTTP [52] для

запроса содержимого у сервера, но сам процесс установки соединения, маршрутизации и надежной доставки данных полностью скрыт от пользователя и реализуется нижележащими уровнями. Прикладной уровень фокусируется на семантике взаимодействия, а не на механизмах передачи, и обладает наибольшим разнообразием протоколов и стандартов, так как непосредственно отражает многообразие сетевых сервисов, востребованных пользователями.

Для обслуживания модели взаимодействия открытых систем используется большое количество сетевых протоколов. Рассмотрим подробнее те из них, которые могут быть использованы в системах изохронной передачи данных в комплексах цифровой обработки сигналов от фазированных антенных решёток.

Протокол TCP (Transmission Control Protocol) – протокол транспортного уровня модели OSI, предназначенный для обеспечения гарантированной доставки данных между узлами сети. Он регламентирует процессы сегментации передаваемой информации, формирования пакетов, их передачи, приема и последующей сборки в пункте назначения [53].

Принципы функционирования TCP стандартизированы в ряде документов RFC, включая RFC 1323, 2581, 2988, 3390 и 5681 [54]. Ниже рассматриваются ключевые аспекты работы данного протокола.

Передача данных осуществляется в виде потока: протокол TCP отправляет данные как непрерывный последовательный поток байтов. Эти байты объединяются в сегменты TCP, которые затем передаются на сетевой уровень (IP) для дальнейшей доставки. При этом TCP самостоятельно выполняет разбиение данных на сегменты [55].

Обеспечение надежности: каждому переданному байту TCP назначает порядковый номер и ждёт подтверждения успешного получения от адресата. Если подтверждение не приходит в течение заданного времени ожидания, *данные отправляются повторно*. Получатель использует порядковые номера

пакетов, чтобы правильно собрать все данные и исключить дублирование пакетов.

Процедура передачи данных показана на рисунке 1.2. Она выполняется следующим образом:

А. Клиент отправляет SYN-сегмент, выбирая случайный начальный порядковый номер (sequence number) в диапазоне от 1 до 2^{32} (ISN – Initial Sequence Number). Этот номер служит точкой отсчёта для нумерации последующих сегментов.

Б. Сервер, получив запрос, отвечает сегментом с флагами SYN и ACK. В поле «номер подтверждения» (acknowledgement number) он указывает ISN клиента, увеличенный на 1 (подтверждая приём SYN), а также генерирует свой собственный случайный порядковый номер.

В. Как только клиент получает SYN+ACK, соединение считается установленным. Клиент отправляет ACK-сегмент с обновлёнными номерами последовательности (без данных), завершая трёхэтапное рукопожатие.

Г. После этого начинается обмен данными.

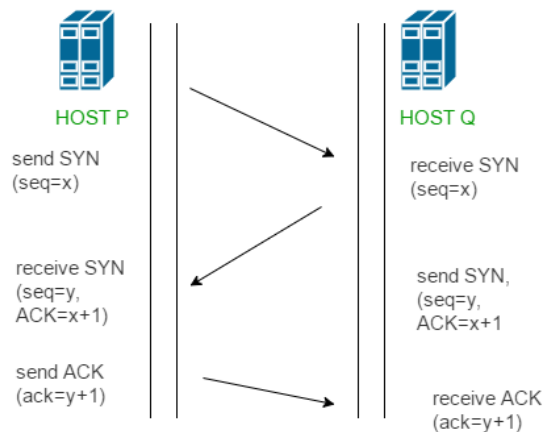


Рисунок 1.2. Процедура обмена данными протокола TCP

Управление потоком происходит следующим образом: при получении пакетов данных TCP отправляет отправителю подтверждение (ACK), указывающее, сколько байтов может быть принято без переполнения внутреннего буфера. Логическое соединение определяется комбинацией

сокетов, порядковых номеров и размеров окон. Оно идентифицируется парой сокетов, используемых для передачи данных между отправителем и получателем. Протокол TCP поддерживает полнодуплексную передачу, позволяя данным передаваться в обоих направлениях одновременно. Для этого каждый TCP-узел использует отдельные буферы отправки и приема, обеспечивая двунаправленный обмен пакетами. TCP является протоколом с установлением соединения.

Необходимо отметить основные недостатки и ограничения данного протокола, которые не позволяют использовать его для решения задач изохронной передачи данных для их последующей когерентной обработки в комплексах цифровой обработки сигналов. Когерентная обработка применяется в задачах, где требуется обработка данных сопоставимой с темпом поступления информации и в системах реального времени, где не допускается большая задержка обработки сигналов. После передачи пакета отправитель ждет подтверждения от получателя. Если после определенного времени ожидания не было подтверждения, то пакет отправляется снова. Все это создает недопустимые задержки передачи данных и нарушает порядок следования пакетов.

Еще одно ограничение применения протокола TCP связано с механизмом работы TCP-сокетов. При отключении интерфейса, связанного с назначенным IP-адресом, происходит остановка TCP-соединения, после чего необходимо инициировать заново весь алгоритм установки как для нового TCP-сеанса, так как приложение может ассоциировать только один IP-адрес с одним TCP-соединением [55]. Данная ситуация критична для работы приложений реального времени, так как возникают большие задержки при передаче данных, что неприемлемо для системы изохронной передачи данных в комплексе ЦОС.

Протокол UDP (User Datagram Protocol) - протокол транспортного уровня, стандарты которого зафиксированы в RFC 768 и утверждены IETF в 1980 году [56]. Он был разработан как альтернатива TCP и отличается тем, что

не требует соблюдения порядка передачи сообщений. Вместо установления соединения UDP использует IP-датаграммы - независимые пакеты данных, которые отправляются без предварительного соединения и подтверждения получения. При передаче по сети дейтаграммы могут теряться, дублироваться или приходить в неправильном порядке. Для того чтобы устранить эти проблемы, разработчики программного обеспечения используют специальные механизмы, повышающие надежность протокола UDP [57].

Протокол UDP работает без установления соединения: при передаче данных по UDP не требуется предварительного создания канала между отправителем и получателем. Пакеты направляются на нужный IP-адрес с указанием порта назначения, при этом ответ от целевого устройства не обязателен. Однако если требуется отправить пакеты обратно, в заголовке UDP может быть указан порт отправителя.

Протокол UDP, как и TCP, применяет порты для того, чтобы пакеты доставлялись нужным протоколам верхнего уровня или соответствующим приложениям на стороне получателя. На рисунке 1.3 продемонстрирован стандартный пример использования UDP-портов: клиентское приложение работает с портом 1234, а серверное — с портом 2345.

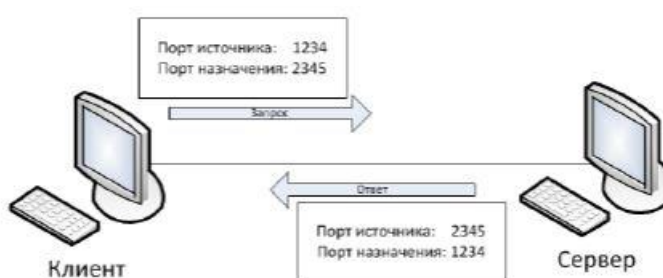


Рисунок 1.3. Процедура установки соединения протокола UDP

Когда клиент отправляет запрос, он идентифицируется как исходящий из порта 1234 на хосте клиента и предназначенный для порта 2345 на хосте сервера. Если серверный процесс отвечает на этот запрос, то реализация UDP

сервера отправляет ответ как исходящий из порта 2345 на хосте сервера и предназначенный для порта 1234 на хосте клиента.

При передаче данных по протоколу UDP взаимодействие между клиентом и сервером осуществляется следующим образом.

Исходный запрос от клиента формируется на транспортном уровне, где в качестве источника указывается порт 1234 на хосте клиента, а в качестве адресата - порт 2345 на хосте сервера. Данные передаются в виде дейтаграммы, содержащей заголовок UDP с указанными портами, а также IP-адресами отправителя и получателя.

При обработке запроса серверный процесс, прослушивающий порт 2345, формирует ответное сообщение. В этом случае серверная реализация UDP инкапсулирует данные в дейтаграмму, где источником выступает порт 2345 на серверном хосте, а адресатом - порт 1234 на клиентском хосте. Таким образом, обеспечивается корректная маршрутизация ответа к исходному отправителю.

Эта процедура демонстрирует принцип работы UDP как ориентированного на обмен дейтаграммами без установления соединения, где каждая передача данных является независимой, а адресация осуществляется исключительно на основе указанных в заголовке портов и IP-адресов.

UDP обеспечивает высокоскоростную передачу данных с минимальными задержками. Этот транспортный протокол не требует установки соединения, что ускоряет обмен данными. Кроме того, его производительность выше, поскольку потеря отдельных пакетов влияет лишь на качество передачи, а не на общую скорость [53].

Хотя отсутствие аутентификации между отправителем и получателем способствует высокой скорости, однако при этом протокол UDP не гарантирует пользователям ни целостность данных, ни их защиту. Также нет уверенности в том, что пакеты данных будут доставлены в правильном порядке [58].

Протокол UDP способен обеспечивать высокоскоростную передачу с минимальными задержками. В отличие от TCP, протокол UDP не реализует

процедуру повторной передачи пакетов, что может быть преимуществом в системах изохронной передачи, где своевременность доставки данных имеет приоритет над надежностью. Однако отсутствие в UDP нумерации пакетов и контроля их последовательности приводит к потенциальному нарушению порядка доставки, что является критическим недостатком для систем изохронной передачи данных в комплексах ЦОС, требующих строгого соблюдения порядка следования передаваемых данных, поэтому данный протокол не может быть использован.

Семейство технологий Real-Time Ethernet (RTE) - это семейство промышленных сетевых технологий с механизмами, обеспечивающими детерминированную и предсказуемую передачу данных в реальном времени. Эти технологии широко применяются в промышленной автоматизации и других областях, где критичны низкие задержки и точная синхронизация.

Семейство RTE включает в себя ряд протоколов, созданных различными организациями и компаниями. К ним относятся, например, EtherCAT [59], Powerlink [60], Profinet [61], SERCOS III [62], LXI [63] и другие. Особое место среди них занимает Profinet, который считается самым перспективным и активно используется в современных промышленных сетях.

Протокол Profinet (Process Field Network) - это промышленный протокол, созданный для автоматизации производственных процессов. Он поддерживает высокоскоростной обмен данными между узлами промышленной сети и включает механизмы, гарантирующие детерминированную передачу информации. Работа протокола Profinet регламентируется международными стандартами, такими как IEC 61784-2 и IEC 61158 [64]. Протокол охватывает физический, канальный, транспортный и прикладной уровни OSI, но основная функциональность сосредоточена на канальном и прикладном уровнях.

Его ключевая особенность - оптимизация для работы в реальном времени, достигаемая за счет минимизации использования сетевого уровня для критичного трафика и акцента на канальный уровень. Протокол

поддерживает несколько режимов передачи данных, каждый из которых предназначен для решения определенных задач.

Режим недетерминированной передачи использует стандартные протоколы TCP/IP и предназначен для задач, не требующих жестких временных ограничений. В данном режиме передаются данные конфигурации, диагностическая информация, а также файлы (например, загрузка прошивок). Поскольку TCP/IP обеспечивает надежную доставку данных с подтверждением и повторной передачей в случае ошибок, он подходит для операций, где важна точность, но не критична задержка. Однако из-за накладных расходов на обработку и возможных задержек этот режим не применяется для управления в реальном времени [65].

Режим реального времени (Real-Time, RT) является ключевым механизмом ProfiNet, обеспечивающим детерминированную передачу данных в промышленных сетях. Этот режим разработан для задач, где критически важны предсказуемые временные характеристики, но не требуется сверхвысокая точность синхронизации.

В основе режима реального времени лежит циклическая модель обмена данными между узлами сети. При этом один узел выполняет роль ведущего устройства, которое организует и контролирует весь процесс коммуникации. Работа происходит по строгому временному графику: через заданные интервалы (от 1 до 10 мс) ведущее устройство рассылает выходные данные всем подчиненным устройствам и одновременно собирает их входные данные. Такой подход обеспечивает строгую периодичность обновления информации в системе.

Вместо протоколов TCP/IP применяется упрощенная схема передачи на основе UDP-протокола, что значительно снижает накладные расходы на обработку пакетов. Дополнительно реализована система приоритизации трафика через VLAN-теги (IEEE 802.1Q), позволяющая выделять RT-пакеты в отдельный приоритетный поток. Временные параметры передачи

поддерживаются на уровне 1-10 мс с вариацией задержки (джиттером) менее одной микросекунды.

Изохронный режим реального времени (Isochronous Real-Time, IRT) представляет собой наиболее совершенную и технологически сложную форму передачи данных в сетях Profinet, обеспечивающую высокий уровень детерминизма и синхронизации. Этот режим специально разработан для задач, где требуются микросекундные точности синхронизации и абсолютная предсказуемость временных характеристик передачи данных [66].

Основу работы IRT-режима составляет принцип временного разделения канала передачи (Time Division Multiple Access, TDMA) [67], кардинально отличающийся от традиционных методов Ethernet-коммуникации. В отличие от стандартного CSMA/CD-подхода, где устройства конкурируют за доступ к среде передачи, в IRT-режиме каждому устройству в сети выделяются строго определенные временные интервалы для передачи данных. Такой подход полностью исключает возможность коллизий и гарантирует, что критические данные всегда будут переданы в точно заданный момент времени.

Сетевой цикл в IRT-режиме делится на два принципиально разных типа временных окон. Изохронные окна предназначены для передачи критически важных данных. Эти данные имеют абсолютный приоритет и передаются с гарантированными временными параметрами. Асинхронные окна используются для менее критичного трафика - диагностической информации, параметров конфигурации, файлов. Они заполняют оставшееся время цикла и могут быть задержаны, если изохронные окна требуют дополнительного времени.

В изохронном режиме реального времени (IRT) Profinet ключевыми параметрами, определяющими качество передачи данных, являются длительность цикла передачи и джиттер (вариация задержки). Длительность цикла передачи в IRT-режиме представляет собой фиксированный временной интервал, в течение которого происходит полный обмен данными между ведущим и ведомыми узлами. Этот параметр является настраиваемым и

обычно варьируется в диапазоне от 31,25 микросекунд до четырех миллисекунд в зависимости от конкретного применения. Джиттер в IRT-режиме характеризует отклонение фактического времени доставки пакета от ожидаемого и достигается на уровне одной микросекунды.

Проведенный анализ протокола PROFINET позволяет заключить, что, несмотря на использование механизмов обеспечения детерминированности задержек при передаче данных, его использование в системах изохронной передачи данных комплекса ЦОС с фазированными антенными решетками является недопустимым. Это связано с тем, что критическим параметром для систем изохронной передачи данных в ЦОС с ФАР является уровень джиттера, который должен находиться в диапазоне десятков наносекунд. Однако даже в оптимальной конфигурации IRT-режима PROFINET минимально достижимый джиттер составляет порядка одной микросекунды, что на порядок превышает допустимые значения.

Кроме того, техническая реализация IRT-режима сопряжена с необходимостью применения специализированных аппаратных решений, на основе ASIC (например, Siemens ERTEC) [68], обеспечивающих обработку данных в строго регламентированные временные интервалы. Данное требование существенно ограничивает гибкость подобной системы изохронной передачи данных и повышает ее стоимость.

Real-time Transport Protocol (RTP) - это стандартный сетевой протокол, разработанный для передачи аудио- и видеоданных в режиме реального времени [69]. Он широко используется в приложениях, где критически важна минимальная задержка, включая голосовую связь, видеоконференции и потоковое вещание. Основным документом, определяющим RTP, является RFC 3550 [70]. Протокол охватывает прикладной уровень модели OSI, а также дополняет транспортный уровень.

Основу RTP составляет транспортный протокол UDP. Такой выбор обусловлен принципиальным отказом от гарантий доставки, которые предоставляет TCP, в пользу минимально возможных задержек. В отличие от

TCP с его механизмами повторной передачи, RTP делегирует обработку потерь пакетов вышестоящим приложениям.

Ключевым элементом архитектуры RTP является система временных меток (timestamps). Каждый передаваемый пакет получает временную отметку, генерируемую на основе часов источника, которые работают с частотой, соответствующей используемой частоте передаче данных. Временные метки позволяют принимающей стороне правильно принимать данные, даже если пакеты прибывают с переменной задержкой или в неправильном порядке [71].

Следующим ключевым элементом архитектуры RTP являются порядковые номера пакетов (sequence numbers), увеличивающиеся на единицу с каждым отправленным пакетом, и решающие проблему возможного нарушения порядка поступления пакетов. Этот механизм позволяет обнаруживать потери пакетов и правильно собирать данные на приемной стороне.

Протокол RTP устраняет ключевой недостаток протокола UDP, связанный с нарушением последовательности доставки пакетов, за счёт механизмов временных меток и порядковой нумерации пакетов. Однако эти механизмы приводят к увеличению джиттера, который даже в лабораторных условиях обеспечивается на уровне сотен микросекунд, что превышает допустимые пределы для систем изохронной передачи данных в комплексах системах обработки сигналов с фазированными антенными решётками. Поэтому его применение для подобных систем является недопустимым.

1.2 Методы и средства синхронизации информационных потоков в архитектуре dataflow (потока данных)

Проблемы обеспечения изохронной передачи данных актуальны не только в компьютерных сетях, но и в вычислительных комплексах (суперЭВМ). Наиболее эффективно синхронизация информационных потоков

реализуется в суперЭВМ, архитектура которых основанная на принципах потока данных (dataflow).

Система архитектур потока данных [72], [73] подразделяется на систему архитектур потока токенов, системы с редуцированными моделями вычислений и системы архитектур потока операндов. При этом механизмы синхронизации данных в каждой архитектуре dataflow принципиально различные.

В основе архитектур потока токенов лежит механизм организации асинхронного вычислительного процесса с помощью токен-структур, используемых для хранения и идентификации разных операндов одной и той же операции. Токен представляет собой структуру данных, которая содержит инструкцию, операнды, особые метки (теги) и контекст, что в совокупности определяет, на каком типе исполнительного устройства должна выполняться операция, в какое время и при каких условиях [74]. В тот момент времени, когда найдена пара токенов с операндами и идентичным тегом, что показывает, что они предназначены для совместной обработки, они сразу передаются в исполнительные устройства для непосредственных вычислений. Можно сказать, что архитектура потока данных работает по готовности операндов.

Вычислительная система с архитектурой потока токенов (ВСПТ), представленная в упрощенном виде на рисунке 1.4, содержит набор исполнительных арифметико-логических устройств, объединенных некоторой коммутационной сетью [75].

Принцип работы представленной системы заключается в следующем: входные данные, представленные в виде токенов, загружаются в оперативную память. Коммутационные узлы КУ1 обеспечивают передачу валидных данных в исполнительные устройства ИУ из оперативной памяти через коммутационные узлы КУ2. В блоках анализа токена АТ определяется тип арифметико-логической операции вместе с иными служебными данными, необходимыми для организации корректного вычислительного процесса. После этого инструкция и операнды поступают в ИУ для непосредственной

обработки. Результат вычислений поступает в блоки формирования токенов ФТ, где происходит формирование новых токенов. Коммутационные узлы КУ2, помимо организации обратной связи, выполняют функции передачи токенов между массивом ИУ и записи токенов в оперативную память. По окончании вычислительного процесса производится выгрузка результата вычислений из оперативной памяти устройства.

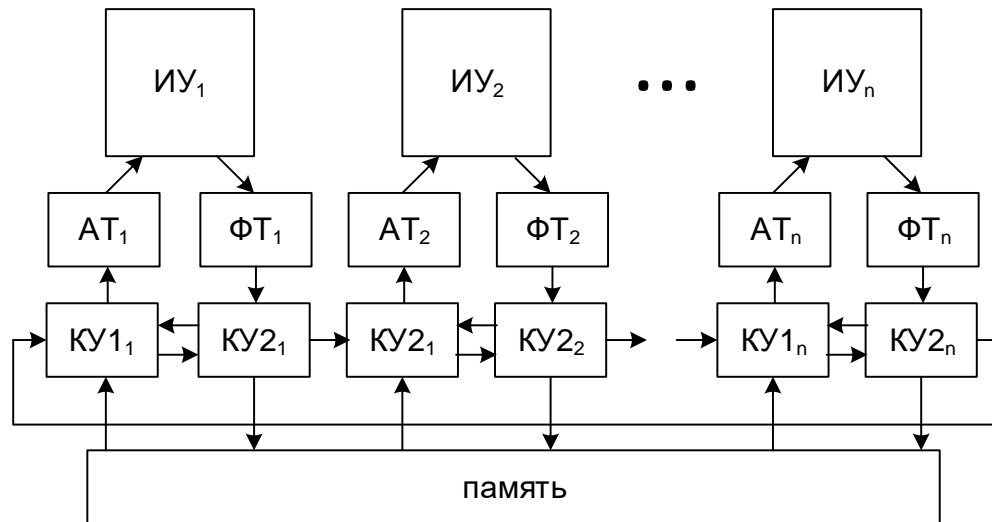


Рисунок 1.4. Упрощенная структурная схема архитектуры потока токенов

Архитектура потока токенов позволяет выполнять операции в порядке, определяемым потоковым графом, в котором вершины представляют собой операции, а дуги – не просто каналы данных, а связи типа ИУ–память–ИУ.

Следует отметить, что значительное время в работе архитектуры потоков токенов занимает организация синхронизации вычислительного процесса, подразумевающая создание токенов, организацию пересылок токенов между узлами вычислительной системы, сохранение операндов в оперативной памяти ВСПТ, поиск нужных операндов в памяти, поиск незанятого ИУ конкретного типа для выполнения непосредственных вычислений, организацию работы условных переходов, а также выполнение иных служебных операций. Поэтому большую часть времени исполнительные устройства простаивают. Необходимость постоянной работы с оперативной памятью, а также невозможность настройки архитектуры ВСПТ под

поточковый граф задачи в совокупности не позволяют организовать эффективные параллельные и конвейерные вычисления [75], [76].

Другой проблемой архитектуры потока токенов является необходимость для каждого поступающего токена выполнить проверку наличия второго операнда в оперативной памяти. Эта операция при решении реальных прикладных задач может на несколько порядков превышать время выполнения непосредственных вычислений [76]. Известно, что ускорить поиск соответствующего операнда можно за счет использования ассоциативной памяти (АП) [77], [78], структура которой приведена на рисунке 1.5.

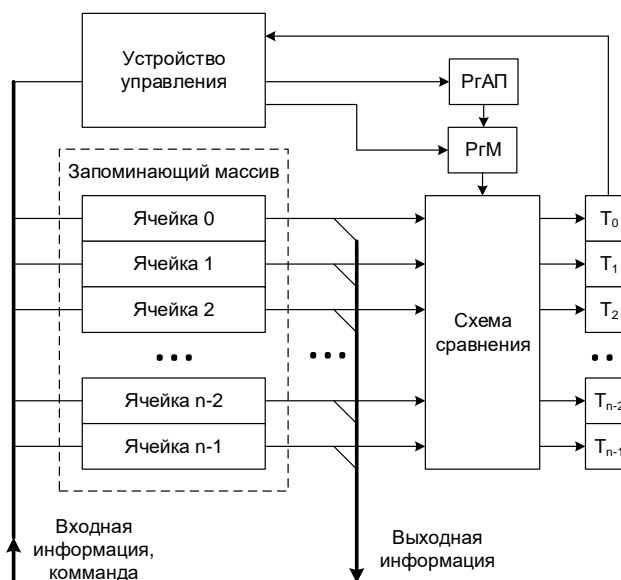


Рисунок 1.5. Схема ассоциативной памяти

В отличие от обычной оперативной памяти типа RAM, которая по заданному пользователем адресу возвращает одно слово данных, АП выполняет поиск данного в запоминающем массиве путем параллельного сравнения его содержимого с новым объектом. Если слово данных присутствует в запоминающем массиве, АП возвращает список из одного или более адресов хранения, по которым слово было найдено [78].

В общем виде ассоциативная память состоит из: запоминающего массива для хранения слов данных, в каждой ячейке которого несколько

разрядов отведены для хранения служебной информации; регистра ассоциативного признака РгАП, куда передается ключ для поиска; регистра маски РгМ, запрещающей сравнение определенных бит слов данных; схемы параллельного сравнения каждого бита всех хранимых слов с соответствующим битом ключа; регистры адресов Т, в которые записывается результат поискового запроса; устройство управления, которое обрабатывает поступившую команду, записывает регистры РгАП и РгМ, а также обрабатывает результат сравнения, записанный в регистры Т.

В настоящее время АП малой емкости повсеместно используются при реализации кэш–памяти универсальных процессоров [79]. Также существуют специализированные виды АП, которые, например, используются при отборе уникальных событий для анализа физических данных от детектора ATLAS [80]. Однако при увеличении емкости АП до уровня современной оперативной памяти (1–50 гигабайт), возникают значительные трудности с организацией. Параллельная реализация сравнений всех имеющихся ячеек массива с новым словом данных требует огромных аппаратных затрат. Вместе с этим энергоэффективность ассоциативной памяти такого объема будет ниже энергоэффективности самих вычислительных элементов.

В системах изохронной передачи данных эта проблема усугубляется тем, что необходимо синхронизировать не пару операндов, а множество операндов, соответствующих одному времени по всем каналам данных. В этом случае аппаратные/временные затраты возрастают кратно количеству синхронизируемых каналов. В этой связи использование механизмов токенов для синхронизации множества потоков операндов неэффективно.

Другой разновидностью архитектуры потока данных является редуцирующая вычислительная система [81]. В ее основе лежит представление вычислительного процесса в виде информационного графа, но механизм управления по запросу состоит в обработке вершин графа снизу–вверх. Данный процесс получил название редукиции графа. В редуцирующей системе вычисления производятся по запросу на результат операции. Известны два

типа моделей редуционных систем: строчная и графовая, отличающиеся тем, что именно передается в операцию – скопированные значения данных или же только указатели мест хранения данных.

Вычислительная система с архитектурой на основе редуционной модели вычислительного процесса (ВСРМ), представлена в упрощенном виде на рисунке 1.6.

ВСРМ представляет собой набор исполнительных устройств ИУ, объединенных коммутационной сетью, причем каждое устройство объединено каналами данных не только с одним соседним ИУ, но и с несколькими.

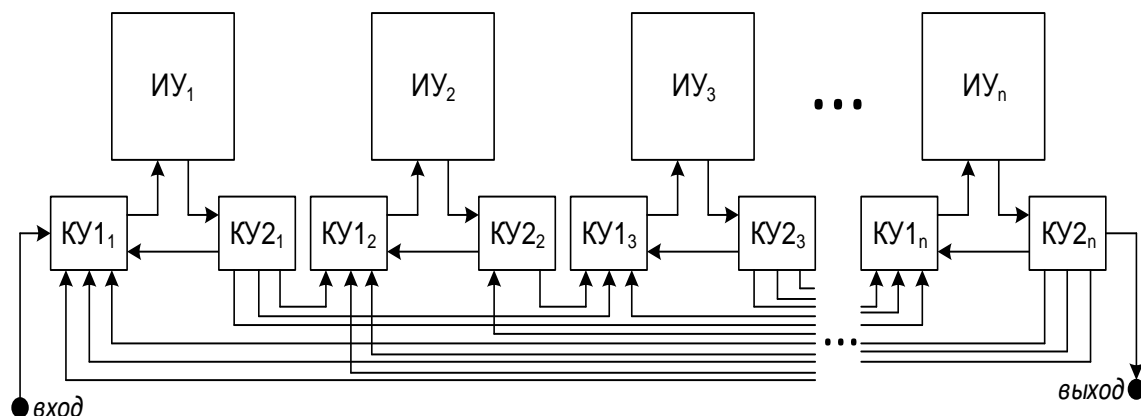


Рисунок 1.6. Упрощенная структурная схема архитектуры вычислительной системы на основе редуционной модели

Перед началом работы валидные данные и идентификаторы операций загружаются в ИУ ВСРМ в виде структур данных, содержащих особые синхронизирующие символы (метки, теги). Коммутационные узлы ВСРМ КУ1 выбирают источник, из которого поступают входные данные на обработку: из внешней памяти системы, из других ИУ, либо из системы обратной связи, которую реализуют коммутационные узлы КУ2. Вместе с этим в КУ2 осуществляется передача данных от конкретного ИУ в один из нескольких других ИУ системы.

Такая архитектура формально не содержит буферной памяти, поэтому результаты промежуточных вычислений «хранятся» в вычислительных узлах системы и постоянно курсируют по коммутационной сети между узлами так, что один вычислительный узел ВСРМ либо выполняет запрошенную операцию, либо хранит одно валидное данное.

Эффективность ВСРМ зависят от выбора модели редуccionных вычислений: графовой или строковой [82].

При реализации строковой редуccionной модели рекурсивный анализ потокового графа прикладной задачи выполняется непосредственно на массиве доступных исполнительных устройств, что делает невозможным расчет оптимальных путей передачи данных и значительно уменьшает число вычислительных узлов, используемых непосредственно для вычислений. В связи с этим архитектура ВСРМ на базе строковой редуccionной модели обладает низкой эффективностью.

В графовой редуccionной модели рекурсивный анализ алгоритма задачи выполняется заранее с помощью эшелонирования потокового графа задачи, в результате чего формируется программа, содержащая индексы используемых ИУ и ссылки на индексы тех ИУ, в которых будут сформированы результаты промежуточных вычислений. Формально графовая редуccionная модель позволяет работать по готовности операндов и теоретически может обеспечить выполнение конвейерных вычислений. Однако реализация ВСРМ на базе графовой редуccionной модели требует значительных дополнительных аппаратных затрат на систему *буферизации последовательности операндов*, инструкций, тегов и указателей на путь к требуемому ИУ. Более того, если число результатов промежуточных вычислений превышает число доступных ИУ, появляется необходимость в организации выгрузки данных в общую память системы, что является «узким горлом» архитектуры ВСРМ.

Основным недостатком при реализации системы изохронной передачи данных на базе редуccionной модели вычислительного процесса является

малое число каналов данных в системе между вычислительными блоками комплекса ЦОС, из-за чего вычислительные узлы будут выполнять полезную работу неэффективно, из-за высоких накладных временных затрат.

Можно утверждать, что увеличить число каналов обмена, практически невозможно, поскольку затраты на коммутацию возрастут квадратично, что явно не соответствует требованию минимизации затрат для системы изохронной передачи данных.

Альтернативой рассмотренным архитектурам dataflow является архитектура потока операндов (рисунок 1.7).

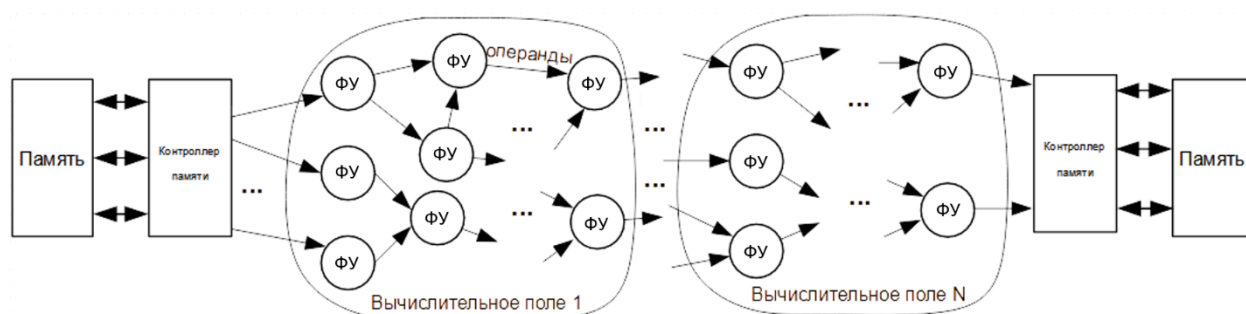


Рисунок 1.7. Архитектура потока операндов

Архитектура потока операндов представляет собой вычислительное поле функциональных устройств (ФУ), связанных между собой КС. При этом каждый операнд перемещается в процессе обработки между ФУ в соответствии с информационным графом задачи.

Функциональные устройства выполняют только информационно значимые преобразования. Все информационно незначимые операции, такие как синхронизация, коммутация, выбор источника данных, реализованы в КС путем организации пространственной коммутации и добавлением необходимых задержек при передаче данных между различными ФУ.

Для синхронизации в системе используется специальный блок операндов БО, структура которого представлена на рисунке 1.8 и состоит из динамических коммутаторов sw_i и sw_o , регистра хранения переменной rg , блока синхронизации потоков операндов $syncf$, интерфейса доступа к внешней

памяти IWR и блока управления ctrl, предназначенного для подсчета величины рассинхронизации, автоматического или принудительного управления режимами работы и формирования соответствующих управляющих сигналов.

Динамические коммутаторы swi и swo являются устройствами типа «мультиплексор/демультиплексор». Сигналы на их управляющих входах определяются результатами работы ФУ.

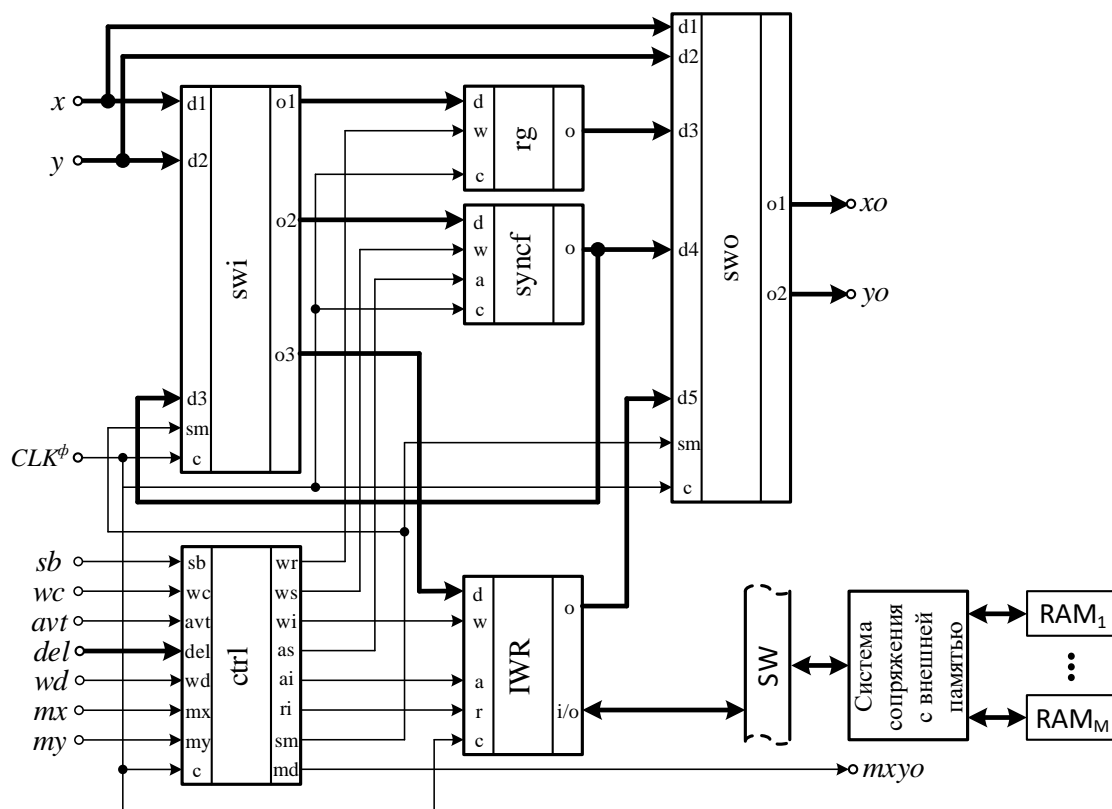


Рисунок 1.8. Структура блока операндов БО

Регистр rg в блоке операндов применяется при выполнении арифметико-логических операций над потоком данных и переменной, которая меняет своё значение за время, сопоставимое со временем обработки потока. При этом разрядность переменной не должна превышать разрядность rg. В данный регистр записывается константа на этапе синтеза программы, или же в процессе решения задачи записывается опережающий операнд.

С учетом возможного разнообразия сценариев синхронизации в блоке операндов предусмотрено четыре основных режима работы. Для этого в блок

ctrl записывается величина синхронизации del, значение которой определяется режимом работы. При этом устанавливается сигнал принудительной синхронизации avt=1. Если avt=0, то в процессе решения задачи определение режима работы будет выполняться автоматически в соответствии с величиной рассинхронизации между маркерами tx и ty потоков операндов x и y.

Режимы работы 1, 2 или 3 устанавливаются по значению величины рассинхронизации между потоками операндов. Режим 1 устанавливается, если в процессе решения задачи потоки операндов приходят синхронно или он установлен принудительно. В этом случае входные потоки операндов x и y подаются сразу на выходы o1 и o2 коммутатора swo. Режим 2 устанавливается при величине рассинхронизации до $(2l-1)$ тактов, где l – длина операндов, или принудительно при операциях с массивами констант объемом до $(2l-1)$ бит. Считается, что операнды подаются поразрядно последовательно. Опережающий поток операндов x (или y) через коммутатор swi подаётся на syncf, а затем синхронно с запаздывающим потоком операндов y (или x) поступает через коммутатор swo на выходы. Режим 3 устанавливается при величине рассинхронизации более $(2l-1)$ тактов или принудительно при операциях с массивами констант более $(2l-1)$ бит, хранимыми во внешней памяти.

Блок операндов обеспечивает автоматическую синхронизацию пары операндов с минимальными аппаратными/временными затратами в условиях недетерминированности информационных потоков. Однако при применении блоков операндов в системах изохронной передачи данных в цифровых обрабатывающих комплексах с фазированными антенными решетками, где требуется синхронизация операндов от множества параллельных каналов, возникает необходимость в каскадировании блоков операндов. При этом на каждом последующем каскаде объем буферной памяти в блоке операндов будет удваиваться, поскольку в буфер загружается не единичный опережающий операнд, а пара. На следующем каскаде требуется записать в

память опережающую группу уже из четырех операндов. Как следствие, происходит рост аппаратных и временных затрат, близкий к квадратичному.

Следует подчеркнуть, что в случае детерминированных задержек в каналах передачи данных возможна тривиальная установка буферов фиксированных размеров, пропорционально задержкам, заданным на этапе трансляции (синтеза схем). Однако в системах изохронной передачи данных в ЦОС с ФАР задержки имеют явно недетерминированный характер, что исключает использование подобных методов.

Таким образом, известные средства синхронизации в системах архитектуры dataflow не обладают необходимой эффективностью для обеспечения изохронной передачи данных. Все они требуют значительных аппаратных и временных затрат. Это обуславливает необходимость разработки и исследования новых методов и средств, способных гарантировать синхронизацию данных для множества каналов при минимальных аппаратных/временных затратах.

1.3 Анализ элементной базы для обеспечения когерентной обработки данных в комплексах ЦОС

Когерентная обработка данных в комплексах цифровой обработки сигналов предъявляет высокие требования к элементной базе вычислительных модулей, особенно в сложных системах, таких как устройства обработки данных от фазированных антенных решеток. Современные ФАР могут включать десятки тысяч антенных элементов, каждый из которых генерирует поток данных с частотой дискретизации в сотни мегагерц [83]. Это создает колоссальную вычислительную нагрузку, требующую специализированных вычислительных модулей с высокой производительностью [84].

Вычислительные модули должны обеспечить параллельную обработку данных от всех антенных элементов с минимальными задержками, выполнять

быстрое преобразование Фурье в реальном времени, динамически рассчитывать весовые коэффициенты для формирования диаграммы направленности и т.д.

Особую сложность представляет необходимость выполнения всех операций в режиме жесткого реального времени. Важно не просто обработать данные за определенный временной интервал, но и обеспечить постоянство этого времени. Учитывая, что период обновления антенной решетки может составлять всего несколько микросекунд, любая задержка сверх установленного лимита может привести к серьезным последствиям – от деградации характеристик до полной неработоспособности системы.

Вычислительных модули для когерентной обработки данных в комплексах ЦОС могут быть реализованы на различной элементной базе: на универсальных и сигнальных процессорах, специализированных интегральных схемах (ASIC) или программируемых логических интегральных схемах (ПЛИС).

Одним из эффективных способов решения сложных задач цифровой обработки сигналов является применение кластера, содержащего вычислители на основе универсальных процессоров с применением методов параллельных вычислений [85]. Основные преимущества такого кластера — это универсальность, поскольку стандартные процессоры поддерживают широкий спектр приложений, и масштабируемость, так как можно добавлять новые узлы для увеличения производительности. Подобные кластеры широко применяются в научных исследованиях, инженерных расчетах, обработке больших данных и других областях, где требуется высокая вычислительная мощность. Их гибкость и поддержка разнообразного программного обеспечения делают их популярным решением для многих задач высокопроизводительных вычислений.

Одним из примеров подобных систем служат кластерные решения AMD на базе процессоров EPYC, такие как Navion AMD EPYC 2U Twin Systems от компании Microway (рисунок 1.9) [86]. Эти высокопроизводительные

вычислительные системы предназначены для параллельной обработки данных и находят применение в различных сферах, включая высокопроизводительные вычисления, такие как научное моделирование и климатические прогнозы. Они также востребованы в области искусственного интеллекта и машинного обучения, где требуется обработка больших объемов данных, а также в задачах рендеринга и обработки медиаконтента, где важна высокая вычислительная мощность.



Рисунок 1.9. Microway Navion AMD EPYC 2U Twin Systems

Однако при их применении в реализации вычислительных узлов для когерентной обработки данных в комплексах ЦОС они имеют ряд существенных недостатков.

Первым недостатком является ограниченная вычислительная мощность микропроцессоров. Алгоритмы когерентной обработки информации от фазированных антенных решеток требуют значительных вычислительных ресурсов, поскольку включают в себя сложные математические операции, такие как быстрое преобразование Фурье, адаптивную пространственную фильтрацию, коррекцию фазовых и амплитудных искажений. Эти вычисления зачастую предполагают обработку больших объемов данных в режиме реального времени, что создает высокую нагрузку на процессор. Процессоры могут оказаться недостаточно производительными для эффективной

реализации таких алгоритмов. Последствия недостаточной производительности процессоров проявляется в увеличении времени обработки сигналов, что может привести к возникновению недопустимых задержек и нарушению условия обработки в реальном масштабе времени.

Вторым недостатком является ограниченная параллельная обработка. В задачах когерентной обработки информации от фазированных антенных решеток требуется параллельная обработка больших объемов данных, что бывает сложно реализовать на процессорах из-за их ограниченной способности к параллельной обработке. Сложность заключается в том, что для оптимальной работы с входными данными необходимо грамотно организовать планирование и управление вычислительными потоками. При многоканальной обработке каждый канал требует определенного объема ресурсов, и процессору приходится постоянно перераспределять их между различными задачами. Это создает дополнительную нагрузку на систему и может приводить к временным задержкам в обработке данных.

Более того, при попытке максимально задействовать все доступные ресурсы процессора возникают значительные накладные расходы, связанные с переключением между потоками и синхронизацией данных. В результате даже при наличии достаточного количества вычислительных мощностей эффективность обработки может существенно снижаться из-за несовершенства механизмов параллельной работы. Это особенно заметно при когерентной обработке информации от ФАР, где требуются высокая производительность и минимальные задержки при обработке поступающих данных.

Третьим недостатком универсальных процессоров с общей шиной памяти является ограниченная пропускная способность самой шины [87]. В ситуации, когда несколько ядер или потоков одновременно обращаются к памяти, шина становится узким местом, что негативно сказывается на общей производительности системы. В современных многопоточных и многозадачных сценариях каждый поток требует доступа к данным,

хранящимся в оперативной памяти. Если шина памяти не обладает достаточной пропускной способностью, возникают задержки, так как запросы начинают конкурировать за доступ к ограниченному ресурсу.

Кроме того, при высокой загрузке шины памяти может наблюдаться эффект насыщения, когда дальнейшее увеличение количества потоков не приводит к росту производительности, а иногда даже ухудшает её из-за накладных расходов на синхронизацию.

Наконец, еще одним недостатком универсальных процессоров в качестве базиса для реализации системы когерентной обработки является сложность их интеграции с аппаратными компонентами, особенно в системах, работающих с фазированными антенными решетками. Для работы таких систем необходима развитая периферия, способная обеспечить множество высокоскоростных каналов передачи данных с минимальными задержками. Кроме того, критически важным является наличие точных механизмов синхронизации, так как даже незначительные расхождения во времени обработки сигналов могут привести к фазовым ошибкам, что ухудшит характеристики диаграммы направленности антенны. Универсальные процессоры часто не справляются с такими требованиями. В результате чего на практике приходится использовать дополнительные аппаратные компоненты, что усложняет архитектуру системы и увеличивает ее стоимость.

В отличие от универсальных процессоров, цифровой сигнальный процессор (ЦСП) имеет узкую специализацию, ориентированную на эффективное выполнение задач цифровой обработки сигналов. Их архитектура оптимизирована для ускорения типовых операций, таких как фильтрация (как с конечной, так и с бесконечной импульсной характеристикой – КИХ и БИХ), дискретное и быстрое преобразования Фурье, а также других математических вычислений, характерных для цифровой обработки сигналов [88].

В сравнении с универсальными процессорами, современные ЦСП обеспечивают более высокую производительность при выполнении

алгоритмов цифровой обработки сигналов, не требуя при этом значительного увеличения тактовой частоты. Это позволяет добиться высокой вычислительной мощности без существенного роста энергопотребления. Цифровые сигнальные процессоры хорошо подходят для выполнения последовательных алгоритмов или алгоритмов, насыщенных циклами, а также при реализации сложных алгоритмов, требующих вычислений с плавающей точкой [89].

Примером современных ЦСП можно отнести процессор TMS320C6657 (рисунок 1.10), являющийся представителем семейства цифровых сигнальных процессоров от Texas Instruments, основанных на высокопроизводительной архитектуре KeyStone. Этот чип предназначен для обработки сложных алгоритмов цифровой обработки сигналов в реальном времени, а также для выполнения высоконагруженных вычислений в телекоммуникационных, промышленных и военных приложениях [90].



Рисунок 1.10. Вычислительный блок TMDSEVM6657L на базе сигнального процессора TMS320C6657

Процессор TMS320C6657 содержит два ядра, работающих на частоте 1,25 ГГц. Благодаря интегрированной многопоточной системе, нагрузка оптимально распределяется между ядрами. Процессор обладает развитой иерархией памяти, включая кэш L1, L2 и общую память L2 для межъядерного

обмена. Ядра построены на архитектуре C66x VLIW с поддержкой операций как в фиксированной, так и в плавающей точке, что делает чип универсальным для разных вычислительных задач [90].

Однако цифровые сигнальные процессоры имеют существенный недостаток при их применении в многоканальных вычислительных комплексах ЦОС. Основным недостатком является недостаточно развитая периферия для задач когерентной обработки данных от множества каналов.

Например, процессоры семейства C7x от Texas instruments имеют несколько высокоскоростных интерфейсов: PCI Express Gen3, Serial RapidIO Gen2, а также Gigabit Ethernet. Наиболее производительным из доступных интерфейсов является PCI Express Gen3 с конфигурацией x4 lanes, обеспечивающий реальную пропускную способность около 31,5 Гбит/с. Но даже такой пропускной способности недостаточно для систем, требующих одновременной обработки данных от сотен или тысяч каналов, как, например, в системах когерентной обработки данных от фазированной антенной решетки. В результате в один сигнальный процессор можно ввести лишь небольшое количество каналов, что вынуждает распределять обработку между множеством микросхем сигнальных процессоров.

Такое распределение неизбежно приводит к увеличению межпроцессорного обмена данными, что создает дополнительные аппаратные и временные затраты. Кроме того, возрастают требования к задержкам и синхронизации, что ограничивает масштабируемость системы и снижает ее эффективность при работе с большими объемами данных в реальном времени. Таким образом, можно утверждать, что сигнальные процессоры не ориентированы на многоканальную обработку и не могут служить в качестве элементной базы для построения систем когерентной обработки.

Альтернативой системам на основе универсальных и сигнальных процессоров в настоящее время являются системы, построенные на основе заказных интегральных схем специального назначения [91]. В отличие от универсальных и сигнальных процессоров, интегральные схемы специального

назначения проектируется под конкретные алгоритмы обработки данных, что позволяет достичь максимальной производительности при выполнении целевой задачи.

Современные технологии дают возможность интегрировать в единый кристалл аналоговые схемы обработки сигналов, цифровые конвейерные блоки, различные интерфейсы, встроенную память и вспомогательные модули. Это обстоятельство сокращает число дискретных компонентов в системе и облегчает проектирование печатной платы вычислительного устройства [92].

Примером ASIC, выполняющей задачи цифровой обработки сигналов является модем Snapdragon X50 от компании Qualcomm (рисунок 1.11) [93]. Модем Qualcomm X50 — это первый коммерческий 5G-модем от Qualcomm, представленный в 2016 году. Он предназначен для обеспечения высокоскоростной беспроводной связи пятого поколения в смартфонах и других устройствах.

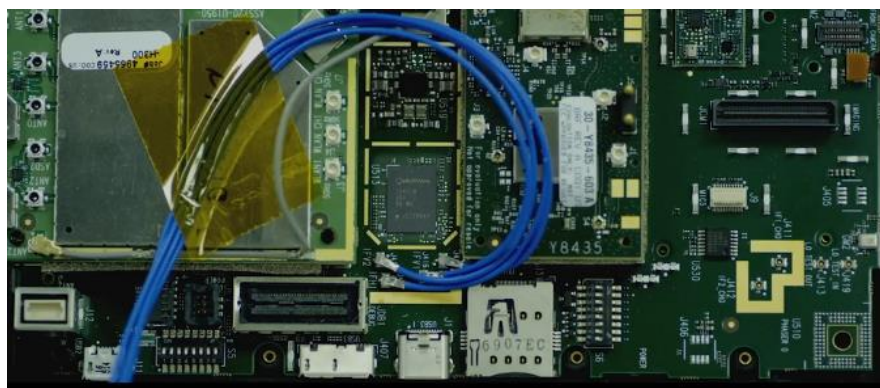


Рисунок 1.11. Модем Qualcomm Snapdragon X50

Модем X50 поддерживает работу в двух ключевых диапазонах 5G: mmWave (24,25–27,50 ГГц) и Sub-6 (3,4–3,8 ГГц). Теоретическая максимальная скорость загрузки данных достигала 5 Гбит/с, что в несколько раз превышало возможности LTE-сетей. Qualcomm Snapdragon X50 сыграл важную роль в развитии 5G, ускорил развертывание 5G-сетей и заложило основу для современных высокоскоростных беспроводных сетей [93].

Можно утверждать, что интегральные схемы специального назначения являются лучшим решением по показателям производительности и энергоэффективности, однако им присущ ряд существенных недостатков.

Основной проблемой ASIC является высокая трудоемкость их разработки и производства, что приводит к значительным финансовым и временным затратам. Полный цикл создания даже относительно простых микросхем занимает от 12 до 18 месяцев. Это связано с необходимостью тщательного проектирования, верификации, изготовления прототипов и их тестирования. В отраслях с высокой динамикой развития технологий, за это время стандарты и алгоритмы могут измениться, и ASIC может устареть ещё до появления на рынке.

Стоимость производства ASIC складывается из единовременных инженерных затрат (NRE), не зависящих от объема выпуска, и переменных расходов, определяемых объемом производства. При небольших партиях NRE составляет значительную часть себестоимости каждого кристалла. Например, если затраты на разработку составили пять миллионов, а выпущено всего 10 тысяч кристаллов, то на каждый из них приходится 500 только за счет NRE. В случае массового производства, исчисляемого миллионами экземпляров, эти затраты распределяются, и себестоимость одного кристалла снижается.

Таким образом, применение ASIC экономически оправдано лишь при очень больших объемах выпуска. Эти микросхемы целесообразно использовать в массовой продукции, но для проблемно-ориентированных решений, таких как вычислительные блоки ЦОС для обработки данных с фазированных антенных решеток, их применение приведет к неоправданному росту стоимости конечного изделия.

Еще одним важным недостатком заказных микросхем является невозможность изменения функциональности после производства. Структура ASIC жёстко зафиксирована в процессе изготовления кристалла. Физическая природа этого ограничения кроется в самом процессе производства ASIC. На этапе фотолитографии создаются транзисторы и соединения между ними,

формируя строго определённую схему. Попытка изменить алгоритмы его работы эквивалентна необходимости перепроектировать и заново изготовить весь кристалл - процесс, требующий повторения всего цикла разработки с соответствующими временными и финансовыми затратами.

В качестве альтернативы ASIC можно использовать программируемые логические интегральные схемы (ПЛИС). ПЛИС представляют собой электронные компоненты, предназначенные для построения цифровых схем. Их внутренняя архитектура и логика работы определяются в процессе программирования (конфигурации) [94]. Фактически это полуфабрикаты сверхбольших интегральных схем, внутреннюю структуру которых пользователь может настраивать под конкретную задачу.

В 1970-х годах были созданы первые реконфигурируемые интегральные схемы - программируемые логические матрицы. Они позволяли реализовать систему логических функций n элементов за счет пережигания перемычек с помощью специальных устройств. Позднее разработали базовые матричные кристаллы, состоящие из набора базовых ячеек с коммутационными зонами между ними [95].

В результате эволюции программируемой электроники появились ПЛИС (FPGA, программируемые пользователем вентильные матрицы). Эти устройства состоят из множества программируемых логических модулей LUT, выполняющих логические функции, и триггерных элементов памяти FF, соединённых между собой разветвлённой системой программируемых связей. Современные ПЛИС включают в себя значительное количество встроенных блоков памяти различного размера, арифметико-логических модулей (DSP), высокоскоростных интерфейсов и периферийных устройств для передачи данных и других функций [96]. Это дает возможность аппаратно реализовывать в рамках одной микросхемы самые разные схемотехнические решения - от универсальных процессоров и схем цифровой обработки сигналов до нейросетей, интерфейсных схем и сложных специализированных

вычислителей. При этом пользователь может переконфигурировать их в соответствии с текущими задачами.

В основном ПЛИС применяют в качестве сопроцессоров или ускорителей для определённых типов вычислений. Однако даже самые высокопроизводительные ПЛИС не способны обеспечить достаточную мощность для цифрового формирования множества диаграмм направленности в реальном времени от фазированных антенных решёток с десятками тысяч элементов. Для таких задач требуется объединение сотен или даже тысяч ПЛИС, соединённых высокоскоростными линиями передачи данных, поскольку только во всей совокупности ПЛИС можно сформировать вычислительную структуру необходимой производительности.

Один из возможных подходов - создание реконфигурируемых вычислительных систем (РВС) с масштабируемой архитектурой на основе ПЛИС [97], [98]. В таких системах вычислительное поле состоит из множества ПЛИС, связанных между собой высокоскоростными каналами обмена данными.

Примером таких систем может служить реконфигурируемая проблемно-ориентированная вычислительная система (РПВС) для цифровой обработки сигналов «Орфей-Т», разработанная НИЦ СЭ и НК [99], где проектируют и производят уникальные высокопроизводительные РВС на базе ПЛИС Xilinx [100].

Реконфигурируемая проблемно-ориентированная вычислительная система «Орфей-Т» (рисунок 1.12) предназначена для моделирования процессов цифрового формирования и управления множеством диаграмм направленности в радиолокаторах с фазированными антенными решетками различных конфигураций. В состав системы входит 16 блоков пространственной обработки (БПО) и 16 блоков временной обработки (БВО), каждый из которых содержит один базовый модуль «ММ475» (рисунок 1.13).

Базовый модуль «ММ475» содержит восемь ПЛИС Virtex-6 «XC6VSX475T» и имеет производительность свыше $0,6 \cdot 10^{15}$ оп/с. ПЛИС

объединены LVDS-каналом из 152 линий, с суммарной скоростью передачи 120 Гбит/с. Внешние коммуникационные возможности базового модуля «ММ475» обеспечиваются наличием 256 независимых интерфейсов на основе RocketGTX, работающих на частотах до 5 ГГц с пропускной способностью до 1 Тбит/с.

РПВС «Орфей-Т» может имитировать 4096 антенных элементов АФАР. Производительность вычислительной части РПВС «Орфей-Т» составляет 27,52 Тфлопс.

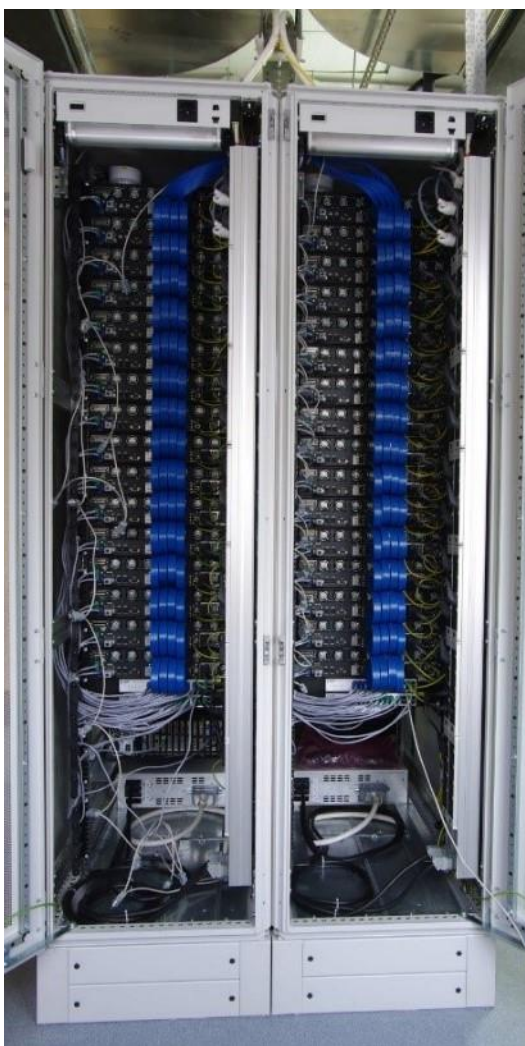


Рисунок 1.12. Реконфигурируемая проблемно-ориентированная вычислительная система «Орфей-Т»

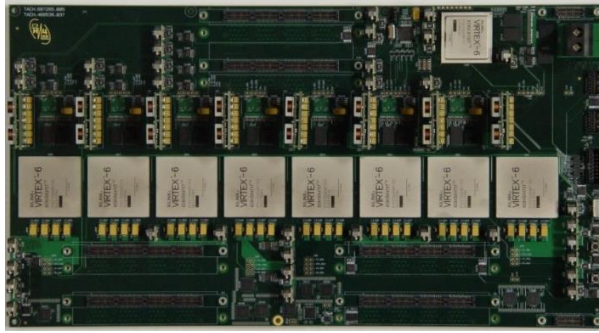


Рисунок 1.13. Многопроцессорный модуль «ММ475» РВС «Орфей-Т»

Перспективной разработкой НИЦ СЭ и НК, применимых для задач когерентной цифровой обработки сигналов, являются реконфигурируемые вычислительные блоки «Арктур».

Разрабатываемый РВБ «Арктур» представляет собой функционально законченное изделие с 16-ю вычислительными модулями на основе ПЛИС Xilinx XCVU35P (рисунок 1.14). В качестве основных связей между ПЛИС используются дифференциальные линии с подключенными к ним мультигигабитными трансиверами (MGT); вспомогательными связями являются дифференциальные линии, подключенные к НР-банкам.



Рисунок 1.14. Вычислительная система нового поколения производства НИЦ СЭ и НК РВБ «Арктур»

В вычислительном модуле ПЛИС связаны «горизонтальными» и «вертикальными» линиями связей. «Горизонтальные» связи между ПЛИС

представлены 28-ю дифференциальными парами со скоростью передачи данных до 25 Гбит/с по одной паре, что обеспечивает пропускную способность до 700 Гбит/с независимо в каждом направлении.

Также имеется канал, связывающий по кольцу первую и последнюю ПЛИС в ВМ с помощью 12-ти дифференциальных пар и со скоростью до 300 Гбит/с независимо в каждом направлении.

Дополнительно имеются 48 дифференциальных линий с максимальной скоростью до 1,2 Гбит/с, каждая из которых может работать в любом направлении.

К «вертикальным» связям относятся 192 дифференциальные пары с допустимой скоростью передачи данных до 20 Гбит/с по одной паре, что обеспечивает пропускную способность до 3840 Гбит/с независимо в каждом направлении.

Кроме того, между платами существуют связи, предназначенные для служебных целей, таких как загрузка конфигурации ПЛИС, передача команд и данных от процессора к любому вычислительному модулю. Пропускная способность данного канала – 64 Гбит/с для обоих направлений независимо.

Общая пропускная способность каналов связи ВМ – 13 Тбит/с, в том числе между ВМ – 7,68 Тбит/с.

Важными достоинствами реконфигурируемых вычислительных систем, разрабатываемых в НИЦ СЭ и НК, являются:

- динамически изменяемая в процессе функционирования архитектура, позволяющая настраивать вычислительное поле системы под структуру алгоритма решаемой задачи, что обеспечивает РВС высокую реальную производительность, близкую к пиковой, в том числе при решении вычислительно-трудоемких задач цифровой обработки сигналов;

- большое число аппаратных ресурсов для построения различных алгоритмов цифровой обработки сигналов и множество внешних интерфейсов дают возможность организовать в системах ЦОС многопоточную когерентную обработку огромного количества данных;

- высокая эффективность при решении вычислительно-трудоемких задач, включая задачи потоковой и многоканальной обработки данных;

- возможность обеспечить минимальную задержку при обработке данных, благодаря возможности оптимизации логики и прямого доступа к аппаратным ресурсам. Это особенно критично для многоканальных систем в реальном времени, где даже небольшие задержки могут быть неприемлемы;

- сокращение времени и финансовых затрат на разработку за счет возможности использования высокоуровневых языков программирования, более быстрой отладки и модификации функционала.

Все эти особенности делают реконфигурируемые вычислительные системы на основе ПЛИС перспективным направлением для построения комплексов когерентной цифровой обработки сигналов.

1.4 Принципы обеспечения гарантоспособной изохронной передачи данных в комплексах ЦОС

В некоторых задачах цифровой обработки сигналов необходима когерентная обработка, предполагающая совместную обработку данных от множества источников в реальном времени. Примером такой обработки является формирование диаграммы направленности фазированной антенной решетки, где требуется синхронная обработка сигналов от всех антенных элементов, принятых в один и тот же момент времени.

Это накладывает строгие требования на элементную базу: устройства должны обладать достаточными вычислительными ресурсами для параллельной обработки множества каналов, а также средствами синхронизации, чтобы гарантировать передачу данных, соответствующих одному моменту времени. Как показал анализ элементной базы для построения систем когерентной обработки данных в комплексах ЦОС, приведенный в предыдущем параграфе, рациональным является

использование ПЛИС, поскольку они предлагают множество внешних интерфейсов и значительные аппаратные ресурсы, позволяющие одновременно реализовывать алгоритмы ЦОС и механизмы синхронизации потоков данных.

Благодаря гибкости настройки логики и прямому доступу к аппаратным ресурсам, ПЛИС обеспечивают минимальные задержки при обработке, что критически важно для работы в режиме реального времени. Однако, учитывая необходимость обработки десятков тысяч каналов с высокой вычислительной нагрузкой и множеством высокоскоростных интерфейсов, когерентную обработку целесообразно реализовать на множестве ПЛИС.

Таким образом, **первый принцип** заключается в следующем: *«Для обеспечения когерентной обработки информации и изохронной передачи данных от множества источников с минимальными задержками целесообразно использовать PBC, построенную на основе множества ПЛИС».*

Задачу обеспечения когерентной обработки данных от множества источников можно решить в ПЛИС процедурно, реализовав множество процессоров на кристалле ПЛИС, но в таком случае не удастся обеспечить обработку в темпе поступления с минимальными задержками из-за высоких накладных расходов на управление потоками и синхронизацию.

Альтернативой является структурно-процедурная организация вычислений, при которой весь информационный граф задачи отражается на имеющийся аппаратный ресурс. В информационном графе задачи вершины соответствуют исходным данным, операциям над данными и результатам информационного преобразования. Дуги информационного графа соответствуют информационным связям алгоритма.

Такой подход обеспечивает естественный параллелизм и предсказуемость временных характеристик, поскольку каждый вычислительный элемент соответствует отдельному узлу графа, а связи между ними реализуются на аппаратном уровне.

В таком случае когерентная обработка данных может быть структурно-процедурно реализована на имеющемся аппаратном ресурсе РВС, что позволит минимизировать временные затраты за счет снижения влияния программных накладных расходов. Кроме того, структурно-процедурная организация позволяет эффективно использовать конвейеризацию и параллелизм, что критически важно для обработки данных в реальном времени.

Отсюда следует **второй принцип**: *«Когерентная обработка данных на РВС должна быть построена на основе структурно-процедурной организации вычислений»*.

Важным условием когерентной обработки данных от множества источников является совместная обработка данных от всех без исключения источников, соответствующих одному моменту времени. Для этого, прежде всего, необходимо обеспечить синхронизацию всех узлов комплекса ЦОС для того, чтобы обеспечить единовременное формирование потоков данных от всех источников и обеспечить привязку потоков данных к физическому времени.

Синхронизацию можно осуществить, используя систему единого времени, сигнал которой будет распространяться от единого источника машинного времени во все узлы комплекса ЦОС по линиям с одинаковой задержкой. Однако такой подход не обеспечит необходимой точности привязки времени для потока данных на частоте сотен мегагерц. Основная проблема заключается в различиях характеристик тактовых генераторов, установленных в разных блоках комплекса ЦОС. Даже при использовании качественных компонентов неизбежны различия в их характеристиках, а также дрейф параметров в процессе работы. В результате возникают фазовые ошибки, нарушающие когерентную обработку данных.

Чтобы устранить это явление, целесообразно комбинировать систему единого времени с системой опорной тактовой частоты, которая будет распространяться из единого центра во все узлы, обеспечив синфазность

тактовых генераторов всех блоков ЦОС, что обеспечит необходимую точность хода машинного времени во всех узлах комплекса ЦОС.

Отсюда следует **третий принцип**: *«Необходимая точность синхронизации для когерентной обработки данных в комплексах ЦОС может быть обеспечена только совмещением систем единого машинного времени и опорной тактовой частоты»*.

Для обеспечения когерентной обработки необходимо согласованно обработать информацию, полученную от множества источников, соответствующую одному моменту физического времени. В процессе передачи данных от блоков приема к блокам цифровой обработки сигналов неизбежно возникают временные рассогласования между информационными потоками. Эти задержки связаны с разбросом параметров компонентов канала данных, включая различия в длине линий связи и дрейф характеристик в процессе работы. Дополнительными факторами, влияющими на разброс значения задержек между разными каналами, являются процессы тренировки канала, в ходе которой каждый приемник данных независимо от других вставляет такты синхронизации. Для обеспечения изохронной передачи данных необходимо обеспечить выравнивание задержек информационных потоков в пределах каждой ПЛИС, осуществляющей прием данных.

Однако реализация выравнивания задержек в непрерывном информационном потоке требует сложных аппаратных решений, таких как глубокие буферы и/или сложные схемы синхронизации. Рациональным решением является сегментация потока данных, в этом случае непрерывный информационный поток разбивается на фиксированные массивы операндов. Для информационных потоков каждого канала используется буферная память. Запись в буфер происходит асинхронно по мере поступления массивов операндов от каждого канала, а чтение выполняется синхронно, обеспечивая выравнивание задержек.

Такой подход позволяет обеспечить предсказуемую латентность и снизить требования к буферной памяти, которая является критическим

ресурсом, так как требуется в большом количестве для реализации алгоритмов ЦОС.

Отсюда следует **четвертый принцип**: *«Для гарантоспособной передачи данных необходимо разбить поток данных на массивы операндов, обеспечить их синхронизацию с помощью буферной памяти».*

При увеличении количества каналов передачи данных наблюдается пропорциональный рост вероятности возникновения аварийных ситуаций, связанных с передачей данных. Простейший способ обеспечения надежности заключается в третировании каналов данных с последующим мультиплексированием по мажоритарному принципу. Однако такой подход ведет к значительному увеличению аппаратных затрат.

Использование процедуры повторного запроса данных также не является оптимальным решением, поскольку оно неизбежно приводит к увеличению временных задержек. Это противоречит требованиям систем реального времени, где критически важна обработка данных в темпе их поступления без дополнительных задержек.

В таких условиях необходимо применение специализированной разметки массивов операндов, способной идентифицировать ошибки передачи данных в реальном времени, например, с использованием контрольных меток размещенных в потоке данных.

Таким образом, формулируется **пятый принцип**: *«Для идентификации ошибок передачи данных в реальном времени необходима специальная разметка массивов операндов, размещенная в информационном потоке».*

На основе предложенных принципов будут разработаны методы и программные инструментальные средства для обеспечения изохронной передачи данных в комплексах ЦОС.

1.5 Выводы

1. Когерентная обработка предполагает совместную обработку данных, соответствующих одному моменту времени от множества источников. Для этого требуется обеспечение изохронной передачи данных от множества источников данных (блоков оцифровки) в блоки ЦОС, осуществляющие когерентную обработку.

2. Анализ показал, что протоколы сетевых технологий не обеспечивают изохронную передачу данных в комплексах ЦОС, содержащих множество информационных каналов, передающих данные на высоких частотах, из-за механизмов повторных запросов, возможного нарушения последовательности поступления пакетов данных, высокого уровня джиттера.

3. Известные решения синхронизации информационных потоков для архитектур потока данных (dataflow) при обеспечении изохронной передачи данных от множества информационных каналов требуют значительных аппаратных и временных затрат, что приводит к низкой эффективности работы комплекса ЦОС.

4. Был проведен анализ элементной базы для средств обеспечения когерентной обработки данных. Анализ показал, что универсальные процессоры не подходят для использования в комплексах ЦОС с большим числом информационных каналов, так как имеют недостаточную вычислительную мощность, ограниченную параллельную обработку, сложность их интеграции с аппаратными компонентами комплексов ЦОС.

5. Сигнальные процессоры не ориентированы на многоканальную обработку и не могут обеспечить высокую эффективность при построении комплексов ЦОС с большим количеством информационных каналов и высокой частотой передачи данных.

6. Применение ASIC экономически оправданно лишь при очень больших объемах выпуска. Такие микросхемы целесообразно использовать в массовой продукции, но для проблемно-ориентированных решений, таких как

вычислительные блоки ЦОС для обработки данных с фазированных антенных решеток, их применение приведет к неоправданному росту стоимости конечного изделия.

7. Современные ПЛИС содержат большое число аппаратных средств (ресурсов) для эффективной реализации алгоритмов цифровой обработки сигналов и множество внешних интерфейсов, что дает возможность организовать в комплексах ЦОС многопоточную когерентную обработку огромного количества данных.

8. Реконфигурируемые вычислительные системы на основе множества ПЛИС являются рациональным решением для построения комплексов когерентной цифровой обработки сигналов с большое число информационных каналов. В то же время отсутствуют методы и средства обеспечения надежной (гарантоспособной) передачи данных и синхронизации множества каналов для РВС.

9. Были разработаны принципы обеспечения гарантоспособной изохронной передачи данных в комплексах ЦОС:

- для обеспечения когерентной обработки информации и изохронной передачи данных от множества источников с минимальными задержками целесообразно использовать РВС, построенную на основе множества ПЛИС;

- когерентная обработка данных на РВС должна быть построена на основе структурной организации вычислений;

- необходимая точность синхронизации для когерентной обработки данных в комплексах ЦОС может быть обеспечена только совмещением систем единого машинного времени и опорной тактовой частоты;

- для гарантоспособной передачи данных необходимо разбить поток данных на массивы операндов обеспечить их синхронизацию с помощью буферной памяти;

- для идентификации ошибок передачи данных в реальном времени необходимо применить специальную разметку массивов операндов, размещенную в информационном потоке.

2 КОМБИНИРОВАННЫЙ МЕТОД ИЗОХРОНОЙ ПЕРЕДАЧИ ДАННЫХ

2.1 Комбинированный метод изохронной передачи данных

Для ряда задач ЦОС требуется когерентная обработка данных, которая предполагает совместную обработку данных от множества каналов, полученных в один момент физического времени. Ключевым применением когерентной обработки является обработка данных, поступающих от антенных элементов фазированных антенных решеток [7]. Для получения одного отсчета сигнала y для i -й диаграммы направленности ФАР, требуется выполнить следующее преобразование:

$$\dot{y}_t^i = \sum_{n=1}^N W_n^i \cdot x_t^n, \quad (2.1)$$

где i – номер диаграммы направленности;

W – коэффициент диаграммоформирования;

x – оцифрованные данные от антенного элемента;

N – количество антенных элементов ФАР;

n – номер антенного элемента.

Из вышесказанного можно сделать вывод, что для формирования диаграммы направленности устройствах ЦОС с ФАР критически важно обеспечить одновременное поступление данных в узлы обработки, соответствующих одному моменту физического времени (такту оцифровки) t от всех без исключения антенных элементов ФАР. Такой тип передачи данных, который гарантирует поступление данных, соответствующих одному моменту физического времени, от множества передающих узлов в узел когерентной цифровой обработки сигналов, в дальнейшем будем называть изохронным.

Обеспечение изохронной передачи данных является сложной научной и технической задачей, для которой требуется синхронизация всех узлов, осуществляющих прием и передачу информации. Здесь под синхронизацией

понимается обеспечение согласованной во времени работы всех источников и приемников данных.

Известно несколько методов для синхронизации узлов комплекса ЦОС.

- все узлы могли бы работать от одного физического генератора. Этот вариант подходит только для систем, которые разнесены друг от друга на небольшое расстояние. Передать высокочастотный сигнал на большие расстояния без потерь невозможно;

- все узлы могли бы использовать собственные генераторы с практически идентичными характеристиками. Этот подход не реализуем на практике из-за того, что найти подобные генераторы и предотвратить дрейф их характеристик во времени очень сложно. И, что особенно важно, рабочие условия генераторов неизбежно будут различаться.

Хорошо зарекомендовавшим себя способом обеспечения синхронизации узлов комплекса ЦОС является использование системы единого времени. Система единого времени состоит из первичных (ведущих) часов и вторичных (ведомых) часов, связанных между собой и позволяющих установить единое и точное время на всех подключенных вторичных часах [101].

Необходимость создания таких систем назрела в начале 1950-х годов в ракетных войсках СССР при испытаниях первых баллистических ракет дальнего действия [102]. Сверхвысокая скорость ракет требовала оперативного и сверхточного управления траекторией полёта в любой момент движения. На начальных этапах испытаний система единого времени полигонного комплекса представляла собой кварцевый генератор частоты, выдающий секундные сигналы по проводам и на радиостанцию для их передачи на удалённые объекты. Такая система являлась односторонней и справлялась со своей задачей, так как требования по дальности и точности пусков были не очень высокими.

Система единого времени получила широкое применение и в других областях науки и техники, таких как космические полеты, авиация, навигация,

радиоастрономия, связь, системы телекоммуникации, системы глобального позиционирования, контрольно-измерительное оборудование, промышленные сети и др. [103].

Системы единого времени постоянно развивались, появилось множество технологий синхронизации времени, вот некоторые из них:

- GPS (Global Positioning System) – глобальная система позиционирования. Синхронизация времени осуществляется во время определения местонахождения устройства, оснащенного GPS-приёмником. Для этого устройство принимает сигнал со спутников, установленных на околоземной орбите. Каждый из спутников имеет атомные часы, за счет чего система GPS обеспечивает хорошую точность. Минусом данного метода является необходимость в GPS-антенне, сигнал от которой может быть нестабильным [104];

- 1PPS (1 pulse per second) – сигнал 1PPS не содержит метки времени. Master-устройство посылает один импульс в секунду по отдельной сети: оптоволоконной линии, витой паре или коаксиальному кабелю. Часы Slave используют этот импульс только для синхронизации начала каждой секунды. Устройства не могут с помощью 1PPS получить информацию по дате и времени, поэтому его чаще всего используют совместно с другими протоколами синхронизации, например NTP [105];

- NTP (Network Time protocol) – протокол сетевого времени широко распространен в сетях Ethernet и Интернет. Принцип работы NTP основан на многоуровневой системе с множеством источников времени [106];

- IRIG-B (Inter Range Instrumentation Group) – с помощью данной технологии передается информация о дате и времени вместе с импульсным сигналом синхронизации. IRIG-B используют выделенную сеть для передачи информации. Сеть может быть построена на оптическом волокне, витой паре или коаксиальном кабеле [107];

- SNTP (Simple Network Time Protocol) – простой протокол сетевого времени. Применяется в локальных сетях для некритичных ко времени

приложений. Формат сообщений, которыми между собой обмениваются устройства в системах SNTP и NTP, идентичен, поэтому протоколы совместимы друг с другом. В отличие от NTP, у SNTP нет сложных алгоритмов сравнения и выбора наилучшего сервера времени, поэтому устройство может быть синхронизировано только с одним сервером времени, и если данные на сервере ошибочные, то конечное устройство не узнает об этом [108].

Точность установки времени, обеспечиваемая вышеописанными технологиями, представлена в таблице 2.1.

Таблица 2.1 Точность установки времени различными технологиями синхронизации

| Название | Точность |
|----------|---------------|
| GPS | ~ 1 мкс |
| 1PPS | ~ 1 мкс |
| NTP | 10 мкс – 1 мс |
| IRIG-B | 0,5 - 100 мс |
| Sntp | 1 – 50 мс |

Из открытых источников известно, что типовое значение частоты оцифровки данных от антенных элементов ФАР в комплексах когерентной цифровой обработки сигналов составляет около 100 МГц. Для получения наиболее точной привязки времени к оцифрованным данным необходима точность привязки времени не менее чем один такт шкалы времени на один такт оцифровки. Следовательно, дискрета шкалы времени должна быть не менее этого значения, а это значит, что необходимо обеспечить точность установки времени в разных узлах менее одного периода частоты шкалы времени, то есть 10 нс для частоты 100 МГц.

На практике это время еще меньше из-за необходимости учитывать время установки (setup) и время удержания (hold) (рисунок 2.1).

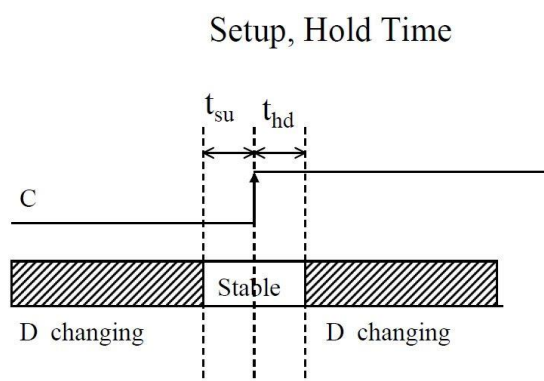


Рисунок 2.1. Временная диаграмма времени установки и удержания

Эти параметры определяют временные интервалы, в которые входные сигналы должны оставаться стабильными относительно тактового сигнала. Время setup указывает на то, как долго перед поступлением тактового сигнала должен быть стабилизирован входной сигнал, а время hold определяет, как долго после окончания тактового сигнала входной сигнал должен оставаться стабильным. Эти параметры критически важны для правильной фиксации сигналов в цифровых системах. Обычно в цифровой электронике для времени setup и hold выделяется примерно 10% от длительности тактового сигнала [109]. Это позволяет обеспечить достаточное время для установки и удержания входных сигналов, чтобы гарантировать правильное считывание данных и предотвращение ошибок в системе. С учетом времени установки и удержания точность установки времени во всех узлах не должна превышать 8 нс для частоты 100 МГц. Из чего следует, что точность методов, приведенных в таблице 2.1, в контексте их применения для обеспечения синхронизации современных систем когерентной обработки информации от ФАР является недостаточной.

Наиболее перспективным протоколом на данный момент является протокол RTP (англ. Precision Time Protocol — «протокол точного времени»)

[110]. РТР был изначально определён как стандарт IEEE 1588-2002 с официальным названием «Стандарт протокола синхронизации точного времени для сетевых измерительных систем и систем управления» и опубликован в 2002 году. В 2008 году была выпущена новая редакция стандарта IEEE 1588-2008. В стандарте определены алгоритмы выбора главных часов, определения задержек и их компенсация, а также процесс обмена сообщениями.

В стандарте IEEE 1588 применяются несколько типов устройств:

- Grandmaster clock (Гроссмейстерские часы) - основной источник точного времени;

- Ordinary Clock (Обычные часы) - устройство с одним портом, которое может быть ведущими часами или ведомыми часами;

- Boundary Clock (Граничные часы) - устройство с несколькими портами, которое может быть ведущим или ведомым. То есть эти часы могут синхронизироваться от вышестоящих ведущих часов и синхронизировать нижестоящие ведомые часы.

Передача сигнала точного времени одним из устройств Grandmaster clock остальным устройствам происходит в широковещательном режиме. Остальные устройства при таком подходе корректируют свои часы в соответствии с информацией, полученной от задающего синхронизацию устройства. Высокую точность установки времен в конечных устройствах обеспечивают алгоритмы определения и компенсации задержек. РТР подразумевает двухсторонний обмен сообщениями с метками времени. На основе полученных меток времени рассчитывается задержка. Алгоритм синхронизации ведомых устройств проиллюстрирован на рисунке 2.2.

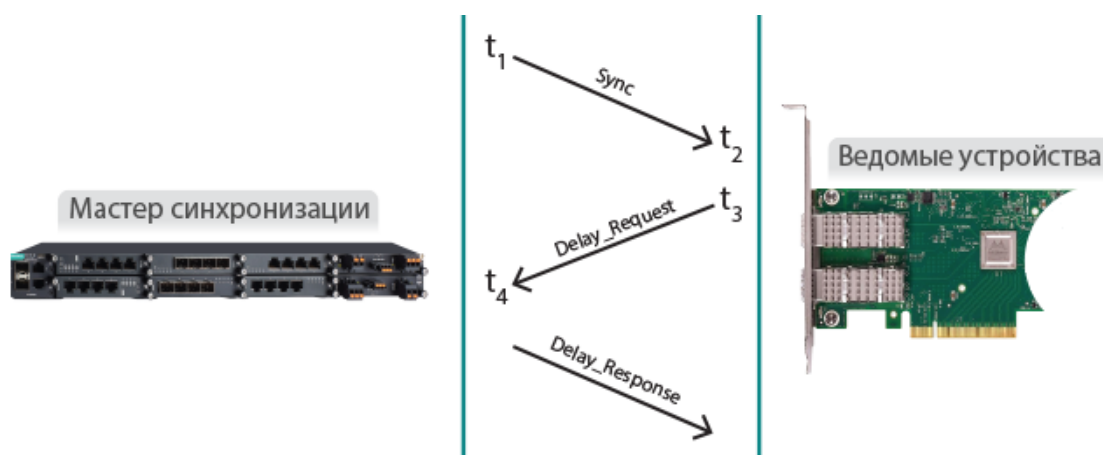


Рисунок 2.2. Алгоритм синхронизации ведомых устройств по протоколу PTP

С помощью данного метода синхронизации с использованием протокола IEEE 1588 удается достигать точности синхронизации около 100 нс [111]. Однако у этого метода есть недостатки.

Во-первых, различные устройства имеют собственные тактовые генераторы, которые не могут обладать одинаковыми характеристиками. Таким образом, временные метки, полученные от ведущих часов, могут захватываться разными фазами тактового сигнала в различных устройствах. К тому же имеет место дрейф частоты тактовых генераторов в различных ведомых устройствах. Это может привести к тому, что в интервалах между обновлениями времени от ведущего устройства время в различных ведомых устройствах может расходиться.

Во-вторых, для определения задержки распространения сигналов синхронизации используется расчет круговой задержки, для этого необходимо, чтобы задержка распространения сигнала от ведущего к ведомому была равна задержке распространения ответного сигнала от ведомого к ведущему. На практике эти условия не всегда могут выполняться.

Реализация метода синхронизации времени на основе стандарта IEEE 1588 не обеспечивает необходимую точность синхронизации времени между различными узлами системы ЦОС. Более того, есть ряд значительных недостатков при его использовании, что делает его непригодным для

применения в качестве подсистемы единого времени в современных комплексах ЦОС с когерентной обработкой информации от ФАР.

Для устранения недостатков реализации метода синхронизации времени на основе стандарта IEEE 1588 и обеспечения требуемой точности, согласно третьему принципу обеспечения изохронной передачи данных, предлагается применить локальную систему единого времени, распространяющую сигналы единого машинного времени и опорной тактовой частоты из единого центра во все узлы по линиям с одинаковой задержкой.

Каждый узел в системе имеет собственную систему машинного времени. У всех узлов имеется свой счетчик текущего времени, который работает от опорной тактовой частоты и периодически обновляется значениями времени от источника единого машинного времени. С помощью каналов управления в определенные регистры узлов записывается желаемое время начала того или иного события. Когда текущее время, указанное в счетчиках машинного времени, достигнуто, все узлы начинают выполнение своих задач одновременно. Аналогично запускаются все остальные процессы и генерируются управляющие сигналы, необходимые для согласованной обработки информации в каждом узле комплекса ЦОС.

Структурная схема системы изохронной передачи данных в комплексе ЦОС с использованием локальной системы единого времени представлена на рисунке 2.3.

Следует отметить, что точная установка времени во всех узлах комплекса, обеспечивающих передачу данных, не обеспечивает синхронное поступление данных в узлы приема. Несмотря на то что для каналов передачи данных используются оптические линии связи [112] как наиболее стабильные и с минимальным дрейфом характеристик, избежать рассогласования потоков данных невозможно. В каналах данных неизбежно возникают задержки, так как канал включает не только линию связи, но и преобразование сигнала из оптического в электрический и обратно. Все эти компоненты вносят задержку, которая может различаться от устройства к устройству. Поэтому

выравнивание этих задержек является еще одним важным условием для обеспечения изохронной передачи данных.

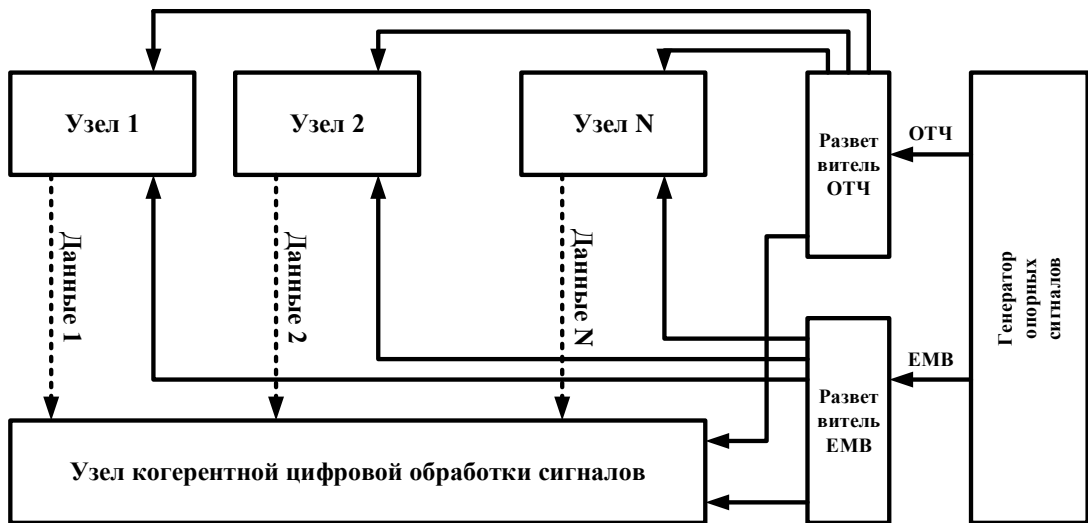


Рисунок 2.3. Структурная схема системы изохронной передачи данных

Для передачи привязанных ко времени данных чаще всего используют различные протоколы сетевых технологий. В главе 1 данной диссертационной работы было доказано, что использование сетевых технологий неприменимо к задаче обеспечения когерентной обработки в комплексе когерентной цифровой обработки сигналов.

Тривиальным решением для передачи данных, привязанных ко времени, является сопровождение каждого операнда временной меткой (рисунок 2.4).

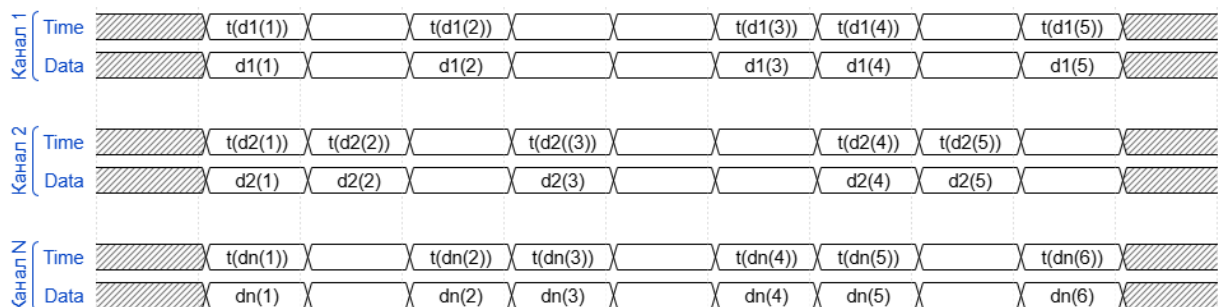


Рисунок 2.4. Временная диаграмма сопровождения каждого операнда временной меткой

Отметим, что понятие операнд в данном контексте понимается как данное, привязанное ко времени и используемое для дальнейшей обработки в узлах когерентной цифровой обработки сигналов.

Данный способ обеспечивает передачу привязанных ко времени операндов, однако, имеет ряд существенных недостатков. Первый - это снижение пропускной способности канала для полезных данных, поскольку для передачи временной метки выделяется некоторая ширина канала, которая может занимать половину от общей ширины или даже больше, все зависит от формата метки времени. Второй недостаток – это усложнение процедуры выравнивания операндов в каналах относительно друг друга. На стороне приемника требуются сложные алгоритмы анализа временных меток и буферизации операндов.

Возможно модернизировать способ следующим образом – сформировать плотный синхронный поток операндов без разрывов (рисунок 2.5). Таким образом, метка времени $t(d1(1))$ может быть передана только для первого операнда ($d1(1)$) в массиве. Так как операнды передаются синхронно, можно получить временную метку любого операнда в потоке.

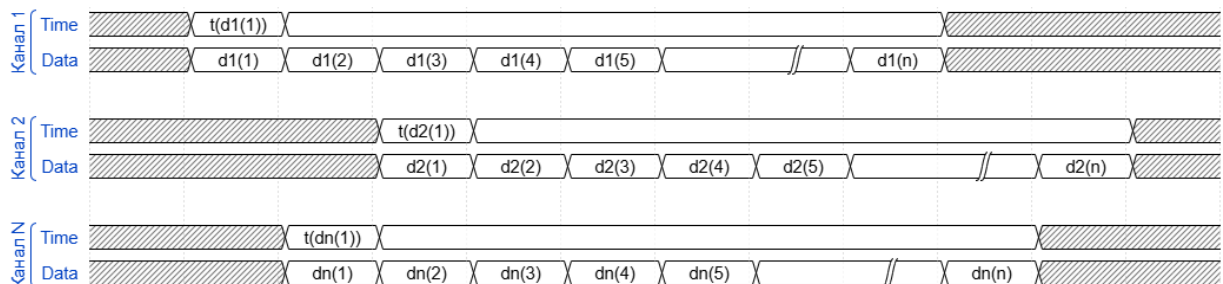


Рисунок 2.5. Временная диаграмма неразрывного потока операндов

Такой способ передачи данных избавляет от сложных алгоритмов выравнивания потоков данных в различных каналах, но проблема сокращения пропускной способности из-за выделения части канала для передачи временной метки сохраняется.

Существует актуальный протокол, применяемый в современных устройствах когерентной цифровой обработки сигналов, который для удобства дальнейшего изложения обозначим как базовый. Он отличается размещением временной метки непосредственно в потоке операндов, что позволяет значительно повысить пропускную способность. Временная диаграмма показана на рисунке 2.6.

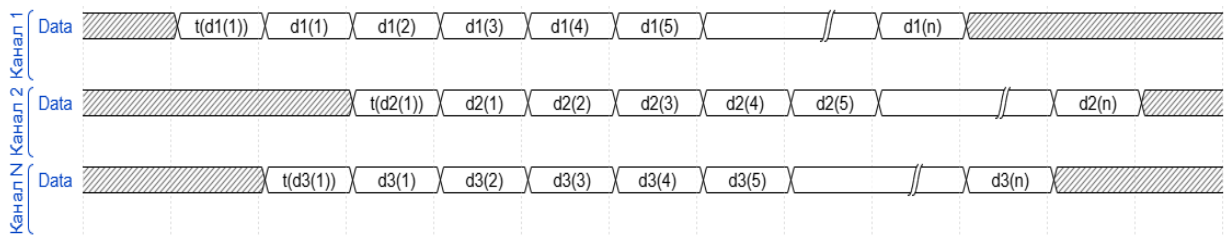


Рисунок 2.6. Временная диаграмма передачи метки времени в потоке операндов

Для того чтобы устранить различия в задержках между каналами, используется следующий прием: во время инициализации подсистемы передачи данных производится измерение задержек для каждого канала. Полученные данные поступают в буферные элементы, которые настраиваются таким образом, чтобы на выходе задержки между каналами были устранены.

Первым недостатком этого метода является вычисление задержек на этапе инициализации. Вычисленные значения задержек фиксируются и не изменяются в процессе работы. На практике же возможно изменение задержек в процессе работы. Способ не учитывает динамическое изменение задержек в различных каналах. Например, при потере соединения и его последующего восстановления задержка может отличаться от той, что была рассчитана ранее, тем самым изохронная передача данных будет нарушена. Потребуется повторить процедуру вычисления новых значений задержек.

Вторым недостатком является невозможность своевременного обнаружения ошибок в потоке операндов, поскольку отсутствуют механизмы контроля и парирования ошибок.

Для устранения вышеописанных недостатков, предложен новый метод обеспечения изохронной передачи данных, отличающийся от предыдущего применением служебных промежутков при передаче массивов операндов. (рисунок 2.7).

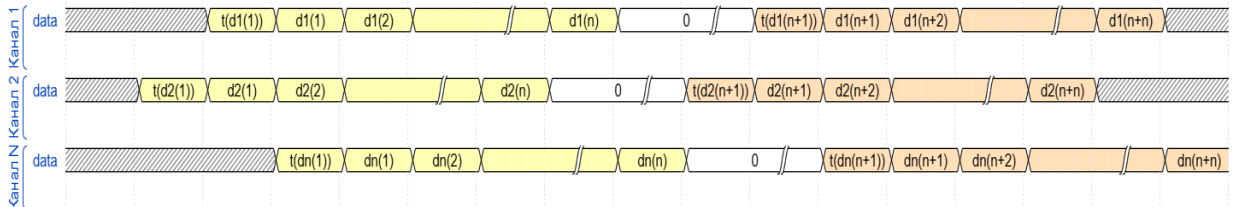


Рисунок 2.7. Временная диаграмма передачи массивов операндов, разделенных служебными промежутками

Суть метода заключается в том, чтобы разделить входной массив операндов на группы одинаковой длины, которые дополняются сервисной информацией и передаются на более высокой частоте со служебными промежутками. В служебном промежутке перед началом массива операндов необходимо поместить заголовок с временной меткой, соответствующей времени получения первого операнда массива, тем самым осуществив его привязку ко времени. Кроме того, необходимо выполнить подсчет контрольной суммы заголовка и массива и разместить ее в служебном промежутке после массива операндов.

Служебные промежутки могут быть образованы при переходе на большую частоту передачи относительно частоты поступления данных и реализуют следующие варианты действий:

- во-первых, позволяют осуществить возможность динамического выравнивания задержек на стороне приемника данных;

- во-вторых, позволяют добавить к каждому фрагменту заголовок с сервисной информацией. Этот заголовок может содержать данные о канале передачи и другие важные параметры для правильной интерпретации данных на стороне получателя. Также в заголовке указывается временная метка,

соответствующая времени получения первого операнда во фрагменте, что позволяет привязать весь массив операндов во времени.

Важным отличием метода передачи операндов со служебными промежутками от предыдущих является возможность динамического выравнивания задержек на стороне приемника данных и наличие средств контроля целостности данных. Для контроля целостности и информации в массиве операндов применяется циклический избыточный код CRC [39]. Контрольная сумма считается для всех операндов группы и размещается в конце массива.

Метод обеспечения изохронной передачи данных с использованием служебных промежутков реализуется следующим образом:

- в передающем узле исходные данные подаются потоком, который с точки зрения интерфейса представляет собой массив операндов. Необходимо разделить входной массив операндов на группы одинаковой длины, которые дополняются сервисной информацией (меткой времени, контрольной суммой и т.д.) и передаются на более высокой частоте со служебными промежутками;

- в служебном промежутке перед началом массива операндов поместить заголовок с временной меткой, соответствующей времени получения первого операнда массива, тем самым осуществив его привязку ко времени;

- выполнить подсчет контрольной суммы заголовка и массива и разместить в служебном промежутке после массива операндов;

- временная диаграмма процесса преобразования в формат массивов данных, разделенных служебными промежутками, представлена на рисунке 2.8.

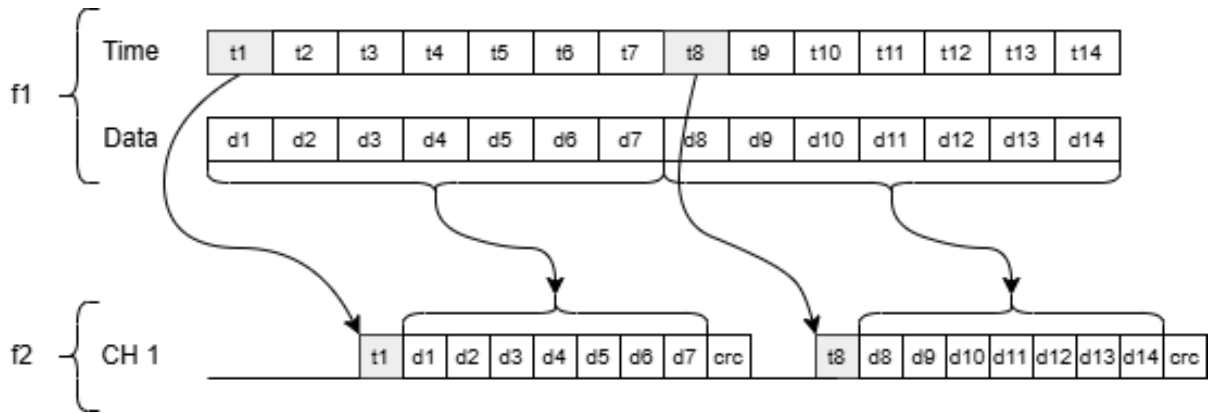


Рисунок 2.8. Временная диаграмма преобразования непрерывного потока операндов

Частота передачи операндов зависит от множества параметров, представленных на рисунке 2.9.

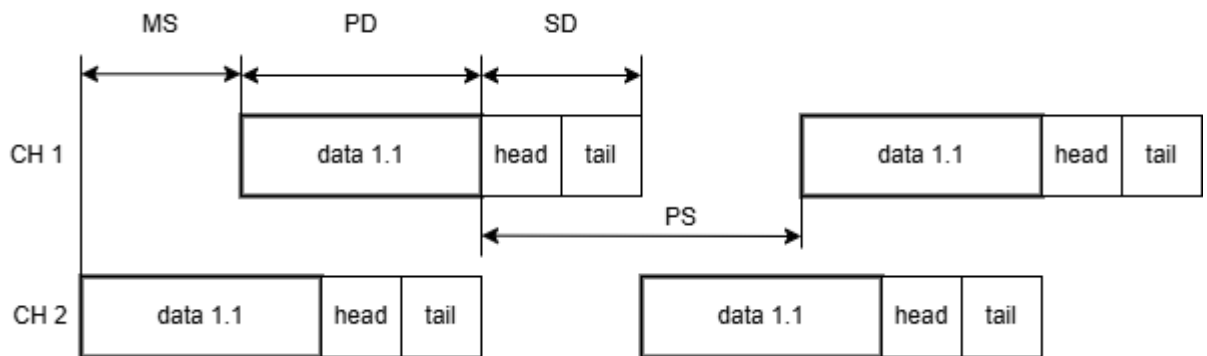


Рисунок 2.9. Основные параметры формата передачи операндов со служебными промежутками

MS – максимальное рассогласование задержек операндов между различными каналами, выражается в количестве тактов частоты передачи;

PD – количество операндов в массиве;

SD – количество сервисных данных, таких как временная метка, контрольная сумма, дескрипторы и т.д.;

PS – служебный интервал, выражается в количестве тактов частоты передачи.

Частота передачи данных определяется по формуле:

$$F_2 = \frac{f_1}{DC}, \quad (2.2)$$

где F_1 – частота поступления данных;

F_2 – частота передачи данных;

DC – коэффициент заполнения.

Коэффициент заполнения DC вычисляется по формуле:

$$DC = \frac{PD}{(PD + PS)}. \quad (2.3)$$

Параметры количества данных в массиве PD и размер служебного интервала PS имеют следующие ограничения.

Во-первых, количество данных в массиве PD должно быть не меньше значения максимального рассогласования задержек MS .

Во-вторых, минимальный размер служебного интервала PS для работы метода и предоставления возможности парировать динамические задержки составляет один такт частоты передачи F_2 . Так как служебный интервал PS используется для добавления сервисной информации, то минимальный размер служебного промежутка будет выражаться следующим образом:

$$PS = SD + 1. \quad (2.4)$$

Получив значения PD и PS , можно найти минимальный коэффициент заполнения D и получить значение минимальной частоты передачи f_2 , при котором будет обеспечена корректная работа метода. Для минимизации потери пропускной способности при передаче массивов операндов со служебными промежутками целесообразно максимально увеличить коэффициент заполнения DC . Так как минимальный размер служебного промежутка PS зафиксирован, то для увеличения коэффициента заполнения DC возможно увеличивать только количество данных в массиве операндов PD .

В соответствии с формулой (2.4) при увеличении количества данных PD значение коэффициента заполнения DC асимптотически стремится к единице (рисунок 2.10).

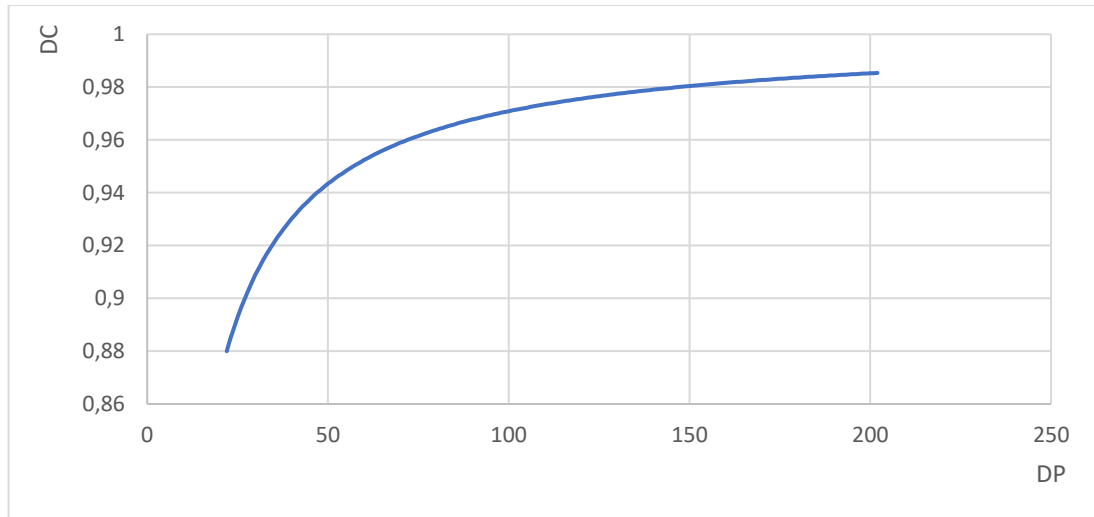


Рисунок 2.10. Зависимость коэффициента заполнения от количества данных в массиве при фиксированном размере служебного интервала

В этой связи предлагается выбрать коэффициент заполнения DC, исходя из нескольких факторов, таких как: допустимый уровень потери пропускной способности, желаемые соотношения частот или заданный уровень гарантоспособности.

В процессе передачи данных в узел приема между различными каналами неизбежно возникнут рассогласования, выравнивание которых и будет произведено в узле приема.

Процесс выравнивания задержек проиллюстрирован на временной диаграмме (рисунок 2.11).

В узле приема данных выполняется следующий алгоритм.

1. В каждом канале, независимо от других, при получении признака начала массива начинается запись данных в буфер этого канала. Фиксируется временная метка из заголовка, и начинается процесс расчета контрольной суммы принятых данных.

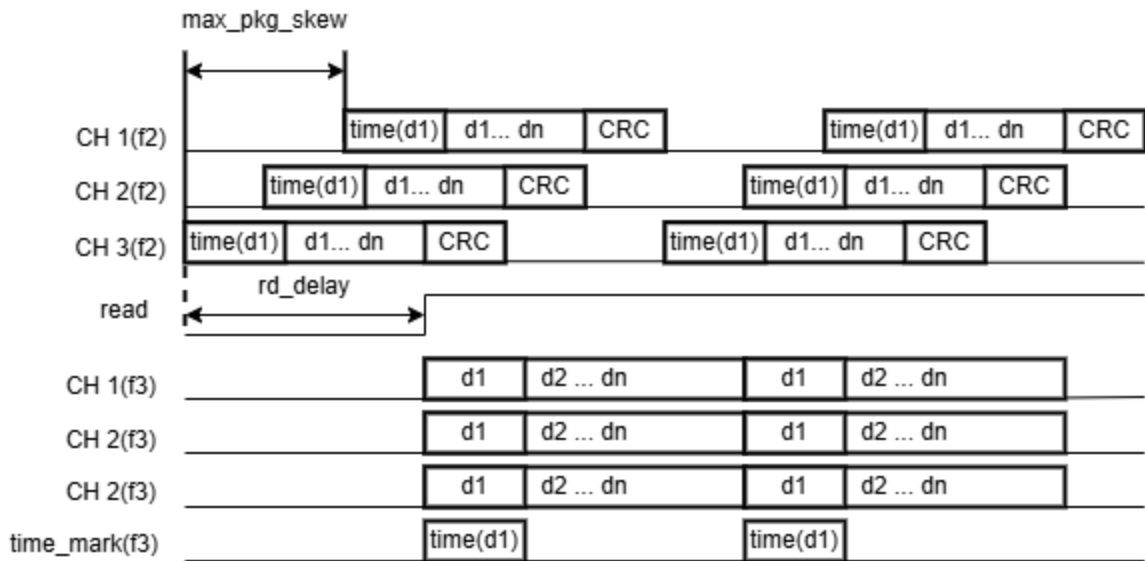


Рисунок 2.11. Временная диаграмма выравнивания задержек

2. При поступлении признака конца массива запись в буфер прекращается и происходит сравнение контрольных сумм. При совпадении контрольных сумм формируется флаг успешной проверки контрольных сумм, при несовпадении - флаг ошибки контрольных сумм.

3. Пункты 1-2 циклически повторяются для каждого следующего массива.

4. Одновременно с началом записи в буфер от первого пришедшего массива начинается отсчет задержки чтения.

5. После задержки чтения начинается одновременное чтение данных из буферов всех каналов. Синхронно с первым вычитанным данным в узлы когерентной цифровой обработки сигналов передается временная метка от первого пришедшего канала.

Таким образом обеспечивается динамическое выравнивание задержек, возникающих при передаче массивов операндов, и происходит восстановление первоначального непрерывного, изохронного потока данных, необходимого для алгоритмов когерентной цифровой обработки сигналов.

Следует отметить, что предельные значения допустимых задержек целесообразно ограничить, поскольку большие задержки приведут к

увеличению латентности данных на выходе программы и необходимости использовать большой объем памяти. Слишком большие задержки могут свидетельствовать о неполадках канала данных, таких как нарушение синхронизации узла, передающего данные, с системой единого машинного времени или с генератором опорной частоты, в результате чего данные в передающих узлах формируются в неправильный момент времени. использование таких данных недопустимо, так как это нарушает принципы изохронной передачи. Необходимо обнаруживать подобные ситуации и не пропускать данные такого канала на выход программы.

Для того чтобы обнаружить недопустимые задержки в каналах данных предлагается сформировать допустимый диапазон рассогласования относительно одного из каналов. В дальнейшем такой канал будем называть опорным.

Для назначения опорного канала, предлагается реализовать процедуру, определяющую значение задержек в каждом канале и вычисляющую среднее арифметическое для этих задержек. В качестве опорного канала следует назначать тот, чья задержка находится ближе к среднеарифметическому значению задержек всех каналов.

Процедура назначения опорного канала осуществляется на начальном этапе работы программы и выполняется следующим образом.

1. Ожидание поступления внешней команды запуска `run`.
2. Ожидание поступление флага строба `valid[any]` первого массива операндов по любому из каналов. При обнаружении строба `valid[any]` входящего массива данных по любому из каналов подается команда `start` на вычисление задержек в каждом канале и запускается счетчик задержек `cntr`.
3. При поступлении строба данных `valid[n]` в соответствующем канале текущее значение счетчика задержек `cntr` фиксируется в массиве `ltch` размером, равным количеству каналов, а его индекс определяет номер канала `n`. Однако если строб массива операндов в каком-либо канале `n` приходит после того, как счетчик достигает значения максимально

допустимого рассогласования задержек MS , то в $ltch[n]$ записывается ноль. Это исключает каналы с чрезмерной задержкой из списка кандидатов на роль опорного.

4. В цикле, число итераций i которого равно числу каналов N , вычисляется среднее арифметическое значение avg^* всех задержек в массиве $ltch$.

5. Находим канал, задержка которого наиболее близка к средней задержке всех каналов. Для этого в цикле, число итераций j которого равно числу каналов N , сравниваем задержку каждого канала $ltch[j]$ со значением avg^* . Если задержка канала $ltch[j]$ равна avg^* , переходим к пункту 6. Если по окончании цикла не найдена задержка $ltch[j]$, равная avg^* , переходим к пункту 8.

6. Задаем вход мультиплексора временных меток $sel = j$.

7. Запускаем счетчик диапазона допустимого рассогласования win_cntr при выполнении условия: $cntr = PD + PS + ltch[j] - MS$, где PD – размер массива операндов, PS – размер служебного промежутка, MS – максимально допустимое рассогласование задержек (счетчик win_cntr сбрасывается каждый раз при выполнении условия: $win_cntr = PD + PS$). Процедура назначения опорного канала завершается.

8. Инкрементируем переменную Si .

9. Проверяем переменную Si на четность. Если значение переменной Si четное, то переходим к пункту 10. Если нечетное, то к пункту 11.

10. Прибавляем к avg^* значение переменной Si . Переходим к пункту 5.

11. Отнимаем от avg^* значение переменной Si . Переходим к пункту 5.

Алгоритмическая схема процедуры назначения опорного канала представлена на рисунке 2.12.

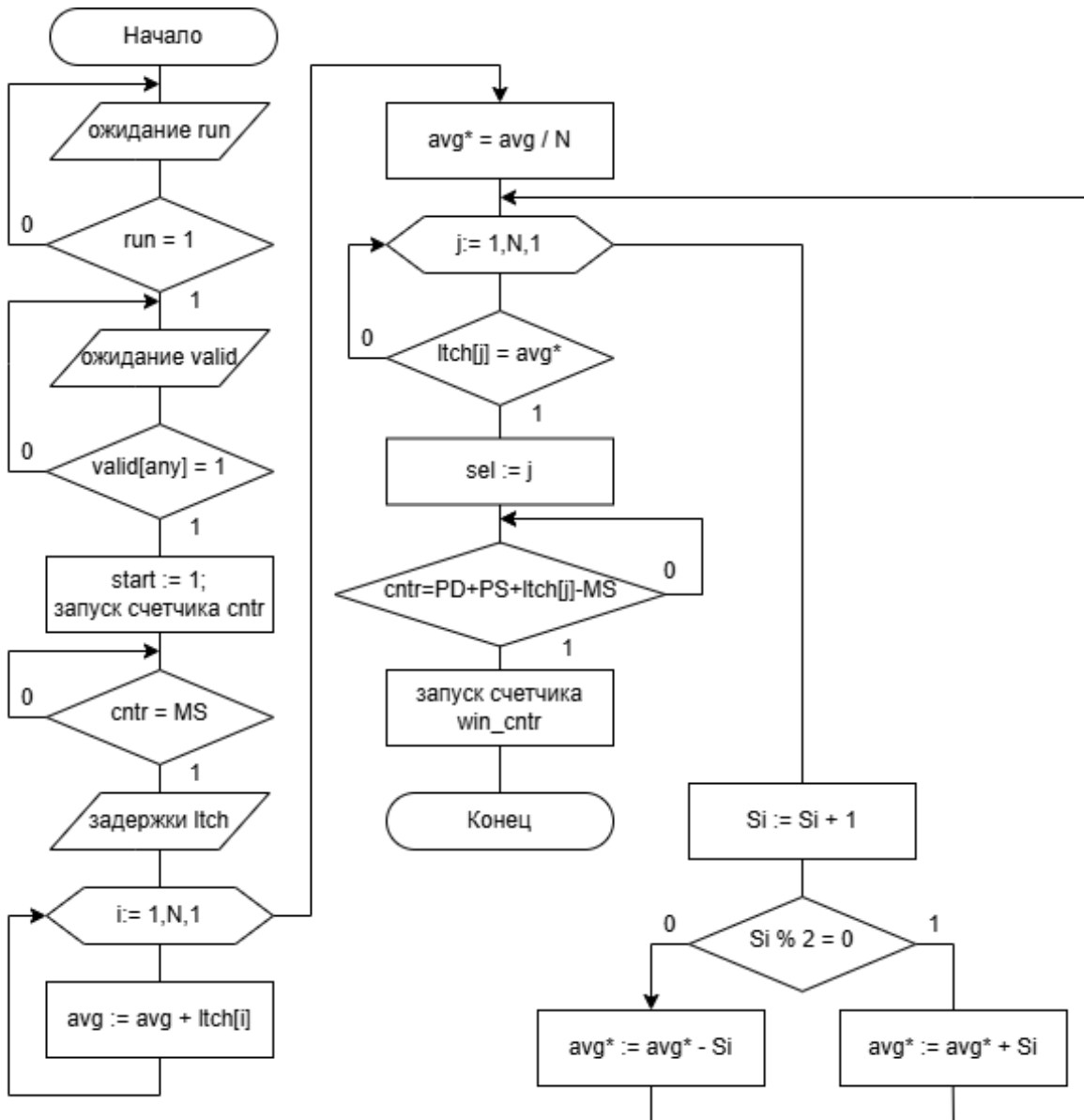


Рисунок 2.12. Алгоритмическая схема процедуры назначения опорного канала

После назначения опорного канала необходимо сформировать флаг диапазона допустимого рассогласования, центрированный относительно начала массива операндов, передаваемых в опорном канале.

Формирование флага диапазона допустимого рассогласования `allow_window` осуществляется на основе счетчика `win_cntnr`. Флаг `allow_window` принимает значение 1 при `win_cntnr = 1` и сбрасывается при достижении `win_cntnr = MS * 2`.

При поступлении массивов операндов в каждом канале проверяется, попадают ли их задержки в допустимый диапазон. Если задержка выходит за пределы диапазона, данные в массивах заменяются на массивы с нейтральной информацией для минимизации негативного влияния на результат обработки.

Временная диаграмма процедуры проверки задержек данных в каналах на соответствие допустимому диапазону рассогласования представлена на рисунке 2.13.

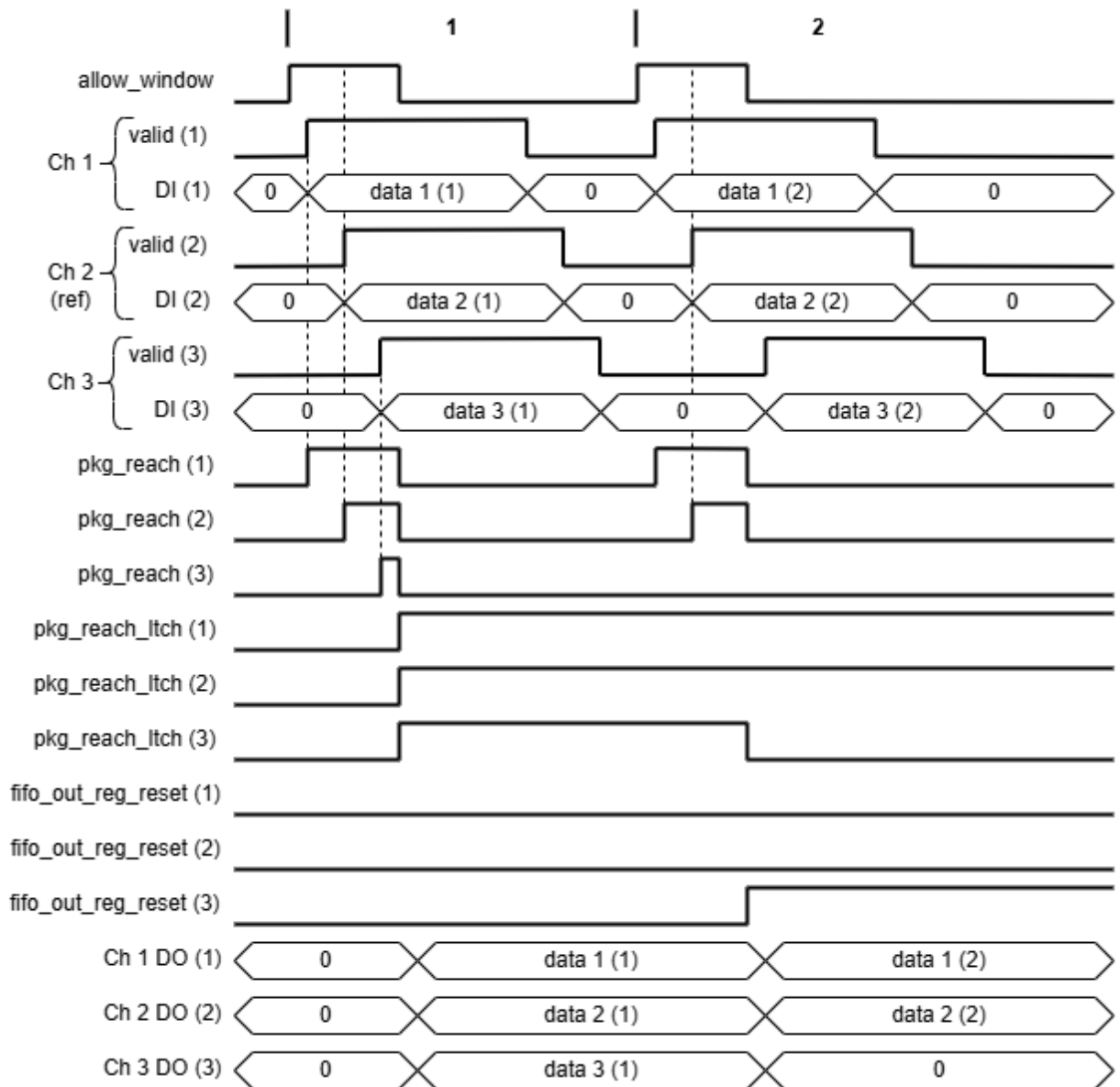


Рисунок 2.13. Временная диаграмма процедуры проверки задержек данных в каналах

На временной диаграмме в качестве примера показаны три канала данных. Канал Ch 2 выбран как опорный. Относительно начала массивов операндов, передаваемых в опорном канале, формируется флаг `allow_window`, определяющий допустимый диапазон рассогласования задержек. Если начало массива операндов поступает в момент активности флага `allow_window`, устанавливается флаг `pkg_reach(n)`, указывающий на попадание в допустимый диапазон. После снятия флага `allow_window` состояние флага `pkg_reach(n)` защелкивается во флаге `pkg_reach_latch(n)`, а сам `pkg_reach(n)` сбрасывается. Флаг `pkg_reach_latch(n)` управляет выходом канала: если он активен, данные передаются из памяти, если сброшен, то в данном примере информация канала обнуляется. Зона 1 демонстрирует работу, когда все каналы находятся в допустимом диапазоне задержек. Зона 2 показывает ситуацию, когда задержка канала Ch 3 вышла за пределы допустимого диапазона, и его данные на выходе канала Ch 3 DO (3) обнулились.

Таким образом процедура назначения опорного канала и формирования диапазона допустимого рассогласования позволяет обнаружить недопустимые задержки поступления данных в каналах и не допустить поступления данных сбойных каналов в блоки цифровой обработки сигналов.

Для обработки ошибок целостности данных предлагается осуществлять статистику ошибок целостности данных в каждом канале для оценки его работоспособности. Предлагается установить порог количества ошибок целостности данных, после которых канал следует отключить, а его данные подменить массивами с нейтральной информацией. После отключения канала предлагается продолжить проверку целостности данных в канале и в случае достижения установленного порога корректно принятых массивов подряд восстановить работу канала.

Временная диаграмма процедуры обнаружения и парирования ошибок целостности данных представлена на рисунке 2.14.

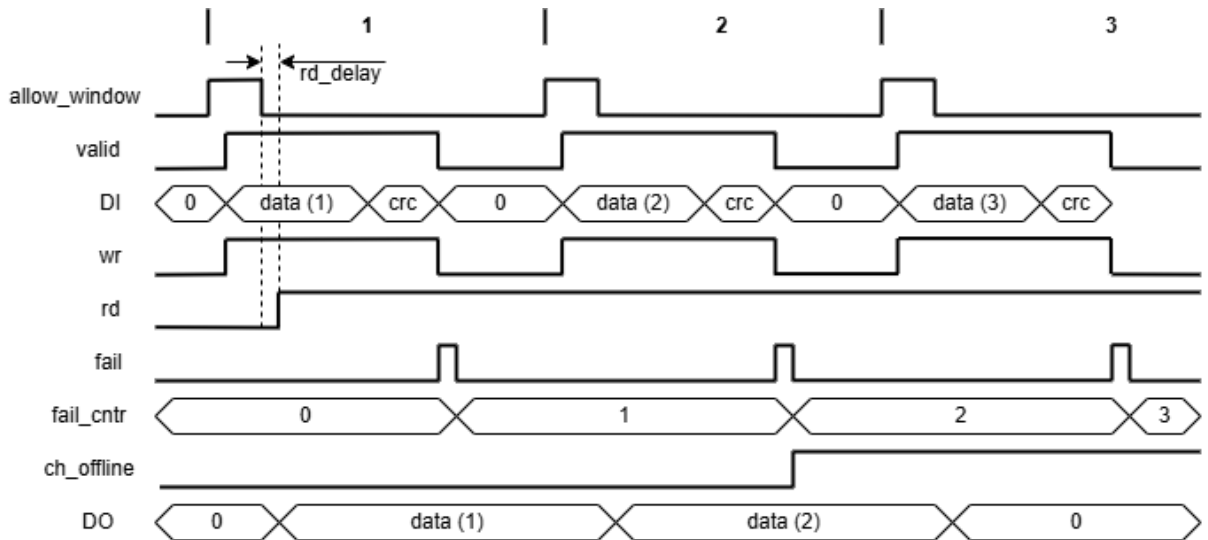


Рисунок 2.14. Временная диаграмма контроля целостности

Входные данные поступают в виде массивов операндов, разделённых служебными промежутками. Каждый массив операндов сначала проходит проверку на рассогласование задержек, после чего начинается его запись в память. Одновременно с записью запускается вычисление контрольной суммы. После того как все массивы операндов, прошедшие контроль на допустимые задержки, записали в память хотя бы одно данное, подается команда чтения. По окончании массива операндов из служебного промежутка считывается контрольная сумма и сравнивается с подсчитанным значением. Если суммы не совпадают, инкрементируется счетчик ошибок. При достижении счетчиком предельного значения, заданного параметром (для примера установим порог ошибок, равным двум), формируется признак, указывающий на необходимость отключения канала.

В зоне I представлена ситуация, когда в канале начинают передаваться массивы операндов с ошибками целостности данных. Массивы 1 и 2 содержат в себе поврежденные при передаче данные. Обнаружив несоответствие контрольных сумм массива 1 сформируется флаг ошибки целостности данных fail, который инкрементирует счетчик ошибок fail_cntr. Ошибка контрольных сумм массива 2 снова инкрементирует счетчик ошибок fail_cntr, который достигнет заданного порога в две ошибки и сформируется признак

необходимости отключения канала `ch_offline`. После чего последующие массивы операндов, передаваемые в канале, будут проигнорированы до восстановления канала.

Предложенная процедура позволяет установить предельное количество ошибок за определенный период времени, при достижении которого канал будет отключен, а также определить временной интервал, в течение которого отсутствие ошибок позволит восстановить функционирование канала.

2.2 Реализация комбинированного метода изохронной передачи данных

Описанный в предыдущем разделе комбинированный метод изохронной передачи данных реализован для ПЛИС Xilinx Virtex-7 XC7VX485T [113]. На вход ПЛИС поступает 24 канала разрядностью 33 бита (32 бита данных и один бит строб) на частоте $F_1 = 255$ МГц.

Выбор частоты 255 МГц был обусловлен эмпирически установленными закономерностями при проектировании цифровых систем на базе программируемых логических интегральных схем XC7VX485T. Эксперименты показали, что при увеличении частоты свыше 255 МГц наблюдается рост количества нарушений временных ограничений (timing constrain), связанных с задержками распространения сигналов по глобальным трассировочным ресурсам ПЛИС. С другой стороны, уменьшение частоты ниже 255 МГц не позволяет в полной мере использовать вычислительный потенциал кристалла, снижая общую производительность системы. В этой связи частота 255 МГц для ПЛИС XC7VX485T позволяет достичь высокого коэффициента заполнения кристалла ПЛИС без возникновения ошибок трассировки и временных нарушений.

Данные на вход ПЛИС поступают массивами операндов, разделенными служебными промежутками. В служебный промежуток перед началом массива операндов вставлено два слова, соответствующих метке времени

первого данного массива операндов. В конце массива размещено одно слово контрольной суммы, посчитанной для всех операндов массива и для метки времени.

В соответствии с комбинированным методом изохронной передачи данных для корректной работы метода необходимо соблюдать определенные условия. Во-первых, количество данных в массиве операндов должно быть больше значения максимального рассогласования задержек между каналами. Во-вторых, минимальный размер служебного промежутка составляет один такт частоты поступления данных.

Экспериментально установлено, что максимальная задержка между каналами составляет 10 тактов, это означает, что минимальный размер массива операндов должен быть не менее 11 тактов. Служебный промежуток используется для хранения сервисной информации, включая три слова (два слова для метки времени и одно слово для контрольной суммы). Следовательно, минимальный размер служебного промежутка составляет четыре такта.

В результате получится, что данные займут 11 тактов, а служебный промежуток - четыре такта. Однако такое соотношение не является оптимальным. Для минимизации потери пропускной способности при передаче массивов операндов со служебными промежутками целесообразно увеличить коэффициент заполнения данных DC.

Коэффициент заполнения данных DC представляет собой отношение числа тактов, используемых для передачи данных, к общему количеству тактов, включая служебные промежутки. Коэффициент заполнения данных рассчитывается по формуле (2.3).

Для увеличения коэффициента заполнения данных можно либо уменьшить служебный промежуток, либо увеличить размер массива операндов. Поскольку минимально возможный размер служебного промежутка PS уже достигнут, то единственным способом увеличения DC является увеличение размера массива операндов DP.

Коэффициент DC влияет на значение частоты выходных данных F_2 , выраженной по формуле:

$$F_2 = F_1 \cdot DC, \quad (2.5)$$

где F_1 – частота входных данных;

F_2 – частота выходных данных.

Опыт реализации программы на ПЛИС показал, что для корректной работы программы обе частоты должны генерироваться от одного и того же генератора MMCM (Mixed-mode clock manager). MMCM - это аппаратный блок управления синхронизацией смешанного режима, который используется в качестве синтезатора широкого диапазона частот [114]. При использовании частот от разных генераторов MMCM может возникнуть «дрейф» частот, при котором их соотношение начнет меняться, что нарушит отношение скорости поступления данных к скорости их считывания, и в конечном итоге может привести к опустошению или переполнению памяти, из-за чего непрерывность выходного потока операндов будет нарушена.

Генерация выходных частот в MMCM имеет ряд особенностей и ограничений. Упрощенная структурная схема блока MMCM представлена на рисунке 2.15.

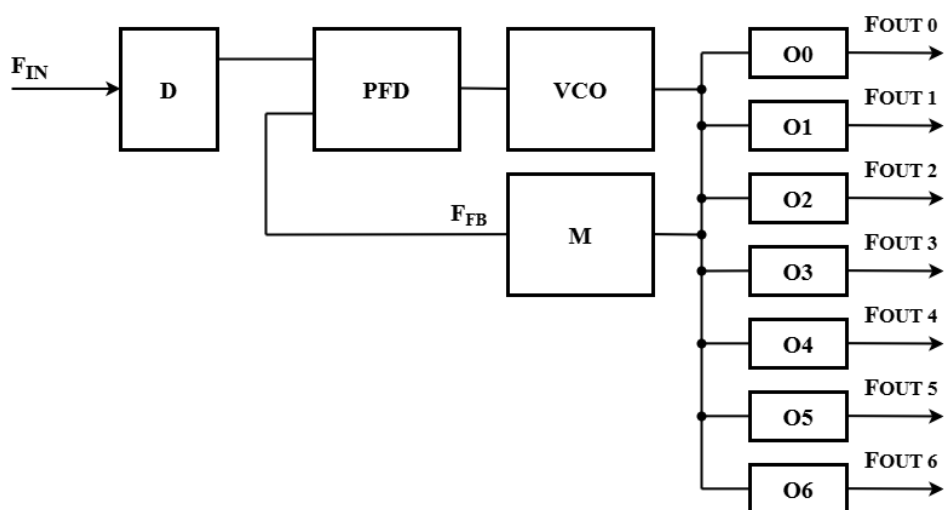


Рисунок 2.15. Упрощенная структурная схема блока MMCM

Блок ММСМ состоит из: фазово-частотного детектора (PFD – phase-frequency detector); генератора, управляемого напряжением (VCO – voltage control oscillator); делителя входной частоты D; делителя частоты обратной связи M; делителей выходной частоты $O0 \div O6$.

На вход генератора ММСМ поступает сигнал опорной тактовой частоты F_{IN} . Фазово-частотный детектор PFD сравнивает фазу и частоту опорного синхросигнала F_{IN} с синхросигналом обратной связи F_{FB} и формирует управляющий сигнал, определяющий частоту работы генератора VCO. Частота генератора VCO F_{VCO} программируется путем установки коэффициентов K_D и K_M и определяется с помощью следующего выражения:

$$F_{VCO} = F_{IN} \cdot \frac{K_M}{K_D}, \quad (2.6)$$

где F_{IN} – значение входной опорной тактовой частоты;

K_M – коэффициент деления делителя частоты обратной связи;

K_D – коэффициент деления делителя входной частоты.

K_M, K_D – целое число.

На коэффициенты K_D и K_M действуют ограничения, использованные для определения разрешенных значений K_D и K_M , которые задаются с помощью следующих выражений:

$$K_{D \text{ MIN}} = \left\lceil \frac{F_{IN}}{F_{PFD \text{ MAX}}} \right\rceil; \quad (2.7)$$

$$K_{D \text{ MAX}} = \left\lfloor \frac{F_{IN}}{F_{PFD \text{ MIN}}} \right\rfloor; \quad (2.8)$$

$$K_{D \text{ MIN}} = \left\lceil \frac{F_{VCO \text{ MIN}}}{F_{IN}} \cdot K_{D \text{ MIN}} \right\rceil; \quad (2.9)$$

$$K_{D \text{ MAX}} = \left\lfloor \frac{F_{VCO \text{ MAX}}}{F_{IN}} \cdot K_{D \text{ MAX}} \right\rfloor; \quad (2.10)$$

где $\lceil x \rceil$ – наименьшее целое, большее или равное x ;

$\lfloor x \rfloor$ – наибольшее целое, меньшее или равное x ;

F_{IN} – значение входной опорной тактовой частоты;

F_{PFDMAX} и F_{PFDMIN} – максимальная и минимальная частоты фазово-частотного детектора PFD;

F_{VCOMAX} и F_{VCOMIN} – максимальная и минимальная частоты генератора VCO.

Параметры F_{VCOMAX} , F_{VCOMIN} , F_{PFDMAX} , F_{PFDMIN} определены в спецификации на MMCM для каждого семейства ПЛИС [99]. Согласно спецификации на MMCM для ПЛИС Xilinx Virtex-7, диапазон допустимых значений генератора VCO (F_{VCO}) лежит в пределах 600-1600 МГц. Для фазово-частотного детектора PFD допустимый диапазон составляет 10-500 МГц.

Для генерирования выходных частот существует семь независимо программируемых выходных делителей $O0 \div O6$. Делитель $O0$ позволяет использовать дробные коэффициенты с шагом 0,125. Значение выходной частоты ($F_{\text{OUT}n}$) определяется следующим выражением:

$$F_{\text{OUT}n} = \frac{F_{\text{VCO}}}{O_n}. \quad (2.11)$$

Следует учитывать, что есть ограничения на значения выходных частот из-за того, что все выходные частоты определяются делителями $O0 \div O6$ от единственной частоты F_{VCO} . Поэтому при установке одной из выходных частот допустимые значения остальных выходных частот будут ограничены.

Рассмотрим настройку MMCM для данной реализации. Имеется заданная опорная частота $F_{\text{IN}} = 100$ МГц, необходимо сгенерировать две выходные частоты, одна из которых задана значением 255 МГц (частота поступления данных F_1). Вторая частота (частота выходных данных F_2) должна быть меньше F_1 и как можно ближе к значению F_2 , чтобы минимизировать потерю пропускной способности. Требуется найти значения коэффициентов входного делителя D , делителя обратной связи M и выходных делителей K_{O0} и K_{O1} .

Имея одну заданную выходную частоту $F_1 = 255$ МГц, найдем возможные значения F_{VCO} , которые определяются следующим выражением:

$$F_{VCO} = K \cdot F_1, \quad (2.12)$$

где K – коэффициент деления выходного делителя.

Для задания выходной частоты F_1 используем делитель с целочисленным коэффициентом деления $O1$. Согласно спецификации на ММСМ для ПЛИС Xilinx Virtex-7, диапазон допустимых значений частоты генератора VCO (F_{VCO}) лежит в пределах 600 - 1600 МГц:

$$600 \leq (F_{VCO} = K_{O1} \cdot F_1) \leq 1600. \quad (2.13)$$

Диапазон возможных значений K_{O1} составит от трех до шести. Получим следующие значения F_{VCO} :

$$K_{O1} = 3, F_{VCO} = 765;$$

$$K_{O1} = 4, F_{VCO} = 1020;$$

$$K_{O1} = 5, F_{VCO} = 1275;$$

$$K_{O1} = 6, F_{VCO} = 1530.$$

Поскольку F_{VCO} задается с помощью коэффициентов K_D и K_M по формуле (2.6), а на коэффициенты K_D и K_M действуют ограничения, показанные в выражениях (2.9) ÷ (2.12), необходимо проверить возможность настройки ММСМ на данные значения F_{VCO} .

Применив формулы (2.9) ÷ (2.12) при заданной входной частоте $F_{IN}=100$ МГц, получим допустимый диапазон коэффициентов K_D : 1-10, K_M : 6÷160. Подставив допустимый диапазон значений коэффициента K_D в формулу (2.6), найдем значения коэффициентов K_M . Результаты расчета коэффициентов K_M представлены в таблице 2.2.

Поскольку коэффициент K_M может быть только целочисленным (см. примечание к формуле (2.6)), из таблицы 2.2 необходимо выбрать целочисленные значения. Из таблицы 2.1 следует, что генератор VCO невозможно настроить на частоту $F_{VCO}=765$ МГц, так как коэффициент K_M для

нее не имеет целочисленных значений. Остальные частоты F_{VCO} доступны для применения.

Таблица 2.2 Результаты расчета коэффициентов K_M

| K_{O1} | F_{VCO} | K_D | | | | | | | | | |
|----------|-----------|-------|------|-------|-----------|-----------|------|-------|------------|--------|------------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 3 | 765 | 7,65 | 15,3 | 22,95 | 30,6 | 38,25 | 45,9 | 53,55 | 61,2 | 68,85 | 76,5 |
| 4 | 1020 | 10,2 | 20,4 | 30,6 | 40,8 | 51 | 61,2 | 71,4 | 81,6 | 91,8 | 102 |
| 5 | 1275 | 12,75 | 25,5 | 38,25 | 51 | 63,75 | 76,5 | 89,25 | 102 | 114,75 | 127,5 |
| 6 | 1530 | 15,3 | 30,6 | 45,9 | 61,2 | 76,5 | 91,8 | 107,1 | 122,4 | 137,7 | 153 |

Найдем значение частоты выходных данных F_2 . F_2 должна быть целым числом, меньшим, чем F_1 , но при этом как можно ближе к значению F_1 . Для задания значения F_2 используем делитель $O0$ с дробным коэффициентом деления. Разрешение делителей для $O0$ имеет градацию $1/8$ или $0,125$.

Так как обе выходные частоты генерируются от одной частоты F_{VCO} , то справедливо выражение:

$$K_{O1} \cdot F_1 = K_{O0} \cdot F_2. \quad (2.14)$$

Отсюда следует что:

$$F_2 = \frac{K_{O1} \cdot F_1}{K_{O0}}, \quad (2.15)$$

где

$$K_{O0} = K_{O1} + \frac{1}{8} \cdot b, \quad b = 1 \dots 7. \quad (2.16)$$

Путем перебора коэффициентов K_{O0} и K_{O1} получим следующие значения всех возможных частот F_2 , представленных в таблице 2.3.

Из таблицы 2.3 можно сделать вывод, что ближайшая к F_1 целочисленная частота F_2 достигается при значениях $K_{O1} = 4$, $K_{O0} = 4.25$ и составляет 240 МГц. С помощью полученного значения K_{O1} можно вычислить все остальные коэффициенты, необходимые для настройки ММСМ, по таблице 2.2: при $K_{O1} = 4$ F_{VCO} составит 1020 МГц, что соответствует коэффициенту входного делителя $D = 5$ и коэффициенту обратной связи $M=51$.

Таблица 2.3 Доступные значения частоты F_2

| К ₀₀ | К ₀₁ | | |
|-----------------|-----------------|----------|----------|
| | 4 | 5 | 6 |
| 3 | 340 | 425 | 510 |
| 3,125 | 326,4 | 408 | 489,6 |
| 3,25 | 313,8462 | 392,3077 | 470,7692 |
| 3,375 | 302,2222 | 377,7778 | 453,3333 |
| 3,5 | 291,4286 | 364,2857 | 437,1429 |
| 3,625 | 281,3793 | 351,7241 | 422,069 |
| 3,75 | 272 | 340 | 408 |
| 3,875 | 263,2258 | 329,0323 | 394,8387 |
| 4 | 255 | 318,75 | 382,5 |
| 4,125 | 247,2727 | 309,0909 | 370,9091 |
| 4,25 | 240 | 300 | 360 |
| 4,375 | 233,1429 | 291,4286 | 349,7143 |
| 4,5 | 226,6667 | 283,3333 | 340 |
| 4,625 | 220,5405 | 275,6757 | 330,8108 |
| 4,75 | 214,7368 | 268,4211 | 322,1053 |
| 4,875 | 209,2308 | 261,5385 | 313,8462 |
| 5 | 204 | 255 | 306 |
| 5,125 | 199,0244 | 248,7805 | 298,5366 |
| 5,25 | 194,2857 | 242,8571 | 291,4286 |
| 5,375 | 189,7674 | 237,2093 | 284,6512 |
| 5,5 | 185,4545 | 231,8182 | 278,1818 |
| 5,625 | 181,3333 | 226,6667 | 272 |
| 5,75 | 177,3913 | 221,7391 | 266,087 |
| 5,875 | 173,617 | 217,0213 | 260,4255 |
| 6 | 170 | 212,5 | 255 |
| 6,125 | 166,5306 | 208,1633 | 249,7959 |

Зная значения частот входных данных и частоты выходных данных, вычисляем коэффициент DC из формулы (2.5). При значениях частоты входных данных $F_1 = 255$ МГц и частоты выходных данных $F_2 = 240$ МГц, коэффициент заполнения данных $DC = 0.941$.

Получив значение коэффициента DC, используем формулу (2.3), чтобы найти размер массива операндов PD. При размере служебного промежутка PS - четыре слова, размер массива операндов составит 64 слова. Структура входных данных для одного канала представлена на рисунке 2.16.

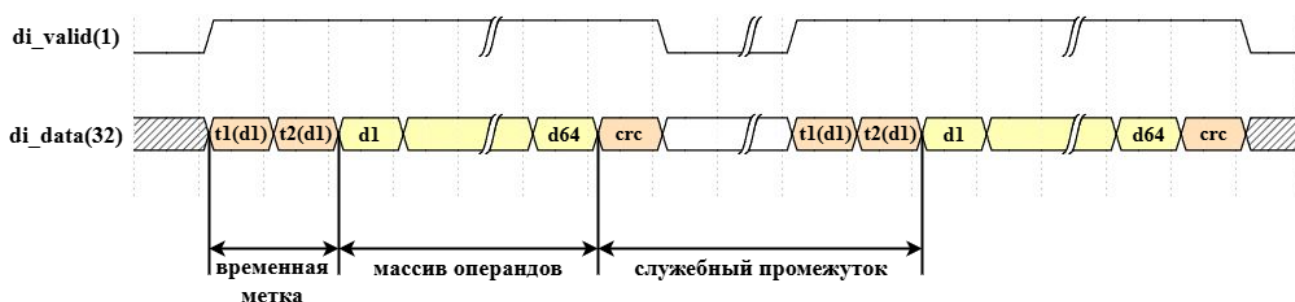


Рисунок 2.16. Структура входных данных

Согласно четвертому принципу обеспечения изохронной передачи данных, сформулированному в главе 1 данной диссертационной работы, необходимо обеспечить выравнивание задержек массивов операндов по всем каналам в пределах одной ПЛИС с помощью буферной памяти. Для этого в каждом из 24-х входных каналов задействована память типа FIFO. Запись в память массивов операндов происходит независимо по каждому каналу при их поступлении. Команда чтения подается на все FIFO одновременно, обеспечивая выравнивание задержек между каналами.

Программа выдает 24 непрерывных потока операндов разрядностью 32 бита на частоте 240 МГц. Все потоки операндов сопровождаются одним стробом и меткой времени первого операнда массива. Структура выходных данных представлена на рисунке 2.17.

Программа состоит из программных блоков, реализованных независимо для каждого канала (В1 – В24) и общего блока (А), управляющего работой всех каналов. Структура программы, реализующего комбинированный метод изохронной передачи данных, представлена на рисунке 2.18.

Общий блок для всех каналов А включает в себя: блок управления (БУПР) и блок синхронизации времени (БСВ).

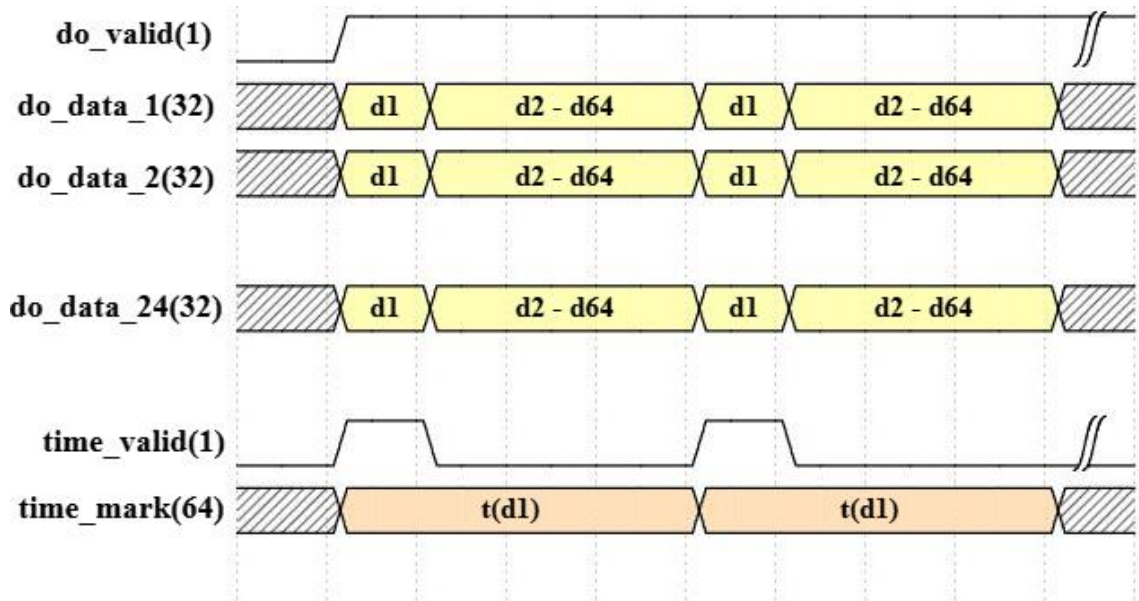


Рисунок 2.17. Структура выходных данных

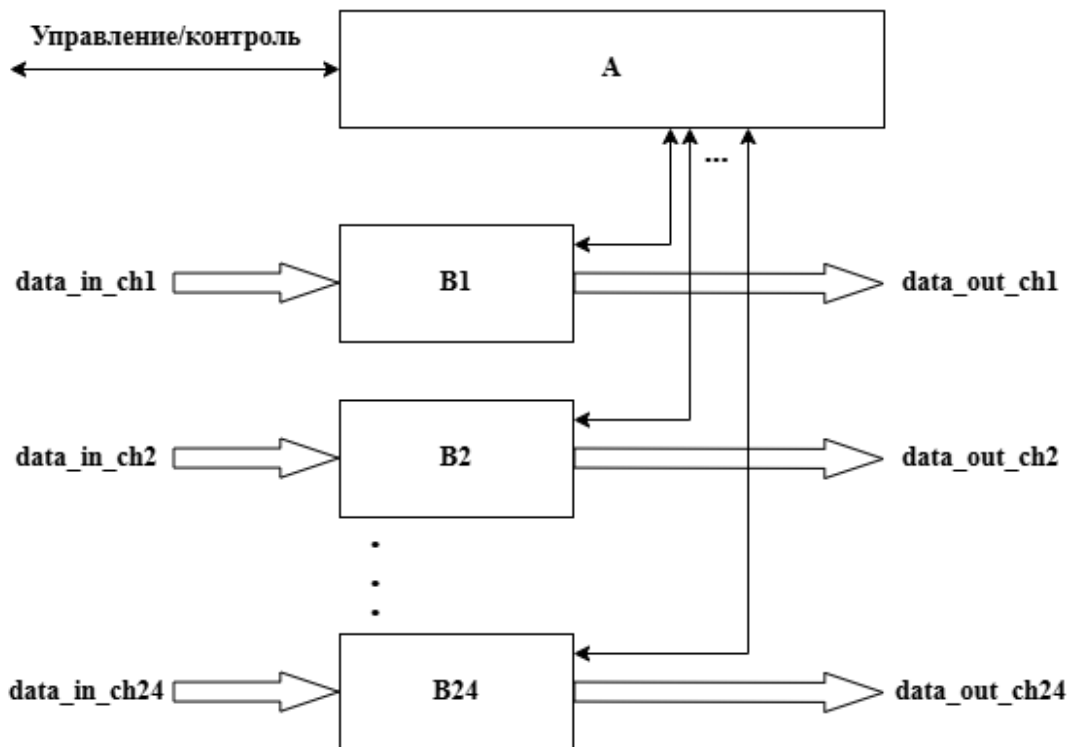


Рисунок 2.18. Общая структура программы

Блок управления (БУПР) – выполняет несколько функций, которые определяют работу всех остальных программных блоков, среди которых: поиск и задание опорного канала, установка допустимого диапазона

рассогласования, запуск процесса одновременного считывания данных из памяти.

В блоке управления реализована процедура назначения опорного канала. Сначала определяется задержка в каждом канале, затем вычисляется среднее арифметическое этих задержек. Опорным выбирается канал с задержкой, наиболее близкой к среднему значению. Если несколько каналов имеют одинаково близкие задержки, то выбирается первый по порядку.

Алгоритм назначения опорного канала выполняется следующим образом:

1. Ожидание внешней команды запуска программы (CMD_RUN).
2. При получении первого пришедшего строба `di_valid` от любого из каналов устанавливается флаг `sop_scan_start` в течении одного такта.
3. Запускается счетчик `sop_scan_cntr`.
4. После достижения счетчиком `sop_scan_cntr` значения максимально допустимого рассогласования, задаваемого параметром `max_pkg_skew`, получаем значения задержек `sop_scan_cntr_ch_latch` от блока приема данных каждого канала.
5. Вычисляем среднее арифметическое `avg_sum` для всех значений `sop_scan_cntr_ch_latch`.
6. В цикле перебираем каналы по порядку и сравниваем значение `sop_scan_cntr_ch_latch` с `avg_sum`, если такой канал найден, присваиваем переменной `ch_selector` индекс этого канала, переменной `window_center` - значение `avg_sum` и заканчиваем выполнение алгоритма, если в течение цикла не было найдено такого канала, переходим следующему пункту.
7. Начинаем поэтапное изменение значения `avg_sum`, для этого используем переменную `center_scan_index`. Если значение `center_scan_index` нечетное, то отнимаем от `avg_sum` значение переменной `center_scan_index`, инкрементируем `center_scan_index` и возвращаемся в пункт 6. Если значение `center_scan_index` четное, то прибавляем от

avg_sum значение переменной center_scan_index, инкрементируем center_scan_index и возвращаемся в пункт 6.

После назначения опорного канала формируется диапазон допустимого рассогласования, центрированный относительно начала массива операндов, передаваемых в опорном канале. Допустимое рассогласование задается параметром PS_max. Диапазон измеряется в тактах частоты поступления данных и должен быть установлен за PS_max тактов до первого данного опорного канала и закончиться через PS_max-1 тактов после.

Формирование диапазона осуществляется на основе счетчика allow_window_cntr. Флаг allow_window принимает значение 1 при allow_window_cntr = 1 и сбрасывается при достижении allow_window_cntr = max_pkg_skew * 2. Момент запуска счетчика allow_window_cntr определяется счетчиком расчета задержки sop_scan_cntr при выполнении условия:

$$sop_scan_cntr = PD + PS + window_center - max_pkg_skew - 3$$

где PD – размер массива операндов;

PS – размер служебного промежутка.

На этом этапе завершается первичная инициализация опорного канала: задается диапазон допустимого рассогласования и конфигурируется вход мультиплексора временных меток, соответствующий опорному каналу.

Рассмотрим подробнее программные блоки для обеспечения работы одного канала. В состав блока обработки потока данных одного канала входят следующие блоки: интерфейс трансивера (ИТ), блок приема данных (БПД), блок расчета контрольных сумм (БРКС), блок обработки ошибок целостности данных (БОУЦД), буферная память FIFO, блок управления записью (БУЗ), блок управления чтением (БУЧ).

Структура программного блока, обеспечивающего работу одного канала, представлена на рисунке 2.19.

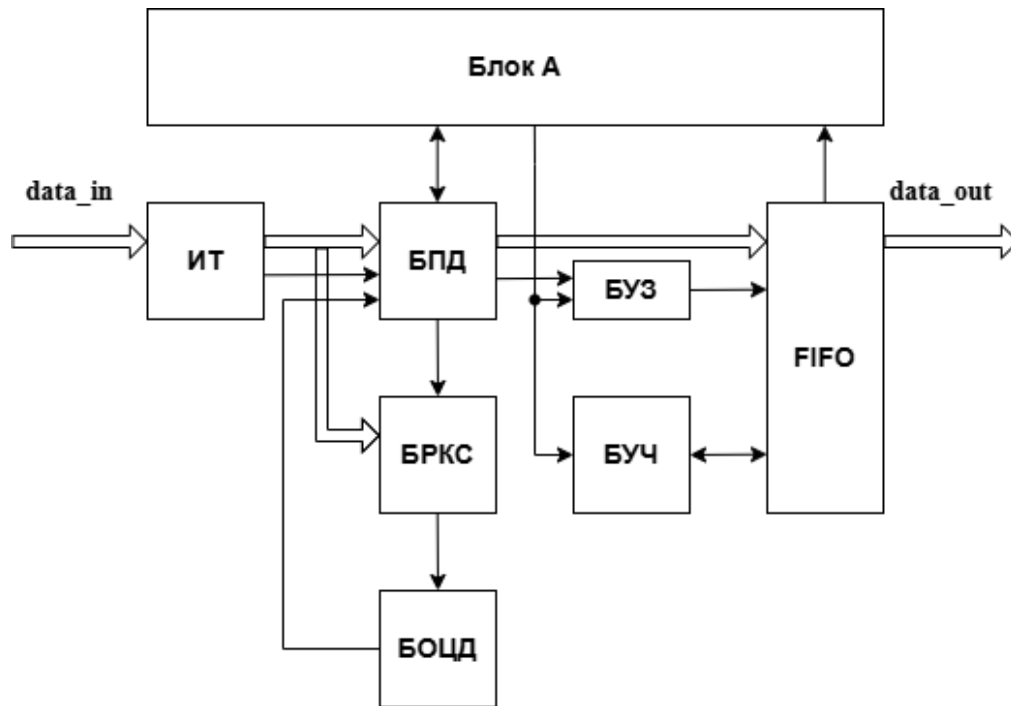


Рисунок 2.19. Структура программного блока одного канала

Интерфейс трансивера (ИТ) – программный блок, управляющий работой аппаратного MGT трансивера (Multigigabit Transceiver), обеспечивает настройку параметров трансивера и проведение тренировки канала. На выходе блока интерфейса выставляются данные на шину `di_data` разрядностью 32 бита в сопровождении строба `di_valid`. До приема данных, после успешного завершения тренировки, выставляется флаг готовности канала к работе `gx_on`, свидетельствующий о том, что канал активен и готов к работе. Флаг остается поднятым до момента получения признака потери соединения от аппаратного MGT трансивера. Флаг `gx_on` в дальнейшем будет использован другими блоками для контроля доступности канала.

Блок приема данных (БПД) – получает данные `di_data` со стробом `di_valid` от интерфейса трансивера, выделяет из заголовка временную метку `time_mark` и формирует флаги, необходимые для работы других блоков. В частности, формируются флаги:

1. `ch_data_sop` – флаг начала массива данных;
2. `ch_data_eop` – флаг конца массива данных;

3. `pkg_reach`, `pkg_reach_ltch` – флаг, устанавливаемый, если задержка канала входит в допустимый диапазон рассогласования;
4. `pkg_miss` – флаг, устанавливаемый, если задержка канала больше допустимого диапазона рассогласования.

Разработан алгоритм получения временной метки и формирования признаков начала массива, конца массива:

1. Если получен строб `di_valid`, установить флаг `ch_data_sop` на 1 такт.
2. Запустить счетчик принятых данных `ch_di_cntr`.
3. Получить первое слово временной метки `time_mark`.
4. Получить второе слово временной метки `time_mark`.
5. Если значение счетчика `ch_di_cntr = pkg_depth - 1`, установить флаг конец массива `ch_data_eop` на 1 такт. `Pkg_depth` - длина массива операндов.
6. Если значение счетчика `ch_di_cntr = pkg_depth`, сбросить счетчик.

Данный алгоритм позволит сформировать признаки начала и конца массива в нужные моменты времени, что необходимо для работы других блоков.

Блок расчета контрольных сумм (БРКС) – подсчитывает контрольную сумму для принятых данных. Блок получает данные `di_data` со стробом `di_valid` от блока интерфейса трансивера и флаг `ch_crc_check` от блока приема данных. При поступлении строба `di_valid` запускается счетчик принятых данных `data_counter` и производится расчет контрольной суммы принимаемых данных. Для расчета контрольных сумм был применен известный алгоритм CRC-16 Modbus [24] с полиномом `0x8005` и начальным значением `0xFFFF`. Контрольная сумма считается для всего массива операндов и вновь инициализируется при каждом новом массиве. Для 32-разрядных операндов массива контрольная сумма получается конкатенацией двух 16-разрядных значений, посчитанных независимо для старшей и младшей частей всех операндов массива. После достижения счетчиком значения, заданного

параметром `pkg_depth`, расчет контрольной суммы прекращается и производится сравнение принятой и рассчитанной контрольных сумм. При совпадении устанавливается флаг `src_pass`, длительностью один такт, иначе выставляется флаг `src_fail`. Флаги `src_fail` и `src_pass` используются в блоке обработки ошибок целостности данных.

Блок обработки ошибок целостности данных (БОЦД) – формирует признак необходимости отключения канала, в котором число ошибок целостности данных превышает заданное параметром `err_to_offline` значение. Для этого в блоке реализован счетчик ошибок `err_cntr`, который инкрементируется при получении флага `src_fail` от блока расчета контрольных сумм. Если значение счетчика ошибок превышает установленное параметром `err_to_offline` значение, то взводится флаг отключения канала `ch_offline`. Флаг `ch_offline` будет сохранять свое состояние до тех пор, пока не будет принято определенное количество безошибочно принятых массивов подряд. Для подсчета безошибочно принятых массивов используется счетчик `good_cntr`, который инкрементируется при поступлении флага `src_pass` и сбрасывается каждый раз при получении флага `src_fail`. По достижении счетчиком `good_cntr` значения, установленного параметром `good_to_online`, флаг `ch_offline` сбрасывается.

Алгоритм обработки ошибок целостности данных выполняется следующим образом:

1. Если получен флаг ошибки целостности данных `src_fail` от блока расчета контрольных сумм, инкрементируем счетчик ошибок `err_cntr`.
2. Если получен флаг безошибочно принятого массива `src_pass` от блока расчета контрольных сумм, инкрементируем счетчик безошибочно принятых массивов `good_cntr`.
3. Если счетчик безошибочно принятых массивов `good_cntr` равен параметру `good_to_online`, то сбрасываем счетчик ошибок `err_cntr` и переходим к пункту 1, иначе переходим к пункту 4.

4. Если счетчик ошибок `err_cntr` равен параметру `err_to_offline`, то устанавливаем флаг отключения канала `ch_offline`, сбрасываем счетчик `err_cntr` и переходим к пункту 5, иначе переходим к пункту 1.
5. Если получен флаг безошибочно принятого массива `src_pass` от блока расчета контрольных сумм, инкрементируем счетчик безошибочно принятых массивов `good_cntr`.
6. Если получен флаг ошибки целостности данных `src_fail`, сбрасываем счетчик безошибочно принятых массивов `good_cntr`.
7. Если счетчик `good_cntr` равен параметру `good_to_online`, то опускаем флаг отключения канала `ch_offline`, сбрасываем счетчик `good_cntr` и переходим к пункту 1, иначе переходим к пункту 5.

Данный алгоритм работает безостановочно, до поступления флага (сигнала) общего сброса (`reset = 1`) и позволяет осуществить статистику ошибок целостности данных в канале для оценки его работоспособности. Формирует признак необходимости отключения канала (`ch_offline = 1`) при превышении заданного количества ошибок, а также признак необходимости включения канала (`ch_offline = 0`) при определенном количестве безошибочно принятых массивов подряд. Флаг `ch_offline` используется в других блоках для прекращения поступления данных отключенного канала.

Буферная память FIFO – блочная двухпортовая память (Dual-port Block RAM), сконфигурированная в режим работы FIFO (first in, first out – первым вошел, первым вышел). Является аппаратным ресурсом ПЛИС. Имеет два независимых набора сигналов, позволяющих проводить операции чтения или записи одновременно. Использование такого типа памяти является архитектурно безопасным способом передачи данных между разными тактовыми доменами.

Структура памяти FIFO представлена на рисунке 2.20.

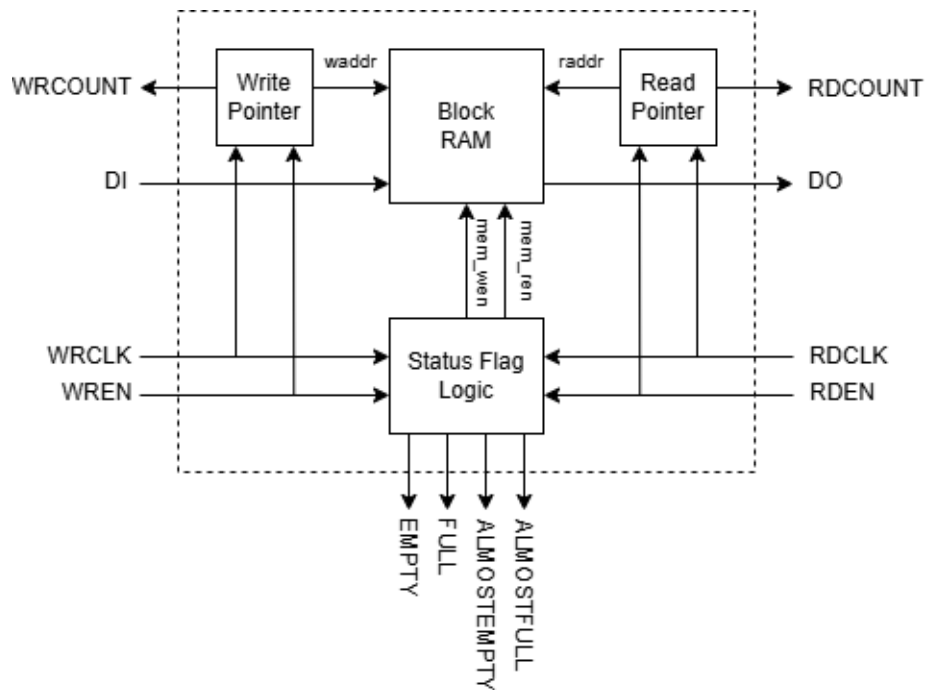


Рисунок 2.20. Структура памяти FIFO

На вход данных FIFO (DIN) подключается шина данных di_data . Вход разрешения записи (WREN) управляется блоком управления записью, а вход разрешения чтения (RDEN) - блоком управления чтением.

Запись операндов, выставляемых на шину данных DIN, происходит каждый такт, пока удерживается сигнал WREN. При операции чтения формирование следующего операнда на выходе DO осуществляется каждый такт, пока удерживается сигнал RDEN. Поток данных формируется автоматически, отсутствует необходимость формирования адресов для блока памяти. Адреса записи (waddr) и чтения (raddr) образуются с помощью блоков формирования указателя записи Write Pointer и указателя чтения Read Pointer.

Блок управления записью (БУЗ) осуществляет управление записью данных в память FIFO. Блок работает непрерывно, до поступления сигнала общего сброса ($reset = 1$). Получает флаги начала массива данных ch_data_sop и конца массива ch_data_eop от блока приема данных, а также флаг разрешения записи $allow_window_flag$ от блока управления. Если флаг начала массива данных ch_data_sop поступает, пока активен флаг разрешения записи $allow_window_flag$, то формируется флаг записи в FIFO ($wr_en = 1$). Флаг записи снимается, когда поступает флаг признака конца массива ch_data_eop ,

и блок ожидает поступления флага `ch_data_sop` от следующего массива данных. Блок позволяет записывать в память только те массивы данных, начала которых попадают в допустимый интервал рассогласования. Массивы данных, начала которых не попадают в допустимый интервал рассогласования, не будут записаны в FIFO.

Блок управления чтением (БУЧ) осуществляет управление чтением данных из памяти FIFO. Блок получает флаг запуска процесса одновременного чтения `read_run` и значение счетчика вычитанных данных `fifo_rd_cntr` от блока управления, флаг признака опустошения FIFO от буферной памяти и флаг признака попадания начала массива данных в допустимый интервал рассогласования `pkg_reach_ltch` от блока приема данных. Формирует флаг разрешения чтения из FIFO `rd_en` и сигнал сброса выходного регистра (`out_reg_reset = 1`). Блок начинает работу после поступления команды запуска процесса одновременного чтения `read_run` и работает до тех пор, пока не будет получен сигнал общего сброса (`reset = 1`).

Алгоритм управления чтением выполняется следующим образом:

1. Ожидание команды `read_run` от блока управления.
2. Если получен признак попадания начала массива данных в допустимый интервал рассогласования `pkg_reach_ltch=1`, то устанавливаем флаг разрешения чтения `rd_en=1` и снимаем сигнал сброса выходного регистра `out_reg_reset=0`, иначе переходим к пункту 5.
3. Проверяем признак опустошения памяти `fifo_empty`. Если `fifo_empty=1` (память опустошена), то переходим к пункту 5.
4. Если значение счетчика вычитанных данных `fifo_rd_cntr=pkg_depth-1`, то переходим в пункт 2.
5. Установка флага запрета чтения `rd_en=1` и сброс выходного регистра `out_reg_reset=1`, возврат к пункту 3.

Применение данного алгоритма в каждом канале позволит обеспечить начало считывания данных из всех FIFO в один момент времени при условии, что массивы данных прошли проверку в блоке приема данных на попадание в допустимый интервал рассогласования. Если канал отключится по разным причинам, данные из FIFO отключенного канала будут считаны до его полного опустошения. При восстановлении работоспособности канала данные из FIFO начнут считываться строго в определенный момент, соответствующий интервалу следования массивов, чтобы не нарушить согласованный поток данных.

Разработана прикладная программа на языке высокого уровня COLAMO [115]. Занимаемый программой ресурс на ПЛИС XC7VX485T представлен в таблице 2.4.

Таблица 2.4. Занимаемый программой ресурс на ПЛИС XC7VX485T

| Ресурс | Всего | Занимаемый ресурс | % |
|--------|--------|-------------------|--------|
| LUT | 303600 | 13134 | 4,33 % |
| FF | 607200 | 11402 | 1,88 % |
| BRAM | 1030 | 24 | 2,33 % |
| GTX | 56 | 24 | 43 % |

В соответствии с данными, представленными в таблице 2.4, разработанная программа обеспечения изохронной передачи данных, основанная на комбинированном методе, демонстрирует высокую эффективность использования ресурсов программируемой логической интегральной схемы (ПЛИС), занимая лишь незначительную их часть. Свободный ресурс может быть задействован для реализации алгоритмов цифровой обработки сигналов. Следует отметить, что память BRAM особенно важна для реализации алгоритмов ЦОС и является критическим ресурсом.

2.3 Анализ гарантоспособности реализации комбинированного метода

Оценим увеличение гарантоспособности программного комплекса при применении разработанного комбинированного метода изохронной передачи данных. Под гарантоспособностью понимается комплексное свойство системы (программного комплекса) предоставлять требуемые услуги, которым можно оправданно доверять [116]. На оценку гарантоспособности влияют следующие компоненты:

- угрозы и воздействия на систему, которые могут привести к негативному изменению ее состояния и характеристик;
- способы обнаружения и парирования угроз и воздействий на систему;
- первичные и производные от них индивидуальные и общие вторичные свойства, отражающие различные аспекты гарантоспособности системы.

К числу первичных свойств (атрибутов гарантоспособности) могут быть отнесены следующие [117]:

- безотказность (reliability) – свойство непрерывно предоставлять корректные (требуемые) услуги;
- готовность (availability) – свойство доступности ресурсов вычислительной системы для предоставления требуемых услуг;
- живучесть (survivability) – свойство минимизировать снижение и сохранять в приемлемых пределах объём и качество предоставляемых услуг при отказах;
- сохранность (safety) – свойство исключать или минимизировать вредные (катастрофические) последствия при отказах для пользователей, других систем или окружающей среды;
- конфиденциальность (confidentiality) – свойство препятствовать неавторизованному доступу к информации об услугах;
- целостность (integrity) – свойство исключать непредусмотренные изменения системы и предоставляемых услуг;

- достоверность (high confidence) – свойство правильно оценивать корректность предоставляемых услуг, т.е. определять степень доверия к услуге;

- обслуживаемость (maintainability) – свойство приспособленности к модификациям и ремонту.

Вышеуказанные атрибуты могут использоваться в большей или меньшей степени в зависимости от свойств исследуемого программного комплекса. Например, готовность ресурсов требуется всегда, хотя и в разной степени, тогда как живучесть, сохранность, могут требоваться, а могут и не требоваться. Система, обладающая атрибутами гарантоспособности, должна интерпретироваться в относительном (вероятностном) смысле, а не в абсолютном, детерминированном смысле: из-за неизбежного наличия или возникновения неисправностей системы никогда не бывают полностью живучими и сохранными.

Гарантоспособный программный комплекс требует совместного использования набора из четырех действий: предотвращение сбоев, живучесть (как обеспечить правильное обслуживание данных при наличии неисправностей), устранение неисправности (как уменьшить количество или серьезность неисправностей), прогнозирование неисправностей (как оценить текущее количество, вероятность возникновения неисправностей в будущем и вероятные последствия).

Для количественной оценки гарантоспособности программного комплекса, как правило, применяется атрибутивная модель гарантоспособности, описывающая общие характеристики системы с помощью атрибутов и метрик [118]. Атрибут объединяет различные свойства программного комплекса, в то время как метрика представляет численное значение определенного параметра атрибута, отражающее только одно из его свойств.

Точное определение метрик и атрибутов программного комплекса позволит подойти к формализации обобщенного критерия уровня достигнутой

гарантоспособности разрабатываемого программного комплекса. Для этого необходимо каждый атрибут модели рассматривать как набор метрик, которые могут быть оценены расчётными, экспериментальными или экспертными методами. На основе количественных оценок метрик вычисляются количественные оценки атрибутов и далее через них вычисляется комплексная количественная оценка достигнутого уровня гарантоспособности рассматриваемого программного комплекса.

Согласно работе [118], в качестве аналитического представления атрибутивной модели гарантоспособности используется функционал G , составляющими которого являются нормированные значения количественных оценок уровней реализации атрибутов и метрик с соответствующими весовыми коэффициентами.

$$G = \sum_{i=1}^n B_i A_i, \quad (2.17)$$

где n – количество атрибутов атрибутивной модели гарантоспособности;

B_i – коэффициент влияния i -го атрибута;

A_i – количественная оценка уровня выполнения i -го атрибута в относительных величинах.

В свою очередь, количественная оценка уровня выполнения i -го атрибута (A_i) определяется по формуле

$$A_i = \sum_{j=1}^{m_j} \beta_{ij} M_{ij}, \quad (2.18)$$

где m_i – количество метрик i -го атрибута;

β_{ij} – вес j -й метрики i -го атрибута;

M_{ij} – количественная оценка уровня выполнения j -й метрики i -го атрибута в относительных величинах.

Величины весовых коэффициентов зависят от особенностей применения разрабатываемого программного комплекса и могут быть оценены экспертным методом.

Для расчета гарантоспособности программного комплекса изохронной передачи данных следует выделить наиболее значимые атрибуты. Рассматриваемый программный комплекс входит в состав программно-аппаратных средств, осуществляющих когерентную обработку данных. Поскольку программный комплекс предназначен исключительно для обеспечения изохронной передачи данных от множества каналов данных и не рассматривает функциональное преобразование данных, будем оценивать только те атрибуты, которые непосредственно влияют на его работу. Экспериментальные исследования показали, что вероятность отказов и сбоев, возникающих в части вычислительных функций на два-три порядка ниже, чем отказы и сбои, возникающие при передаче данных от множества каналов.

Следует отметить, что в контексте программного комплекса передачи данных наличие атрибутов, таких как конфиденциальность и обслуживаемость, не имеет значения. Обслуживаемость относится к способности к модификациям и ремонту, что обычно связано с аппаратным обеспечением. Конфиденциальность, в свою очередь, предотвращает несанкционированный доступ к информации об услугах, но в данном случае физический доступ к услугам снаружи невозможен, поэтому в дальнейшем влияние этих атрибутов на гарантоспособность программного комплекса не учитывается.

Такие атрибуты, как безотказность, сохранность и достоверность, для рассматриваемого программного комплекса являются интегральными показателями, базируемыми на других атрибутах гарантоспособности.

Можно утверждать, что наиболее важными атрибутами для программного комплекса изохронной передачи данных от множества каналов являются:

- целостность, которая для рассматриваемого программного комплекса понимается как целостность передаваемых данных;

- готовность, которая подразумевает доступность каналов передачи данных;

- живучесть, которая понимается как способность сохранять работоспособность в приемлемых пределах при превышении допустимых значений ошибок.

Согласно атрибутивной модели оценки гарантоспособности, необходимо выделить метрики для каждого атрибута, представляющих собой выражение только одного из свойств атрибута. Поскольку процессы носят вероятностно-недетерминированный характер, в качестве метрик выберем вероятность возникновения возможной причины.

Целостность данных может быть нарушена из-за возникновения битовых ошибок или из-за нарушения формата передаваемых данных, поэтому в качестве метрик для атрибута «целостность» будем использовать вероятность битовых ошибок данных в канале и вероятность нарушения целостности формата данных.

Готовность каналов может быть нарушена из-за отключения канала по причине неисправности или из-за неверной коммутации вследствие ошибок оператора или обслуживающего персонала, поэтому в качестве метрик для атрибута готовность используем вероятность отключения канала и вероятность ошибки оператора.

Живучесть отражает способность программного комплекса сохранять работоспособность при превышении допустимых значений ошибок. Программный комплекс изохронной передачи данных имеет возможность парировать ошибки рассогласования задержек между каналами в определенных заданных пределах. Однако могут возникать ситуации, при которых значение рассогласования превышают допустимый уровень. Кроме того, есть вероятность маскирования ошибки рассогласования если ее значение совпадет с периодом следования массивов. В качестве метрик,

отражающих свойства атрибута «живучесть», будем использовать вероятность превышения допустимого значения рассогласования задержки канала и вероятность совпадения задержки с периодом следования массивов данных. В таблице 2.5 приведены атрибуты и метрики, используемые для оценки гарантоспособности программного комплекса изохронной передачи данных.

Таблица 2.5 – Атрибуты и метрики

| Атрибут | Вес атрибута | Метрика | Вес метрики | Границы |
|-------------|--------------|---|-------------|---------|
| Целостность | 0,98 | Коэффициент битовых ошибок | 0,98 | 0..1 |
| | | Вероятность нарушения целостности формата данных | 0,96 | 0..1 |
| Готовность | 0,96 | Вероятность отключения канала | 0,97 | 0..1 |
| | | Вероятность ошибки оператора | 0,94 | 0..1 |
| Живучесть | 0,99 | Вероятность превышения допустимого значения рассогласования задержки канала | 0,99 | 0..1 |
| | | Вероятность совпадения задержки с периодом следования массивов данных | 0,9 | 0..1 |

Весовые коэффициенты атрибутов и метрик определены на основе экспертных оценок, учитывающих экспериментальные данные.

Очевидно, что гарантоспособность программного комплекса изохронной передачи данных в значительной степени определяется требуемой длительностью безошибочной работы и числом каналов передачи. На практике для сложных программных комплексов в системах с ФАР принимают уровень гарантоспособности 0,95, а минимальное время безотказной работы системы составляет не менее шести часов.

Вероятность каждого типа ошибок в зависимости от количества каналов P_m^N комплекса рассчитывается по формуле

$$P_m^N = (1 - (1 - P_m)^N) \cdot W_m, \quad (2.19)$$

где P_m - вероятность ошибки в одном канале;

N – количество каналов;

W_m – весовой коэффициент метрики.

Общая вероятность для каждого атрибута P_a определяется по формуле

$$P_a = (1 - (1 - P_{m1}^N) \cdot (1 - P_{m2}^N)) \cdot W_a, \quad (2.20)$$

где W_a – весовой коэффициент атрибута.

Общий уровень гарантоспособности оценивается по формуле

$$G = 1 - (1 - (1 - P_{a1}) \cdot (1 - P_{a2}) \cdot (1 - P_{a3})). \quad (2.21)$$

Проведен анализ гарантоспособности базового метода. Результаты показали, что требуемый уровень гарантоспособности 0,95 за время работы комплекса 6 часов достигается при количестве каналов данных в системе до 10900.

Для оценки гарантоспособности программной реализации комбинированного метода изохронной передачи данных необходимо учесть коэффициенты парирования ошибок. В отличие от базового метода, комбинированный метод включает процедуры обнаружения и устранения неисправностей, что требует дополнительных корректировок в расчетах.

В программе, использующей комбинированный метод, предусмотрены механизмы для обработки наиболее вероятных ошибок, таких как аварийное отключение канала, повреждение данных и выход за допустимые пределы рассогласования. Остальные ошибки не отслеживаются и не обрабатываются, поскольку их возникновение маловероятно и несущественно влияет на общую гарантоспособность системы в течение заданного времени работы при фиксированном числе каналов. Включение дополнительных механизмов для редких ошибок приведет к увеличению используемых ресурсов ПЛИС без значительного повышения гарантоспособности.

В таблице 2.6 приведены коэффициенты парирования наиболее вероятных ошибок для комбинированного метода изохронной передачи данных. Эти коэффициенты, определяемые экспертными оценками,

варьируются от 0 до 1, где 0 означает отсутствие защиты от соответствующей ошибки.

Таблица 2.6 – Коэффициенты парирования для реализации комбинированного метода

| Тип ошибки | Коэффициент парирования |
|---|-------------------------|
| Нарушение целостности | 0,55 |
| Нарушения целостности формата данных | 0,4 |
| Аварийное отключение канала | 0,77 |
| Ошибка оператора | 0,6 |
| Превышение допустимого значения рассогласования задержки канала | 0,85 |
| Совпадение задержки с периодом следования массивов данных | 0 |

Добавим коэффициент парирования ошибок K_p в формулу (2.19) и получим следующее выражение:

$$P_m^N = (1 - (1 - P_m)^N) \cdot W_m \cdot K_p. \quad (2.22)$$

Расчет гарантоспособности выполним по той же методике, что и для базового метода. Для корректного сравнения двух методов вычислим гарантоспособность при одинаковом числе каналов. Количество каналов возьмем то, при котором реализация базового метода обеспечивает требуемый уровень гарантоспособности 0,95, определим этот уровень как G1. Получим значение гарантоспособности для реализации комбинированного метода G2=0,987 при числе каналов 10900.

Определим, насколько улучшилась гарантоспособность программного комплекса, реализующего комбинированный метод. Для этого воспользуемся коэффициентом эффективности улучшения (гарантоспособности) IEF, который определяется по ГОСТ Р 51901.16-2017 «Повышение надежности. Статистические критерии и методы оценки» как

$$IEF = \frac{\lambda_1(t)}{\lambda_2(t)}, \quad (2.23)$$

где λ – интенсивность отказов;

t – время работы.

Интенсивность отказов определяется как отношение отказавших объектов n в единицу времени t к среднему числу исправно работающих объектов N_{cp} в данный отрезок времени t

$$\lambda(t) = \frac{n(t)}{N_{cp} \cdot \Delta t}. \quad (2.24)$$

Подставим в формулу (2.1) получим

$$IEF = \frac{n_1(t)}{N_{cp1} \cdot \Delta t} : \frac{n_2(t)}{N_{cp2} \cdot \Delta t}. \quad (2.25)$$

Поскольку сравнение реализаций базового и комбинированного методов производится в одинаковых условиях, на одном и том же оборудовании, за одинаковое время работы и при фиксированном числе каналов, отличия между N_{cp1} и N_{cp2} будут только в четвертом-пятом знаке, следовательно $N_{cp1} \approx N_{cp2}$. Получаем следующее выражение:

$$IEF \approx \frac{n_1(t)}{n_2(t)}. \quad (2.26)$$

Коэффициент эффективности улучшения (гарантоспособности) IEF можно выразить через отношение вероятностей отказа программного комплекса Q

$$Q(t) = \frac{n(t)}{N_0}, \quad (2.27)$$

где n – число отказавших объектов за время t ;

N_0 – общее число работоспособных элементов в начальный момент времени, приблизительно равно N_{cp} при большом числе элементов.

Тогда

$$\frac{Q_1}{Q_2} = \frac{n_1(t)}{N_0} : \frac{n_2(t)}{N_0} = \frac{n_1(t)}{n_2(t)}. \quad (2.28)$$

Данная формула неразличима с выражением (2.4), следовательно:

$$IEF \approx \frac{Q_1}{Q_2} = \frac{1 - G_1}{1 - G_2}. \quad (2.29)$$

Формула (2.29) коррелирует с параметром RIF, описанным в известной работе А. Авижениса [119], определяющим коэффициент улучшения надежности.

Подставив значения гарантоспособности для реализации базового метода G1 и гарантоспособности для реализации комбинированного метода G2, получим коэффициент эффективности улучшения (гарантоспособности) $IEF = 3,85$.

Таким образом, применение комбинированного метода изохронной передачи данных вместе с процедурами обнаружения и парирования типовых аварийных ситуаций позволяет повысить гарантоспособность программного комплекса обеспечения изохронной передачи данных в 3,85 раз по сравнению с базовым методом.

Зависимость гарантоспособности от количества каналов показана на рисунке 2.19.

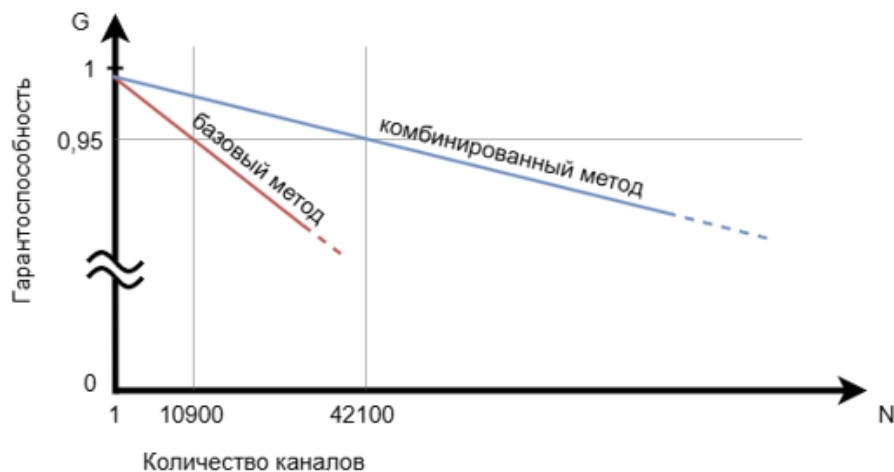


Рисунок 2.21. Зависимость уровня гарантоспособности от количества каналов

Согласно графику, при требуемом уровне гарантоспособности 0,95 комбинированный метод обеспечивает этот показатель при 42 500 каналах.

2.4 Выводы

1. Известные методы изохронной передачи данных в комплексах ЦОС основаны на использовании системы единого машинного времени и/или опорной тактовой частоты.

2. Существующие методы и средства обеспечения изохронной передачи данных в комплексах ЦОС с заданным уровнем гарантоспособности имеют ряд недостатков:

- существенное снижение пропускной способности за счет выделения части ширины канала для передачи временных меток;

- рост аппаратных и/или временных затрат, связанный с необходимостью реализации сложных процедур синхронизации данных между разными каналами;

- отсутствие динамической компенсации задержек между разными каналами при передаче данных.

3. Разработан комбинированный метод изохронной передачи данных, сочетающий использование единого машинного времени и опорной тактовой частоты, отличающийся от известных введением служебных промежутков при передаче массивов операндов.

4. Протокол комбинированного метода предполагает передачу данных на несколько большей частоте относительно частоты поступления данных. В частности, для передачи данных от антенных элементов ФАР, оцифрованных на частоте 240 МГц, предложена частота 255 МГц.

5. В служебные промежутки каждого информационного потока для реализации процедуры выравнивания задержек в каналах данных следует

разместить заголовок массива операндов с временной меткой, а в конце подмассива добавить контрольную сумму для проверки целостности данных.

6. Определены целесообразные параметры размера массива операндов и служебного промежутка для достижения оптимальной пропускной способности и уменьшения аппаратных затрат. Размер массива операндов составил 64 слова, служебного промежутка – 4 слова.

7. Выделены наиболее вероятные аварийные ситуации, характерные для системы изохронной передачи данных:

- аварийное отключение канала;
- нарушение целостности передаваемой информации;
- превышение допустимого порога рассогласования задержек поступления массивов операндов.

8. Разработаны алгоритмы и создана структура программы для РВС, реализующая комбинированный метод изохронной передачи данных.

9. В программе для ПЛИС реализованы процедуры обнаружения и парирования аварийных ситуаций:

- процедура назначения опорного канала и формирования диапазона допустимого рассогласования;
- процедура проверки целостности данных с использованием контрольных сумм;
- процедура зануления данных в сбойных каналах.

Программа выравнивания потоков данных на ПЛИС XC7VX485T использовала не более 5% от общего объема доступных ресурсов.

10. Выполнен анализ гарантоспособности разработанной программы, реализующей комбинированный метод изохронной передачи данных. Применение комбинированного метода позволяет повысить гарантоспособность программного комплекса обеспечения изохронной передачи данных в 3,85 раз по сравнению с базовым методом, что обеспечивает заданный уровень гарантоспособности (0,95) в течение шестичасового

интервала работы комплекса ЦОС с ФАР при использовании не более 42100 антенных элементов (информационных каналов).

3. МОДЕРНИЗИРОВАННЫЙ МЕТОД ИЗОХРОННОЙ ПЕРЕДАЧИ ДАННЫХ

3.1 Необходимость усовершенствования комбинированного метода изохронной передачи данных

Современные радиолокационные системы с фазированными антенными решетками (ФАР) в настоящее время активно развиваются. Они находят применение не только в традиционных сферах, таких как воздушное наблюдение, но и в новых задачах, таких как мониторинг космического мусора, количество которого неуклонно растет с каждым годом.

Проблема загрязнения околоземного космического пространства становится всё более острой. С каждым годом количество техногенных объектов на орбите Земли неуклонно увеличивается (рисунок 3.1).

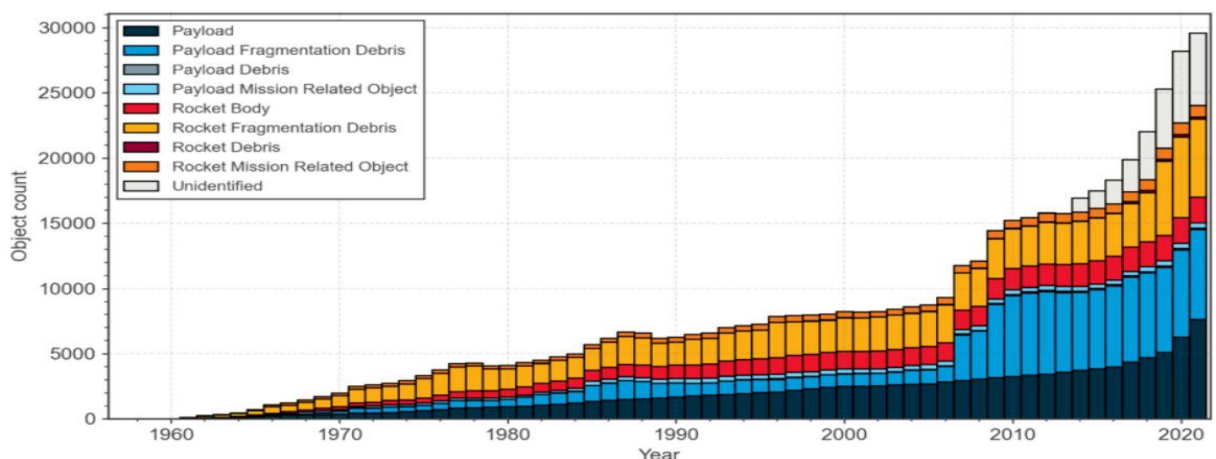


Рисунок 3.1. Динамика роста количества объектов в околоземном пространстве

На это есть две ключевые причины. Во-первых, бурное развитие космической отрасли приводит к значительному росту числа запусков. Как государственные, так и частные компании регулярно выводят на орбиту новые спутники и другие космические аппараты. Во-вторых, современная космонавтика пока не располагает эффективными способами очистки орбиты от мусора. Отработавшие свой срок космические аппараты и их фрагменты продолжают накапливаться в околоземном пространстве, создавая

потенциальную угрозу для действующих космических объектов. Конечно, обломок со временем теряет высоту из-за сопротивления атмосферы, пока не сгорит в плотных слоях. Но из-за нелинейной зависимости скорости падения от высоты этот процесс может занять сотни или тысячи лет.

С увеличением объёма космического мусора на околоземной орбите значительно возрастает вероятность возникновения синдрома Кесслера - опасного каскадного эффекта, при котором столкновение объектов порождает новые обломки, что, в свою очередь, приводит к цепной реакции разрушений [120]. В результате даже небольшое количество мусора может со временем превратиться в плотное облако опасных фрагментов, делающее ближний космос практически непригодным для эксплуатации, угрожая спутникам, международной космической станции и будущим космическим миссиям. Важно отметить, что этот процесс носит экспоненциальный характер: чем больше мусора накапливается, тем быстрее происходит его размножение при столкновениях.

Столкновения на орбите уже происходят. Ярким примером является столкновение американского коммерческого спутника Iridium 33 с российским космическим аппаратом Космос 2251 в 2009 году. Эта авария привела к образованию более 2000 обломков, которые до сих пор представляют серьезную угрозу для действующих спутников и международной космической станции. Инцидент подчеркнул критическую необходимость разработки систем предотвращения столкновений.

В таблице 3.1 приведены данные о количестве объектов космического мусора на низкой околоземной орбите (НОО), согласно информации Национального управления по аэронавтике и исследованию космического пространства (НАСА) по состоянию на 2025 год [121].

Современные радиолокационные станции, занимающиеся мониторингом околоземного космического пространства, способны различать объекты размеров 10 см. Как видно из таблицы 3.1, существует множество мелких фрагментов космического мусора, которые, несмотря на свои

небольшие размеры, представляют серьезную опасность. Дело в том, что даже крошечные фрагменты, разогнанные до высокой скорости, обладают колоссальной кинетической энергией, способной причинить значительные повреждения [10]. Поэтому обнаружение, а также отслеживание таких объектов является важной задачей для обеспечения безопасности космических полетов. Необходимо создать новые РЛС, которые будут работать в составе системы предупреждения о столкновениях, и способные различать более мелкие фрагменты космического мусора.

Таблица 3.1. Количество и размеры мусора на НОО

| Размер обломков | Количество |
|-----------------|------------|
| Больше 10 см | 30 тыс. |
| От 1 до 10 см | 0,5 млн. |
| От 1 мм до 1 см | 100 млн. |

Для того чтобы РЛС могла обнаруживать объекты с меньшими размерами, необходимо повысить её разрешающую способность по угловым координатам (азимуту и углу места) [1]. Для этого необходимо уменьшить ширину основного лепестка диаграммы направленности, являющегося областью излучения или приема антенны, в которой концентрируется основная мощность сигнала. [6].

Чаще всего ширина луча определяется шириной луча половинной мощности (HPBW - Half Power Beam Width). Для определения HPBW требуется измерить угловой размер луча на уровне, который на 3 дБ ниже максимального [122]. В случае эквидистантных фазированных антенных решёток (с равным расстоянием между элементами) HPBW может быть приближённо вычислена по следующей формуле:

$$\theta_{HPBW} \approx \frac{0.886 \cdot \lambda}{N \cdot d \cdot \cos\theta_0}, \quad (3.1)$$

где λ – длина волны;

N – количество антенных элементов ФАР;

d – расстояние между антенными элементами (как правило, $\lambda/2$);

$\cos\theta_0$ – угол сканирования.

Из формулы (3.1) видно, что уменьшение ширины основного лепестка диаграммы направленности достигается двумя способами: уменьшением длины волны λ и увеличением количества антенных элементов ФАР.

Уменьшение длины волны λ приводит к необходимости увеличения тактовой частоты аналого-цифровых преобразователей, осуществляющих дискретизацию принимаемых сигналов, в соответствии с теоремой Котельникова. С ростом числа антенных элементов пропорционально увеличивается и количество информационных каналов, требующих изохронной передачи данных. Следствием данных факторов является увеличение объема передаваемой информации, что предъявляет повышенные требования к гарантоспособности системы изохронной передачи данных.

Согласно формуле (3.1), рассчитаем оптимальное количество каналов для достижения высокого разрешения фазированной антенной решетки. Рассмотрим антенну с равным числом излучающих элементов по горизонтали и вертикали, что обеспечит одинаковое разрешение по азимуту и углу места. В качестве рабочей частоты выберем типовое для подобных радиолокационных систем значение - 3 ГГц. Расстояние между элементами антенны составит $\lambda/2$. Расчет проведем без отклонения луча, что позволит получить оптимальный результат, поскольку при сканировании ширина основного лепестка диаграммы направленности увеличивается. Результат представлен на рисунке 3.2.

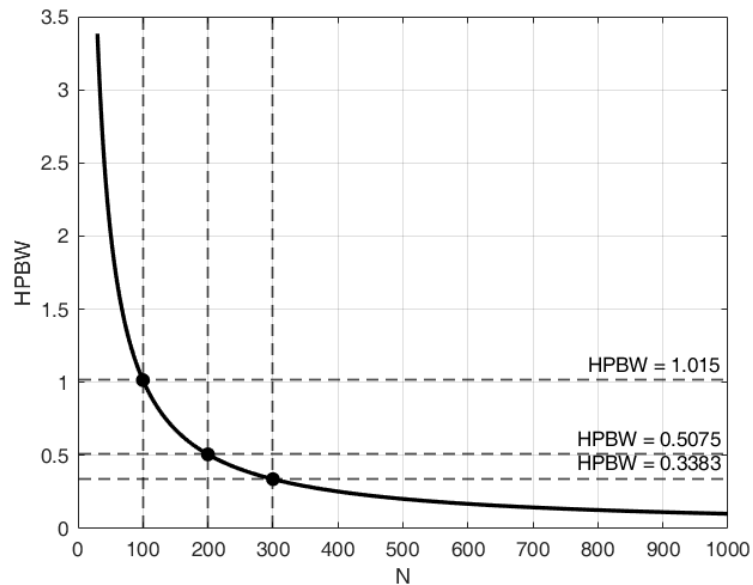


Рисунок 3.2. Зависимость углового разрешения ФАР от количества антенных элементов

Согласно графику, наращивание количества антенных элементов целесообразно лишь до определенного уровня, поскольку дальнейшее увеличение не приводит к существенному повышению разрешающей способности. Следует выбрать разумное количество, обеспечивающее достаточный прирост разрешения. Компромиссным вариантом можно считать 90000 элементов - при этом значении прирост разрешающей способности еще остается значительным.

В таких условиях комбинированный метод изохронной передачи перестает справляться с обеспечением требуемого уровня гарантоспособности. Для сохранения заданного уровня гарантоспособности при увеличении числа каналов до 90000 необходима модернизация комбинированного метода.

Основным параметром гарантоспособности, на который можно влиять и который напрямую определяет общий уровень гарантоспособности системы, является коэффициент парирования ошибок. Следовательно, ключевым направлением модернизации комбинированного метода является повышение эффективности процедур парирования. Кроме того, важно разработать новые

процедуры обработки ошибок, которые ранее не учитывались. Это особенно актуально в условиях масштабирования системы – при значительном увеличении числа каналов даже ранее незначительные ошибки могут оказывать существенное влияние на общую гарантоспособность.

3.2 Модернизированный метод изохронной передачи данных

В системе изохронной передачи данных одной из наиболее серьезных проблем являются ошибки, возникающие при аварийном отключении каналов передачи. В большинстве случаев такие ошибки успешно обнаруживаются и устраняются с помощью предусмотренных процедур парирования, которые были рассмотрены во второй главе данной диссертационной работы. Однако существует особый случай, представляющий наибольшую опасность для корректной работы системы, - аварийное отключение канала, назначенного в качестве опорного.

Опорный канал играет критически важную роль в функционировании всей системы изохронной передачи данных. Он служит в качестве базиса для генерации управляющих сигналов, координирующих работу всех обслуживаемых каналов. Одной из ключевых функций опорного канала является задание допустимого диапазона рассогласования задержек.

Как правило, диапазон допустимого рассогласования задержек формируется при инициализации. В случае аварийного отключения опорного канала границы диапазона допустимого рассогласования фиксируются на последнем известном значении. Такая ситуация приводит к значительному снижению гарантоспособности системы изохронной передачи данных. Вследствие возможного временного дрейфа задержек в обслуживаемых каналах их значения могут выйти за установленные пределы, что приведет к ложному распознаванию ошибок рассогласования и некорректному

нивелированию (обнулению) исправно функционирующих каналов. Пример такой ситуации продемонстрирован на временной диаграмме (рисунок 3.3).

На временной диаграмме в качестве примера изображены три канала данных. Опорным выбран канал Ch 2. В зоне 1 отображен штатный режим работы: начала массивов операндов всех каналов находятся в пределах допустимого рассогласования задержек, поэтому на выходе присутствуют данные от каждого из них. В зоне 2 показан случай аварийного отключения опорного канала. Видно, что его выходные данные заменяются нулевой информацией. Однако диапазон допустимого рассогласования задержек, определенный относительно этого канала, остался и еще какое-то время будет выполнять свою функцию, пока не произойдет ситуация, представленная в зоне 3. Зона 3 иллюстрирует возможный сценарий, при котором со временем возникает дрейф задержек каналов Ch 1 и Ch 3. Поскольку диапазон допустимого рассогласования больше не обновляется, задержки данных этих каналов выйдут за его пределы, что приводит к обнулению их выходных значений.

Следует отметить, что опорный канал используется для привязки ко времени выходного потока данных. Временные метки, извлекаемые из массива данных опорного канала, являются временными метками для выходных операндов (значение временных меток опорного канала прямо переносится на выход). Временная диаграмма, демонстрирующая процесс привязки выходных данных ко времени, представлена на рисунке 3.4.

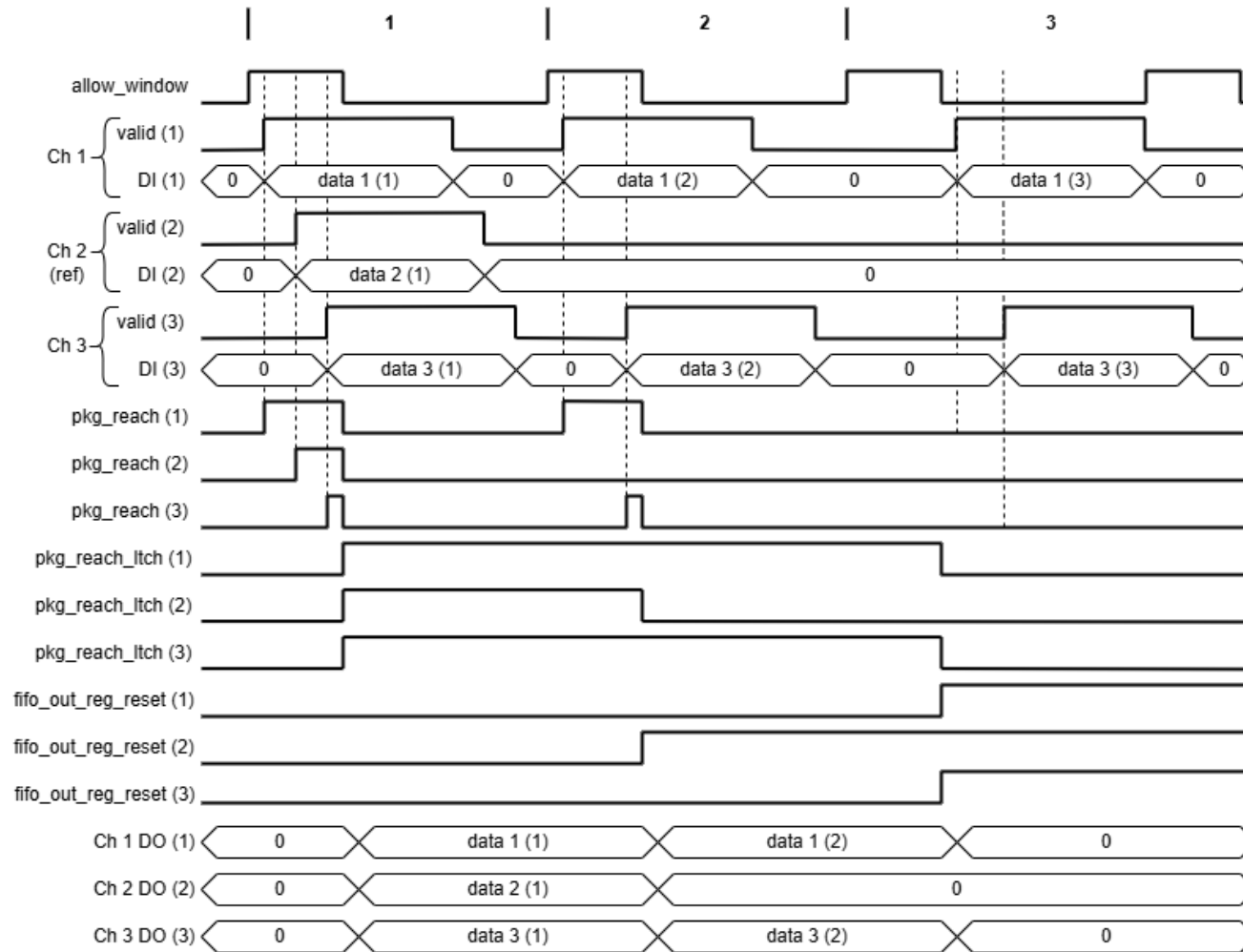


Рисунок 3.3. Временная диаграмма некорректной работы

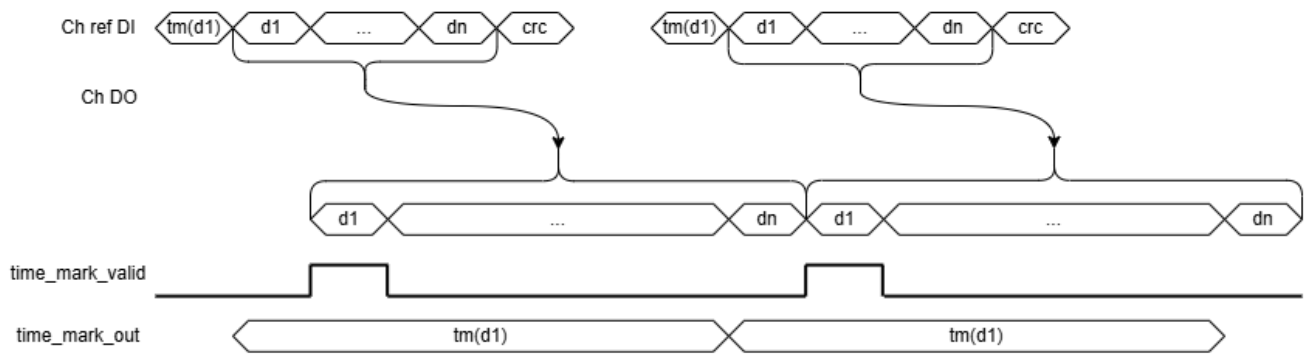


Рисунок 3.4. Временная диаграмма привязки ко времени выходного потока операндов

На представленной временной диаграмме изображен опорный канал Ch ref DI, принимающий массивы данных, разделенные служебными промежутками. При поступлении массива данных из его заголовка извлекается временная метка, соответствующая первому операнду в массиве, которая затем выставляется на выходную шину $time_mark_out$. В процессе формирования выходного потока операндов сигналы строба временной метки $time_mark_valid$ активируется синхронно с каждым первым передаваемым данным подмассива, обеспечивая временную привязку выходного потока операндов.

В случае аварийного отключения опорного канала процесс временной привязки выходных операндов критически нарушается из-за отсутствия источника временных меток. В результате выходной поток операндов теряет точную временную привязку, что нарушает изохронную передачу.

Для того чтобы избежать негативных последствий аварийного отключения опорного канала, необходимо реализовать процедуру переназначения опорного канала в случае отказа текущего. Этот процесс должен происходить в реальном времени без прерывания передачи данных, с сохранением изохронности потока операндов.

Предлагаемая процедура динамического переключения каналов является модернизацией процедуры поиска и назначения опорного канала, описанной во второй главе данной работы.

В качестве нового опорного канала следует назначать тот канал, чья задержка наиболее близка к задержке опорного канала. Для этого предлагается после инициализации системы изохронной передачи и назначения опорного канала продолжать непрерывно мониторить значения задержек всех обслуживаемых каналов и фиксировать их в массиве переменных C . Задержка канала определяется по его номеру, совпадающему с номером элемента массива. Такой подход позволит постоянно проверять задержки каналов относительно опорного и обеспечивает возможность оперативно переключиться на ближайший по задержке канал в случае аварийного отключения текущего опорного канала.

Для определения задержки между каналами предлагается задействовать существующий счетчик, формируемый относительно опорного канала после его назначения и определяющий допустимый интервал рассогласования. Определение задержки выполняется исключительно для каналов, у которых она находится в пределах заданного диапазона. Это исключит возможность выбора в качестве опорного канала того, чья задержка выходит за допустимые границы.

При аварийном отключении опорного канала инициируется процедура назначения нового опорного канала. Первым шагом в этой процедуре является поиск канала с задержкой, максимально приближенной к задержке текущего опорного канала. Для этого выполняется сравнение задержки каждого канала $C[n]$ с задержкой опорного $C[r]$, где r — номер опорного канала. Если задержка канала n превышает задержку опорного, то устанавливается специальный флаг. Затем вычисляется разница между задержками и сохраняется в массиве C^* по следующим правилам:

$$C^*[n] = |C[n] - C[r]|. \quad (3.2)$$

Выбираем в качестве нового опорного канала тот, у которого значение задержки $\min_{i=1,N} C_i^*$, где N — размер массива равный числу каналов, потенциально рассматриваемых в качестве опорных. Далее корректируется

допустимый диапазон рассогласования a относительно нового опорного канала r^* по следующим правилам:

$$a^* = a + k \cdot C^*[r^*], \quad (3.3)$$

$$k = \begin{cases} 1, & \text{если } C[n] \geq C[r] \\ -1, & \text{если } C[n] < C[r] \end{cases}$$

Заключительным этапом процедуры переключения опорного канала является активация соответствующего выхода мультиплексора временных меток, соответствующего новому опорному каналу.

Благодаря процедуре переназначения опорного канала, система изохронной передачи данных останется работоспособной, даже если произойдет аварийное отключение опорного канала.

В процессе передачи данных может возникнуть критическая ситуация, когда расхождение между массивами операндов, которые соответствуют одному временному интервалу, достигает величины, равной периоду их следования. Такая ситуация возникает, если обрабатываемый массив существенно отстает или, наоборот, опережает остальные массивы на один или даже несколько периодов их следования. Пример такой критической ситуации продемонстрирован на временной диаграмме (рисунок 3.5).

На временной диаграмме показаны три канала данных. Канал Ch 2 выбран как опорный. Относительно начала массивов операндов, передаваемых в опорном канале, формируется флаг `allow_window`, определяющий допустимый диапазон рассогласования задержек.

В зоне 1 отображен штатный режим работы: начала массивов операндов всех каналов находятся в пределах допустимого рассогласования задержек, поэтому на выходе присутствуют данные от каждого из них.

В зоне 2 показано отставание массива операндов в канале Ch 3 относительно опорного канала на значение, близкое к периоду следования массивов операндов (размер массива + длина служебного промежутка). При этом начало массива в канале Ch 3 попадает в диапазон допустимого

рассогласования `allow_window`, и данные этого массива будут отправлены на Выход.

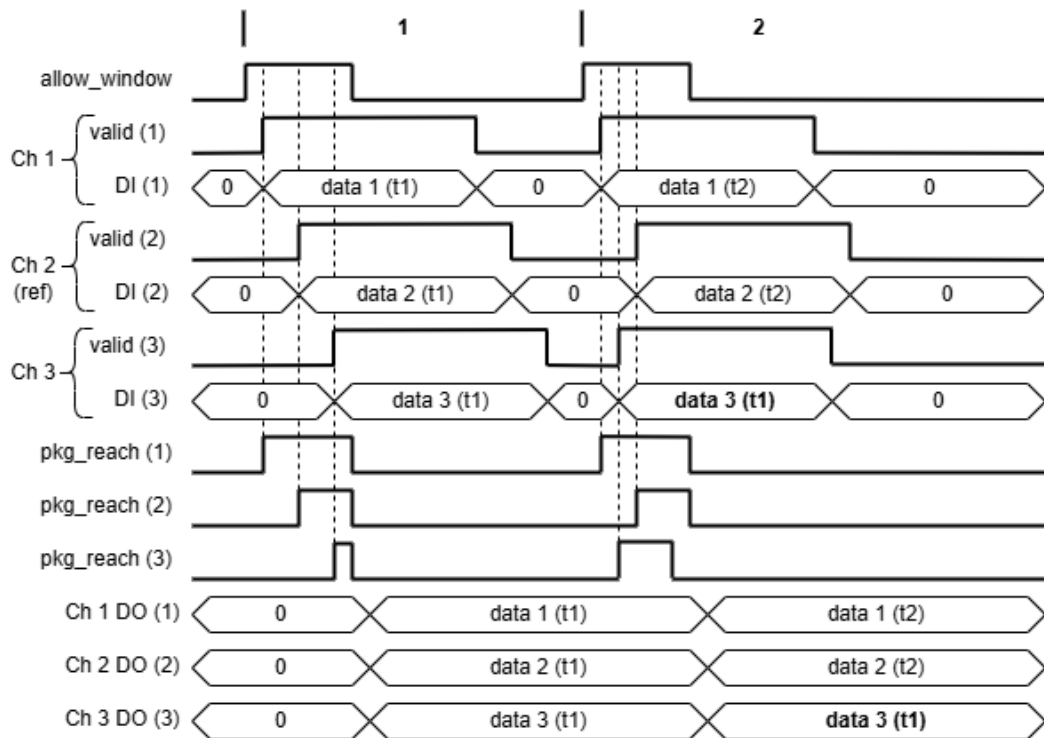


Рисунок 3.5. Временная диаграмма, совпадение с периодом следования

Это означает, что базовая процедура проверки задержек в каналах данных, описанная во второй главе данной диссертации, не может выявить подобные критические ошибки, поскольку формально условие проверки на допустимое рассогласование выполняется. В такой ситуации передача данных из отстающего(опережающего) канала на вход блоков обработки данных категорически недопустима, поскольку это не только нарушает принципы изохронной передачи данных, но и может нарушить формирование диаграммы направленности фазированной антенной решётки.

Важно отметить, что подобная ошибка обычно носит не единичный характер. Если в системе синхронизации возникли неисправности, приведшие к рассогласованию, критическое значение задержки сохранится до момента перезапуска или повторной синхронизации всего комплекса. Это означает, что в течение всего этого времени канал будет генерировать некорректные

данные, что может привести к серьёзным сбоям в работе всего комплекса ЦОС.

Для того чтобы избежать подобных ситуаций, следует разработать новую процедуру проверки допустимого расхождения между массивами.

Для различения массивов операндов, относящихся к разным моментам времени, модернизированный передающий узел должен использовать специальный счетчик, присваивающий каждому массиву уникальный порядковый номер перед отправкой. Этот номер записывается в служебный интервал перед массивом данных.

На принимающей стороне необходимо процедурно реализовать проверку: сравнить полученный номер массива с номером, переданным по опорному каналу. Если обнаруживается несоответствие, то у соответствующего канала формируется признак превышения допустимого рассогласования задержки, а на его выходе формируются нейтральные данные. Нейтральными данными для задач цифровой обработки сигналов, как правило, являются нулевые данные. Процедура обнаружения и парирования такого типа ошибок представлена на временной диаграмме (рисунок 3.6).

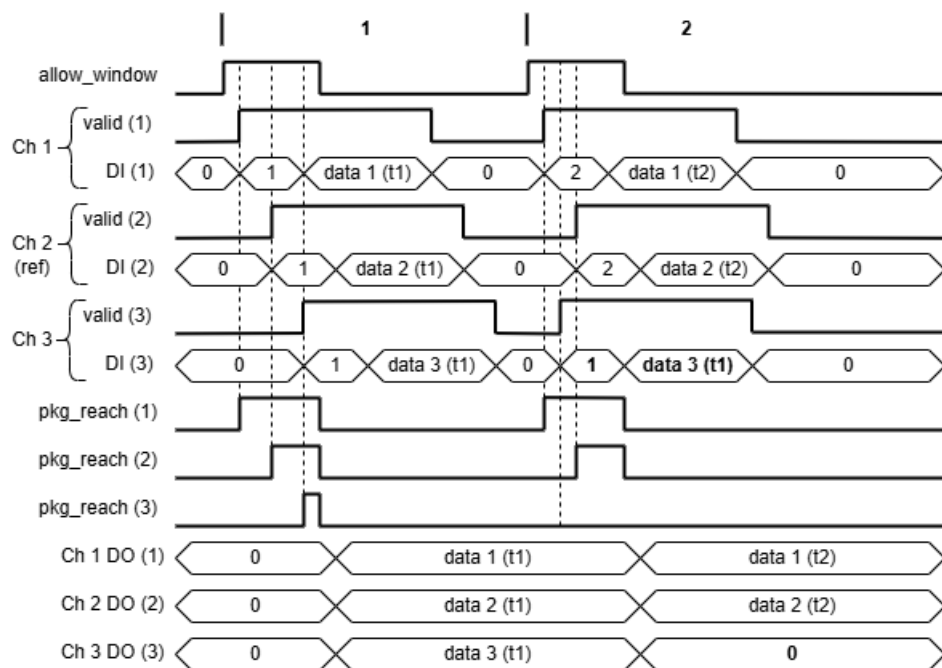


Рисунок 3.6. Временная диаграмма проверки номера массива

На временной диаграмме представлены три канала данных, канал Ch 2 используется в качестве опорного. Относительно начала передаваемых в этом канале массивов операндов формируется сигнал `allow_window`, который задаёт допустимый диапазон задержек.

Зона 1 демонстрирует штатный режим работы: начала массивов операндов во всех каналах находятся в пределах допустимого рассогласования задержек, поэтому на выходе присутствуют данные из каждого из них.

В зоне 2 наблюдается случай, когда задержка канала Ch 3 сдвигается на значение, близкое к периоду следования массивов операндов (равному сумме размера массива и длины служебного интервала). При этом начало массива операндов оказывается в пределах диапазона допустимого рассогласования задержек. Поскольку в данном случае порядковый номер массива операндов не совпадает с номером массива операндов в опорном канале, флаг `pkg_reach` (3), указывающий на попадание массива в допустимый диапазон рассогласования, не устанавливается. В результате на выходе этого канала будут формироваться нулевые данные.

Предложенная процедура позволяет выявлять ранее не обнаруживаемые ошибки рассогласования, которые могли бы привести к существенному снижению работоспособности системы изохронной передачи данных.

Еще одним типом наиболее вероятных ошибок для системы изохронной передачи данных является ошибки, связанные с нарушением целостности данных. Применение комбинированного метода позволяет эффективно выявлять такие ошибки посредством проверки контрольных сумм, однако их парирование (подмена сбойных массивов на массив с нейтральными данными) осуществляется лишь после достижения определённого порога количества ошибок.

Такой подход к обработке ошибок целостности данных обеспечил незначительное увеличение задержки обработки данных (латентности) по сравнению с базовым методом, при этом ошибки, возникающие до

достижения порогового значения, не оказывали значимого влияния на общую гарантоспособность при небольшом количестве каналов данных.

Однако при масштабировании системы с увеличением числа каналов данный подход не обеспечит требуемого уровня гарантоспособности, что обусловлено ростом общей вероятности возникновения ошибок целостности данных, способных негативно повлиять на итоговый результат обработки. В этой связи предлагается новая процедура проверки целостности данных, суть которой заключается в предварительной проверке контрольных сумм фрагментов массивов операндов перед их отправкой в блоки обработки. Основное отличие от предыдущего подхода заключается в том, что команда на чтение данных из памяти подаётся только после успешной проверки контрольных сумм всех массивов операндов, прошедших контроль задержек. Если контрольные суммы не совпадают, вместо отключения канала формируется признак, указывающий на необходимость обнуления выходных данных. Временная диаграмма модернизированной процедуры обнаружения и парирования ошибок целостности данных представлена на рисунке 3.7.

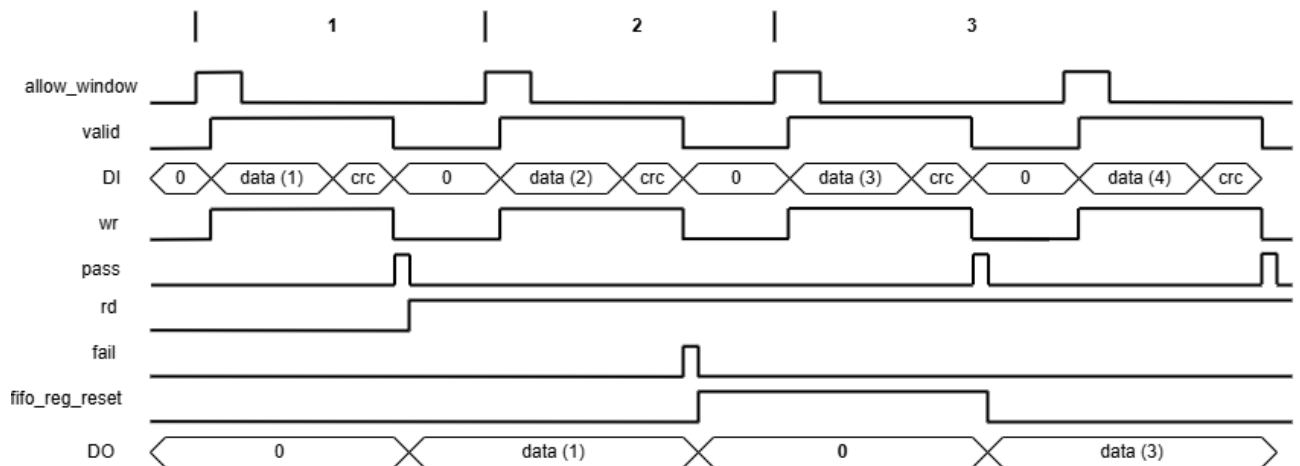


Рисунок 3.7. Временная диаграмма модернизированного контроля целостности

В зоне 1 осуществляется нормальная работа: массив операндов 1 не содержит ошибок целостности данных. После прохождения проверки на рассогласование задержек, начинается его запись в память. Параллельно с

записью запускается процесс подсчета контрольной суммы. По окончании массива операндов из служебного промежутка считывается контрольная сумма и сравнивается с посчитанным значением. Массив 1 успешно проходит проверку целостности, после чего поступает команда на чтение, и его данные выводятся на выход.

В зоне 2 показана обработка массива операндов 2, содержащего поврежденные при передаче данные. При выявлении несовпадения контрольных сумм этого массива операндов формируется флаг ошибки целостности данных, что приводит к обнулению выходных данных, соответствующих этому массиву операндов.

Предложенный алгоритм обработки ошибок целостности данных обеспечит их обнаружение до вывода массивов операндов, что увеличит процент успешного исправления ошибок и в целом повысит гарантоспособность системы.

3.3 Реализация модернизированного метода изохронной передачи данных

Модернизированный метод изохронной передачи данных был реализован для ПЛИС Xilinx Virtex-7 XC7VX485T. На вход ПЛИС поступает 24 канала разрядностью 33 бита (32 бита данных и 1 бит строб) на частоте $F_1=255$ МГц. На выходе формируется непрерывный поток операндов на частоте $F_2 = 240$ МГц. Данные поступают массивами операндов, разделенными служебными промежутками. В служебный промежуток перед началом массива операндов вставлено два слова, соответствующих метке времени первого данного массива операндов, и одно слово, соответствующее порядковому номеру массива. В конце массива размещено одно слово контрольной суммы, посчитанной для всех операндов массива и служебных слов.

Определены размер массива операндов и длина служебного промежутка. Следует отметить, что для корректного сравнения реализаций комбинированного и модернизированного методов, необходимо обеспечить одинаковый объем переданных данных за установленное время доверительной работы. Следовательно, необходимо сохранить тот же коэффициент заполнения данных $DC=0.941$, что и в реализации комбинированного метода. Минимальный размер служебного промежутка составляет пять слов, в который входят: два слова для метки времени, одно слово для порядкового номера массива, одно слово для контрольной суммы и одно пустое слово для разделения фрагментов операндов (массив операндов + временная метка и контрольная сумма).

Используя формулу (2.3), находим размер массива операндов. При длине служебного промежутка P_s – пять слов размер массива операндов составит 80 слов. Структура входных данных для одного канала представлена на рисунке 3.8.

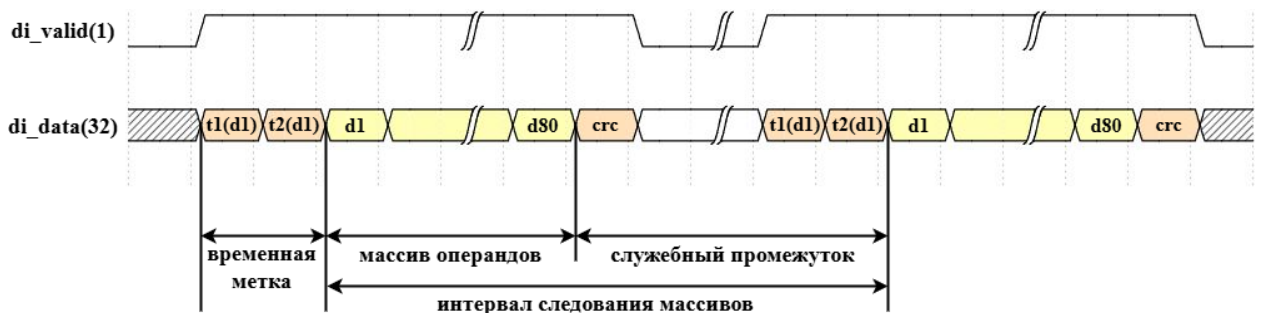


Рисунок 3.8. Структура входных данных

Программа, реализующая модернизированный метод изохронной передачи данных, обеспечивает выравнивание задержек между разными каналами, аналогично программе, описанной во второй главе диссертации. На выходе программы формируется 24 непрерывных потока операндов разрядностью 32 бита на частоте 240 МГц. Все потоки операндов сопровождаются одним стробом и меткой времени первого операнда массива. Структура выходных данных представлена на рисунке 3.9.

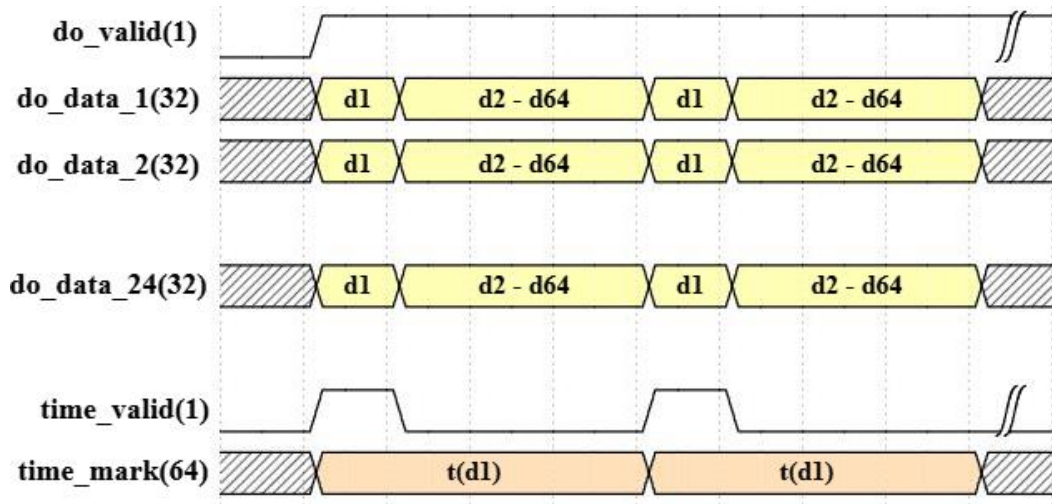


Рисунок 3.9. Структура выходных данных

Программа состоит из программных блоков, реализованных независимо для каждого канала ($V1 \div V24$), и общего блока (А), управляющего работой всех каналов (Рисунок 2.18).

Общий блок для всех каналов А включает в себя блок управления (БУПР) и блок синхронизации времени (БСВ).

Блок управления (БУПР) выполняет несколько функций, которые определяют работу всех остальных программных блоков, среди которых: поиск и задание опорного канала, установка допустимого диапазона рассогласования, запуск процесса одновременного считывания данных из памяти. В отличие от реализации комбинированного метода, описанного во второй главе диссертации, реализована процедура переключения опорных каналов при аварийном отключении текущего.

Одной из ключевых функций блока управления является назначение опорного канала и его переключение при возникновении отказа. Для этого используются базовая процедура первичной инициализации опорного канала и две новых процедуры мониторинга задержек и переназначения опорного канала.

На начальном этапе работы программы выполняется алгоритм первичной инициализации, аналогичный используемому в реализации комбинированного метода, описанного во второй главе диссертации.

После завершения процедуры первичной инициализации и назначения опорного канала, в отличие от реализации комбинированного метода, запускается алгоритм постоянного мониторинга задержек во всех обслуживаемых каналах. Это необходимо для того, чтобы блок управления всегда располагал актуальной информацией о текущих задержках во всех каналах и мог оперативно назначить новый опорный канал с минимальным отклонением по задержке в случае отказа текущего опорного канала.

Для отслеживания задержек всех обслуживаемых каналов используется счетчик диапазона допустимого рассогласования `allow_window`. Значение этого счетчика передается на все блоки приема данных. В каждом блоке при появлении строба данных соответствующего канала текущее значение `allow_window` фиксируется в массиве `sop_scan_ltch`. Если строб массива операндов приходит вне допустимого диапазона `allow_window` или если канал отключен (флаг `rx_on = 0`), устанавливается флаг `sop_miss[n] = 1`, в противном случае `sop_miss[n] = 0`. Этот флаг необходим для исключения отключенных каналов и каналов с превышенной задержкой из списка кандидатов на роль опорного.

При обнаружении отказа опорного канала (что определяется по сбросу флага готовности `rx_on[ref]`, получаемого от интерфейсного блока трансивера) запускается алгоритм переключения опорного канала. В первую очередь алгоритм определяет, задержка какого из каналов наиболее близка к задержке текущего опорного канала. Для этого в цикле от 1 до m (где m - общее число каналов) выполняется сравнение задержки опорного канала `sop_scan_ltch[ref]` с задержкой каждого канала `sop_scan_ltch[n]`.

На каждой итерации цикла выполняются следующие действия:

1. Проверяется флаг `sop_miss[i]` (i - номер текущей итерации). Если флаг установлен в 1, канал пропускается.

2. Если значение $sop_scan_ltch[ref] < sop_scan_ltch[i]$, то относительная задержка $rel_dly[i] = sop_scan_ltch[i] - sop_scan_ltch[ref]$, флаг $is_dly_higher[i]$ устанавливается в единицу. Иначе $rel_dly[i] = sop_scan_ltch[ref] - sop_scan_ltch[i]$, флаг $is_dly_higher[i]$ сбрасывается в ноль.

В результате выполнения цикла сформирован массив rel_dly с задержками каналов относительно опорного, а также массив is_dly_higher , содержащий признаки опережения задержки канала относительно опорного. Этот признаки в дальнейшем будут необходимы для определения направления корректировки счетчика допустимого рассогласования $allow_window_cntr$ при смене опорного канала.

Далее определяется канал с минимальной задержкой относительно опорного канала. Для этого применяется процедура линейного поиска:

1. Присвоить переменной min значение задержки первого канала $rel_dly[1]$.
2. В цикле от 2 до m (где m - количество каналов) выполнить:
 - 2.1 . Сравнить задержку текущего канала $rel_dly[j]$ (где j — номер итерации) с min .
 - 2.2 . Если $rel_dly[j] < min$, то:
 - обновить min , присвоив ему значение $rel_dly[j]$;
 - записать в $ch_selector$ текущий индекс j ;
 - присвоить $is_dly_higher_var$ значение $is_dly_higher[j]$.
 - 2.3 . Перейти к следующей итерации.

В результате выполнения процедуры линейного поиска формируются три переменные, которые содержат необходимую информацию для переключения на новый опорный канал. Переменная min хранит относительную задержку нового канала, $ch_selector$ определяет его номер, а $is_dly_higher_var$ указывает, является ли его задержка больше или меньше текущей опорной.

Завершающий этап процедуры переключения канала включает в себя настройку мультиплексора временной метки, согласно значению `ch_selector`, и коррекцию счетчика `allow_window_cntr`, определяющего диапазон допустимого рассогласования `allow_window` по следующим правилам:

Если `is_dly_higher_var = 1`, то `allow_window_cntr + min`;

Если `is_dly_higher_var = 0`, то `allow_window_cntr - min`.

Следующей важной функцией блока управления является инициализация одновременного чтения данных по всем каналам из памяти. Данная функция несколько изменена по сравнению с той, что присутствует в реализации комбинированного метода, из-за новой процедуры проверки целостности данных. Инициализация одновременного чтения обеспечивает формирование изохронных потоков данных на выходе программы. Команда на параллельное чтение подается однократно в начале работы программы и сбрасывается по сигналу общего сброса (`reset = 1`).

Процедура запуска одновременного чтения начинается при старте программы и ожидает, пока все массивы операндов, прошедшие проверку допустимого диапазона рассогласования (`allow_window`), будут также проверены на целостность данных через контрольные суммы. Для этого сравниваются два массива: `pkg_reach_ltch` - содержит флаги успешной проверки допустимого рассогласования по каждому каналу; `src_pass_ltch` - содержит флаги успешной проверки контрольных сумм данных по каждому каналу. Если массивы совпадают, формируется команда на параллельное чтение путем установки флага `read_run`. Одновременно запускается счетчик `fifo_rd_cntr`, отслеживающий количество считанных данных. Счетчик сбрасывается при достижении значения `pkg_depth - 1`, где `pkg_depth` - размер массива операндов. Флаг `read_run` и счетчик `fifo_rd_cntr` в дальнейшем будут использоваться блоками управления чтением для корректного формирования выходного изохронного потока операндов.

В состав блока обработки потока данных одного канала входят следующие блоки: интерфейс трансивера (ИТ), блок приема данных (БПД), блок расчета контрольных сумм (БРКС), буферная память FIFO, блок управления записью (БУЗ), блок управления чтением (БУЧ).

Структура программного блока, обеспечивающего работу одного канала, представлена на рисунке 3.10.

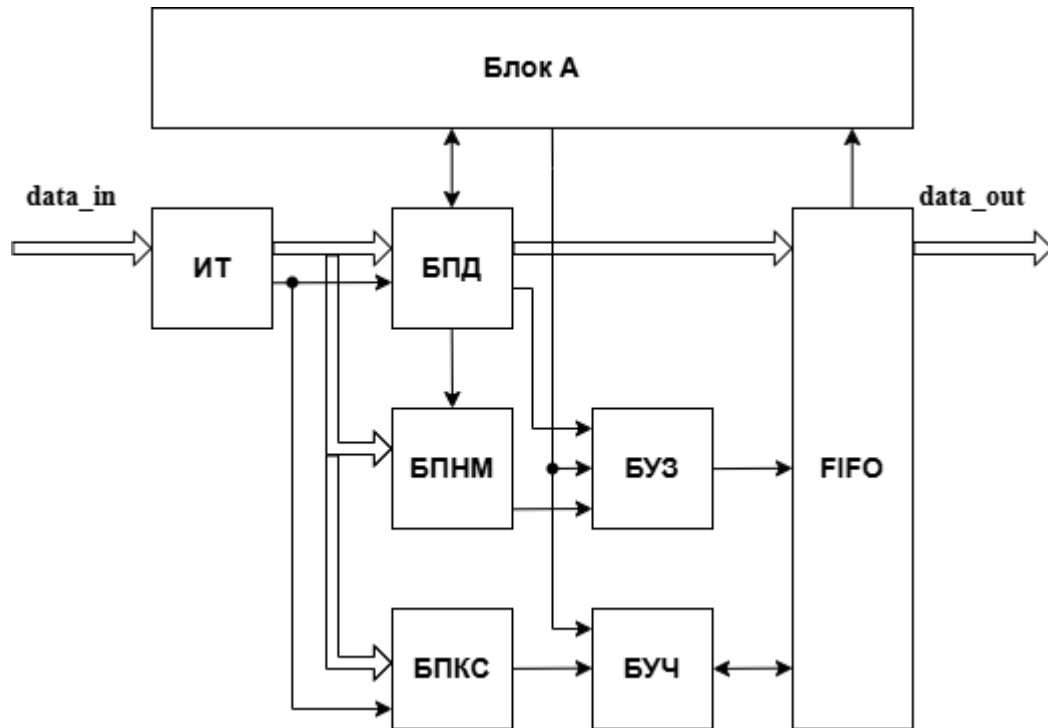


Рисунок 3.10. Структура программного блока одного канала

Интерфейс трансивера (ИТ) полностью повторяет блок, описанный в реализации комбинированного метода.

Блок приема данных (БПД) является модернизированной версией блока, описанного в реализации комбинированного метода. Он принимает данные `di_data` вместе со стробом `di_valid` от интерфейса трансивера. Он извлекает из заголовка временную метку `time_mark` и номер массива операндов `argau_num`, а также формирует управляющие флаги для работы остальных блоков системы. К этим флагам относятся:

1. `ch_data_sop` – сигнал начала массива данных;
2. `ch_data_eop` – сигнал конца массива данных;

3. `pkg_reach`, `pkg_reach_ltch` – флаг, указывающий, что задержка канала находится в допустимом диапазоне рассогласования.

Алгоритм работы блока приема данных включает следующие шаги:

1. При поступлении строба `di_valid` флаг `ch_data_sop` устанавливается на 1 такт.
2. Запускается счетчик принятых данных `ch_di_cntr`.
3. Считывается первое слово временной метки `time_mark`.
4. Считывается второе слово временной метки `time_mark`.
5. Считывается номер массива `array_num`.
6. Если значение счетчика `ch_di_cntr` достигает `pkg_depth - 1` (где `pkg_depth` – длина массива операндов), флаг `ch_data_eop` активируется на 1 такт.
7. При достижении счетчиком `ch_di_cntr` значения `pkg_depth` он сбрасывается.

Этот алгоритм обеспечивает формирование флагом начала и конца массива в требуемые моменты времени, что необходимо для синхронизации работы других модулей.

Блок проверки номера массива (БПНМ) является новым блоком, выполняющим дополнительный контроль рассогласования задержек между массивами операндов в разных каналах. Его задача - обнаруживать аварийную ситуацию, когда задержка достигает значения, равного периоду следования массивов. Это происходит, если обрабатываемый массив значительно отстаёт или, наоборот, опережает другие массивы на один или несколько периодов.

На вход блока подаётся переменная `array_num` - номер массива, полученный от блока приёма данных. Этот номер сравнивается с эталонным значением `array_num_ref`, поступающим от опорного канала. Если значения совпадают, формируется флаг `array_num_pass`, подтверждающий успешную проверку на рассогласование. В дальнейшем этот флаг используется блоком управления записью: если проверка не пройдена, соответствующий массив операндов не будет записан в память.

Блок проверки контрольных сумм (БПКС) является модернизированной версией блока расчета контрольных сумм, описанного во второй главе диссертации. Блок подсчитывает контрольную сумму для полученных данных и формирует флаги обнуления выходных данных. Он принимает данные `di_data` вместе со стробом `di_valid` от интерфейсного блока трансивера. При активации строба `di_valid` запускается счетчик принятых данных `data_counter`, и начинается расчет контрольной суммы.

Для расчёта контрольной суммы используется алгоритм CRC-16 Modbus [24] с полиномом `0x8005` и начальным значением `0xFFFF`, аналогично, как и в описанном ранее блоке расчета контрольных сумм.

Когда счетчик `data_counter` достигает значения, заданного параметром `pkg_depth`, расчет контрольной суммы завершается, и происходит сравнение принятой и вычисленной сумм. Если они не совпадают, устанавливается флаг ошибки `crc_fail`, который остается активным до тех пор, пока не будет успешно обработан массив данных без ошибок целостности. Флаг `crc_fail` будет использован в блоке управления чтением, чтобы предотвратить передачу некорректных данных на выход программы.

Буферная память FIFO блочная двухпортовая память (Dual-port Block RAM), сконфигурированная в режим работы FIFO (first in, first out – первым вошел, первым вышел).

Модернизированный блок управления записью (БУЗ) осуществляет управление записью данных в память FIFO. Блок работает непрерывно, до поступления сигнала общего сброса (`reset = 1`). Получает флаги начала массива данных `ch_data_sop` и конца массива `ch_data_eop` от блока приема данных. Флаг успешной проверки номера массива `array_num_pass` от блока проверки номера массива, а также флаг разрешения записи `allow_window_flag` от блока управления.

Реализован следующий алгоритм:

1. Ожидание флага начала массива данных `ch_data_sop`.

2. Если флаг начала массива данных `ch_data_sop` поступил, пока активен флаг разрешения записи `allow_window_flag`, то переходим к пункту 3, иначе возврат к пункту 1.
3. Проверяем наличие флага `array_num_pass`, если он присутствует, то формируется флаг записи в FIFO (`wr_en = 1`) и переходим к пункту 4, иначе `wr_en = 0` и возврат к пункту 1.
4. Ожидание флага признака конца массива `ch_data_eop`.
5. Если поступил флаг `ch_data_eop`, то снимаем флаг записи в FIFO (`wr_en = 0`) и переходим к пункту 1.

Процедура сохраняет в память только те массивы операндов, у которых начало попадает в допустимый диапазон рассогласования, и номер массива совпадает с номером массива операндов опорного канала. Данные, не соответствующие этим условиям, в FIFO не записываются.

Модернизированный блок управления чтением (БУЧ) предназначен для управления процессом чтения данных из памяти FIFO. Он принимает от блока управления флаг запуска процесса одновременного чтения `read_run` и значение счетчика прочитанных данных `fifo_rd_cntr`. От буферной памяти FIFO блок получает флаг, сигнализирующий о пустом состоянии FIFO (`fifo_empty`). Блок приема данных передает флаг `pkg_reach_ltch`, указывающий на попадание начала массива данных в допустимый интервал рассогласования. Кроме того, от блока проверки контрольных сумм поступает флаг `src_fail`, свидетельствующий об ошибке целостности данных.

На основе полученных сигналов блок управления чтением формирует флаг разрешения чтения FIFO `rd_en` и сигнал сброса выходного регистра `out_reg_reset`. Блок начинает работу при поступлении сигнала `read_run` и продолжает функционировать до тех пор, пока не будет подан сигнал общего сброса `reset`.

Алгоритм управления чтением выполняется следующим образом:

1. Ожидание команды `read_run` от блока управления.

2. Если начало массива данных попадает в допустимый интервал рассогласования ($pkg_reach_ltch=1$) и нет ошибки целостности данных ($crc_fail = 0$), то устанавливаем флаг разрешения чтения $rd_en=1$ и снимаем сигнал сброса выходного регистра $out_reg_reset=0$, иначе переходим к пункту 5.
3. Проверяем признак опустошения памяти $fifo_empty$. Если $fifo_empty=1$ (память опустошена), то переходим к пункту 5.
4. Если счётчик прочитанных данных $fifo_rd_cntr$ достигает значения $pkg_depth-1$, возврат к пункту 2.
5. Установка флага запрета чтения $rd_en=0$ и сброс выходного регистра $out_reg_reset=1$, возврат к пункту 3.

Реализация этого алгоритма в каждом канале гарантирует одновременное начало чтения данных из всех FIFO при условии, что массивы операндов прошли проверку в блоке приёма на соответствие допустимому интервалу рассогласования и проверку целостности в блоке контроля контрольных сумм. В случае отключения канала данные из его FIFO будут считываться до полного опустошения. После восстановления канала чтение возобновится строго в заданный момент, соответствующий интервалу следования массивов (размер массива операндов плюс длина служебного промежутка), чтобы не нарушить выходной изохронный поток данных. Для каналов, данные которых не были считаны, будут сгенерированы нулевые значения, что минимизирует влияние сбоя на результаты обработки в блоках цифровой обработки сигналов.

Разработана прикладная программа на языке высокого уровня COLAMO [100]. Занимаемый программой ресурс на ПЛИС XC7VX485T представлен в таблице 3.2.

Согласно данным таблицы 3.2, программа использует лишь небольшую часть ресурсов ПЛИС, что позволяет задействовать значительную часть кристалла для реализации блоков цифровой обработки сигналов.

Таблица 3.2. Занимаемый программой ресурс на ПЛИС XC7VX485T

| Ресурс | Всего | Занимаемый ресурс | % |
|--------|--------|-------------------|--------|
| LUT | 303600 | 18823 | 6,2 % |
| FF | 607200 | 13965 | 2,3 % |
| BRAM | 1030 | 24 | 2,33 % |
| GTX | 56 | 24 | 43 % |

Согласно данным таблицы 3.2, программа использует лишь небольшую часть ресурсов ПЛИС, что позволяет задействовать значительную часть кристалла для реализации блоков цифровой обработки сигналов.

Разработанная программа, реализующая модернизированный метод изохронной передачи данных, не только сохраняет все основные возможности комбинированного метода, но и добавляет усовершенствованные алгоритмы выявления ошибок, а также обеспечивает обработку новых типов ошибок, которые ранее не поддерживались.

3.4. Анализ гарантоспособности реализации модернизированного метода

Проведена оценка гарантоспособности реализации модернизированного метода. Для корректного сравнения с реализациями базового и комбинированного методов тестирование проводилось в идентичных условиях: при равном объёме передаваемых данных, одинаковом времени работы и на одном и том же оборудовании.

Для оценки гарантоспособности были определены новые коэффициенты парирования определенных типов ошибок. Результаты представлены в таблице 3.3.

Представленные коэффициенты были получены методом экспертных оценок по результатам экспериментов.

Таблица 3.3. Коэффициенты парирования для реализации модернизированного метода

| Тип ошибки | Коэффициент парирования | |
|---|-------------------------|---------------|
| | Комб. метод | Модерн. метод |
| Нарушение целостности данных | 0,55 | 0,98 |
| Нарушения целостности формата данных | 0,4 | 0,95 |
| Аварийное отключение канала | 0,77 | 0,88 |
| Ошибка оператора | 0,6 | 0,86 |
| Превышение допустимого значения рассогласования задержки канала | 0,85 | 0,85 |
| Совпадение задержки с периодом следования массивов данных | 0 | 0,95 |

Благодаря модернизации процедуры проверки целостности данных с предварительным контролем массивов на целостность информации, удалось значительно повысить коэффициент парирования ошибок, связанных с нарушением целостности и формата данных. Внедрение новой процедуры переключения опорного канала в случае его аварийного отключения позволила повысить коэффициенты парирования ошибок, связанных с аварийным отключением канала. Ранее не учитываемая в реализации комбинированного метода ошибка, связанная с совпадением задержки с периодом следования массивов данных, благодаря новой процедуре проверки номера массива, с высокой вероятностью успешно обнаруживается и парируется.

Применив методику оценки гарантоспособности, использованной для комбинированного метода из главы 2 диссертации, была выполнена оценка гарантоспособности реализации модернизированного метода, которая составила $G_3 = 0.99488$ в тех же условиях (10900 каналов и 6 часов работы).

Определим, насколько улучшилась гарантоспособность программного комплекса, реализующего модернизированный метод, по сравнению с реализацией комбинированного метода. По формуле (2.29) вычислим коэффициент эффективности улучшения (гарантоспособности)

$$IEF = \frac{1 - 0,987}{1 - 0,99488} = 2,54.$$

Можно сделать вывод, что применение модернизированного метода изохронной передачи данных вместе с новыми разработанными и модернизированными процедурами обнаружения и парирования аварийных ситуаций позволяет повысить гарантоспособность программного комплекса обеспечения изохронной передачи данных в 2,54 раза по сравнению с комбинированным методом.

На рисунке 3.13 показана зависимость гарантоспособности от количества каналов.

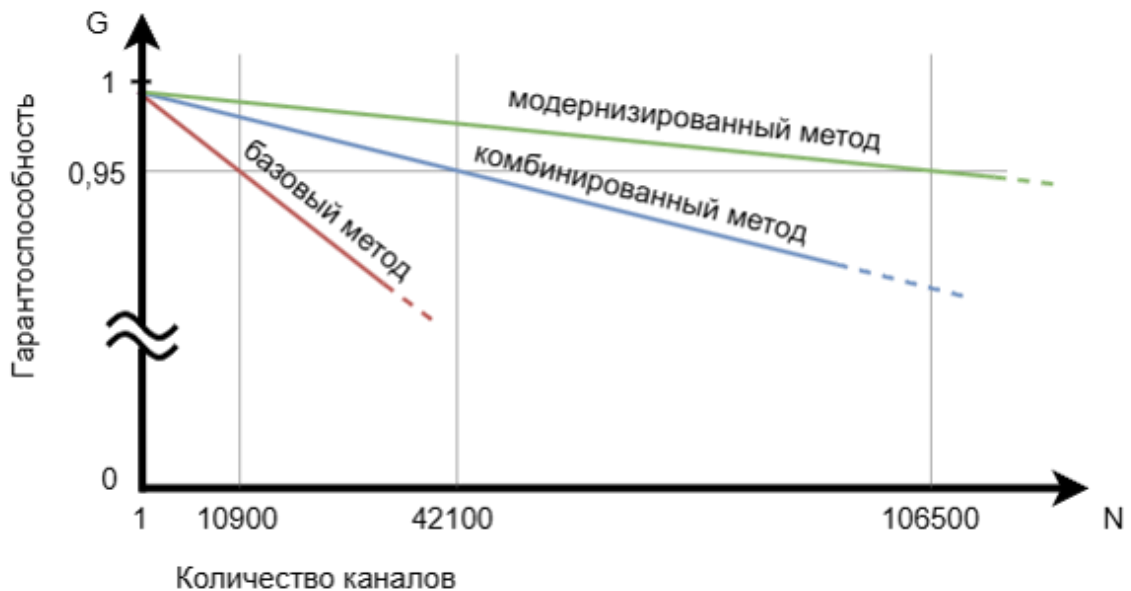


Рисунок 3.11. Зависимость гарантоспособности от количества каналов

При увеличении количества каналов на гарантоспособность начинают значительно влиять все типы ошибок, включая те ошибки, вероятность которых была очень низкой и которые не контролировались в реализации комбинированного метода. Благодаря разработанным и модернизированным процедурам удалось значительно повысить качество парирования различных типов ошибок, что привело к увеличению гарантоспособности программного комплекса, реализующего модернизированный метод.

Из графика видно, что заданный уровень гарантоспособности - 0,95, программа, реализующая модернизированный метод, обеспечивает уровень при числе каналов 106500, что удовлетворяет выдвинутым требованиям для модернизированного метода в 90000 каналов.

3.5 Выводы

1. Перспективные задачи для РЛС с ФАР требуют повышения разрешающей способности ФАР, для чего, как правило, увеличивают количество антенных элементов.
2. Выполнен анализ рационального значения количества антенных элементов, при котором достигается значительный прирост разрешения ФАР, который составляет 90000 антенных элементов.
3. Для сохранения заданной гарантоспособности при увеличении числа каналов до 90000 необходима модернизация комбинированного метода, так как в его текущем виде он не соответствует требуемым показателям при таком количестве каналов. Ключевым направлением модернизации комбинированного метода является повышение эффективности процедур парирования и разработка процедур парирования новых типов ошибок, ранее не учитываемых.
4. Разработан алгоритм переключения опорных каналов при отказе текущего опорного канала, позволяющий системе изохронной передачи данных оставаться работоспособной, даже если произойдет аварийное отключение опорного канала.
5. Разработан алгоритм предварительной проверки контрольных сумм массивов операндов перед их отправкой в блоки обработки. Реализация алгоритма увеличивает процент успешного парирования ошибок целостности данных.

6. Разработан алгоритм обнаружения и парирования ситуации, в которой расхождение между массивами операндов, которые соответствуют одному временному интервалу и достигает величины, равной периоду их следования (размер массива операндов + служебный промежуток). Реализация алгоритма позволяет выявлять ранее не обнаруживаемые ошибки рассогласования, которые могли бы привести к существенному снижению гарантоспособности системы изохронной передачи данных.
7. Разработана и создана программа для РВС на базе ПЛИС XC7VX485T, реализующая модернизированный метод изохронной передачи данных. Занимаемый программой ресурс на ПЛИС не превысил 7% от общего ресурса.
8. Выполнен анализ гарантоспособности разработанной программы, реализующей модернизированный метод изохронной передачи данных. Применение модернизированного метода позволяет повысить гарантоспособность программного комплекса обеспечения изохронной передачи данных в 2,54 раза по сравнению с комбинированным методом, что обеспечивает заданный уровень гарантоспособности (0,95) в течение шестичасового интервала работы комплекса ЦОС с ФАР при использовании не более 106500 антенных элементов (информационных каналов).

ЗАКЛЮЧЕНИЕ

Основные полученные результаты диссертационной работы заключаются в разработке методов и программных инструментальных средств для реконфигурируемых вычислительных систем, повышающих гарантоспособность программного комплекса ЦОС при большом числе каналов и высоких частотах дискретизации. На основании разработанных методов созданы алгоритмы, позволяющие повысить гарантоспособность прикладных программ для комплексов ЦОС с изохронной передачей данных.

При проведении исследований и разработок по теме диссертации получены следующие теоретические и прикладные результаты, обладающие научной новизной:

1. Комбинированный метод изохронной передачи данных, сочетающий использование единого машинного времени и опорной тактовой частоты, отличающийся от известных введением служебных промежутков при передаче массивов операндов.

2. Алгоритм выбора опорного канала и формирования диапазона допустимого рассогласования задержек данных, отличающийся процедурой назначения опорного канала, у которого задержка данных наиболее близка к математическому ожиданию задержек данных всех каналов.

3. Создана инструментальная программа, реализующая комбинированный метод для ПЛИС Xilinx Virtex-7 XC7VX485T. (Свидетельство о государственной регистрации программы для ЭВМ №2025660549, РФ). Оценка гарантоспособности программы изохронной передачи данных показала повышение гарантоспособности в 3,85 раз по сравнению с реализацией базового метода, что обеспечивает заданный уровень гарантоспособности 0,95 на доверительном интервале шесть часов при 42100 антенных элементах (информационных каналах).

4. Модернизированный метод изохронной передачи данных, отличающийся процедурами переключения опорных каналов,

предварительной проверкой контрольных сумм и нумерацией массивов операндов.

5. Алгоритм переключения опорного канала без прерывания передачи данных, отличающийся процедурой автоматического назначения нового опорного канала с задержкой данных в канале, максимально приближенной к задержке данных текущего опорного канала в случае его аварийного отключения.

6. Создана инструментальная программа, реализующая модернизированный метод для ПЛИС Xilinx Virtex-7 XC7VX485T. Оценка гарантоспособности программы изохронной передачи данных показала повышение гарантоспособности в 2,54 раз по сравнению с реализацией комбинированного метода, что обеспечивает заданный уровень гарантоспособности 0,95 на доверительном интервале шесть часов при 106500 антенных элементах (информационных каналах).

Основные научные результаты диссертации опубликованы работах [123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133].

Основные положения докладывались и получили одобрение на восьми всероссийских и международных конференциях и семинарах. По теме исследования получено свидетельство об официальной регистрации программ для ЭВМ.

Предложенные в диссертации научные результаты строго аргументированы и оценены по сравнению с известными работами в рассматриваемой области.

Результаты диссертационного исследования внедрены при выполнении ряда работ в Научно-исследовательском центре супер-ЭВМ и нейрокомпьютеров (ООО «НИЦ СЭ и НК»), а именно: СЧ ОКР «Разработка цифровой диаграммообразующей системы фазированной антенной решетки приемного аппаратурно-антенного комплекса», акт от 29 мая 2025 г., утвержден техническим директором ООО «НИЦ СЭ и НК».

Результаты диссертационного исследования внедрены в учебном процессе кафедры Интеллектуальных и многопроцессорных систем (ИМС) Института компьютерных технологий и информационной безопасности (ИКТИБ) Южного федерального университета (ЮФУ), акт от 30 мая 2025 г., утвержден директором ИКТИБ ЮФУ. Материалы диссертации используются в лекционном курсе по дисциплине «Методы цифровой обработки сигналов, изображений и ситуаций» (тема № 5 «Спектральный анализ. Дискретное преобразование Фурье») и «ПЛИС-технологии и методы создания эффективных прикладных программ для РВС» (тема № 20 «Параллельные системы» модуля 1 «Основы работы с ПЛИС») для подготовки магистров направления подготовки 01.04.02 Прикладная математика и информатика (образовательная программа «Прикладная математика для высокопроизводительных вычислительных систем»).

Все научные результаты, полученные при решении научной задачи разработки методов и инструментальных программных средств обеспечения изохронной передачи данных, получены автором **лично**.

Таким образом, в диссертации решена актуальная **научная задача** – разработка методов создания инструментальных программных средств для реконфигурируемых вычислительных систем, повышающих гарантоспособность программного комплекса ЦОС при большом числе каналов и высоких частотах дискретизации, – являющаяся актуальной и имеющей существенное научно-практическое значение. Решенная научная задача обладает новизной и имеет заметное научно-практическое значение для программирования РВС.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Воскресенский, Д.И. Устройства СВЧ и антенны / Д.И. Воскресенский, В.О. Гостюхин, В.М. Максимов, Л.И. Пономарев ; под ред. Д.И. Воскресенского. — 2-е изд., доп. и перераб. — Москва : Радиотехника, 2006. — 376 с.
2. Гомин, Л. Фазированная антенная решетка с цифровой схемой формирования многолучевой диаграммы направленности / Л. Гомин, П.Н. Захаров, А.Ф. Королев // Известия Российской академии наук. Серия физическая. — 2023. — Т. 87, № 10. — С. 1473-1477. — DOI 10.31857/S0367676523702575.
3. Многолучевые антенны для систем радиолокации и связи / А.В. Шишлов, Б.А. Левитан, С.А. Топчиев [и др.] // Журнал радиоэлектроники. — 2018. — № 7. — С. 1. — DOI: 10.30898/1684-1719.2018.7.6.
4. Бакулев, П.А. Радиолокационные системы. Учебник для вузов / П.А. Бакулев. — 3-е изд., перераб. и доп. — Москва : Радиотехника, 2015. — 440 с. : ил.
5. Сколник, М. Введение в технику радиолокационных систем / М. Сколник ; пер. с англ. П.К. Горохова, Ф.С. Соловейчика ; под ред. К.Н. Трофимова. — Москва : Мир, 1965. — 747 с. : ил. — 27 см.
6. Тепликова, В.И. Анализ диаграммы направленности плоской многоэлементной активной фазированной антенной решетки / В.И. Тепликова, А.А. Сенцов, В.А. Ненашев, В.Б. Поляков // Труды МАИ. — 2022. — № 125. — URL: <https://cyberleninka.ru/article/n/analiz-diagrammy-napravlennosti-ploskoj-mnogoelementnoy-aktivnoy-fazirovannoy-antennoy-reshetki> (дата обращения: 28.11.2024).
7. Григорьев, Л.Н. Цифровое формирование диаграммы направленности в фазированных антенных решетках / Л.Н. Григорьев. — Москва : Радиотехника, 2010. — 144 с. : ил. — ISBN 978-5-88070-243-5.

8. Теоретические и физические основы радиолокации и специального мониторинга : учебник / А.Н. Фомин, В.Н. Тяпкин, Д.Д. Дмитриев [и др.] ; под общ. ред. И.Н. Ищука. — Красноярск : Сиб. федер. ун-т, 2016. — 292 с. — ISBN 978-5-7638-3389-8.

9. Дзюба, А.П. Перспективы развития фазированных антенных решеток / А.П. Дзюба // Вестник ДГТУ. Технические науки. — 2013. — № 3. — URL: <https://cyberleninka.ru/article/n/perspektivy-razvitiya-fazirovannyh-antennyh-reshetok> (дата обращения: 05.09.2024).

10. Вениаминов, С.С. Космический мусор угрожает планете / С.С. Вениаминов // Воздушно-космическая сфера. — 2016. — № 1 (86). — С. 45-53. — URL: <https://cyberleninka.ru/article/n/kosmicheskiy-musor-ugrozhaet-planete> (дата обращения: 14.07.2024).

11. Рыбалкина, О.И. Обоснование применения цифрового диаграммообразования в радиолокационных станциях наземного базирования / О.И. Рыбалкина // Известия Тульского государственного университета. Технические науки. — 2016. — № 12-2. — С. 177-187.

12. Родзин, С.И. Отказоустойчивые вычислительные системы / С.И. Родзин. — Таганрог : Таганрогский государственный радиотехнический университет, 2001. — 272 с. — ISBN 5-8327-0095-3.

13. Лапри, Ж.-К. Гарантоспособные вычисления и отказоустойчивость: концепции и терминология = Dependable Computing and Fault Tolerance: Concepts and Terminology / Ж.-К. Лапри // Proceedings of the 15th International Symposium on Fault-Tolerant Computing (FTCS-15). — Ann Arbor, MI : IEEE Computer Society Press, 1985. — P. 2-11.

14. Авиженис, А. Проектирование отказоустойчивых компьютеров = Design of Fault-Tolerant Computers / А. Авиженис // Труды конференции AFIPS. Осенняя объединенная компьютерная конференция. - Вашингтон : Thompson, 1967. - Т. 31. - С. 733-743.

15. Рэнделл, Б. Структура системы для обеспечения отказоустойчивости программного обеспечения / Б. Рэнделл // IEEE Transactions on Software Engineering. - 1975. - Vol. SE-1, no. 1. - P. 220-232.
16. Lee, P. A. Fault Tolerance: Principles and Practice / P. A. Lee, T. Anderson. - 2nd ed., rev. - Vienna : Springer, 2012. - 320 p. : ill. - (Dependable Computing and Fault-Tolerant Systems ; vol. 3).
17. Discover DO-178C Guidance [электронный ресурс]. – Режим доступа: <https://do178.org/index.html> (дата обращения: 11.04.2025)
18. Functional safety and IEC 61508 [электронный ресурс]. – Режим доступа: https://www.ida.liu.se/~simna73/teaching/SCRTS/IEC61508_Guide.pdf (дата обращения: 11.04.2025)
19. Харитонов, В.А. Показатели отказоустойчивости вычислительных систем для случая нестационарного потока отказов / В.А. Харитонов // Известия вузов России. Приборостроение. - 1993. - № 2. - С. 7-10.
20. Царев, Р.Ю. Многоатрибутивное формирование гарантоспособных систем управления и обработки информации / Р. Ю. Царев, Д.В. Капулин, Д.В. Машурова, Я.А. Тынченко, Д.Н. Ковтанюк // Сибирский аэрокосмический журнал. - 2012. - № 5 (45). - URL: <https://cyberleninka.ru/article/n/mnogoatributivnoe-formirovanie-garantosposobnyh-sistem-upravleniya-i-obrabotki-informatsii> (дата обращения: 15.05.2024).
21. Игнатов, В.А. Методы расчёта показателей гарантоспособности компьютерных сетей / В.А. Игнатов, Н.Н. Гузий, М.А. Соряя // Проблемы информатизации и управления. - 2011. - Т. 2, № 34. - С. 58-63. - DOI: 10.18372/2073-4751.2.9202.

22. Харченко, В.С. Парадигмы и принципы гарантоспособных вычислений: состояние и перспективы развития / В.С. Харченко // Радиоэлектронные и компьютерные системы. - 2009. - № 2. - С. 91-100.

23. Слока, В.К. Вопросы обработки радиолокационных сигналов / В.К. Слока // - Москва : Советское радио, 1970. - 255 с. : ил. - Библиогр.: с. 248-252.

24. Слока, В.К. Новые радиодигитальные платформы для воздушно-космических систем высокоскоростной передачи информации / В.К. Слока // Вестник Московского авиационного института. - 2005. - Т. 12, № 2.

25. Созинов, П.А. Основные подходы к применению компьютерного моделирования при создании вооружения, военной и специальной техники воздушно-космической обороны / П.А. Созинов, А.П. Коновальчик // Вестник воздушно-космической обороны. - 2022. - № 1 (33). - С. 101-110.

26. Топчиев, С.А. Разработка в ПАО "Радиофизика" РЛС с цифровыми АФАР / С.А. Топчиев, М.В. Никитин // Радиолокация и связь - перспективные технологии. - Москва : ПАО "Радиофизика", 2015. - С. 7-14.

27. Семерников, Е.А. Реализация быстрых алгоритмов цифровой обработки сигналов в реконфигурируемых вычислительных системах / Е.А. Семерников // Высокопроизводительные вычислительные системы. ВПВС-2008 : Часть 1. Материалы Пятой Международной научной молодежной школы. Часть 2. Материалы Международной молодежной научно-технической конференции., Таганрог, 01–05 сентября 2008 года. – Таганрог: Таганрогский технологический институт ЮФУ, 2008. – С. 283-308.

28. Дордопуло, А.И. Программные средства решения вычислительно трудоемких задач на реконфигурируемых вычислительных системах / А.И. Дордопуло, О.В. Катаев, И.И. Левин // Научный сервис в сети Интернет: решение больших задач : труды Всероссийской научной конференции

(Новороссийск, 22-28 сентября 2008 г.). - Москва : Изд-во Моск. ун-та, 2008. - С. 114-121.

29. Левин, И. И. Отказоустойчивое функционирование реконфигурируемых многопроцессорных вычислительных систем / И.И. Левин, Г.Л. Трунов // Известия высших учебных заведений. Электромеханика. – 2004. – № 4. – С. 63-67.

30. Каляев И.А. Высокопроизводительные реконфигурируемые вычислительные системы для цифровой обработки сигналов / И.А. Каляев, И.И. Левин, Е.А. Семерников // Труды Российского научно-технического общества радиотехники, электроники и связи имени А.С. Попова. Серия: Цифровая обработка сигналов и ее применение. - Москва, 2010. - Вып. XII-1. - С. 13-18. - ISBN 978-5-904602-07-9.

31. Дордопуло А.И. Высокопроизводительные многопроцессорные системы с реконфигурируемой архитектурой для цифровой обработки сигналов / А.И. Дордопуло, И.А. Каляев, И.И. Левин, Е.А. Семерников // Вестник Концерна ПВО "Алмаз-Антей". - 2011. - № 2 (6). - С. 88-104. - 17 с.

32. Таненбаум, Э.С. Компьютерные сети / Э.С. Таненбаум, Д. Уэзеролл. — 5-е изд. — Санкт-Петербург : Питер, 2012. — 960 с. — ISBN 978-5-459-00342-0.

33. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов / В.Г. Олифер, Н.А. Олифер. — Санкт-Петербург : Питер, 2001. — 672 с. : ил. — ISBN 978-5-4461-1426-9.

34. Линь, Л.Т., Дык, Б.М., Хуи, Н.Н., Чыонг, Н.Д., Хю, Н.Б., Хыонг, Л.Ч. Сетевая модель OSI // Научные исследования. — 2017. — № 1 (12). — URL: <https://cyberleninka.ru/article/n/setevaya-model-osi> (дата обращения: 19.03.2025).

35. Вишневский, В.М. Теоретические основы проектирования компьютерных сетей / В.М. Вишневский. — Москва : Техносфера, 2003. — 512 с.
36. Тучкин, А.В. Принципы функционирования протокола канального уровня для пакетной передачи разнородного трафика по низкоскоростным каналам / А. В. Тучкин // Т-Comm: Телекоммуникации и транспорт. — 2008. — Т. 2, № 3. — С. 31–33.
37. Кучерявый, А.Е. Пакетная сеть связи общего пользования / А.Е. Кучерявый, Л. З. Гильченко, А. Ю. Иванов. — Санкт-Петербург: Наука и техника, 2004. — 274 с.
38. Standard Group MAC Addresses: A Tutorial Guide [электронный ресурс]. — Режим доступа: <https://standards.ieee.org/wpcontent/uploads/import/documents/tutorials/macgrp.pdf> (дата обращения: 11.03.2025)
39. Клименко, С.В. Исследование реализаций алгоритмов контрольной суммы CRC32 / С.В. Клименко, В.В. Яковлев, Е.А.Благовещенская // Известия Петербургского университета путей сообщения. — 2018. — № 3. — С. 471–477
40. Logical Link Control (LLC) Protocol Data Unit [электронный ресурс]. — Режим доступа: <https://www.geeksforgeeks.org/logical-link-control-llc-protocol-data-unit/> (дата обращения: 15.04.2025)
41. Шиманов, С.Н. Научно-методический подход к моделированию процесса функционирования сети связи с коммутацией пакетов на сетевом уровне OSI / С.Н. Шиманов, Ф.Г. Кирдяшов // Информационные технологии в проектировании и производстве. — 2008. — № 1. — С. 75–77.

42. Internet Protocol, Version 6 (IPv6) Request for Comments: 2460 [электронный ресурс]. – Режим доступа: <https://tools.ietf.org/html/rfc2460> (дата обращения: 12.03.2025)
43. Hussain, I. Dynamic MTU: A smaller path MTU size technique to reduce packet drops in IPv6 / I. Hussain, Ja. Bashir // Journal of King Saud University. Computer and Information Sciences. – 2022. – Vol. 34, No. 9. – P. 7070-7088. – DOI 10.1016/j.jksuci.2021.06.011.
44. Воротницкий, Ю.И. Оценка параметров качества сервиса протоколов транспортного уровня в локальных сетях произвольной древовидной структуры. Ч. 2 / Ю.И. Воротницкий, Д.А. Стрикелев // Вестник БГУ. Серия 1, Физика. Математика. Информатика. – 2009. – № 2. – С. 34-38.
45. Лесная, Н.С. Анализ процессов управления передачи данных в корпоративных компьютерных сетях / Н.С. Лесная, Е.С. Иевлев // Восточно-Европейский журнал передовых технологий. – 2012. – Т. 4, № 9(58). – С. 33-36.
46. Зотов, С.В. Тестирование обмена сообщениями на сеансовом уровне / С. В. Зотов // Научно-исследовательские публикации. — 2013. — № 4-2 (4). — С. 99–102.
47. Кручинин, С.В. Реализация модели OSI/ISO телекоммуникационным модулем сопряжения для MESH-сетей / С.В.Кручинин // Научно-исследовательские публикации. – 2016. – № 5(37). – С. 27-32.
48. Кручинин, С.В. Интерпретация модели ЭМВОС применительно к стеку TCP/IP / С.В. Кручинин // Научно-исследовательские публикации. – 2016. – № 2(34). – С. 37-43.
49. Wakoli, L.W. Secure Sockets Layer/Transport Layer Security for E-Commerce / L.W. Wakoli // International Journal of Scientific Research and

Management. – 2024. – Vol. 12, No. 12. – P. 8047-8052. – DOI 10.18535/ijssrm/v12i12.em03.

50. Прокопенко, Л.Л. Модель OSI для организации компьютерных сетей / Л.Л. Прокопенко, Ю.Э. Ионан // Вестник образовательного консорциума Среднерусский университет. Информационные технологии. – 2021. – № 1(17). – С. 40-42. – DOI 10.52374/52100412_2021_17_1_40.

51. Subbotin, N. A. Network application layer protocol security / N.A.Subbotin, V. V. Bukhtoyarov, O. V. Maslova // Молодежь. Общество. Современная наука, техника и инновации. – 2015. – No. 14. – P. 76-77.

52. Попов, И.В. Обзор и сравнение протоколов HTTP/1.1 и HTTP/2 / И. В. Попов // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 04–06 декабря 2023 года. — Воронеж : Общество с ограниченной ответственностью «Вэлборн» : Издательство «Научно-исследовательские публикации», 2024. — С. 930–933.

53. Лоншаков, С.С. Особенности практического применения сетевых протоколов UDP и TCP / С.С. Лоншаков, Т.А. Серебрякова // КИП и автоматика: обслуживание и ремонт. – 2023. – № 7. – С. 4-6.

54. RFC 1323 [электронный ресурс]. – Режим доступа: <https://www.freesoft.org/CIE/RFC/1323/> (дата обращения: 16.04.2025)

55. Протоколы информационно-вычислительных сетей : справочник / Аничкин С. А., Белов С. А., Бернштейн А. В. [и др.] ; под ред. Мизина И. А., Кулешова А. П. — М. : Радио и связь, 1990. — 504 с.

56. RFC 768 [электронный ресурс]. – Режим доступа: <https://www.freesoft.org/CIE/RFC/768/> (дата обращения: 17.03.2025)

57. Felk, V.S. TCP and UDP protocols / V.S. Felk, E.V. Tyubaeva // Молодежь. Общество. Современная наука, техника и инновации. – 2022. – No. 21. – P. 33-34.
58. Борисов, В.В. Протоколы передачи данных и связи: TCP, UDP и QUIC / В.В. Борисов, Д.А. Иванова // Промышленность и сельское хозяйство. – 2024. – № 5(70). – С. 95-98.
59. Дайняк, И.В. Распределенная система управления реального времени на основе технологии Ethercat / И.В. Дайняк // Информационные системы и технологии: управление и безопасность. – 2014. – № 3. – С. 83-90.
60. Андреев, С.Г. Ethernet POWERLINK Industry 4.0 / С.Г. Андреев, А.В. Наседкин, П.М. Тихон // Всероссийская научная конференция по проблемам управления в технических системах. – 2015. – Т. 1. – С. 108-110.
61. Шульга, Р.Н. Стандарт PROFINET для железнодорожного транспорта / Р. Н. Шульга // Электрооборудование: эксплуатация и ремонт. – 2022. – № 12. – С. 54-62.
62. Бессонов, Я. Интерфейс Sercos для управления двигателями / Я. Бессонов // Компоненты и технологии. – 2006. – № 5(58). – С. 142-144.
63. LXI - Общие сведения о стандарте и технологии [электронный ресурс]. – Режим доступа: <http://lxi.ru/about-standart/common-info/> (дата обращения: 22.03.2025)
64. IEC standards for Fieldbus and Real Time Ethernet [электронный ресурс]. – Режим доступа: <https://andry-tino.gitbooks.io/basics-on-industrial-protocols/content/rte/iec.html> (дата обращения: 22.03.2025)
65. Mulaosmanovic, A.M. Communication in industry by using the PROFINET protocol / A.M. Mulaosmanovic // Military Technical Courier. – 2015. – Vol. 63, No. 2. – P. 146-160. – DOI 10.5937/vojtehg63-6586.

66. Козенко, И.А. Программное обеспечение моделирования цифровых систем управления с поддержкой Profinet / И.А. Козенко, С.В.Рязанцев, Е.А. Хромых // Моделирование энергоинформационных процессов, Воронеж, 26–28 декабря 2022 года. – Воронеж: Воронежский государственный университет инженерных технологий, 2023. – С. 177-179.

67 Time Division Multiple Access (TDMA) [электронный ресурс]. – Режим доступа: <https://www.techtarget.com/searchnetworking/definition/TDMA> (дата обращения: 15.03.2025)

68. ERTEC 400 [электронный ресурс]. – Режим доступа: <https://simatic-market.ru/catalog/Siemens-CA01/10021897/info/?ysclid=mavjzwdvgy896914836> (дата обращения: 15.03.2025)

69. State estimation in networked control systems with a real-time transport protocol / F. Jin, L. Ma, Ch. Zhao, Q. Liu // Systems Science and Control Engineering. – 2024. – Vol. 12, No. 1. – DOI 10.1080/21642583.2024.2347885.

70. RTP: A Transport Protocol for Real-Time Applications [электронный ресурс]. – Режим доступа: <https://www.ietf.org/rfc/rfc3550.pdf> (дата обращения: 23.03.2025)

71. Han, W.Y. Real-time transport protocol in a wireless environment / W.Y. Han, 2002. – 1 p.

72. Dennis, J.B. Programming generality, parallelism and computer architecture // Information Processing 68 : proceedings of the IFIP Congress 1968 / edited by A.J.H. Morrell. — Amsterdam : North-Holland Publishing Co., 1969. — P. 484–492.

73 Dennis, J.B. A preliminary architecture for a basic dataflow processor / J.B. Dennis, D.P. Misunas // Proceedings of the 2nd Annual Symposium on Computer Architecture. — 1975. — P. 126–132.

74. Arvind, Culler, D.E. Dataflow architectures / Arvind, D. E. Culler. — Cambridge, MA : MIT Laboratory for Computer Science, 1986. — 294 p. — (Technical memorandum ; no. 294).

75. Lee, B. Dataflow architectures and multithreading / B. Lee, A.R. Hurson // IEEE Computer. — 1994. — Vol. 27, No. 8. — P. 27–39.

76. Arvind, Culler, D.E., Ekanadham, K. The price of asynchronous parallelism: an analysis of dataflow architectures / Arvind, D.E. Culler, K. Ekanadham. — 1988.

77. Петров, Д.А. Технология ассоциативной памяти / Д.А. Петров, А.Д. Прищепа, Д.Н. Васильев // Инновационный потенциал развития науки в современном мире: достижения и инновации: Сборник научных статей по материалам III Международной научно–практической конференции, Уфа, 09 июня 2020 года. – Уфа: Общество с ограниченной ответственностью "Научно–издательский центр "Вестник науки", 2020. – С. 101–107.

78. Кохонен, Т. Ассоциативная память / Т. Кохонен ; пер. с англ. [под ред. А. И. Галушкина]. — М. : Мир, 1980. — 390 с.

79. Орлов, С.П. Организация вычислительных машин и систем : учебное пособие / С.П. Орлов, Н.В. Ефимушкина ; Министерство образования и науки Российской Федерации, Самарский государственный технический университет. — 2-е изд., перераб. и доп. — Самара : СамГТУ, 2016. — 280 с. : ил.

80. Grabowska-Bold, I. Highlights from the ATLAS experiment / I. Grabowska-Bold // Nuclear Physics A. – 2019. – Vol. 982. – P. 8-14. – DOI 10.1016/j.nuclphysa.2018.08.024.

81. Триливен, Ф.К. Модели параллельных вычислений / Ф.К. Триливен // Системы параллельной обработки : сборник статей / под ред.

Д. Ивенса ; пер. с англ. [под ред. А. Н. Терехова]. — М. : Мир, 1985. — С. 277–284.

82. Орлов, С.А. Организация ЭВМ и систем : учебник для вузов / С.А. Орлов, Б.Я. Цилькер. — 2-е изд. — СПб. : Питер, 2011. — 688 с. : ил.

83. Задорожный, В.В. Современное состояние и перспективы развития излучающих систем активных фазированных антенных решеток / В.В. Задорожный, А. Ю. Ларин, Н. И. Чиков // Антенны. — 2022. — № 1(275). — С. 5-25. — DOI 10.18127/j03209601-202201-01.

84. Ступин, Д.Д. Использование суперкомпьютеров в дальней радиолокации: проблемы и перспективы / Д.Д. Ступин // Суперкомпьютерные технологии: разработка, программирование, применение (СКТ-2010) : материалы Международной научно-технической конференции : в 2 т. — Таганрог : Изд-во ТТИ ЮФУ, 2010. — Т. 2. — С. 244–250.

85. Яблонский, С.В. Развитие высокопроизводительных систем обработки информации / С.В. Яблонский, Н.Е. Конева, Ф.Б. Конев // Естественные и математические науки в современном мире. — 2014. — № 19. — С. 49-56.

86. Navion AMD EPYC 2U Twin Systems [электронный ресурс]. — Режим доступа: <https://www.microway.com/product/navion-amd-epyc-2u-twin-systems/> (дата обращения: 5.04.2025)

87. Механов, В.Б. Особенности архитектуры универсальных микропроцессоров : учебное пособие / В.Б. Механов. — Пенза : Изд-во ПГУ, 2010. — 176 с. — Текст : непосредственный.

88. Васюков, В.Н. Цифровая обработка сигналов и сигнальные процессоры в системах подвижной радиосвязи : учебник для студентов, обучающихся по специальности 201200 - «Средства связи с подвижными

объектами» / В. Н. Васюков. — Новосибирск : Изд-во НГТУ, 2003. — 268 с. — (Учебники НГТУ). — ISBN 5-7782-0314-4.

89. Поветкин, И.С. Исследование возможностей современных цифровых сигнальных процессоров / И.С. Поветкин // Актуальные проблемы авиации и космонавтики. — 2018. — Т. 1, № 14. — С. 387–389.

90. Wale, K. A comparative evaluation of the TMS320C67X digital signal processor / K. Wale // Electronic Engineering. — 1998. — Vol. 70, No. 862. — P. 21-22.

91. ASIC Technology: Different Types, Design Flow, Applications [электронный ресурс]. — Режим доступа: <https://www.electronicshub.org/introduction-to-asic-technology/> (дата обращения: 7.04.2025)

92. Golshan, K. Physical design essentials: an ASIC design implementation perspective / K. Golshan. — New York : Springer, 2007. — 212 p.

93. Павлов, А.Е. Первый антенный модуль 5G mmwave для смартфонов / А.Е. Павлов, Д. Алдабергенов, С.В. Губин // Передовые инновационные разработки. Перспективы и опыт использования, проблемы внедрения в производство : сборник научных статей по итогам третьей международной научной конференции, Казань, 30 апреля 2019 года / ПАО ГАЗПРОМ; ООО Газпром трансгаз Казань. Том Часть 1. — Казань: ООО «Конверт», 2019. — С. 176-177.

94. Максфилд, К. Проектирование на ПЛИС. Курс молодого бойца / К. Максфилд ; [пер. с англ. под ред. А. В. Шишкова]. — Москва : Додэка-XXI, 2007. — 408 с. : ил. — (Серия «Программируемые системы»). — ISBN 978-5-94120-147-1.

95. Каляев, И. А. Реконфигурируемые вычислительные системы на основе ПЛИС / И. А. Каляев, И. И. Левин. — Ростов-на-Дону : Издательство Южного научного центра РАН, 2022. — 506 с. — ISBN 978-5-4358-0232-0.

96. Тарасов, И. Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования / И. Е. Тарасов. — Москва : Горячая линия—Телеком, 2020. — 538 с. : ил.

97. Гузик, В.Ф. Реконфигурируемые вычислительные системы / В.Ф. Гузик, И.А. Каляев, И.И. Левин ; под редакцией И.А. Каляева ; Южный федеральный университет. — Таганрог : Издательство Южного федерального университета, 2016. — 472 с.

98. Левин, И.И. Реконфигурируемые вычислительные системы с открытой масштабируемой архитектурой / И.И. Левин // Труды Пятой Международной конференции «Параллельные вычисления и задачи управления» РАСО'2010 : [материалы конференции]. — Москва : ИПУ им. В.А. Трапезникова РАН, 2010. — С. 83–95.

99. Научно-исследовательский центр супер-ЭВМ и нейрокompьютеров [электронный ресурс]. – Режим доступа: <http://superevm.ru> (дата обращения 11.05.2025).

100. Левин, И.И. Высокопроизводительные реконфигурируемые вычислительные системы на основе ПЛИС VIRTEX-7 / И.И. Левин, А.И. Дордопуло, И.А. Каляев, В.А. Гудков // Программная инженерия. — 2014. — № 6. — С. 3–8.

101. Перспективы развития систем синхронизации и единого времени / В.Д. Беспалый, А.Е. Тюляков, Л.Я. Белов [и др.] // Метрология времени и пространства : материалы 6-го Международного симпозиума, Менделеево, 17–19 сентября 2012 года / Федеральное государственное унитарное предприятие «Всероссийский научно-исследовательский институт физико-

технических и радиотехнических измерений». — Менделеево : ВНИИФТРИ, 2012. — С. 155.

102. Кудрявцев, А.А. Проблема поддержания единого времени в распределенных системах / А.А. Кудрявцев, В.Г. Промыслов // Управление развитием крупномасштабных систем MLSD 2010 : материалы Четвертой международной конференции : в 2 т. — Москва : ИПУ им. В. А. Трапезникова РАН, 2010. — Т. 2. — С. 218.

103. Гапанович, В.А. Система единого времени ОАО "РЖД" / В.А. Гапанович, А.Н. Слюняев // Автоматика, связь, информатика. — 2018. — № 12. — С. 2-6.

104. Аврамчиков, В. М. Применение спутниковых навигационных систем ГЛОНАСС и GPS для определения местоположения воздушных судов: достоинства и недостатки / В.М. Аврамчиков, Р.А. Акзигитов, А.И. Марчук // Актуальные проблемы авиации и космонавтики : материалы IX Международной научно-практической конференции, посвященной Дню космонавтики : в 3 т. — Красноярск : СибГУ им. М. Ф. Решетнёва, 2023. — С. 674–676.

105. Quality evaluation of the pulse per second (PPS) signals from commercial GNSS receivers / X. Niu, K. Yan, T. Zhang [et al.] // GPS Solutions. — 2015. — Vol. 19, No. 1. — P. 141-150. — DOI 10.1007/s10291-014-0375-7.

106. Welcome to the home of the Network Time Protocol (NTP) Project [электронный ресурс]. — Режим доступа: <https://www.ntp.org/> (дата обращения 12.05.2025).

107. What is IRIG Standard [электронный ресурс]. — Режим доступа: <https://www.irig.org/> (дата обращения 12.05.2025).

108. Simple Network Time Protocol (SNTP) Version 4 [электронный ресурс]. — Режим доступа: <https://2rfc.net/2030> (дата обращения 12.05.2025).

109. Melikyan, V.S. Data-clock setup and hold times margins correction method in high-speed serial links / V.S. Melikyan, A.H. Shishmanyanyan, N.V. Melikyan [et al.] // CSIT 2013 : 9th International Conference on Computer Science and Information Technologies : revised selected papers, Yerevan, 23–27 September 2013. — Yerevan, 2013.— DOI: 10.1109/CSITechnol.2013.6710331.

110. Андреев, Д.А. Протокол синхронизации времени RTP / Д.А. Андреев // Лучшая исследовательская статья 2024 : сборник статей V Международного научно-исследовательского конкурса, Пенза, 30 января 2024 года. — Пенза : Наука и Просвещение, 2024. — С. 7–10.

111. Валяев, Д.А. Сравнение NTP и RTP протоколов в синхронизации часов компьютерных сетей / Д.А. Валяев, И.Г. Стахеев, Н.И. Фокин // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2021) : сборник научных статей : в 4 т. — Санкт-Петербург : СПбГУТ им. проф. М. А. Бонч-Бруевича, 2021. — Т. 4. — С. 40–44.

112. Чичёв, С.И. Теория и технология волоконно-оптических линий связи / С.И. Чичёв, В.Ф. Калинин, Е.И. Глинкин // Агротехника и энергообеспечение. — 2014. — № 4 (4). — С. 8-10.

113. 7 series product selection guide [электронный ресурс]. – Режим доступа: <https://www.amd.com/content/dam/xilinx/support/documents/selection-guides/7-series-product-selection-guide.pdf> (дата обращения 20.04.2025).

114. 7 Series FPGAs Clocking Resources [электронный ресурс]. – Режим доступа: https://www.amd.com/content/dam/xilinx/support/documents/user_guides/ug472_7Series_Clocking.pdf (дата обращения 11.04.2025).

115. Левин, И.И. Программирование реконфигурируемых вычислительных узлов на языке COLAMO : учебное пособие / И.И. Левин, А.И. Дордопуло, В.А. Гудков. — 2-е изд., доп. — Ростов-на-Дону : Издательство Южного федерального университета, 2016. — 114 с.

116. Avizienis, A. Dependable computing: from concepts to application / A. Avizienis, J.-C. Laprie // IEEE Transactions on Computers. — 1986. — Vol. 35, no. 5. — P. 629–638.
117. Avizienis, A. Basic concepts and taxonomy of dependable and secure computing / A. Avizienis, J.-C. Laprie, B. Randell [et al.] // IEEE Transactions on Dependable and Secure Computing. — 2004. — Vol. 1, no. 1. — P. 11–33.
118. Муха, А.А. Количественная оценка уровня гарантоспособности компьютерных систем / А.А. Муха // Математические машины и системы. — 2019. — № 4. — С. 146-153.
119. Avizienis, A. Fault-tolerance: The survival attribute of digital systems / A. Avizienis // Proceedings of the IEEE. — 1978. — Vol. 66, no. 10. — P. 1109-1125. — DOI: 10.1109/PROC.1978.11107.
120. Ключников, В.Ю. Синдром Кесслера: будет ли закрыта дорога в космос? / В.Ю. Ключников // ВКС. — 2021. — № 4 (109). — URL: <https://cyberleninka.ru/article/n/sindrom-kesslera-budet-li-zakryta-doroga-v-kosmos> (дата обращения: 25.10.2024).
121. NASA ORBITAL DEBRIS PROGRAM OFFICE [электронный ресурс]. — Режим доступа: <https://orbitaldebris.jsc.nasa.gov/> (дата обращения 19.04.2025).
122. Zhou, Y. Influence of antenna half-power beam width on equipment under test volume in radiation disturbance measurement / Y. Zhou, D. Zhang, X. Xu, J. Li // Journal of Physics: Conference Series. — 2021. — Vol. 2108, No. 1. — Art. 012078.
123. Левин, И. И. Система передачи данных на основе программируемых логических интегральных схем в устройствах цифровой обработки сигналов с фазированными антенными решетками / И. И. Левин, Е.А. Семерников, Д. С. Буряков // Вестник компьютерных и информационных

технологий. – 2022. – Т. 19. – № 3(213). – С. 42-50. – DOI 10.14489/vkit.2022.03.pp.042-050.

124. Левин, И.И. Некоторые методы синхронизации информационных потоков в системах цифровой обработки сигналов / И.И. Левин, Д.С. Буряков // Известия ЮФУ. Технические науки. – 2024. – № 5(241). – С. 243-254. – DOI 10.18522/2311-3103-2024-5-243-254.

125. Буряков, Д.С. Подсистема межблочной синхронной передачи данных на основе ПЛИС в комплексах цифровой обработки сигналов / Д.С. Буряков, И.И. Левин // Молодой ученый. — 2022. — № 15 (410). — С. 14–20. — URL: <https://moluch.ru/archive/410/90408/> (дата обращения: 25.10.2024).

126. Буряков, Д.С. Система передачи данных на основе ПЛИС в устройствах цифровой обработки сигналов с фазированными антенными решетками / Д.С. Буряков, И.И. Левин // Многопроцессорные вычислительные и управляющие системы: материалы Всероссийской научно-технической конференции (МВУС-2022), Таганрог, 27–30 июня 2022 года. – Ростов-на-Дону: Южный федеральный университет, 2022. – С. 138-142.

127. Буряков Д. С., Подход к повышению пропускной способности линий связи в комплексах когерентной цифровой обработки сигналов (276 с.) // Материалы научных мероприятий: Всероссийской конференции с международным участием «Угрозы и риски на Юге России в условиях геополитического кризиса»: материалы докладов (г. Ростов-на-Дону, 15–18 марта, 26–29 апреля 2023 г.); XIX Ежегодной молодежной научной конференции «Достижения и перспективы научных исследований молодых ученых Юга России»: тезисы докладов (г. Ростов-на-Дону, 17–28 апреля 2023 г.) / [гл. ред. акад. Г.Г. Матишов]. – Ростов-на-Дону: Изд-во ЮНЦ РАН, 2023. – 352 с. – ISBN 978-5-4358-0251-1.

128. Буряков, Д.С. Организация когерентной обработки сигналов в комплексах ЦОС / Д.С. Буряков // Всероссийская школа-семинар

Национального центра физики и математики для студентов старших курсов, аспирантов, молодых ученых и специалистов «Центр исследования архитектур суперкомпьютеров» : тезисы. – Саров : ФГУП «РФЯЦ-ВНИИЭФ», 2023. – С. 22–23. – 63 с. : ил.

129. Буряков Д.С., Методы повышения пропускной способности линий связи в комплексах когерентной цифровой обработки сигналов (133 – 136 с.) // XVI Всероссийская мультikonференция по проблемам управления (МКПУ-2023): материалы мультikonференции (г. Волгоград, 11-15 сентября 2023 г.). В 4 т. Т. 2. Управление в распределенных и сетевых системах (УРСС-2023) / редкол.: И.А. Каляев, В.Г. Пешехонов, С.Ю. Желтов [и др.]. – Волгоград: ВолгГТУ, 2023. – 368 с. – ISBN 978-5-9948-4703-9, ISBN 978-5-9948-4705-3 (Т.2).

130. Буряков Д. С. Методы повышения гарантоспособности систем передачи данных в комплексах ЦОС / Д. С. Буряков // XIV Всероссийское совещание по проблемам управления (ВСПУ-2024): сборник научных трудов, 17-20 июня 2024 г. – М.: ИПУ РАН, 2024. – С. 2513-2517.

131. Буряков Д. С. Методы согласованной цифровой обработки сигналов / Д. С. Буряков // Наука Юга России: достижения и перспективы: Тезисы докладов XX Всероссийской ежегодной молодежной научной конференции с международным участием, Ростов-на-Дону, 15–26 апреля 2024 года. – Ростов-на-Дону: Южный научный центр РАН, 2024. – С. 90. – ISBN 978-5-4358-0258-0.

132. Буряков Д. С. Обнаружение и устранение аварийных ситуаций в каналах передачи данных систем ЦОС // Супервычисления и математическое моделирование. Тезисы XIX Международной конференции (г. Саров, 20-24 мая 2024 г.). – Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2024. – С. 38-39 – ISBN 978-5-9515-0559-0.

133. Буряков Д. С. Реализация комбинированного метода изохронной передачи данных на РВС / Д. С. Буряков // Наука Юга России: достижения и перспективы: Тезисы докладов XXI Всероссийской ежегодной молодежной научной конференции с международным участием, Ростов-на-Дону, 14–25 апреля 2025 года. – Ростов-на-Дону: Южный научный центр РАН, 2024. – С.104. – ISBN 978-5-4358-0270-2.

ПРИЛОЖЕНИЕ 1

**АКТЫ О ВНЕДРЕНИИ И ИСПОЛЬЗОВАНИИ РЕЗУЛЬТАТОВ
ДИССЕРТАЦИИ**

БУРЯКОВ ДМИТРИЙ СЕРГЕЕВИЧ

**МЕТОДЫ И ПРОГРАММНЫЕ СРЕДСТВА ОБЕСПЕЧЕНИЯ
ИЗОХРОННОЙ ПЕРЕДАЧИ ДАННЫХ В КОМПЛЕКСАХ ЦИФРОВОЙ
ОБРАБОТКИ СИГНАЛОВ**

2.3.5 - Математическое и программное обеспечение вычислительных систем,
комплексов и компьютерных сетей

Диссертация на соискание ученой степени кандидата технических наук

Научный руководитель:

доктор технических наук, профессор

Левин И.И.

Таганрог – 2025

УТВЕРЖДАЮ

Технический директор

ООО «НИЦ СЭ и НК»



АКТ

о внедрении результатов кандидатской диссертации «Методы и программные средства обеспечения изохронной передачи данных в комплексах цифровой обработки сигналов» конструктора 1 категории Общества с ограниченной ответственностью «НИЦ супер-ЭВМ и нейрокомпьютеров» (ООО «НИЦ СЭ и НК»)

Бурякова Дмитрий Сергеевича

Комиссия в составе председателя, начальника отдела к.т.н Сорокина Д.А., и членов комиссии: начальника отдела к.т.н Раскладкина М.К., ведущего научного сотрудника к.т.н Семерникова Е.А., составила настоящий акт о том, что в разработках ООО «НИЦ СЭ и НК» внедрены результаты диссертации Бурякова Дмитрия Сергеевича на тему «Методы и программные средства обеспечения изохронной передачи данных в комплексах цифровой обработки сигналов»:

1. Комбинированный метод изохронной передачи данных, сочетающий использование единого машинного времени, опорной тактовой частоты и служебных промежутков между массивами операндов.

2. Модернизированный метод изохронной передачи данных с использованием процедур переключения опорных каналов, предварительной проверки контрольных сумм и нумерации массивов операндов.

Указанные результаты диссертации Бурякова Д.С. были внедрены при реализации инструментальных программных средств, обеспечивающих изохронную передачу данных в комплексах ЦОС.

Основываясь на результатах диссертации совместно с Буряковым Д.С. создана «Программа обработки изохронных потоков данных» (свидетельство о государственной регистрации программ для ЭВМ №2025618684, РФ). Левин И.И., Буряков Д.С., Чекина М.Д. Зарегистр. в Реестре программ для ЭВМ 16.04.2025 г. Правообладатель: Общество с ограниченной ответственностью «НИЦ супер-ЭВМ и нейрокомпьютеров» (ООО «НИЦ СЭ и НК»).

Результаты диссертации Бурякова Д.С. использовались при выполнении СЧ ОКР «Разработка цифровой диаграммообразующей системы фазированной антенной решетки приемного аппаратурно-антенного комплекса».

Внедрение методов и программных средств, разработанных в диссертации Бурякова Д.С. «Методы и программные средства обеспечения изохронной передачи данных в комплексах цифровой обработки сигналов» позволило повысить гарантоспособность программных комплексов ЦОС в 3,8 раз по сравнению с известными решениями.

ПРЕДСЕДАТЕЛЬ КОМИССИИ

Начальник отдела ООО «НИЦ СЭ и НК»,
к.т.н.



Д.А. Сорокин

ЧЛЕНЫ КОМИССИИ

Начальник отдела ООО «НИЦ СЭ и НК»,
к.т.н.



М.К. Раскладкин

Ведущий научный сотрудник ООО «НИЦ СЭ и НК»,
к.т.н.



Е.А. Семерников

«УТВЕРЖДАЮ»

Директор



Г.Е. Веселов

30» 05 2025 г.

АКТ

об использовании результатов диссертации на соискание ученой степени
кандидата технических наук
Бурякова Дмитрия Сергеевича
в учебном процессе кафедры интеллектуальных и многопроцессорных систем
Института компьютерных технологий и информационной безопасности
Южного федерального университета

Комиссия в составе:

председателя Горбунова А.В.

и членов комиссии Механцева Б.Е., Михайлова Д.В.

составили настоящий акт о том, что результаты диссертационной работы
**«Методы и программные средства обеспечения изохронной передачи данных в
комплексах цифровой обработки сигналов»,**
представленной на соискание ученой степени кандидата технических наук,
использованы в учебном процессе кафедры интеллектуальных и многопроцессорных
систем (ИМС) Института компьютерных технологий и информационной
безопасности (ИКТИБ) Южного федерального университета (ЮФУ).

Настоящий акт подтверждает, что в учебном процессе кафедры ИМС ИКТИБ
ЮФУ используются следующие научно-теоретические и практические результаты
кандидатской диссертации Бурякова Д. С.:

- комбинированный метод изохронной передачи данных, сочетающий
использование единого машинного времени, опорной тактовой частоты и служебных
промежутков при передаче массивов операндов;
- алгоритмы обнаружения и парирования аварийных ситуаций в системе
изохронной передачи данных.

Указанные материалы используются в лекционном курсе по дисциплине «Методы цифровой обработки сигналов, изображений и ситуаций» (тема № 5 «Спектральный анализ. Дискретное преобразование Фурье») и «ПЛИС-технологии и методы создания эффективных прикладных программ для РВС» (тема № 20 «Параллельные системы» модуля 1 «Основы работы с ПЛИС») для подготовки магистров направления подготовки 01.04.02 Прикладная математика и информатика (образовательная программа «Прикладная математика для высокопроизводительных вычислительных систем»).

Председатель комиссии:

Зам. директора ИКТИБ ИТА ЮФУ
по учебной работе, к.т.н., доцент

А.В. Горбунов

Члены комиссии:

Лектор дисциплины
«Методы цифровой обработки сигналов,
изображений и ситуаций»,
доцент кафедры ИМС ЮФУ,
к.т.н.

Д.В. Михайлов

Лектор дисциплины
«ПЛИС-технологии и методы создания
эффективных прикладных программ для РВС»,
доцент кафедры ИМС ЮФУ,
к.п.н.

Б.Е. Механцев

ПРИЛОЖЕНИЕ 2

**СВИДЕТЕЛЬСТВО О ГОСУДАРСТВЕННОЙ РЕГИСТРАЦИИ
ПРОГРАММ ДЛЯ ЭВМ**

БУРЯКОВ ДМИТРИЙ СЕРГЕЕВИЧ

**МЕТОДЫ И ПРОГРАММНЫЕ СРЕДСТВА ОБЕСПЕЧЕНИЯ
ИЗОХРОННОЙ ПЕРЕДАЧИ ДАННЫХ В КОМПЛЕКСАХ ЦИФРОВОЙ
ОБРАБОТКИ СИГНАЛОВ**

2.3.5 - Математическое и программное обеспечение вычислительных систем,
комплексов и компьютерных сетей

Диссертация на соискание ученой степени кандидата технических наук

Научный руководитель:

доктор технических наук, профессор

Левин И.И.

Таганрог – 2025

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2025660549

Программа обработки изохронных потоков данных

Правообладатель: *Общество с ограниченной ответственностью «НИЦ супер-ЭВМ и нейрокомпьютеров» (RU)*

Авторы: *Левин Илья Израилевич (RU), Буряков Дмитрий Сергеевич (RU), Чекина Мария Дмитриевна (RU)*

Заявка № 2025618684

Дата поступления 16 апреля 2025 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 24 апреля 2025 г.



Руководитель Федеральной службы
по интеллектуальной собственности

документ подписан электронной подписью
Сертификат 04424e1e493008584241667068a2006
Владелец: **Зубов Юрий Сергеевич**
Действителен с 11.07.2024 по 03.10.2025

Ю.С. Зубов