

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э.Баумана)

ФАКУЛЬТЕТ РК «Робототехника и комплексная автоматизация»

КАФЕДРА РК9 «Компьютерные системы автоматизации производства»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ДИПЛОМНОМУ ПРОЕКТУ

НА ТЕМУ:

*Разработка автоматизированной системы
регистрации потока данных радиолокационной
станции.*

Студент РК9-Д2
(Группа)

(Подпись, дата)

(И.О.Фамилия)

Руководитель дипломного проекта

(Подпись, дата)

(И.О.Фамилия)

Консультант по конструкторской части

(Подпись, дата)

(И.О.Фамилия)

Консультант по технологической части

(Подпись, дата)

(И.О.Фамилия)

Консультант по исследовательской части

(Подпись, дата)

(И.О.Фамилия)

Консультант по
организационно-экономической части

(Подпись, дата)

(И.О.Фамилия)

Консультант по охране труда и экологии

(Подпись, дата)

(И.О.Фамилия)

Нормоконтролер

(Подпись, дата)

(И.О.Фамилия)

2015 г.

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э.Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой _____
(Индекс)

(И.О.Фамилия)
« ____ » _____ 20 ____ г.

З А Д А Н И Е
на выполнение дипломного проекта

Студент группы РК9-Д2

Чернов Алексей Олегович

(фамилия, имя, отчество)

Тема дипломного проекта Разработка автоматизированной системы регистрации потока данных радиолокационной станции.

Источник тематики (НИР кафедры, заказ организаций и т.п.)

Заказ организации ОАО «Всероссийский научно-исследовательский институт радиотехники»

Тема дипломного проекта утверждена распоряжением по факультету
№ _____ от « ____ » _____ 20__ г.

Часть 1. Технико-экономическое обоснование

Проведение целевого обследования. Обзор существующих средств регистрации.
Выделение системы из среды. Выбор концепции системы.

Часть 2. Научно-исследовательская

Исследование влияния задержки между пакетами на скорость передачи данных.

Часть 3. Проектно-конструкторская

Формирование технического задания. Разработка алгоритма программной части подсистемы приема и хранения данных. Разработка модульной архитектуры программной части подсистемы передачи данных. Разработка интерфейсов модулей. Разработка схемы электрической принципиальной платы ячейки.

Часть 4. Технологическая

Реализация программной части подсистемы приема и хранения данных. Реализация программной части подсистемы передачи данных. Апробирование системы.

Часть 5. Организационно-экономическая

Определить трудоемкость и затраты на разработку программного комплекса автоматизированной системы регистрации потока данных радиолокационной станции

Консультант _____

(Подпись, дата)

(И.О.Фамилия)

Часть 6. Охрана труда и экология

Провести анализ опасных и вредных факторов, воздействующих на инженера при работе с регистратором данных. Выполнить расчет освещения на рабочем месте оператора. Выбрать рациональный способ утилизации печатных плат

Консультант _____

(Подпись, дата)

(И.О.Фамилия)

Оформление дипломного проекта:

Расчетно-пояснительная записка на 154 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)
Листов А1 – 10, из них: 3 плаката, 1 Функциональная схема IDEF0, 1 Схема принципиальная электрическая, 1 Функциональная схема электрическая, 1 Диаграмма состояния, 1 Блок схема, 1 Диаграмма последовательности, 1 Диаграмма активности.
Листов А2 – 3, из них: 1 Дерево целей, 1 Структурная схема, 1 Диаграмма развертывания.
Листов А3 – 2, из них: 1 Диаграмма классов, 1 Диаграмма компонентов

Дата выдачи задания « ___ » _____ 20__ г.

В соответствии с учебным планом дипломный проект выполнить в полном объеме в срок до « ___ » _____ 20__ г.

Руководитель дипломного проекта

(Подпись, дата)

(И.О.Фамилия)

Студент

(Подпись, дата)

(И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э.Баумана)**

ФАКУЛЬТЕТ _____

УТВЕРЖДАЮ

КАФЕДРА _____

Заведующий кафедрой _____
(Индекс)

ГРУППА _____

(И.О.Фамилия)
« ____ » _____ 20 ____ г.

**КАЛЕНДАРНЫЙ ПЛАН
выполнения дипломного проекта**

студента: _____
(фамилия, имя, отчество)

Тема дипломного проекта _____

№ п/п	Наименование этапов дипломного проекта (работы)	Сроки выполнения этапов		Отметка о выполнении	
		план	факт	Должность	ФИО, подпись
1.	Задание на выполнение проекта. Формулирование проблемы, цели и задач проекта	Планируемая дата		Руководитель	
2.	1 часть _____	Планируемая дата		Руководитель	
3.	Утверждение окончательных формулировок решаемой проблемы, цели и задач проекта	Планируемая дата		Заведующий кафедрой	
4.	2 часть _____	Планируемая дата		Руководитель	
5.	3 часть _____	Планируемая дата		Руководитель	
6.	Организационно-экономическая часть	Планируемая дат		Консультант	
7.	Охрана труда и экология	Планируемая дат		Консультант	
8.	1-я редакция проекта	Планируемая дата		Руководитель	
9.	Подготовка доклада и презентации	Планируемая дата			
10.	Заключение руководителя	Планируемая дата		Руководитель	
11.	Допуск работы к защите на ГЭК (нормоконтроль)	Планируемая дата		Нормоконтролер	
12.	Внешняя рецензия	Планируемая дата			
13.	Защита работы на ГЭК	Планируемая дата			

Студент _____
(подпись, дата)

Руководитель проекта _____
(подпись, дата)

**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э.Баумана)**

**НАПРАВЛЕНИЕ НА ЗАЩИТУ
выполнения дипломного проекта**

**Председателю
Государственной Экзаменационной Комиссии № _____**

факультета _____ МГТУ им. Н.Э.Баумана

Направляется студент _____ группы _____

на защиту дипломного проекта _____

Декан факультета _____
« ____ » _____ 20__ г.

Справка об успеваемости

Студент _____ за время пребывания в МГТУ имени Н.Э.Баумана
с 20 ____ г. по 20 ____ г. полностью выполнил учебный план со следующими оценками:
отлично – _____ %, хорошо – _____ %, удовлетворительно – _____ %.

Инспектор деканата _____

Заключение руководителя дипломного проекта

Студент _____

Руководитель _____
« ____ » _____ 20__ г.

Заключение кафедры о дипломном проекте

Дипломный проект просмотрен и студент _____ может быть допущен к
защите этой работы в Государственной Экзаменационной Комиссии.

Зав. кафедрой _____
« ____ » _____ 20__ г.

Оглавление

Реферат	10
Перечень сокращений, символов и специальных терминов	11
1. Введение.	13
2. Этап предпроектного исследования.	15
2.1. Целевое обследование.	15
2.2. Обзор существующих средств регистрации.	17
2.2.1. Системы с автоматизированной записью.	17
2.2.2. USB устройства для компьютера.	19
2.2.3. PCI-express платы расширения для компьютера.	20
3. Концептуальный этап проектирования.	22
3.1. Выбор общесистемной методологии проектирования.	22
3.2. Выделение системы из среды.	23
3.3. Разработка архитектуры и состава системы.	26
3.4. Разработка укрупненной функциональной структуры системы.	28
3.5. Выбор концепции.	29
3.5.1. Обзор внешних интерфейсов компьютера.	29
3.6. Окончательный выбор концепции.	33
4. Формирование технического задания.	35
4.1. Общие сведения.	35
4.1.1. Наименование системы.	35
4.1.2. Основания для проведения работ.	35
4.1.3. Наименование организаций — Заказчика и Разработчика	35
4.1.4. Плановые сроки начала и окончания работы.	35

4.1.5. Порядок оформления и предъявления заказчику результатов работ.	35
4.2. Назначение и цели создания системы.	36
4.2.1. Назначение системы.	36
4.2.2. Цели создания системы.	36
4.3. Характеристика объектов автоматизации.	36
4.3.1. Краткие сведения об объекте автоматизации.	36
4.3.2. Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.	37
4.4. Требования к системе.	38
4.4.1. Требования к системе в целом.	38
4.4.2. Требования к функциям (задачам), выполняемым системой.	45
4.4.3. Требования к видам обеспечения.	47
4.5. Состав и содержание работ по созданию системы.	48
4.6. Порядок контроля и приемки системы.	48
4.6.1. Виды, состав, объем и методы испытаний системы.	48
4.6.2. Общие требования к приемке работ по стадиям.	49
4.7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.	50
4.7.1. Технические мероприятия.	50
4.7.2. Организационные мероприятия.	50
5. Технический этап проектирования.	51
5.1. Разработка алгоритма программной части подсистемы приема и хранения данных.	55

5.2. Разработка модульной архитектуры программной части подсистемы передачи данных и интерфейсов взаимодействия.....	57
5.3. Разработка схемы электрической принципиальной платы ячейки.....	76
6. Рабочее проектирование.	79
6.1. Реализация программной части подсистемы приема и хранения данных.	79
6.1.1. Диаграмма классов программной части подсистемы приема и хранения данных.	79
6.1.2 Диаграмма компонентов программной части подсистемы приема и хранения данных	80
6.2. Реализация программной части подсистемы передачи данных.	82
7. Исследовательская часть.....	89
7.1. Исследование влияния задержки между пакетами на скорость передачи данных.	90
8. Апробирование.	96
9. Организационно-экономическая часть.	97
9.1. Введение.....	97
9.2. Трудоемкость и затраты на разработку программного комплекса автоматизированной системы регистрации потока данных радиолокационной станции.	97
9.2.1. Трудоемкость разработки технического задания.....	101
9.2.2. Трудоемкость разработки эскизного проекта	102
9.2.3. Трудоемкость разработки технического проекта	103
9.2.4. Трудоемкость разработки рабочего проекта	104
9.2.5. Трудоемкость выполнения стадии “Внедрение”	106

9.3. Определение цены программной продукции.....	107
9.3.1. Расчет амортизации нематериальных активов и оборудования.....	108
9.3.2. Расчет основной заработной платы.....	109
9.3.3. Расчет дополнительной заработной платы.....	110
9.3.4. Отчисления на социальные нужды.....	110
9.3.5. Накладные расходы.....	111
9.4. Итоговые результаты.....	111
10. Охрана труда и экология.....	113
10.1. Анализ опасных и вредных факторов, воздействующих на инженера при работе с регистратором данных.....	113
10.1.1. Движущиеся машины и механизмы.....	114
10.1.2. Повышенная или пониженная температура воздуха рабочей зоны.....	114
10.1.3. Повышенная или пониженная подвижность воздуха... ..	116
10.1.4. Повышенная или пониженная влажность воздуха.....	117
10.1.5. Повышенный уровень вибрации.....	118
10.1.6. Недостаточная освещенность рабочей зоны.....	119
10.1.7. Отсутствие или недостаток естественного света.....	120
10.1.8. Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.....	120
10.1.9. Повышенный уровень электромагнитных излучений.....	122
10.1.10. Монотонность труда.....	124
10.1.11. Эмоциональные перегрузки.....	125
10.2. Расчет освещенности в помещении.....	125
10.3. Выбор рационального способа утилизации печатных плат.....	128
Заключение.....	132

Список литературы	134
Список используемого программного обеспечения.....	138
Приложение 1. Листинг программного кода, реализующего класс SocketService.....	139
Приложение 2. Функциональная схема модуля PACKET_RECEIVER_FSM	142
Приложение 3. Листинг программного кода, реализующего класс FpgaToPcLink.....	143
Приложение 4. Листинг программы	149
Приложение 5. Акт о рассмотрении и проверке работоспособности программного продукта	151

Реферат

Отчет 151 с., 49 рис., 24 табл., 33 источника, 5 прил.

РАДИОЛОКАЦИЯ, РЕГИСТРАЦИЯ ДАННЫХ, ПЕРЕДАЧА ДАННЫХ, ПЛИС, ПК, ГИГАБИТ, ETHERNET, UDP.

Объект разработки — автоматизированная система регистрации потока данных радиолокационной станции.

Цель работы — разработка регистратора — инструмента для отладки и исправления ошибок алгоритмов обработки сигналов РЛС. Задачей этого устройства является запись входного потока данных блока обработки сигналов на жесткий диск в фоновом режиме работы РЛС.

При создании системы с помощью обработки экспериментальной выборки проведены исследования, целью которых было выбрать оптимальную задержку между пакетами.

Результат разработки — гибкая, автоматизированная система не требовательная к ресурсам, область применения которой распространяется далеко дальше радиолокации и охватывает любые задачи, при которых требуется передать данные для записи с ПЛИС на компьютер.

Эффективность работы системы невозможно посчитать сравнением с чем-либо ввиду отсутствия полных аналогов, следует отметить, что модуль для ПЛИС занимает всего 714 логические ячейки интегральной схемы и позволяет передавать данные со скоростью 931079324 Бит/с

Перечень сокращений, символов и специальных терминов

ВНИИРТ — Всероссийский научно-исследовательский институт радиотехники

РЛС — Радиолокационная станция

ЗРК — Зенитно-ракетный комплекс

ТТХ — Тактико-технические характеристики

ФАР — Фазированная антенная решетка

БОС — Блок обработки сигналов

КДО — Блоки оцифровки сигналов с ФАР

Сумматор — Блок объединения потока с группы КДО

ПК — Персональный компьютер

ПЛИС — Программируемая логическая интегральная схема

USB — последовательный интерфейс передачи данных ПК

PCI (PCIe, PCI-Express) — компьютерная шина

IT — Information Technology — Информационные технологии

API — Интерфейс программирования приложений

GTX — Мультигигабитные передатчики ПЛИС

ООП — Объектно-ориентированное программирование

ПС — Программное средство

ОС — Операционная система

ISA, EISA, VLB, AGP, COM, LPT, HDMI, VGA, DVI, DisplayPort —

Компьютерные интерфейсы

GPU — Graphics Processing Unit - Графический процессор

Ethernet — Семейство технологий пакетной передачи данных

ARC, ARM, AVR32, Alpha AXP, SPARC, ETRAX CRIS, SuperH, PA-RISC, FR-V, H8300, IP7000, m68k, MIPS, mn10300, PowerPC, Xtensa —
Архитектуры процессоров

SSD — Твердотельные накопители (Тип жестких дисков)

ТТЗ — Тактико-техническое задание
ТУ — Технические условия
TCP, UDP — Протоколы передачи данных по средством Ethernet
OSI — Базовая эталонная модель взаимодействия систем
MAC — Подуровень канального уровня модели OSI, согласно стандартам IEEE 802. (Ethernet)
GMII — Независимый от среды передачи интерфейс, реализующий MAC
IHL — Идентификатор
TTL — Время жизни пакета
CRC — Циклический избыточный код, контрольная сумма
ТЗ — Техническое задание
ЭП — Эскизное проектирование
ТП — Техническое проектирование
РП — Рабочее проектирование
ПП — Программный продукт
ПО — Программное обеспечение
ПЭВМ — Персональная электронная вычислительная машина

1. Введение.

Всероссийский научно-исследовательский институт радиотехники (ВНИИРТ) является одним из старейших оборонных научно-исследовательских предприятий. С момента основания, Институтом разработано и вместе с заводами поставлено в войска и за рубеж свыше 17 тысяч РЛС, более 120 типов радиолокационных станций, многие из которых освоены в серийном производстве и нашли применение в нашей стране и за рубежом.[1]

Группа сотрудников ВНИИРТ в настоящее время занимается разработкой радиолокационных станций (РЛС) нового поколения, таких как РЛС для ЗРК «Панцирь-С1», программируемой РЛС для управления дорожным движением «Фодоком-2С», мобильной трёхкоординатной РЛС Гамма-ДЕ, а так же трёхкоординатной корабельной РЛС обнаружения и целеуказания Фуркэ-Э.

Одну из ключевых функций в каждой из этих станций выполняет система обработки информации, сердцем которой является блок обработки сигналов (БОС). Очевидно, что качество обработки зависит от алгоритмов, заложенных в БОС, поэтому сотрудники ВНИИРТ непрерывно работают над разработкой новых алгоритмов обработки и улучшением работы старых. Несмотря на сильный математический аппарат, разработчики алгоритмов нуждаются в моделировании на реальных данных, а также не застрахованы от ошибок, исправление которых влечет за собой необходимость в ведении логов на различных уровнях рабочего цикла станции.

Поэтому мне была поставлена задача — разработка автоматизированной системы регистрации потока данных радиолокационной станции. Задачей этого устройства является запись входного потока данных БОС на жесткий диск в фоновом режиме работы

РЛС, что обеспечит возможность в дальнейшем воспроизводить выбранный сеанс на имитационном стенде. Это необходимо для воссоздания режимов работы, полученных во время испытаний станции в полевых условиях, что позволит проверять часть нововведений, не выходя из здания института и не организовывая дорогостоящих выездов на полигон. А также даст разработчикам алгоритмов необходимый инструмент для отладки и исправления ошибок.

Актуальность работы сопоставима с актуальностью задачи регистрации в целом. Ведь гибкой автоматизированной системе не важно, где она установлена: на военном корабле или в автоматизированном цехе, передает поток данных РЛС или информационно-измерительного канала.

2. Этап предпроектного исследования.

2.1. Целевое обследование.

В это непростое для России время, оборонные предприятия, такие как ОАО «ВНИИРТ», должны в первую очередь заботиться о повышении качества своей продукции и улучшении ТТХ своей продукции.

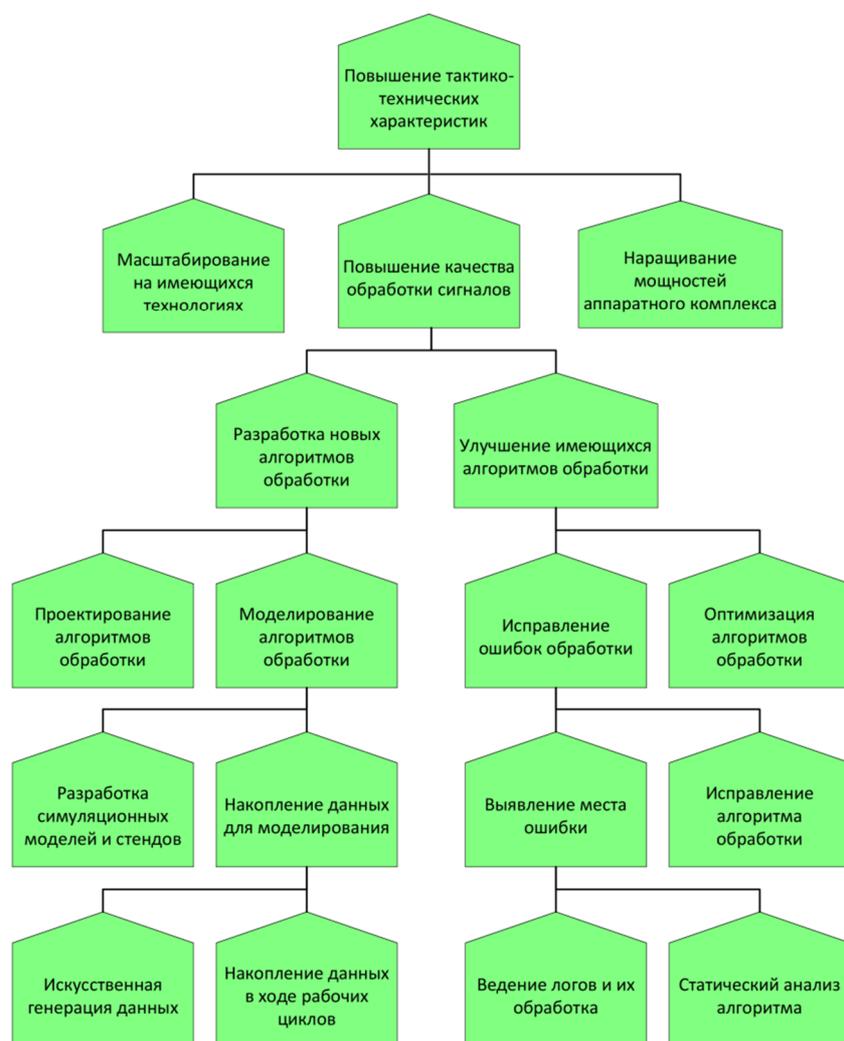


Рис. 1. Дерево целей

Накопление данных в ходе рабочих циклов и введение логов и их обработка — это удобный и необходимый инструмент для разработчика.

Причем для разработчика желательно иметь доступ к данным на любом этапе обработки.

К сожалению, РЛС Фуркэ-Э, которая устанавливается на корветы проекта 20380, не имеет доступа к данным с ФАР непосредственно перед обработкой, что затрудняет отладку и моделирование алгоритмов.



Рис. 2. Корвет проекта 20380

Фазированная антенная решётка (ФАР) — Антенная решетка, направление излучения и форма соответствующей диаграммы направленности которой регулируются изменением амплитудно-фазового распределения токов и полей возбуждения на излучающих элементах.[2]

Данные с ФАР оцифровываются в тракте обработки сигналов и объединяются в единый поток непосредственно перед обработкой. Проблема заключается в отсутствии механизма донесения этих данных до разработчика.

Для решения этой проблемы необходим регистратор. Устройство, которое выполняет единственную цель — сбор и хранение потока радиолокационных данных в удобном формате в виде файла, для последующего его использования на ПК.

Конструкция военных судов довольно специфична. Здесь нет просторных офисов с компьютерами, объединенными в локальную сеть. Это накладывает ограничение на конструкцию регистратора, он должен быть самодостаточен, в том числе и в плане электропитания, а также обладать небольшими размерами для удобства транспортировки оператором регистратора на рабочее место.

2.2. Обзор существующих средств регистрации.

Существующие решения в области регистрации данных:

- Системы с автоматизированной записью
- Устройства-расширения для компьютера

2.2.1. Системы с автоматизированной записью.

Рассматривались универсальные портативные регистраторы данных, способные регистрировать данные восьмибитной шины или восьми однобитных каналов, причем у второго способа имеется недостаток в виде необходимости последующей обработки файла.

Преимущества таких систем:

- Высока степень автоматизации
- Простота разработки системы
- Простота установки
- Лаконичность системы

Общие ограничения для систем с автоматизированной записью:

- Ограничение по доступной памяти
- Ограничения по доступным интерфейсам
- Привязанность к сторонним разработкам
- Ограничения по скорости регистрируемого потока.

Были рассмотрены решения компаний, продукция которых можно было использовать в поставленной задаче:

- Yokogawa Electric (Япония)
- Hottinger Baldwin Messtechnik (НВМ) (Германия)
- MegaSensor (Россия)
- Eсograph (Германия)
- Janitzu (Германия)
- Kyowa (Япония)
- Mitsubishi Electric (Япония)

Лидер среди рассмотренных вариантов универсальный регистратор GEN3i производства НВМ обладает ограничением в 133Мб/с на восьмибитный поток, что значительно ниже требуемой.

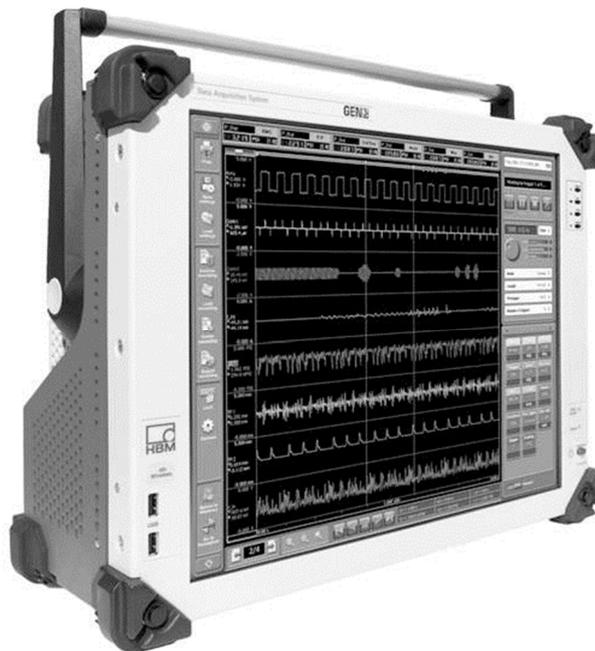


Рис. 3. GEN3i производства НВМ (Германия)

2.2.2. USB устройства для компьютера.

Рассматривались универсальные портативные регистраторы данных, подключаемые к компьютеру по шине USB, способные регистрировать восьмибитные шины данных или восемь однобитных каналов.

Преимущества такого рода систем:

- Высокая степень автоматизации
- Простота разработки системы
- Простота установки
- Компактность регистратора

Общие ограничения устройств-расширений для компьютера с интерфейсом USB:

- Ограничения по доступным интерфейсам
- Ограничения по скорости регистрируемого потока.
- Наличие стороннего устройства, подключаемого к компьютеру
- Наличие стороннего программного обеспечения
- Необходимость наличия компьютера с USB портом

Были рассмотрены решения компаний:

- Hantek (Китай)
- НПК «Рэлсиб» (Россия)
- Eсоgraph (Германия)
- Fourtec (США)

Наличие стороннего устройства со сторонним же программным обеспечением — это даже большая проблема, чем стороннее полное законченное устройство. И новости из мира IT, в которых с завидной регулярностью проскальзывают заметки о нахождении так называемых уязвимостей нулевого дня, только увеличивают такие опасения.

И если с фактом наличия стороннего устройства, после проверки на наличие аппаратных закладок можно смириться, а программное обеспечение при наличии API можно переписать, то с максимальной скоростью регистрируемого потока в 480 Мб/с, что примерно в 2 раза ниже необходимой, смириться не получится.

2.2.3. PCI-express платы расширения для компьютера.

Преимущества такого рода систем:

- Высокая степень автоматизации
- Простота разработки системы
- Простота установки
- Высокая скорость, с большим запасом на развитие

Общие ограничения для плат-расширений для компьютера с интерфейсом PCI-express:

- Ограничения по доступным интерфейсам
- Наличие стороннего устройства, подключаемого к компьютеру
- Необходимость наличия компьютера с PCI-express

Были рассмотрены решения компаний:

- ЗАО «Руднев-Шиляев» (Россия)
- «ТомИУС-ПРОЕКТ» (Россия)
- ЗАО «ИнСис» (Россия)
- Spectrum (Германия)

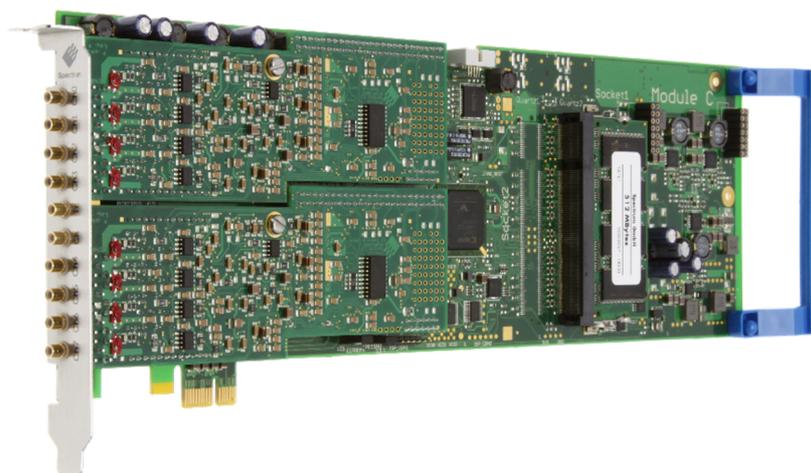


Рис. 4. Плата-расширение PCI-express

Готовая продукция представленных компаний способна предоставить возможность регистрации данных на скоростях до 3 Гб/с. Однако, в таком случае передающая сторона должна быть оснащена передатчиками GTX, похвастаться которыми могут только ПЛИС новых поколений Virtex 6, Virtex 7. Кроме того подключаются такие платы в полноразмерный размер PCI-express, что влечет за собой ограничение использования портативных компьютеров и негативно влияет на габариты регистратора в целом.

Эту проблему можно устранить, ведь ЗАО «Руднев-Шиляев», «ТомИУС-ПРОЕКТ», ЗАО «ИнСис» (Россия) предлагают разработку плат на базе ПЛИС, соединенных с компьютером по PCI-Express, в том числе через разъем mini PCIe, что позволит использовать портативные компьютеры. Однако, такую разработку способен произвести и ОАО «ВНИИРТ», кроме того наличие ПЛИС на стороне передатчика ставит под сомнение необходимости второй ПЛИС в качестве посредника.

Обзор существующих средств разработки привел к ситуации, при которой очевидно, что необходима разработка своей системы регистрации данных. Причем система должна состоять из портативного компьютера, подключенного к ПЛИС на входной ячейке БОС при помощи одного из общедоступных интерфейсов.

3. Концептуальный этап проектирования.

3.1. Выбор общесистемной методологии проектирования.

Задача, поставленная на этапе предпроектного исследования, может быть решена на основе следующих концепций:

- Модульность
- Объектная ориентированность

Модульность — это свойство системы, связанное с возможностью ее декомпозиции на ряд связанных между собой модулей. Применительно к конструированию технических систем модульность — принцип, согласно которому функционально связанные части группируются в законченные узлы — модули. В свою очередь модульность в программировании — принцип, согласно которому программное средство (ПС) разделяется на отдельные именованные сущности, называемые модулями. Модульность часто является средством упрощения задачи проектирования ПС и распределения процесса разработки ПС между группами разработчиков. При разбиении ПС на модули для каждого модуля указывается реализуемая им функциональность, а также связи с другими модулями.

Объектно-ориентированное программирование (ООП) — парадигма программирования, в которой основными концепциями являются понятия объектов и классов.

Класс в ООП — разновидность абстрактного типа данных, характеризуемый способом своего построения.

Объект - это сущность, которой можно посылать сообщения, и которая может на них реагировать, используя свои данные. Объект — это экземпляр класса. Данные объекта скрыты от остальной программы. Скрытие данных называется инкапсуляцией. Наличие инкапсуляции достаточно для объектности языка программирования, но ещё не означает

его объектной ориентированности — для этого требуется наличие наследования. Но даже наличие инкапсуляции и наследования не делает язык программирования в полной мере объектным с точки зрения ООП. Основные преимущества ООП проявляются только в том случае, когда в языке программирования реализован полиморфизм, то есть возможность объектов с одинаковой спецификацией иметь различную реализацию. Выбранные концепции проектирования позволили производить разработку поэтапно и распределить разработку по времени, однако, для эффективной работы необходимо на этапе концептуального проектирования после выделения системы из среды разработать её архитектуру, обособить её модули.

3.2. Выделение системы из среды.

В ходе концептуального проектирования была составлена структурная схема тракта обработки сигналов. На ней прослеживается структура блоков и путь, по которому проходит сигнал от фазированной антенной решетки к блоку обработки сигналов и далее на рабочее место оператора.

В состав тракта входят 56 блоков КДО, в которых происходит оцифровка сигнала с группы антенных приемников и передается дальше по тракту на блоки суммирования. В Сумматорах поток объединяется и передается далее на блок обработки сигналов. БОС — модульная система, состоящая из входной, выходной и других ячеек, занимающихся непосредственно обработкой сигналов. На входной ячейке весь поток данных РЛС объединяется, чтобы потом распределиться по восьми ячейкам обработки, поэтому это наиболее удачная точка, в которой можно записать на диск поток данных РЛС. Общий поток данных достигает 800 Мб/с.

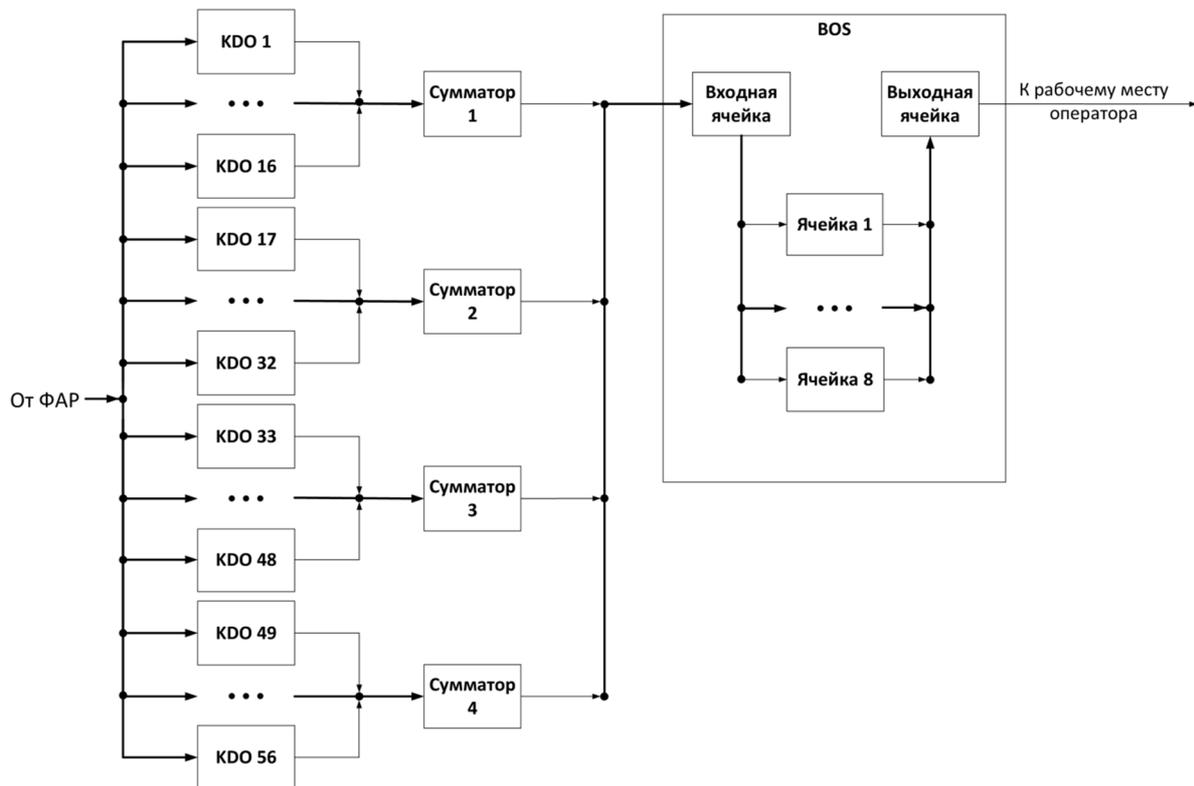


Рис. 5. Структурная схема тракта обработки сигналов РЛС по ГОСТ 24.302-80

Входная ячейка работает под управлением ПЛИС семейства Spartan-6 XC6SLX150t, характеристики которой указаны в Таблица 2. Эта логическая интегральная схема имеет связь с 96 выводами на корпус блока обработки сигналов

Таблица 1. Выводы на корпусе блока

Выводы, связанные с сумматорами	64
Выводы диагностики и управления	16
Незанятые выводы	16

При разработке модуля для ПЛИС входной ячейки следует также учитывать то, что она несет в себе рабочую прошивку. Занимаемые при этом ресурсы указаны в Таблица 2.

Таблица 2. Основные параметры ПЛИС XC6SLX150t семейства Spartan-6

Логические ресурсы	Количество секций Slices	23 038	10 141
	Общее число триггеров CLB	184 304	20 783
	Число логических ячеек	147 443	21 188
Ресурсы памяти	Объем распределенной памяти	1355 К	32 К
	Количество модулей блочной памяти Block RAM емкостью 18 кбит	268	69
	Объем блочной памяти (1К = 1024 бит) Block RAM	4824 К	1242 К
Модули синхронизации	Количество блоков управления синхронизацией (CMT)	6	0
	Число цифровых модулей управления синхронизацией (DCM)	12	1
	Число аналоговых модулей фазовой автоподстройки частоты (PLL)	6	1
Встроенные аппаратные модули	Число аппаратных секций DSP48A1	180	4
	Число аппаратных контроллеров интерфейса памяти (MCB)	4	0
Ресурсы ввода/вывода	Количество банков ввода/вывода	6	5
	Максимальное число пользовательских выводов	570	246
	Максимальное число дифференциальных пар выводов	285	80
Конфигурационная память	Объем конфигурационной памяти, Мбит	33,7	

3.3. Разработка архитектуры и состава системы.

Итогом предпроектного исследования и выделения системы из среды стало получения ряда тезисов, касающихся будущей системы:

- Поток данных необходимо брать с ПЛИС в составе ячейки БОС
- ПЛИС ячейки имеет связь с 16 выводами на корпусе блока
- Регистратор должен быть небольшого размера
- Регистратор независим и самодостаточен
- Регистратор должен использовать общедоступные технологии передачи данных

Учитывая эти факты, на этапе концептуального проектирования была разработана архитектура системы, структурная схема которой изображена на Рис. 6. Непосредственно регистратор выделен на схеме пунктирным прямоугольником.

Состав системы:

Модуль для ПЛИС, задача которого привести поток данных к интерфейсу драйвера физического уровня и передать на него данные.

Среда передачи данных, состоящая из двух драйверов физического уровня для приемной и передающей стороны соответственно и кабеля, по которому передаются данные.

Компьютер с жестким диском и программой, управляющей приемом и записью на диск. В качестве компьютера можно использовать ноутбук, что придаст системе мобильность, компактность и благодаря встроенной батарее — энергонезависимость. Или портативное устройство на базе сигнального процессора под управлением ОС Linux, обладающее нужной средой передачи данных, драйверами и жестким диском и обрабатывающее в цикле программу регистрации.

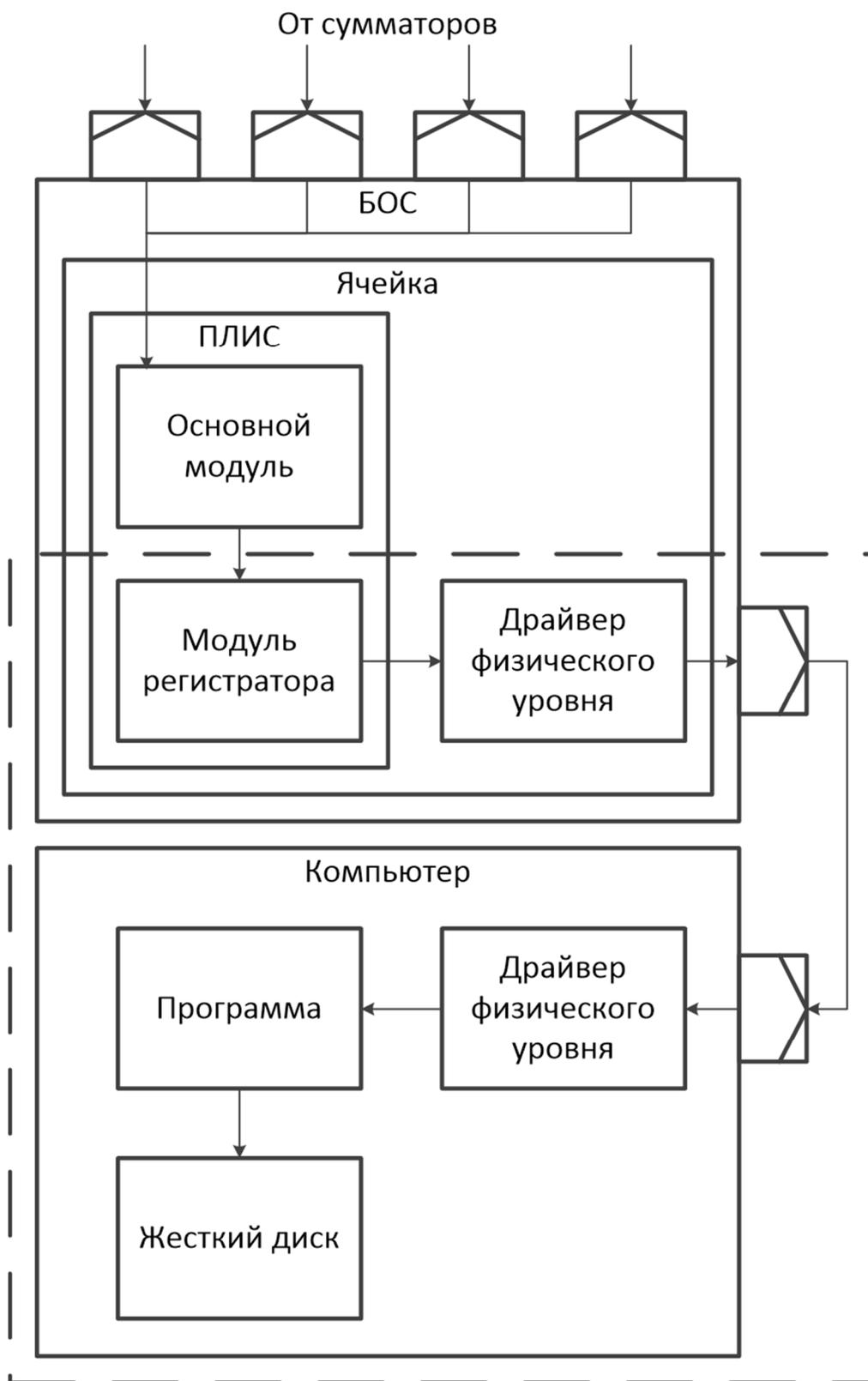


Рис. 6. Структурная схема регистратора

3.4. Разработка укрупненной функциональной структуры системы.

Как уже было сказано ранее, задача регистратора — запись входного потока данных БОС на жесткий диск, что обеспечит возможность в дальнейшем воспроизводить выбранный сеанс на отладочном стенде. Для полученной на предыдущих этапах структуры системы была разработана функциональная модель, выполненная по методологии IDEF0.

Укрупненная функциональная структура изображена на Рис. 7 и Рис. 8. Более подробная декомпозиция представлена на листе 5 графического материала дипломного проекта.

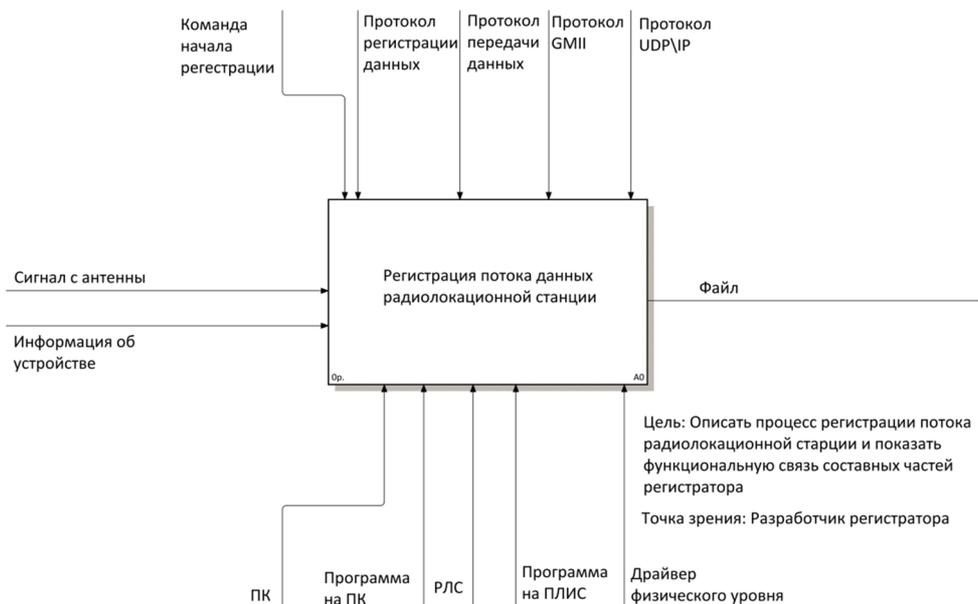


Рис. 7. Функциональная схема регистратора

На структуре системы, изображенной на Рис. 6, прослеживается, насколько тесно интегрирован регистратор в блок обработки сигналов. Функциональная схема показывает в свою очередь и функциональную интеграцию.

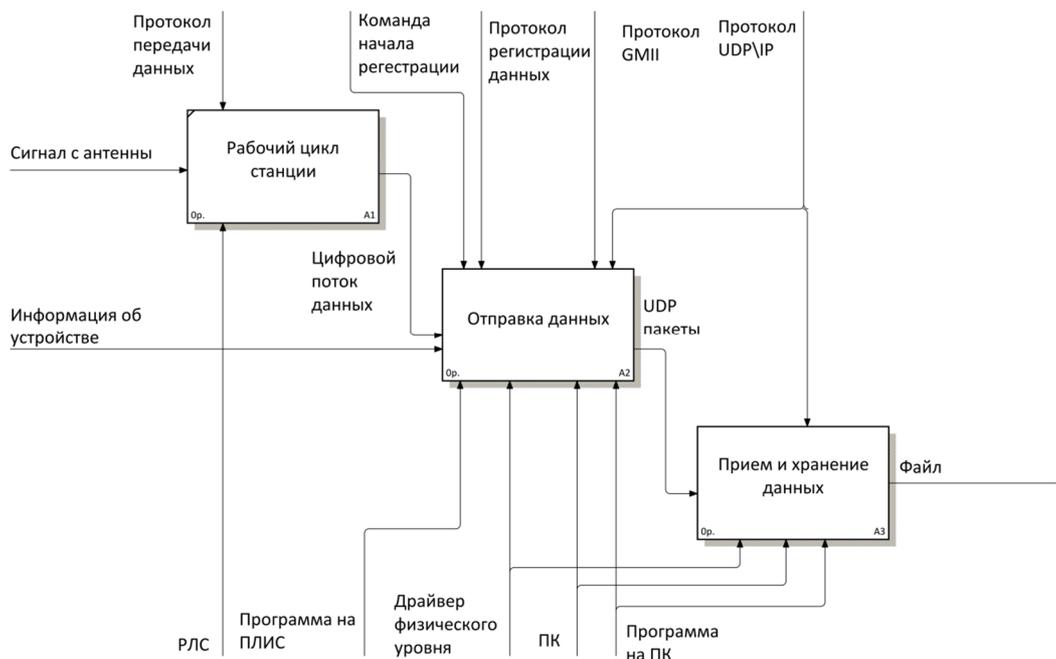


Рис. 8. Функциональная схема регистратора

3.5. Выбор концепции.

На этапе концептуального проектирования удалось определиться с приблизительной архитектурой и функциональной структурой системы. Однако окончательная концепция зависит от выбора среды передачи данных и приемной части (портативного компьютера).

3.5.1. Обзор внешних интерфейсов компьютера.

В рамках концептуального проектирования были рассмотрены большинство внешних интерфейсов. Следует сразу отметить, что интерфейсы ISA, EISA, VLB, AGP, COM, LPT не рассматриваются подробно из-за маленькой скорости и морального устаревания технологий. Кроме того, интерфейсы HDMI, VGA, DVI, DisplayPort, не смотря на их развитие и большую скорость передачи данных, не рассматриваются из-за их специфичности. Перечисленные интерфейсы предназначены для передачи видеосигнала и обрабатываются графической памятью компьютера, считается нецелесообразным использование GPU для записи

на жесткий диск при доступности универсальных интерфейсов, которые будут рассмотрены ниже.

USB или **Universal Serial Bus** — «универсальная последовательная шина» - последовательный интерфейс передачи данных.[3] Один из самых распространенных на сегодняшний день внешних интерфейсов компьютера.

Из-за распространённости интерфейса, существует большое количество библиотек для работы с USB, драйверов, контролеров и кабелей. Большая конкуренция на этом рынке повлияла и на цены. Однако следует учитывать, что сейчас распространены две версии спецификации 2.0 и 3.0. Отличается старшая версия от младшей тем, что используются на четыре линии связи больше, а значит другие драйверы, контроллеры и кабели. Это изменение спецификации в 2008 году дало десятикратный прирост скорости (с 480 Мб/с для версии 2.0 до 5 Гб/с для версии 3.0), а вместе с тем примерно такой же прирост сложности разработки систем.

PCI Express, или **PCIe**, **PCI-E** — компьютерная шина, хотя на физическом уровне шиной не является, будучи соединением типа «точка-точка», использующая программную модель шины **PCI** и высокопроизводительный физический протокол, основанный на последовательной передаче данных.[4]

Доступны 5 способов подключения x1, x4, x8, x16, x32, от этого зависит максимальная скорость передачи, которая достигает 40 Гб/с, однако доступный на портативных компьютерах разъем mini PCIe соответствует PCIe x1, максимальная скорость передачи для которого составляет 2.5 Гб/с. Несмотря на повсеместное использование PCIe, открытых проектов с его использованием мало, что затрудняет разработку драйвера и программы (как со стороны ПЛИС, так и со стороны компьютера). Серьёзным недостатком PCIe является ограничение на длину кабеля.

Thunderbolt (ТВ) — аппаратный интерфейс, ранее известный как Light Peak, разработанный компанией Intel в сотрудничестве с Apple, для подключения периферийных устройств к компьютеру.[5]

Максимальная скорость передачи данных до 10 Гбит/сек по медному проводу и до 20 Гбит/сек при использовании оптического кабеля.

Thunderbolt комбинирует интерфейсы PCIe и DisplayPort в одном кабеле, поэтому сложность разработки выше чем PCIe, которая при этом не дает преимуществ, кроме компактного разъема. Поэтому Thunderbolt в дальнейшем не рассматривается.

SATA (Serial ATA) — последовательный интерфейс обмена данными с накопителями информации[6]. И **eSATA** (External SATA) — интерфейс подключения внешних устройств, поддерживающий режим «горячей замены». [6]

eSATA отнюдь не гибкий интерфейс, поэтому подключение платы ячейки с ПЛИС к компьютеру по этому интерфейсу возможна, однако, требует внесения очень больших изменений в плату. Фактически при этом на плату потребуется установить жесткий диск, подключаемый к компьютеру для передачи информации. При этом будут потрачены большие ресурсы платы, неиспользуемые при этом в рабочих циклах станции.

Возможен и другой вариант подключения, а именно жесткий диск без компьютера к плате ячейки. Это потребует добавление в конструкцию платы компоненты, отвечающие за физический, канальный, транспортный и прикладной уровни. Кроме того, это потребует разводки дополнительной силовой линии под жесткий диск и другие значительные изменения.

IEEE 1394 (FireWire) — последовательная высокоскоростная шина, предназначенная для обмена цифровой информацией между компьютером и другими электронными устройствами.[7]

Спецификация S3200 позволяет передавать данные на скорости до 3.2 Гб/с. Однако стандарт не нашел широкого распространения, что влечет за собой высокую стоимость компонентов и разработки, высокую сложность реализации драйверов и программной части.

Ethernet — семейство технологий пакетной передачи данных для компьютерных сетей, которое описывается стандартами IEEE группы 802.3 [8], описывающими соединение на скорости до 40 Гб/с, однако общедоступными до сих пор являются лишь физические устройства по стандартам, поддерживающим скорость до 10 Гб/с, а самыми распространенными остаются устройства, работающие на скорости 1 Гб/с

Для окончательного выбора концепции был произведен анализ представленных интерфейсов. Критерии и оценки указаны в таблице.

Таблица 3. Характеристики интерфейсов

	USB	PCIe	Ethernet	IEEE 1394	SATA
Максимальная скорость	5 Гб/с 4/5	40 Гб/с 5/5	10 Гб/с 5/5	3.2 Гб/с 4/5	6 Гб/с 5/5
Скорость передачи данных реальных проектов	5 Гб/с 5/5	2.5 Гб/с 5/5	1 Гб/с 4/5	400 Мб/с 3/5	3 Гб/с 5/5
Распространённость	5/5	5/5	5/5	2/5	4/5
Простота написания драйвера	3/5	2/5	5/5	1/5	2/5
Простота написания программы	4/5	3/5	5/5	3/5	2/5
Стоимость кабеля	5/5	4/5	5/5	4/5	4/5
Стоимость изменения платы	4/5	3/5	5/5	2/5	2/5
Итого	30/35	27/35	34/35	19/35	24/35

По итогам анализа был выбран интерфейс Ethernet

3.6. Окончательный выбор концепции.

Мы живем в то замечательное время, когда даже некоторые модели чайников работают под управлением Unix-подобных операционных систем.

В самом деле, на сегодняшний день ядро Linux собирается не только для процессоров ПК и серверов, но и для архитектур ARC, ARM, AVR32, Alpha AXP, SPARC, ETRAX CRIS, SuperH, PA-RISC, FR-V, H8300, IP7000, m68k, MIPS, mn10300, PowerPC, Xtensa и многих других [9].

Выбранный подход к проектированию, а также среда передачи данных позволяет разрабатывать специализированные устройства или использовать уже готовые (ноутбуки, портативные компьютеры). Единственное требование к принимающей стороне регистратора - наличие контроллера физического уровня стандарта IEEE 802.3ab (Гигабитный Ethernet) и наличие жесткого диска, способного записать такой объем информации. Предлагается использовать SSD диски, подключенные по интерфейсу SATA. Операционная система не несет большой роли, принято считать, для что Unix-подобных ОС проще разрабатывать ПО и драйверы, чем под Windows¹, однако при выбранной концепции драйверы чаще всего поставляются вместе в системой независимо от архитектуры процессора. Тем не менее, в рамках дипломного проекта необходимо выбрать концепцию. Так как в распоряжении ВНИИРТ имеются ноутбуки под управлением ОС Windows, окончательно выбираем его в качестве компонента подсистемы приема и хранения информации регистратора. Следует отметить, что разрабатываемая система гибкая и не требовательная в плане приёмной части.

¹ Лёгкость написания драйверов объясняется тем, что исходный код Linux хорошо задокументирован и доступен в виде исходного кода, а в сети множество примеров и литературы на тему разработки драйверов под Linux. Для создания драйвера необходим только дистрибутив и исходники ядра и компилятор под нужную архитектуру.

В итоге получим окончательную концепцию, структурная и функциональные схемы которой были разработаны ранее на этапе концептуального проектирования:

- Модуль для ПЛИС, тесно интегрированный с основной прошивкой и предназначенный для передачи данных
- Драйвер физического уровня Ethernet для преобразования уровней, соединенный с ПЛИС и разъемом на корпусе БОС
- Кабель для соединения драйверов физического уровня между собой
- Компьютер и входящие в его состав:
 - Ответный драйвер физического уровня
 - Жесткий диск для хранения потока данных
 - Программа для записи данных с драйвера на диск

4. Формирование технического задания.

4.1. Общие сведения.

4.1.1. Наименование системы.

4.1.1.1. Полное наименование системы.

Автоматизированная система регистрации потока данных радиолокационной станции

4.1.1.2. Краткое наименование системы.

Регистратор потока данных РЛС, регистратор

4.1.2. Основания для проведения работ.

Договор на выполнение разовой работы №160, именуемый в дальнейшем «Договор»

4.1.3. Наименование организаций — Заказчика и Разработчика

ОАО «Всероссийский научно-исследовательский институт радиотехники» (ОАО «ВНИИРТ»), именуемый в дальнейшем «Заказчик», Чернов Алексей Олегович, далее именуемый «Разработчик»

4.1.4. Плановые сроки начала и окончания работы.

Сроки начала работ: 03.08.2014

Сроки окончания работ: 31.12.2014

4.1.5. Порядок оформления и предъявления заказчику результатов работ.

Работы по созданию регистратора сдаются Разработчиком поэтапно в соответствии с календарным планом Проекта. По окончании каждого из этапов работ Разработчик сдает Заказчику соответствующие отчетные документы этапа, состав которых определен Договором.

4.2. Назначение и цели создания системы.

4.2.1. Назначение системы.

Задачей регистратора является запись потока данных от блока обработки сигналов (БОС) на жесткий диск в фоновом режиме работы РЛС.

4.2.2. Цели создания системы.

Целью создания регистратора является получение возможности записи и хранения потока данных от блока обработки сигналов РЛС на жесткий диск для дальнейшей обработки и воспроизведения на отладочном стенде. Это необходимо для отладки в случае отказа оборудования и имитации режимов работы, полученных во время испытаний станции в полевых условиях, что позволит проверять часть нововведений, не выходя из здания института и не организовывая дорогостоящих выездов на полигон.

4.3. Характеристика объектов автоматизации.

4.3.1. Краткие сведения об объекте автоматизации.

Все радиолокационные данные станции объединяются в единый поток в блоке обработки сигналов, используя программируемые логические интегральные схемы (ПЛИС) ячейки блока обработки сигналов. Подсистема передачи данных интегрирована с этой ПЛИС.

Выбранная концепция системы подразумевает гибкость приемной части системы. В качестве устройства регистрации может выступать плата под управлением цифрового сигнального процессора или ПК. Оба устройства могут управляться Unix-подобной операционной системой. Так как система разворачивается на мощностях Заказчика, в качестве подсистемы приема и хранения данных используется ноутбук под управлением ОС Windows.

Для автоматизированной регистрации данных от системы требуется отправление данных с БОС РЛС (подсистема передачи данных) и прием с последующей записью на устройстве с жестким диском (подсистема приема и хранения).

Таблица 4. Сведения о подсистемах

Подсистема	Наименование процесса	Возможность автоматизации	Решение об автоматизации в ходе проекта
Передачи данных	Отправление потока данных	Возможна	Будет автоматизировано
Приема и хранения данных	Прием данных и запись на диск	Возможна	Будет автоматизировано

4.3.2. Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.

Синусоидальная вибрация:

Амплитуда ускорения, м/с^2 - 20

Диапазон частот, Гц - 1-60

Механический удар одиночного действия:

Пиковое ударное ускорение, м/с^2 (g) - По ТТЗ (ТЗ БОС)

Длительность действия ударного ускорения, мс - 0,5-2

Качка:

Амплитуда качки, град - ± 45

Период, с - 7-16

Максимальный угол наклона, град

длительный - 15

кратковременный (3 мин) - 30

Повышенная температура среды:

Рабочая, °С - 40

Предельная, °С - 70

Пониженная температура среды

Рабочая, °С - 0

Предельная, °С - минус 50

Диапазон изменение температуры среды:

От пониженной предельной до повышенной предельной температуры среды

Повышенная влажность воздуха

Относительная влажность, % - 98

Температура, °С - 35

Соляной (морской) туман:

По ГОСТ РВ 20.57.306

Агрессивные среды, Концентрация, мг/м³:

Сернистый газ - 2,0

аммиак - 1,0

двуокись азота - 2,0

сероводород - 1,0

4.4. Требования к системе.

4.4.1. Требования к системе в целом.

Система должна представлять собой программно-аппаратный комплекс, позволяющий записывать данные с ПЛИС на жесткий диск.

- Модуль на языке описания аппаратуры для прошивки ПЛИС БОС
- Программное обеспечение для приема и записи данных и аппаратная часть для связи этих двух подсистем.

4.4.1.1. Требования к структуре и функционированию системы.

4.4.1.1.1. Перечень подсистем, их назначение и основные характеристики.

Подсистема передачи данных, предназначенная для отправки потока РЛС с ячейки блока обработки сигналов, должна иметь пропускную способность не менее 800 Мб/с

Сердцем этой подсистемы является ПЛИС ячейки, на которой она базируется, основные характеристики ПЛИС Xilinx XC6SLX150t семейства Spartan-6 приведены на этапе концептуального проектирования. Она используется также и в боевой работе станции, поэтому программная часть подсистемы передачи данных должна занимать ограниченное количество ресурсов ПЛИС.

Подсистема приема и хранения данных, представленная в виде программно-аппаратного комплекса, предназначена для приема данных на стороне компьютера с последующей записью на диск.

Характеристики компьютера должны обеспечивать прием и запись потока, скорость составляет 800 Мб/с. Характеристики компьютера, имеющегося в распоряжении Заказчика:

Процессор:	Intel Core i3 4005U 1700 МГц
Оперативная память:	DDR3L 4 ГБ
Жесткий диск:	SSD 500ГБ
Адаптер LAN:	1000 Мб/с

4.4.1.1.2. Требования к способам и средствам связи для информационного обмена между компонентами системы.

Интерфейс между верхним уровнем модуля и драйвером Ethernet — GMI.

Передачи данных на физическом уровне от драйвера к устройству регистрации — экранированный кабель Cat. 5e.

Интерфейс подключения жесткого диска SATA, тип накопителя SSD.

4.4.1.1.3. Требования к характеристикам взаимосвязей создаваемой системы со смежными системами.

Интерфейс между верхним уровнем модуля и остальной частью прошивки (входящий поток данных), должен иметь вид, приведенный на этапе технического проектирования.

4.4.1.1.4. Требования к режимам функционирования системы.

Система имеет один режим работы, однако должна быть гибкой и легко настраиваемой для нужд тестирования и дальнейшего использования на других станциях.

4.4.1.1.5. Требования по диагностированию системы.

Не предъявляются.

4.4.1.1.6. Перспективы развития, модернизации системы.

В перспективе развития системы, предлагается использовать гарантированную доставку, используя протокол передачи данных TSP или др.

При модернизации радиолокационной станции и возрастании скорости потока данных предлагается использовать драйвер физического уровня стандарта IEEE 802.3ae, работающий на скорости до 10 Гб/с, на стороне БОС и соответствующее сетевое оборудование на стороне подсистемы приема и хранения данных.

4.4.1.2. Требования к численности и квалификации персонала системы.

4.4.1.2.1. Требования к численности персонала.

В составе персонала, необходимого для обеспечения эксплуатации разрабатываемой системы, достаточно одного человека — оператора регистратора, который обладает следующими обязанностями:

- Подключение принимающей стороны регистратора к БОС
- Запуск процесса регистрации в программе, входящей в состав комплекс системы
- После завершения необходимого количества рабочих циклов станции, отключение процесса регистрации и последующее отсоединение принимающей стороны

4.4.1.2.2. Требования к квалификации персонала.

Оператор должен обладать знаниями предметной области, базовыми навыками работы с ПК.

Обладать доступом к секретной информации по третьей форме.

4.4.1.3. Показатели назначения.

4.4.1.3.1. Параметры, характеризующие степень соответствия системы назначению.

Система должна обеспечивать следующие количественные показатели, которые характеризуют степень соответствия ее назначению:

- Размер модуля ПЛИС согласно Таблица 2:
- Достигаемая скорость не менее 800 Мб/с

4.4.1.3.2. Требования к приспособляемости системы к изменениям.

Обеспечение приспособляемости системы должно выполняться за счет:

- Гибкости системы и её универсальности
- Модернизации или замены частей системы в соответствии с новыми требованиями

4.4.1.3.3. Требования к сохранению работоспособности системы в различных вероятных условиях.

Таблица 5. Требования к сохранению работоспособности

Вероятное условие	Требование
Нарушения в работе системы внешнего электроснабжения продолжительностью до 4 часов	Функционирование в полном объеме.
Выход из строя составных частей	Не предъявляется

4.4.1.4. Требования к надежности.

Надежность системы осуществляется за счет надежности отдельных ее компонентов. Особое внимание уделяется компонентам подсистемы передачи данных, ведь она тесно интегрирована в БОС и должна выполнять свои функции даже после рабочих циклов в боевых условиях. Все компоненты, установленные на плату ячейки, должны пройти приемку 5, климатические, механические и другие испытания согласно ТУ БОС.

4.4.1.5. Требования к безопасности.

При внедрении, эксплуатации и обслуживании технических средств системы должны выполняться меры электробезопасности в соответствии с

«Правилами устройства электроустановок» и «Правилами техники безопасности при эксплуатации электроустановок потребителей».

Аппаратное обеспечение системы должно соответствовать требованиям пожарной безопасности в производственных помещениях по ГОСТ 12.1.004-91. «ССБТ. Пожарная безопасность. Общие требования».

Должно быть обеспечено соблюдение общих требований безопасности в соответствии с ГОСТ 12.2.003-91. «ССБТ. Оборудование производственное. Общие требования безопасности» при обслуживании системы в процессе эксплуатации.

4.4.1.6. Требования к эргономике и технической эстетике.

Интерфейсы модуля передающей части регистратора должны быть простыми и понятными.

Все требуемые взаимодействия оператора с программой должны сопровождаться визуальными подсказками и однозначными комментариями.

Аппаратная часть подсистемы приема и хранения данных должна быть легкой и компактной, при этом позволять обладать комфортными для работы экраном и устройствами ввода.

4.4.1.7. Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы.

Условия эксплуатации, а также виды и периодичность обслуживания технических средств Системы должны соответствовать требованиям по эксплуатации, техническому обслуживанию, ремонту и хранению, изложенным в документации завода-изготовителя (производителя) на них.

Для обеспечения выполнения требований по надежности должен быть использован существующий комплект запасных изделий и приборов (ЗИП), состав, место и условия хранения описан в ТУ БОС.

4.4.1.8. Требования к защите информации от несанкционированного доступа.

Выполнение требований к защите информации обеспечивается проверкой компонентов подсистемы передачи данных на наличие аппаратных закладок.

В качестве принимающей стороны используются только устройства, допустимые к работе с секретной информации.

4.4.1.9. Требования по сохранности информации при авариях.

Не предъявляются.

4.4.1.10. Требования к защите от влияния внешних воздействий.

К компонентам подсистемы передачи данных предъявляются требования согласно ТУ платы ячейки, на которой они установлены.

Остальные компоненты системы должны быть сертифицированы и быть устойчивы к внешним воздействиям.

Система должна иметь возможность функционирования в диапазоне допустимых температур окружающей среды, установленных изготовителем аппаратных средств.

Система должна иметь возможность функционирования в диапазоне допустимых значений влажности окружающей среды, установленных изготовителем аппаратных средств.

Система должна иметь возможность функционирования в диапазоне допустимых значений вибраций, установленных изготовителем аппаратных средств.

4.4.1.11. Требования к патентной чистоте.

Не предъявляются.

4.4.1.12. Требования по стандартизации и унификации.

При изготовлении системы должны использоваться компоненты из списка допустимых при наличии их в этом списке.

Остальные компоненты, в том числе программные, должны быть стандартизированы и иметь доступную спецификацию.

4.4.2. Требования к функциям (задачам), выполняемым системой.

4.4.2.1. Подсистема передачи данных.

4.4.2.1.1. Перечень функций и задач, подлежащих автоматизации.

Таблица 6. Перечень функций и задач

Функция	Задача
Передача данных	Сбор информации посредством взаимодействия с основным модулем ПЛИС
	Согласование интерфейсов и обрамление заголовком UDP
	Передача данных

4.4.2.1.2. Требования к качеству реализации функций, задач

Таблица 7. Требования к задачам

Задача	Требования
Сбор информации посредством взаимодействия с основным модулем ПЛИС	Не предъявляются
Согласование интерфейсов и обрамление заголовком UDP	Соответствие протоколу UDP
Передача данных	Достижимая скорость 800Мб/с

4.4.2.1. Подсистема приема и хранения данных.

4.4.2.1.1. Перечень функций и задач подлежащих автоматизации.

Таблица 8. Перечень функций и задач

Функция	Задача
Прием данных	Прием данных с драйвера физического уровня
	Передача данных в буфер файла
	Ведение статистики
Хранение данных	Запись данных в файл

4.4.2.1.3. Требования к качеству реализации функций, задач

Таблица 9. Требования к задачам

Задача	Требования
Прием данных с драйвера физического уровня	Достижимая скорость 800Мб/с
Передача данных в буфер файла	Не предъявляются

Задача	Требования
Ведение статистики	Отображение объема записываемых данных, скорости передачи данных и времени работы
Запись данных в файл	Достижимая скорость 800Мб/с

4.4.3. Требования к видам обеспечения.

4.4.3.1 Требования к математическому обеспечению системы.

Не предъявляются.

4.4.3.2. Требования информационному обеспечению системы.

Для выполнения эксплуатационного назначения системе требуется реализовать информационный поток:

- Согласование интерфейсов между основным рабочим модулем ПЛИС и драйвером физического уровня
- Передача между драйверами физического уровня
- Сохранение информации, полученной с драйвера физического уровня, на жесткий диск

Интерфейс информационного обмена между компонентами системы разрабатывается на этапе технического проектирования.

4.4.3.3. Требования к лингвистическому обеспечению системы.

При реализации подсистемы передачи должны применяться языки высокого уровня описания аппаратуры интегральных схем: Verilog и/или VHDL.

При реализации подсистемы приема и хранения информации должны применяться языки программирования высокого уровня: C++, Java и/или др.

4.4.3.4. Требования к программному обеспечению системы.

Не предъявляется.

4.4.3.5. Требования к техническому обеспечению.

Подсистема передачи данных должна быть интегрирована в БОС.

4.4.3.6. Требования к метрологическому обеспечению.

Не предъявляется.

4.4.3.7. Требования к организационному обеспечению.

Не предъявляется.

4.4.3.8. Требования к методическому обеспечению.

Не предъявляется.

4.5. Состав и содержание работ по созданию системы.

Работы по созданию системы выполняются согласно Договору.

4.6. Порядок контроля и приемки системы.

4.6.1 Виды, состав, объем и методы испытаний системы.

Виды, состав, объем и методы испытаний подсистемы передачи данных аналогичны испытаниям платы, на которой она установлена, и описаны в ТУ на БОС.

Система в целом подвергается испытаниям следующих видов:

1. Приемо-сдаточные испытания.

2. Предварительные испытания.
3. Опытная эксплуатация.

4.6.2. Общие требования к приемке работ по стадиям.

Таблица 10. Стадии приемки работ

<p>Приемо-сдаточные испытания</p>	<p>Проведение приемо-сдаточных испытаний. Фиксирование выявленных неполадок в Протоколе испытаний. Устранение выявленных неполадок. Проверка устранения выявленных неполадок. Принятие решения о возможности передачи системы к предварительным испытаниям. Составление и подписание Акта приёмки системы к предварительным испытаниям.</p>
<p>Предварительные испытания</p>	<p>Проведение предварительных испытаний. Фиксирование выявленных неполадок в Протоколе испытаний. Устранение выявленных неполадок. Проверка устранения выявленных неполадок. Принятие решения о возможности передачи системы в опытную эксплуатацию. Составление и подписание Акта приёмки системы в опытную эксплуатацию.</p>
<p>Опытная эксплуатация</p>	<p>Проведение опытной эксплуатации. Фиксирование выявленных неполадок в Протоколе испытаний. Устранение выявленных неполадок. Проверка устранения выявленных неполадок. Составление и подписание Акта о завершении опытной эксплуатации системы. Составление и подписание Акта о завершении приемочных испытаний и передаче системы в эксплуатацию. Оформление Акта завершения работ.</p>

4.7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.

4.7.1. Технические мероприятия.

Внесение изменений в конструкцию и ввод в эксплуатацию платы ячейки БОС.

Закупка и проверка покупных изделий.

4.7.2. Организационные мероприятия.

Заказчик обязан обеспечить следующие мероприятия:

- Организация доступа к БОС
- Определение регламента работы оператора регистратора
- Выделение ответственных специалистов для взаимодействия с Разработчиком по вопросам взаимодействия с системами-источниками данных.

5. Технический этап проектирования.

На техническом этапе проектирования разрабатывается основная документация проектируемой системы, её подсистем и модулей.

На предыдущих этапах были разработаны архитектура системы и её укрупненная функциональная модель, что является входной информацией для этого этапа.

На этапе технического проектирования была разработана диаграмма развертывания (см. Рис. 9), в которой уточняется структура системы. Для соединения модуля регистратора на ПЛИС ячейки и драйвера физического уровня был выбран интерфейс GMII.

GMII (Gigabit Media Independent Interface) — независимый от среды передачи интерфейс для гигабитных Ethernet интерфейсов, обеспечивающий передачу данных между устройствами, реализующими подуровень MAC канального уровня с устройствами, реализующими физический уровень (PHY) модели OSI [10] (Таблица 11). [11]

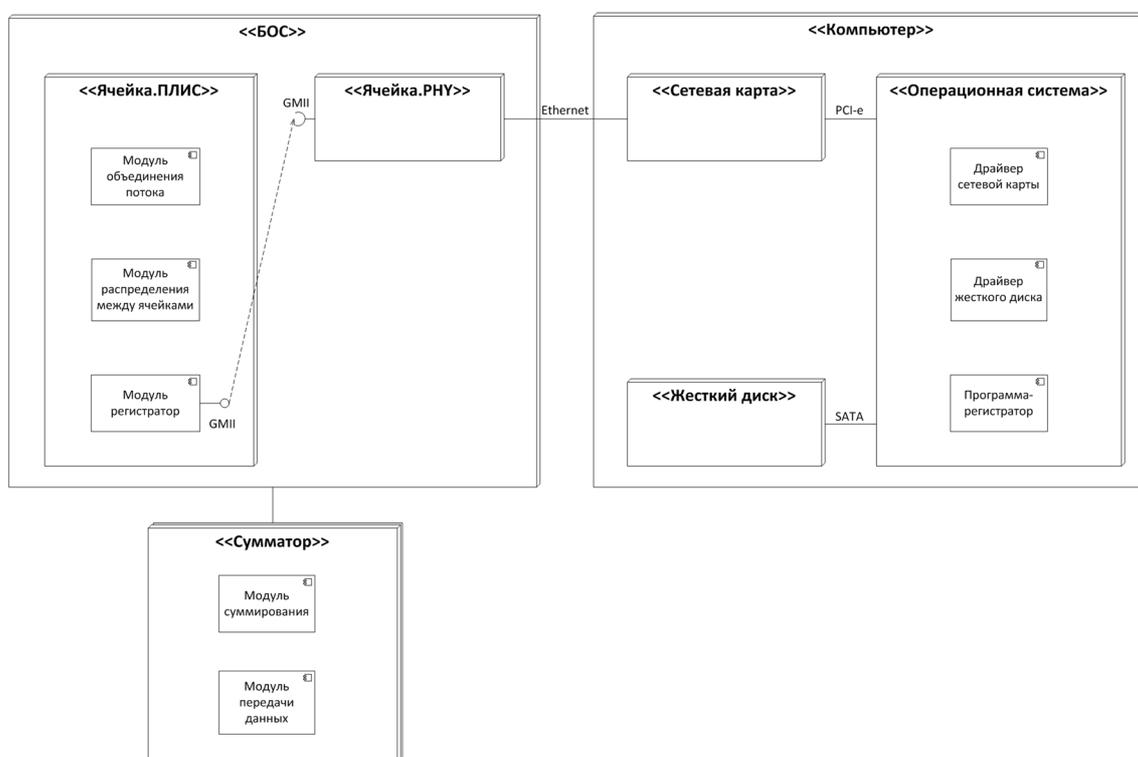


Рис. 9. Диаграмма развертывания системы регистрации

Таблица 11. Модель OSI

Модель OSI			
Уровень (layer)		Тип данных	Функции
Host layers	7. Прикладной (application)	Данные	Доступ к сетевым службам
	6. Представительский (presentation)		Представление и шифрование данных
	5. Сеансовый (session)		Управление сеансом связи
	4. Транспортный (transport)	Сегменты(segment)	Прямая связь между конечными пунктами и надежность
Media layers	3. Сетевой (network)	Пакеты(packet) Датаграммы(datagram)	Определение маршрута и логическая адресация
	2. Звена данных (канальный)(data link)	Биты (bit) Кадры (frame)	Физическая адресация
	1. Физический (physical)	Биты (bit)	Работа со средой передачи, сигналами и двоичными данными

На предыдущих этапах был выбран интерфейс Ethernet в качестве среды передачи данных. Следует напомнить, что Ethernet — это семейство технологий пакетной передачи, которое описывается стандартами IEEE группы 802.3, представление этого стандарта относится к канальному уровню модели OSI и охватывает функции работы со средой передачи данных и физической адресации.

В таком случае программная часть системы должна реализовывать сетевой, транспортный, сеансовый, представительский и прикладные уровни. На сегодняшний день имеется множество стандартизованных протоколов передачи данных, реализующих транспортный уровень модели OSI, отвечающий, в том числе, за прямую связь между конечными пунктами.

Стек протоколов **TCP/IP** — набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет. Название TCP/IP происходит из двух протоколов семейства — Transmission Control Protocol (TCP) и Internet Protocol (IP). [12 **Ошибка! Источник ссылки не найден.**]

Стек протоколов (англ. stack) означает, что протокол, располагающийся на уровне выше, работает поверх нижнего. При этом используются механизмы инкапсуляции. Например, протокол TCP работает поверх протокола IP.

Стек протоколов TCP/IP включает в себя три уровня:

- транспортный уровень (transport layer) TCP,
- сетевой уровень (network layer) IP,
- канальный уровень (link layer) Ethernet.

UDP (User Datagram Protocol — протокол пользовательских дейтаграмм) — набор сетевых протоколов передачи данных. Как и TCP/IP является стеком технологий с тем же сетевым уровнем IP. С UDP компьютерные приложения могут посылать дейтаграммы другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных. [13]

UDP использует простую модель передачи, без неявных «рукопожатий» для обеспечения надёжности, упорядочивания или целостности данных. UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении.

Чувствительные ко времени приложения часто используют UDP, так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся пакеты. В рамках решаемой задачи использовать полезное отличие TCP/IP от UDP в виде гарантированной доставки не представляется возможным. Гарантированная доставка требует значительных буферов данных, реализация которых на стороне подсистемы передачи данных затруднительна, переполнение буферов к тому же все равно влечет за собой потерю данных. Необходимость ожидания ответов, дополнительные «рукопожатия» влекут за собой потерю полезной скорости передачи данных. Все вышеперечисленное, а также сложность реализации TCP/IP по сравнению с UDP, делает последнюю предпочтительней.

SCTP (Stream Control Transmission Protocol — протокол передачи с управлением потоком) — протокол транспортного уровня в компьютерных сетях. [14]

Будучи более новым протоколом, SCTP имеет несколько нововведений, таких как многопоточность, защита от ddos атак, синхронное соединение между двумя хостами по двум и более независимым физическим каналам (multi-homing). Эти нововведения, однако, не применительны к решаемой задаче.

SPX (Sequenced Packet Exchange) — протокол транспортного уровня модели OSI. В отличие от остальных вариантов в стеке протоколов SPX на сетевом уровне находится IPX, разработанный компанией «Novell» для её проприетарной сетевой ОС NetWare [15], поэтому применение его в разрабатываемой системе затруднительно.

После выбора UDP в качестве протокола передачи данных, на этапе технического проектирования разрабатываются:

- Схема электрическая принципиальная платы ячейки, после внесения необходимых изменений

- Модульная структура программной части подсистемы передачи данных, реализующая подуровень МАС
- Интерфейсы между модулями программной части подсистемы передачи данных
- Алгоритм программной части подсистемы приема и хранения данных

5.1. Разработка алгоритма программной части подсистемы приема и хранения данных.

На Рис. 10 изображена блок-схема разработанного алгоритма программной части подсистемы приема и хранения данных. В алгоритме учтены наработки предыдущих этапов разработки системы. Взаимодействие оператора с программой минимально, так как в алгоритме был достигнут высокий уровень автоматизации. Все предполагаемые взаимодействия с пользователем сопровождаются достаточной информацией.

Блок-схема описывает работу подсистемы, которая должна выполнять прием и хранение информации. Прием данных и их запись на диск производится в цикле, прерываемом оператором.

Простота алгоритма достигается использованием библиотек работы с сокетами, позволяющих не реализовывать сетевой и транспортный уровни модели OSI, а сконцентрировать внимание разработчика на разработке удобного инструмента. Канальный уровень реализован программным драйвером, входящим в состав операционной системы, а физический с помощью драйвера физического уровня, входящим в состав целевого для программы компьютера.

Программа реализует недостающие для модели OSI уровни:

- Прикладной (application)

- Представительский (presentation)
- Сеансовый (session)

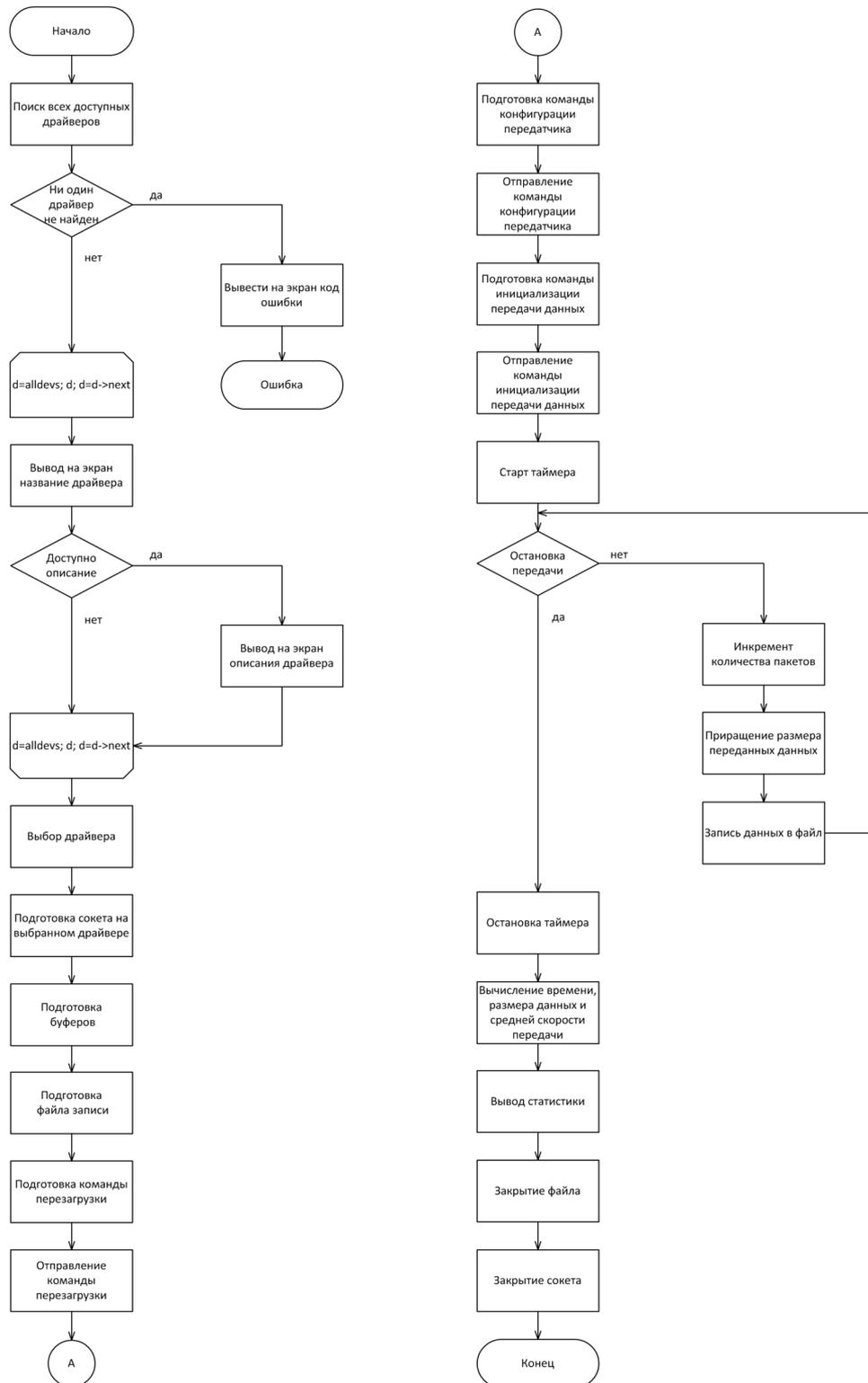


Рис. 10. Блок-схема алгоритма работы программы.

Структура кадра показывает, что для согласования с протоколом UDP, данные необходимо обрамлять заголовками разного уровня, причем некоторые из этих полей статичны:

- Preamble
- Ethernet Type
- Ver.
- IHL
- Service
- Identification
- TTL
- Protocol

Некоторые зависят от параметров подключения:

- Destination MAC Address
- Source MAC Address
- Source IP Address
- Destination IP Address
- Source Port
- Destination Port

А другие — от передаваемых данных:

- Total Length
- Header CRC
- Length
- Data
- CRC

Модули, необходимые для решения задачи, и их укрупнённая структура изображены на Рис. 11

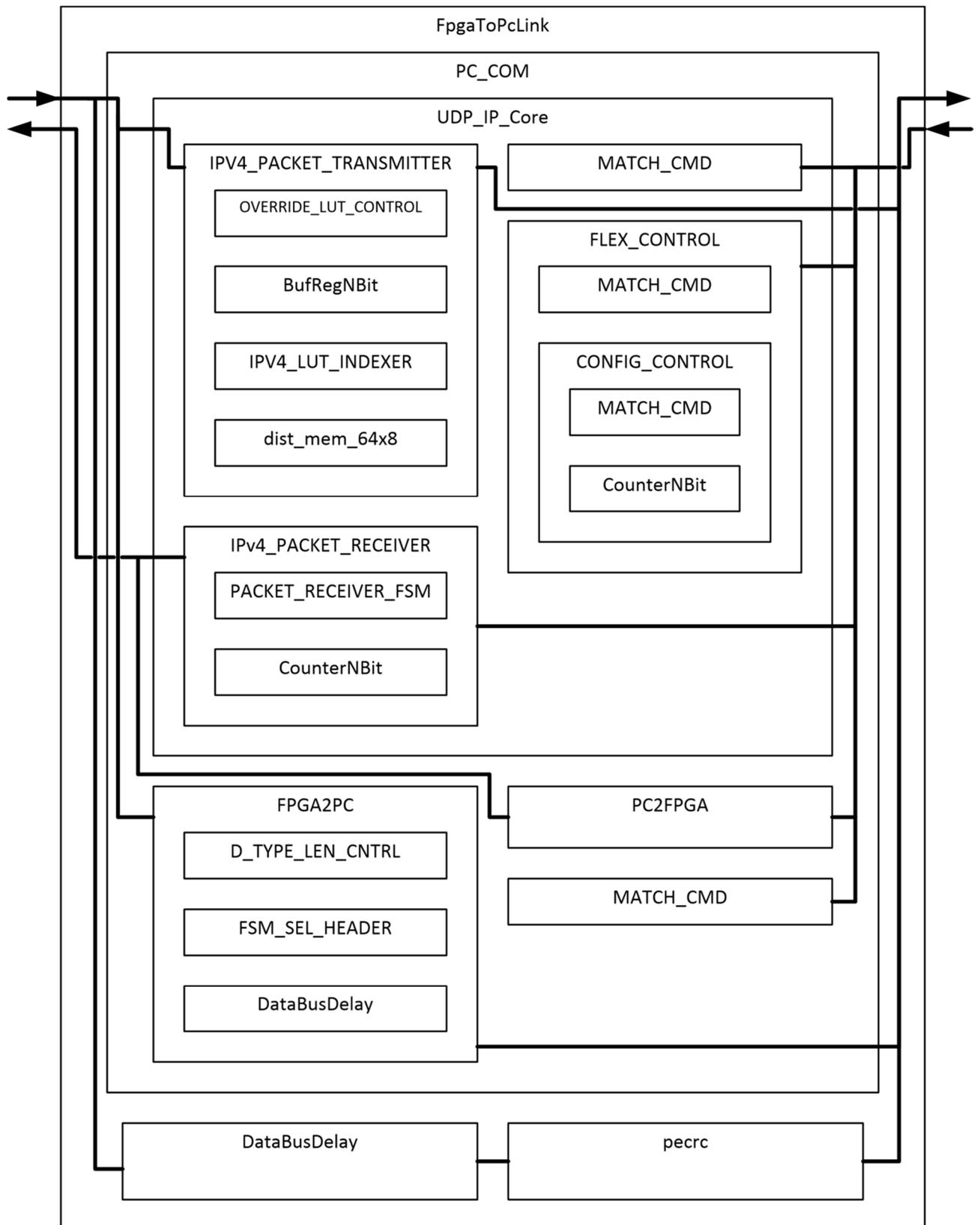


Рис. 11. Структура модулей ПЛИС

FpgaToPcLink — модуль верхнего уровня, согласующий интерфейсы между основным модулем ПЛИС ячейки БОС и драйвером физического уровня. Модуль также обрамляет UDP пакеты Ethernet преамбулой и CRC.

Исходя из назначения модуля, была разработана диаграмма состояния.

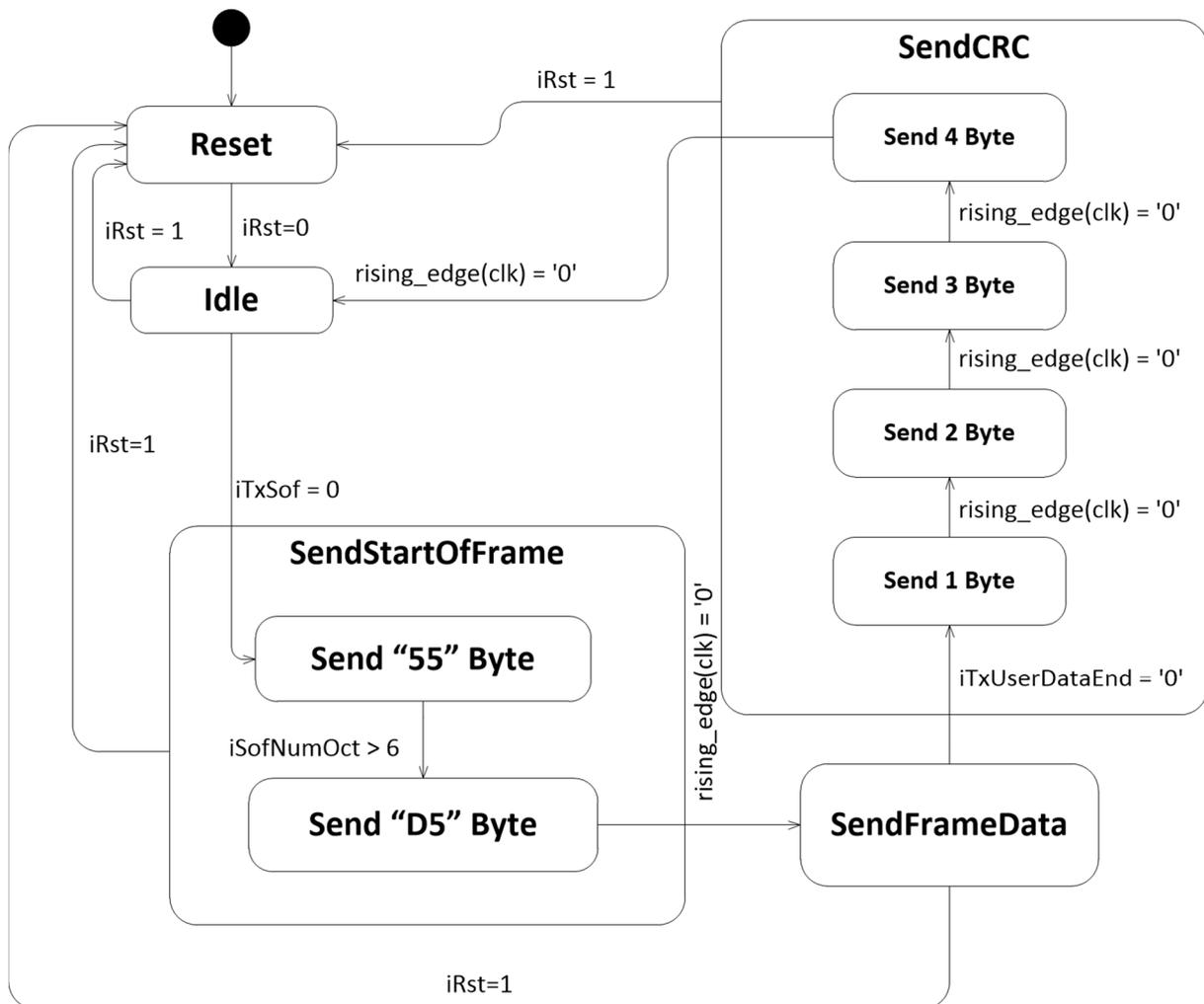


Рис. 12. Диаграмма состояния модуля FpgaToPcLink

Интерфейс модуля изображен на Рис. 13.

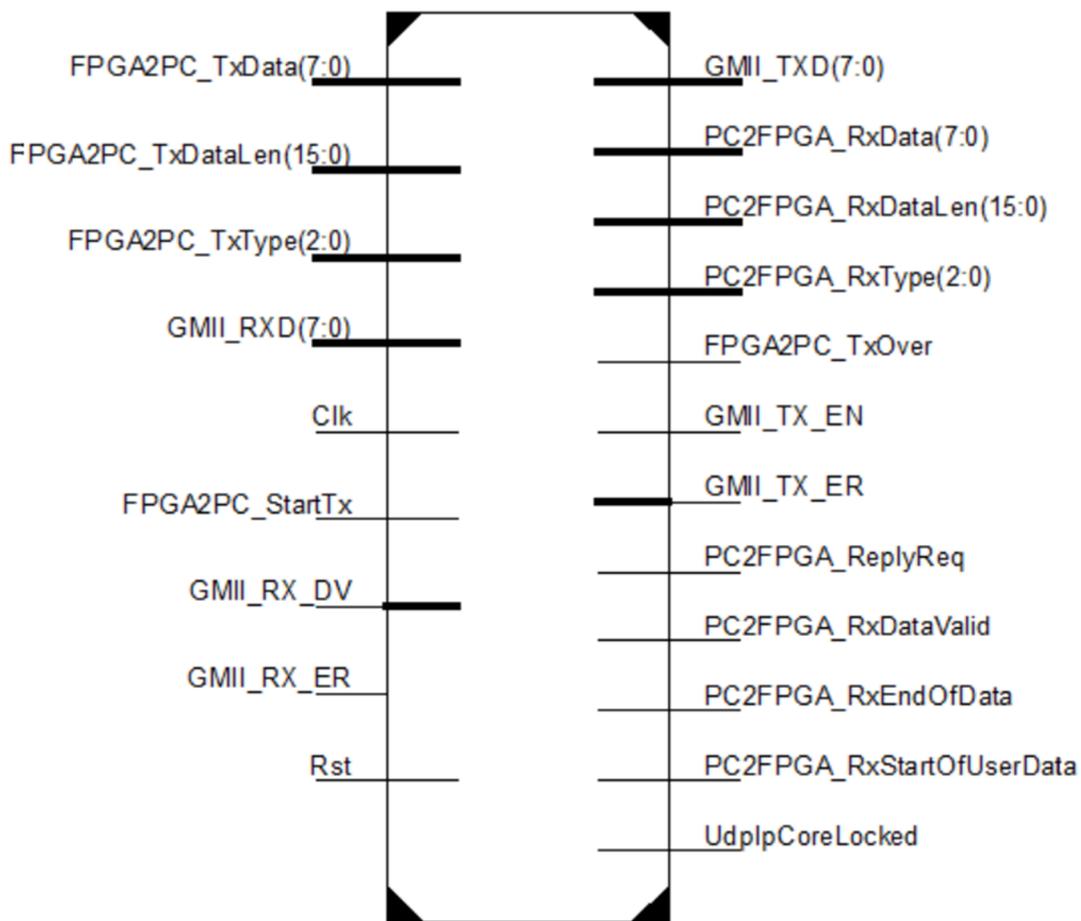


Рис. 13. Интерфейс модуля FpgaToPcLink

FPGA2PC_StartTx	— Начало передачи данных
FPGA2PC_TxType	— Тип данных
FPGA2PC_TxDataLen	— Размер данных
FPGA2PC_TxData	— Шина данных
FPGA2PC_TxOver	— Окончание передачи данных
Clk	— Тактирующий сигнал
Rst	— Сброс
GMI_RXD	— Шина данны GMI
GMI_RX_DV	— Действительность данных
GMI_RX_ER	— Ошибка передачи данных
GMI_TXD	— Принимающая шина данных
GMI_TX_EN	— Действительность данных
GMI_TX_ER	— Ошибка передачи данных

PC2FPGA_RxStartOfUserData	— Начало передачи данных
PC2FPGA_RxEndOfData	— Тип данных
PC2FPGA_RxDataValid	— Действительность данных
PC2FPGA_RxDataLen	— Размер данных
PC2FPGA_RxType	— Тип данных
PC2FPGA_RxData	— Шина данных
UdpIpCoreLocked	— Флаг конфигурации UDP
PC_COM	— модуль, реализующий стек UDP, без согласования его с

GMII

Интерфейс модуля:

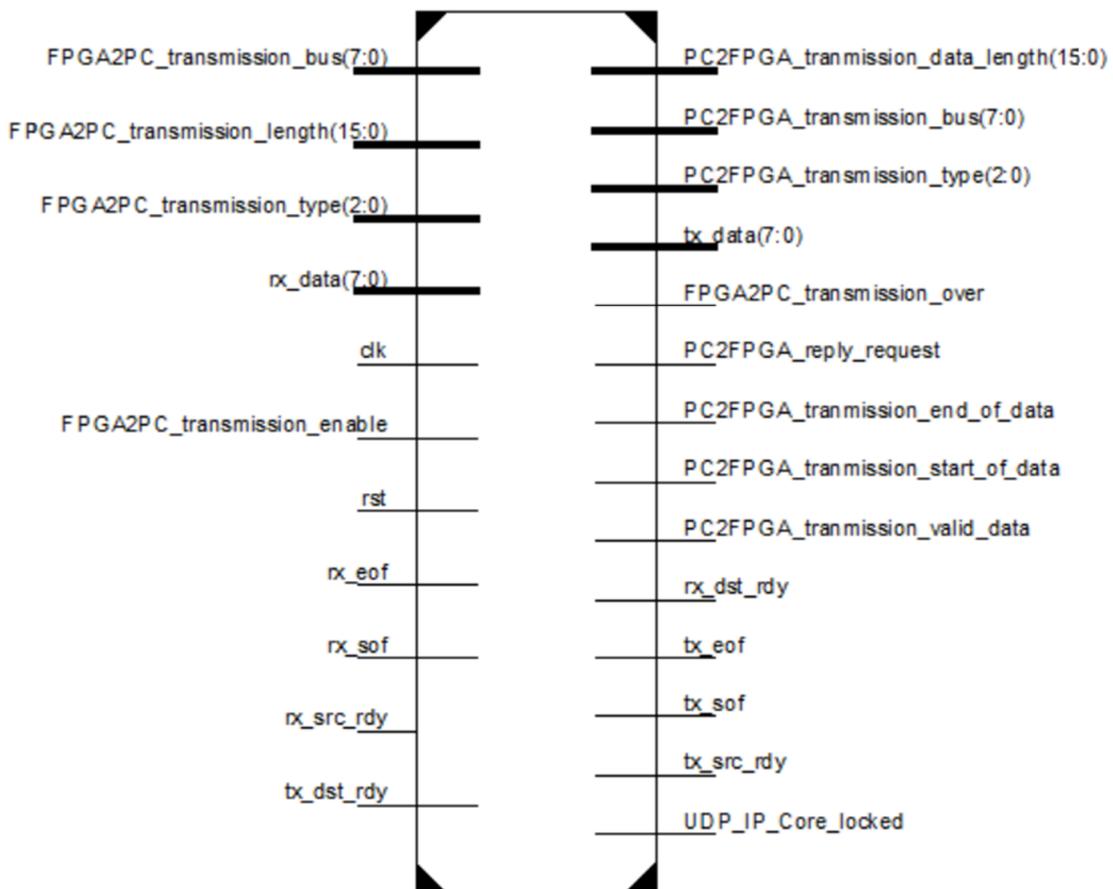


Рис. 14. Интерфейс модуля PC_COM

FPGA2PC_transmission_enable	— Разрешение передачи
FPGA2PC_transmission_type	— Тип данных
FPGA2PC_transmission_length	— Размер данных

FPGA2PC_transmission_bus — Шина данных
FPGA2PC_transmission_over — Окончание передачи данных
clk — Тактирующий сигнал
rst — Сброс

rx_eof — Окончание передачи данных
rx_sof — Начало передачи данных
rx_src_rdy — Готовность передатчика
rx_dst_rdy — Готовность приемника
rx_data — Шина данных

PC2FPGA_tranmission_start_of_data — Начало передачи данных
PC2FPGA_tranmission_end_of_data — Окончание передачи данных
PC2FPGA_tranmission_valid_data — Действительность данных
PC2FPGA_tranmission_data_length — Размер данных
PC2FPGA_transmission_type — Тип данных
PC2FPGA_transmission_bus — Шина данных

tx_sof — Начало передачи данных
tx_eof — Окончание передачи данных
tx_src_rdy — Готовность передатчика
tx_dst_rdy — Готовность приемника
tx_data — Шина данных

UDP_IP_Core_locked — Флаг конфигурации UDP

DataBusDelay — модуль, необходимый для синхронизации потоков, используется для задержки потока данных

pcrc — модуль, отвечающий за подсчет CRC

UDP_IP_Core — модуль, реализующий большинство особенностей стека UDP.

Интерфейс модуля:

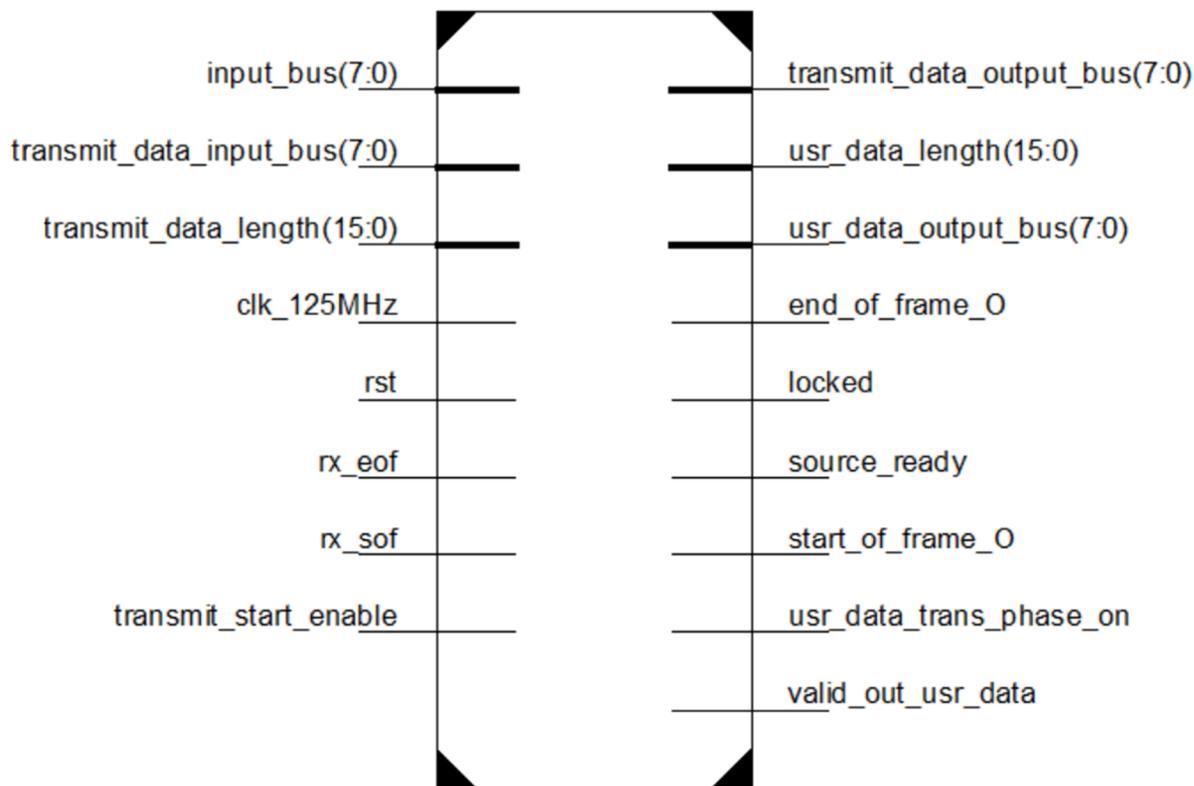


Рис. 15. Интерфейс модуля UDP_IP_Core

input_bus	— Шина данных
transmit_data_input_bus	— Шина данных
transmit_data_length	— Размер данных
clk_125MHz	— Тактирующий сигнал
rst	— Сброс
rx_sof	— Начало передачи данных
rx_eof	— Окончание передачи данных
transmit_start_enable	— Начало передачи данных
transmit_data_output_bus	— Шина данных
usr_data_length	— Размер данных
usr_data_output_bus	— Шина данных

end_of_frame_O	— Окончание передачи данных
locked	— Флаг конфигурации
source_ready	— Готовность передатчика
start_of_frame_O	— Начало передачи данных
usr_data_trans_phase_on	— Передача данных
valid_out_usr_data	— Действительность данных

IPv4_PACKET_RECEIVER — модуль, реализующий приемную часть сетевого уровня IP в модуле UDP_IP_Core

Интерфейс модуля:

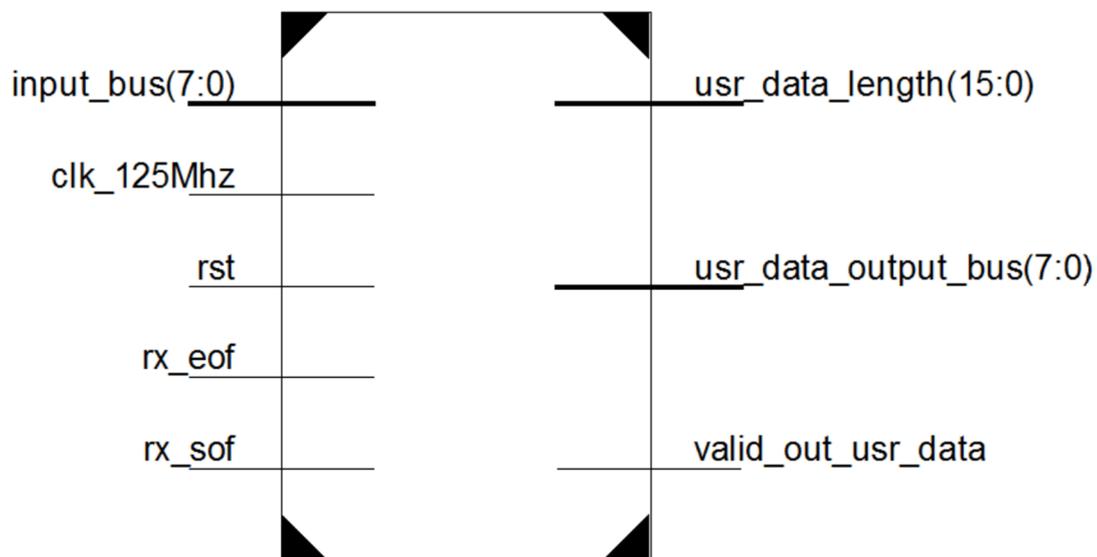


Рис. 16. Интерфейс модуля IPv4_PACKET_RECEIVER

input_bus	— Шина данных
clk_125Mhz	— Тактирующий сигнал
rst	— Сброс
rx_sof	— Окончание передачи данных
rx_eof	— Начало передачи данных
usr_data_length	— Размер данных
usr_data_output_bus	— Шина данных
valid_out_usr_data	— Действительность данных

PACKET_RECEIVER_FSM — модуль, используемый в модуле IPv4_PACKET_RECEIVER для определения длины принимаемого пакета с последующей проверкой.

Исходя из назначения модуля, был разработан конечный автомат

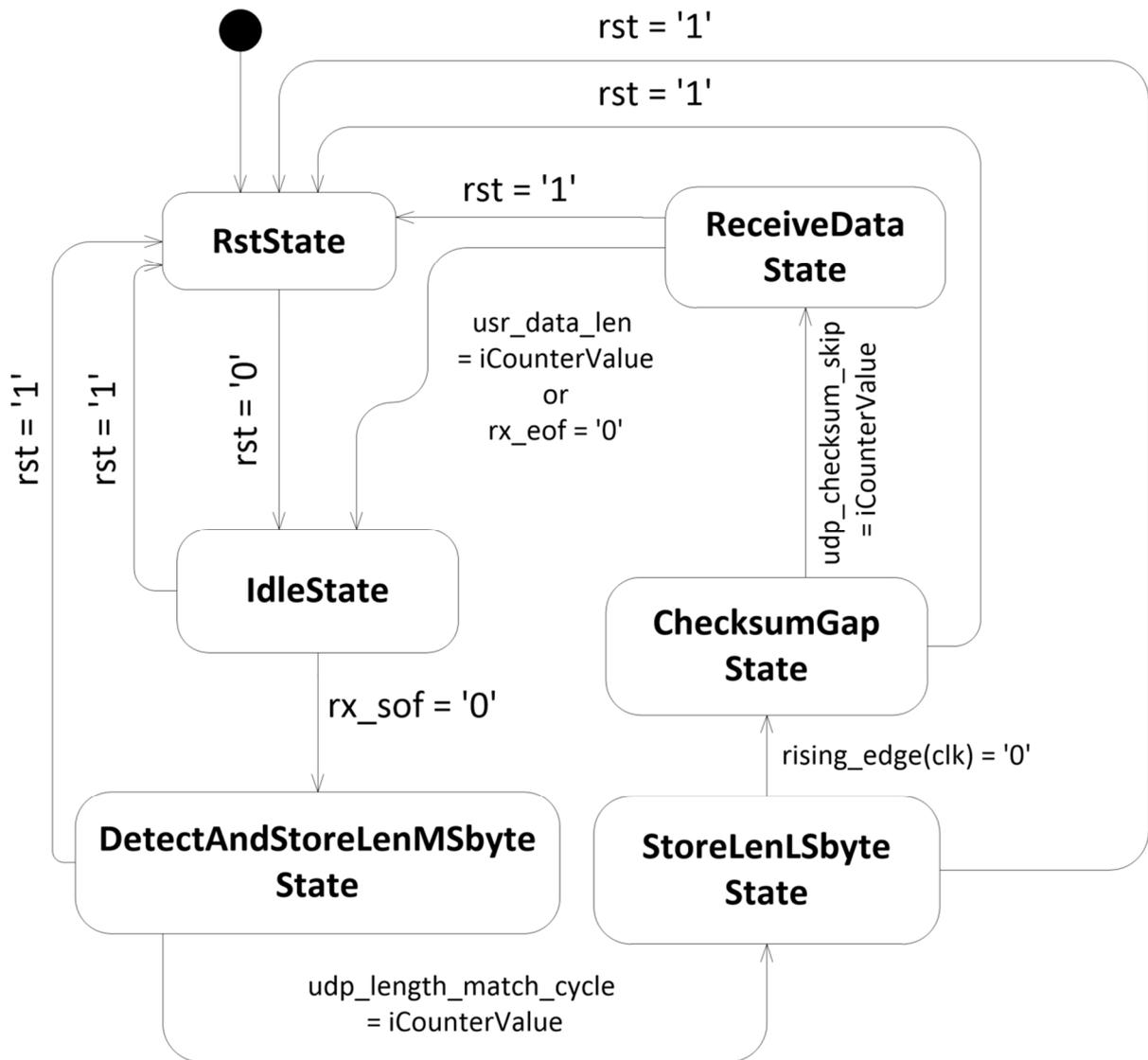


Рис. 17. Диаграмма состояния модуля PACKET_RECEIVER_FSM

Интерфейс модуля:

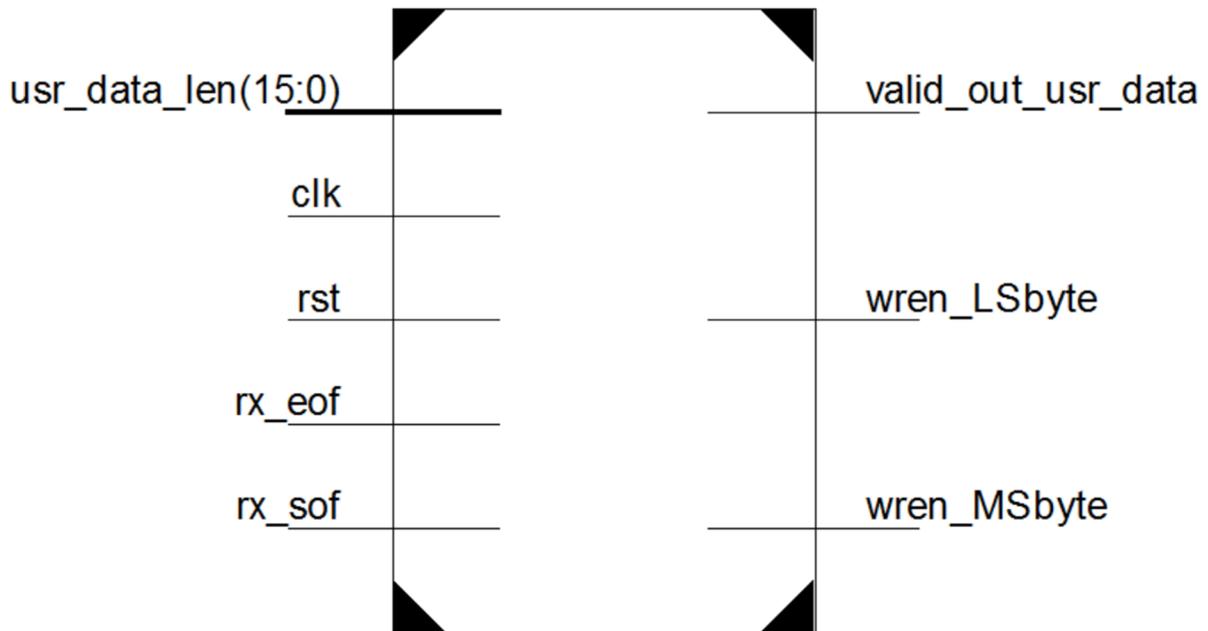


Рис. 18. Интерфейс модуля PACKET_RECEIVER_FSM

- `usr_data_len` — Проверочная шина данных
- `clk` — Тактирующий сигнал
- `rst` — Перезапуск
- `rx_eof` — Конец кадра
- `rx_sof` — Начало кадра
- `wren_MSbyte` - Старший байт длины кадра
- `wren_LSbyte` — Младший байт длины кадра

CounterNBit — параметризованный модуль, реализующий счётчик.

IPV4_PACKET_TRANSMITTER модуль, реализующий передающую часть сетевого уровня IP в модуле `UDP_IP_Core`

Интерфейс модуля представлен на Рис. 19

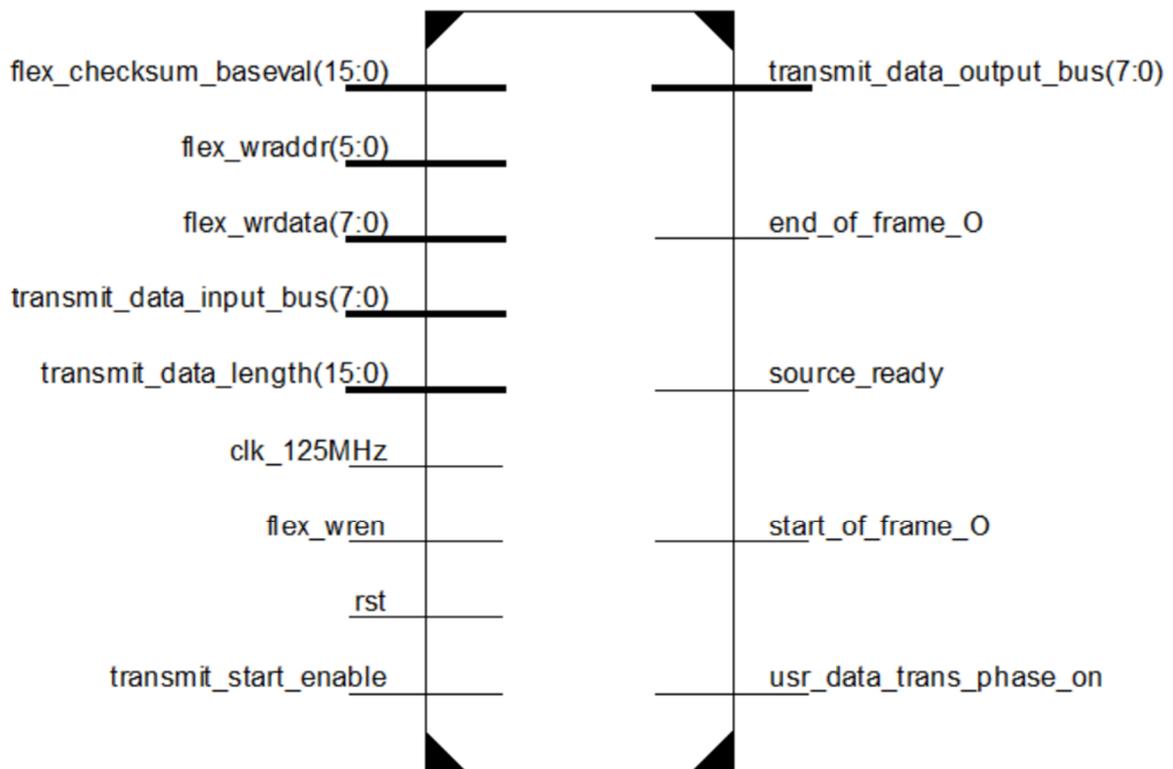


Рис. 19. Интерфейс модуля IPV4_PACKET_TRANSMITTER

flex_checksum_baseval	— Управление CRC
flex_wraddr	— Адресная шина управления
flex_wrdata	— Шина данных управления
transmit_data_length	— Размер данных
transmit_data_input_bus	— Шина данных
clk_125MHz	— Тактирующий сигнал
rst	— Сброс
flex_wren	— Разрешение управления
transmit_start_enable	— Начало передачи данных
transmit_data_output_bus	— Шина данных
end_of_frame_O	— Окончание передачи данных
source_ready	— Готовность передатчика
start_of_frame_O	— Начало передачи данных
usr_data_trans_phase_on	— Передача данных

BufRegNBit — Параметризованный модуль, предназначенный для буферизации шин данных произвольной размерности

IPV4_LUT_INDEXER — Модуль, предназначенный для управления адресацией ячеек памяти конфигурационной информации

OVERRIDE_LUT_CONTROL — Модуль, управляющей записью конфигурационной информацией

dist_mem_64x8 — Модуль реализующий энергозависимую память на логических ячейках.

FLEX_CONTROL — Модуль, предназначенный для управления конфигурацией, обертка над **CONFIG_CONTROL**.

Интерфейс модуля:

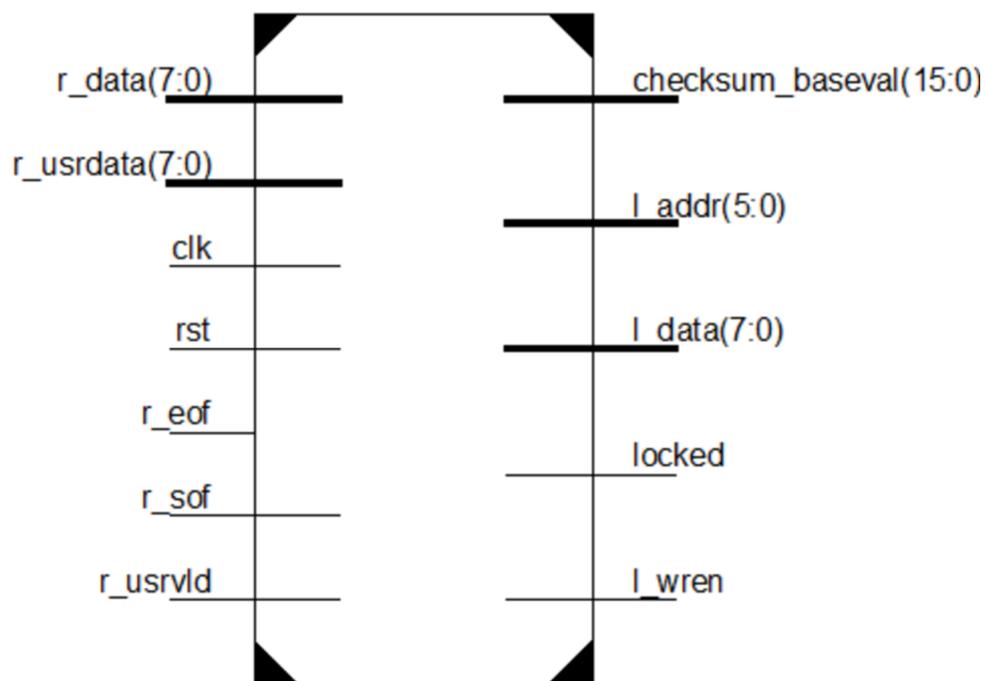


Рис. 20. Интерфейс модуля FLEX_CONTROL

r_data	— Шина данных
r_usrdata	— Шина данных
clk	— Тактирующий сигнал
rst	— Сброс
r_eof	— Окончание передачи данных

r_sof	— Начало передачи данных
r_usrvld	— Действительность данных
checksum_baseval	— CRC
l_addr	— Адресная шина управления
l_data	— Шина данных управления
locked	— Флаг конфигурации
l_wren	— управление конфигурацией

MATCH_CMD — модуль, разработанный для отслеживания в потоке принятых от компьютера данных команд управления, необходимых для регулирования процесса передачи.

Исходя из назначения модуля, была разработана диаграмма состояния.

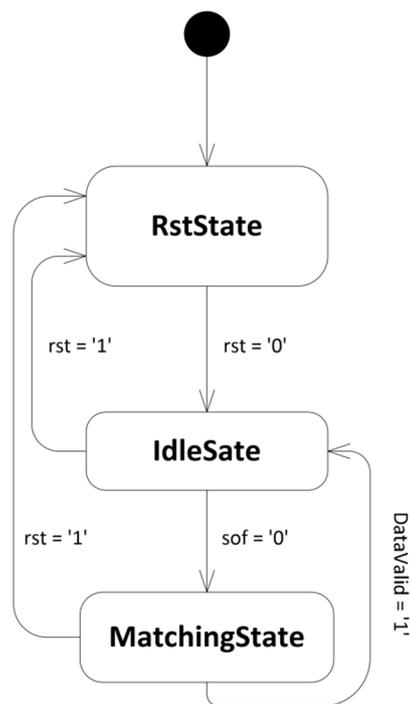


Рис. 21. Диаграмма состояния модуля MATCH_CMD

Интерфейс модуля:

CmdToMatch(7:0) — Входная шина данных

DataToMatch(7:0) — Выявляемая команда

clk — Тактирующий сигнал

DataValid	— Действительность данных
rst	— Перезапуск
sof	— Старт кадра
CmdMatched	— Команда найдена

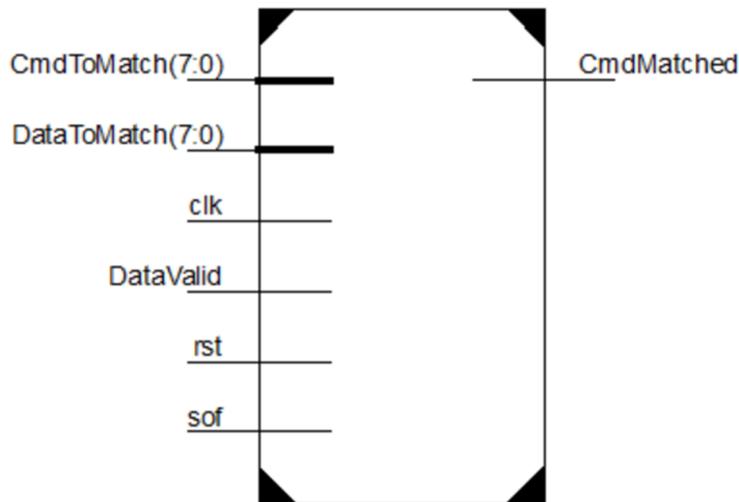


Рис. 22. Интерфейс модуля MATCH_CMD

CONFIG_CONTROL — модуль, отвечающий за конфигурацию подсистемы передачи данных.

Исходя из назначения модуля, была разработана диаграмма состояния, представленная на Рис. 24

Интерфейс модуля изображен на Рис. 23:

clk	— тактирующий сигнал
config_en	— разрешение конфигурации
nxt_sof	— начало кадра
rst	— перезагрузка
unlock_en	— сброс конфигурации
addr	— адрес конфигурируемого параметра
locked	— флаг конфигурации
wren	— разрешение записи параметра конфигурации
wren_checksum_1	— разрешение записи CRC 1
wren_checksum_2	— разрешение записи CRC 2

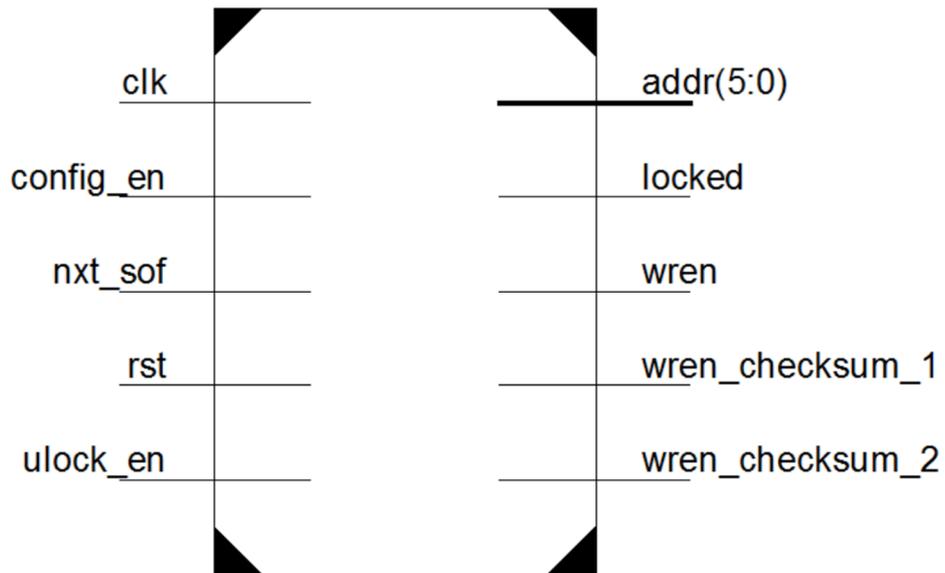


Рис. 23. Интерфейс модуля CONFIG_CONTROL

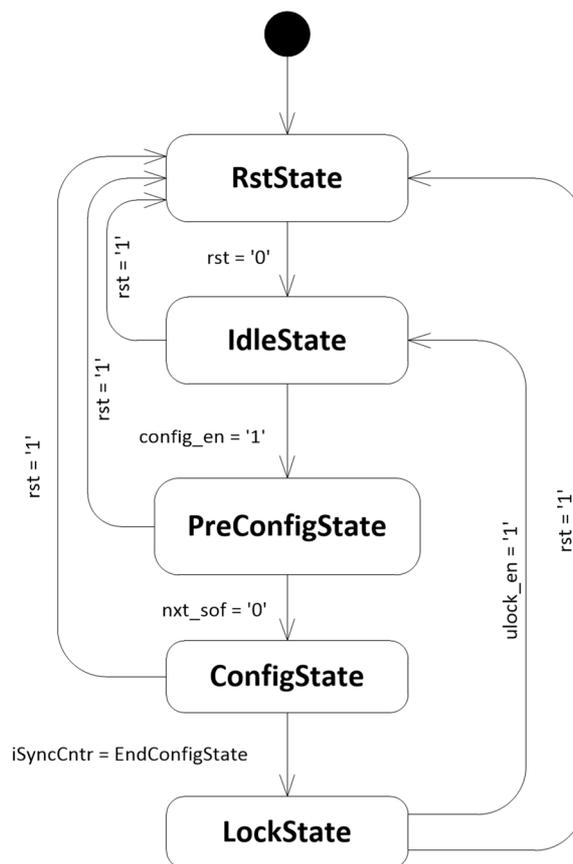


Рис. 24. Диаграмма состояния CONFIG_CONTROL

FPGA2PC — Модуль, реализующий передающую часть модуля PC_COM

Интерфейс модуля:

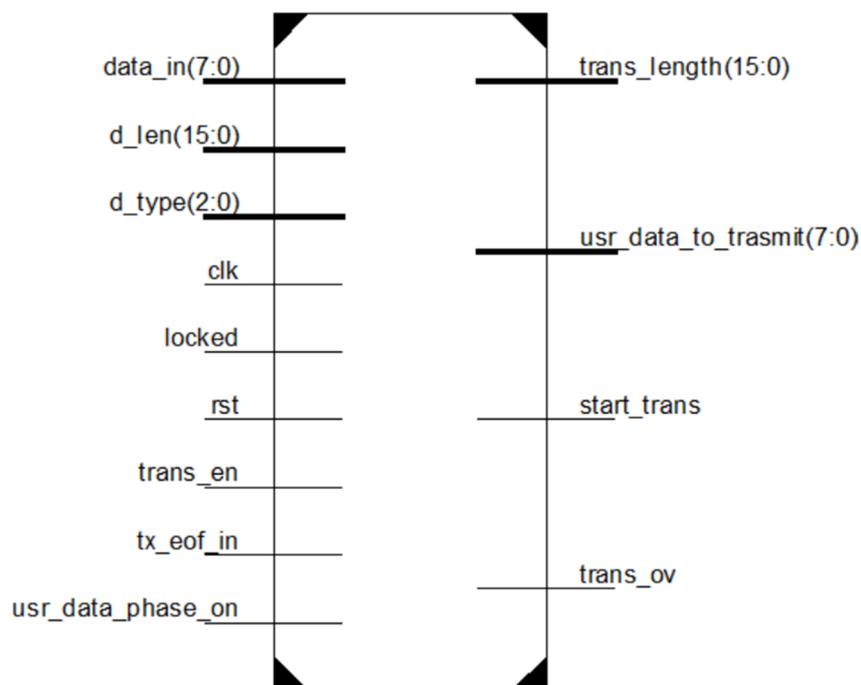


Рис. 25. Интерфейс модуля FPGA2PC

<code>data_in</code>	— Шина данных
<code>d_len</code>	— Размер данных
<code>d_type</code>	— Тип данных
<code>clk</code>	— Тактирующий сигнал
<code>locked</code>	— Флаг конфигурации
<code>rst</code>	— Сброс
<code>trans_en</code>	— Разрешение передачи
<code>tx_eof_in</code>	— Окончание передачи данных
<code>usr_data_phase_on</code>	— Начало передачи данных
<code>trans_length</code>	— Размер данных
<code>usr_data_to_transmit</code>	— Шина данных
<code>start_trans</code>	— Начало передачи данных
<code>trans_ov</code>	— Окончание передачи данных

FSM_SEL_HEADER — модуль, управляющий обрамлением данных заголовком при передаче.

Исходя из назначения модуля, была разработана диаграмма состояния:

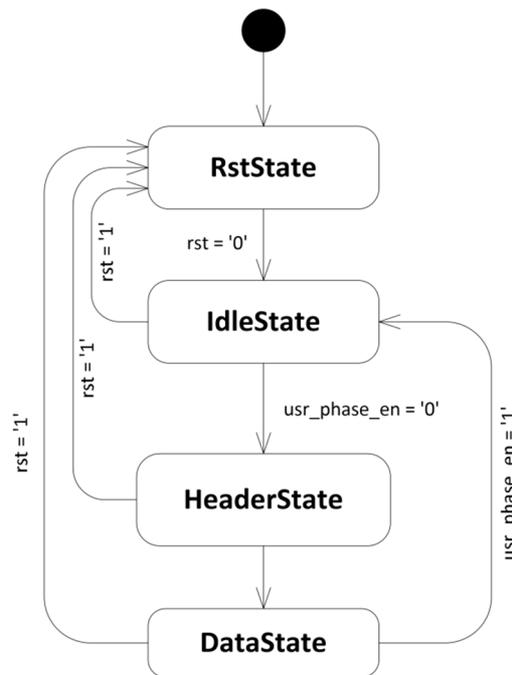


Рис. 26. Диаграмма состояния FSM_SEL_HEADER

Интерфейс модуля:

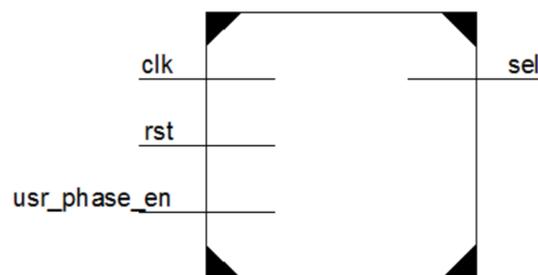


Рис. 27. Интерфейс модуля FSM_SEL_HEADER

- clk — Тактирующий сигнал
- rst — Сброса
- usr_phase_en — Разрешение работы
- sel — Управляющий сигнал

D_TYPE_LEN_CNTRL — Модуль, контролирующий размер переданных данных и их тип.

PC2FPGA — Модуль, реализующий передающую часть модуля **PC_COM**

Интерфейс модуля:

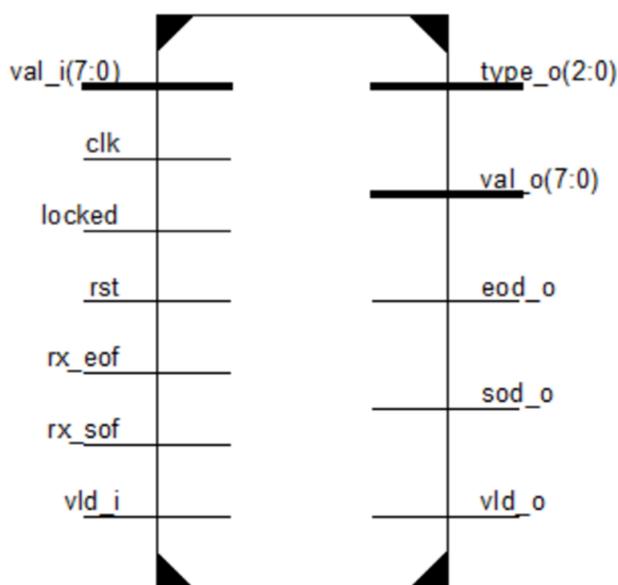


Рис. 28. Интерфейс модуля PC2FPGA

val_i	— Шина данных
clk	— Тактирующий сигнал
locked	— Флаг конфигурации
rst	— Сброс
rx_eof	— Окончание передачи данных
rx_sof	— Начало передачи данных
vld_i	— Действительность данных
type_o	— Тип данных
val_o	— Шина данных
eod_o	— Окончание передачи данных
sod_o	— Начало передачи данных
vld_o	— Действительность данных

5.3. Разработка схемы электрической принципиальной платы ячейки.

В схему платы добавлен, так называемый, логический банк. Этот элемент позволяет показывать на схеме только необходимую информацию, не показывая массивную схему с 900 выводами. Кроме того, такой подход упрощает разработку, позволяя легче вносить изменения в уже готовый проект.

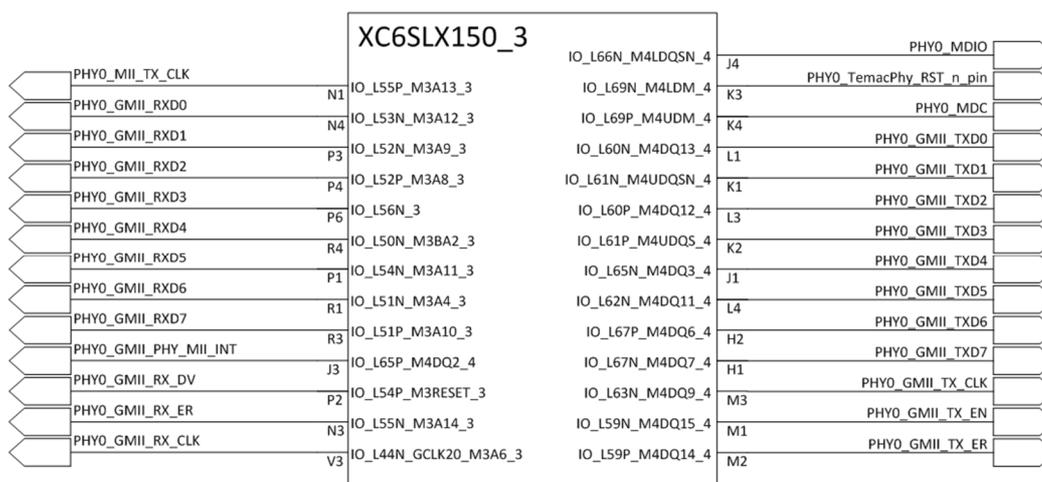


Рис. 29. Принциальная схема логического банка ПЛИС

На Рис. 29 изображен блок, который показывает линии, соединенные напрямую к ПЛИС. Среди них линии интерфейса GMII для связи с драйвером физического уровня. И линии управления этим драйвером.

На Рис. 30 изображена принципиальная электрическая схема драйвера физического уровня. На которой показаны:

- Линии питания и заземления
- Светодиоды, информирующие о наличии соединения
- Линии интерфейса GMII, необходимые для соединения с ПЛИС
- Линии соединяющие дифференциальные линии с внешним трансформатором

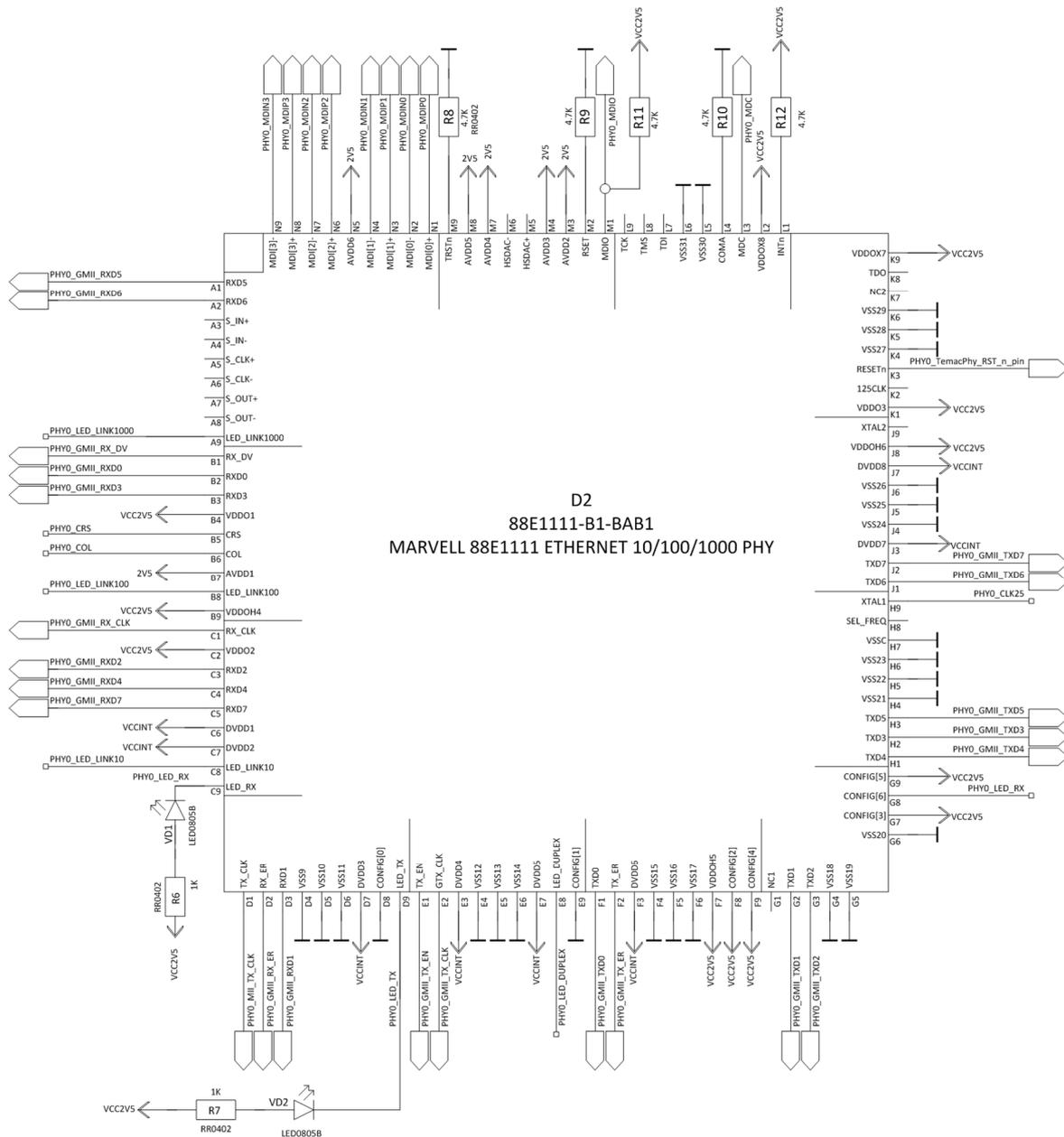


Рис. 30. Принциальная схема драйвера физического уровня

На Рис. 31 изображена схема принципиальная электрическая, относящаяся к внешнему трансформатору, который необходим для согласования уровней и predания дифференциальной паре необходимой индуктивности.

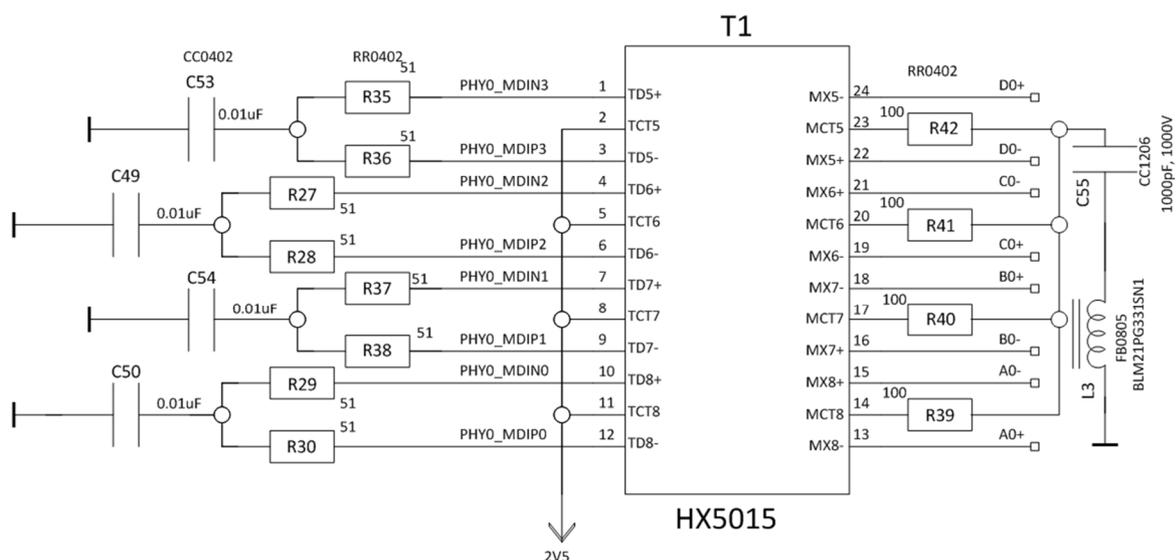


Рис. 31. Принципиальная схема внешнего трансформатора

На принципиальной схеме разъема на плате изображено соединение выходной дифференциальной пары трансформатора с разъемом на плате, который, в свою очередь, соединен жгутом с разъемом на корпусе БОС.

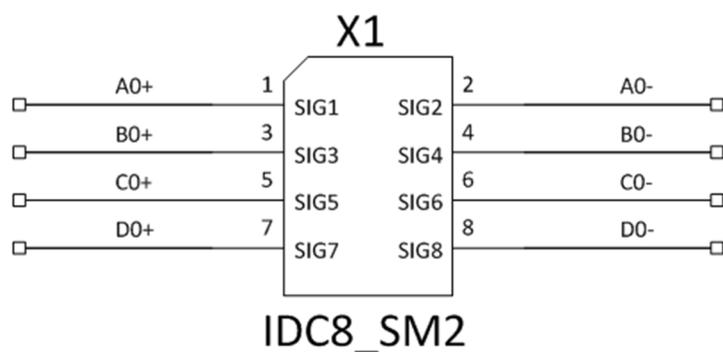


Рис. 32. Принципиальная схема разъема на плате

6. Рабочее проектирование.

6.1. Реализация программной части подсистемы приема и хранения данных.

На этапе технического проектирования, исходя из назначения подсистемы, был разработан алгоритм. На этапе рабочего проектирования он был реализован в виде программы. Исходя из выбранной общесистемной концепции и технического задания, которое является входным документом для данного этапа, программа была написана на языке C++, который реализует парадигму объектно-ориентированного программирования.

6.1.1. Диаграмма классов программной части подсистемы приема и хранения данных.

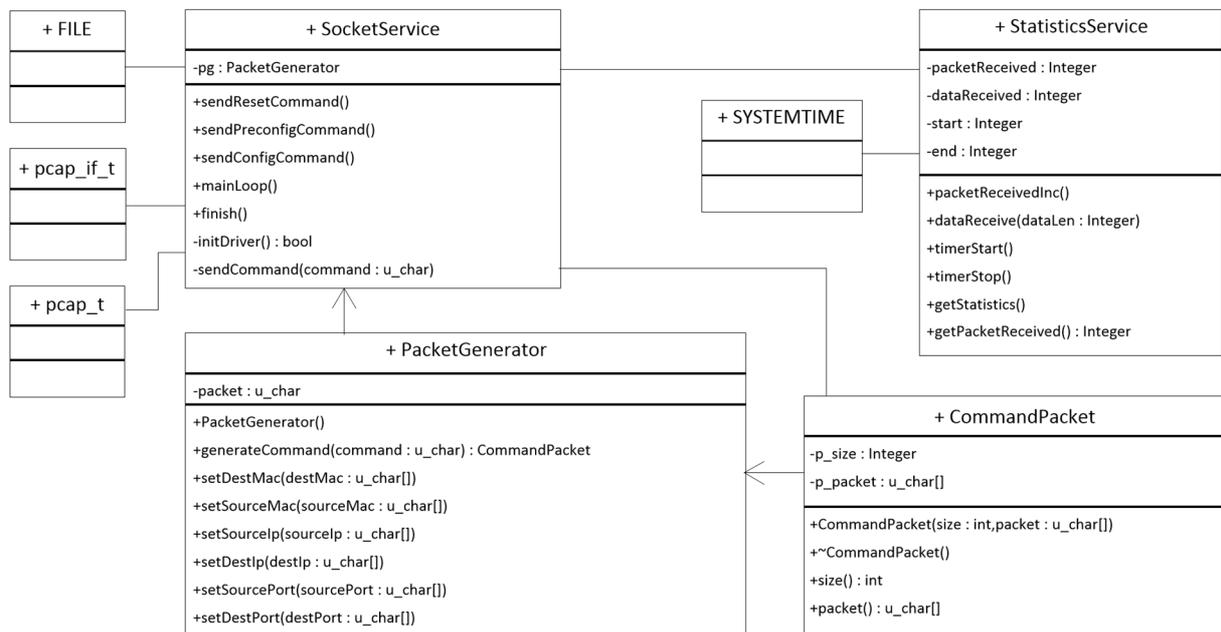


Рис. 33. Диаграмма классов программы

StatisticsService — Класс, реализующий необходимые для статистики функции.

SYSTEMTIME — Класс, стандартный для Windows API, используется для замера времени работы программы в классе **StatisticsService**

PacketGenerator — Класс, реализующий функции, необходимые для создания управляющих команд (объекты класса **CommandPacket**) и конфигурации этих команд.

CommandPacket — Класс, объекты которого используются для передачи управляющих команд.

SocketService — Основной класс программы, владеющий экземплярами классов **PacketGenerator**, **StatisticsService**, **pcap_t**, **FILE** и реализующий функционал подсистемы приема и хранения данных.

Программный код, реализующий класс **SocketService**, представлен в Приложении 1.

FILE — Стандартный класс, необходимый для работы с файлом.

pcap_if_t — Класс из библиотеки WinPCap, необходимый для работы с драйверами физического уровня.

pcap_t — Класс из библиотеки WinPCap, необходимый для работы с сокетами.

6.1.2 Диаграмма компонентов программной части подсистемы приема и хранения данных

Как уже было сказано ранее, в программе используются классы **pcap_if_t** и **pcap_t** из библиотеки WinPCap.

Эта библиотека позволяет создавать программы анализа сетевых данных, поступающих на сетевую карту компьютера. Примером программного обеспечения, использующего библиотеку Pcap, служит программа Wireshark, а также другие программы мониторинга и тестирования сети и sniffеры [16]. Она предназначена для разработки программ на языке C++, на котором были написаны остальные

компоненты подсистемы приема и хранения данных разрабатываемого регистратора.

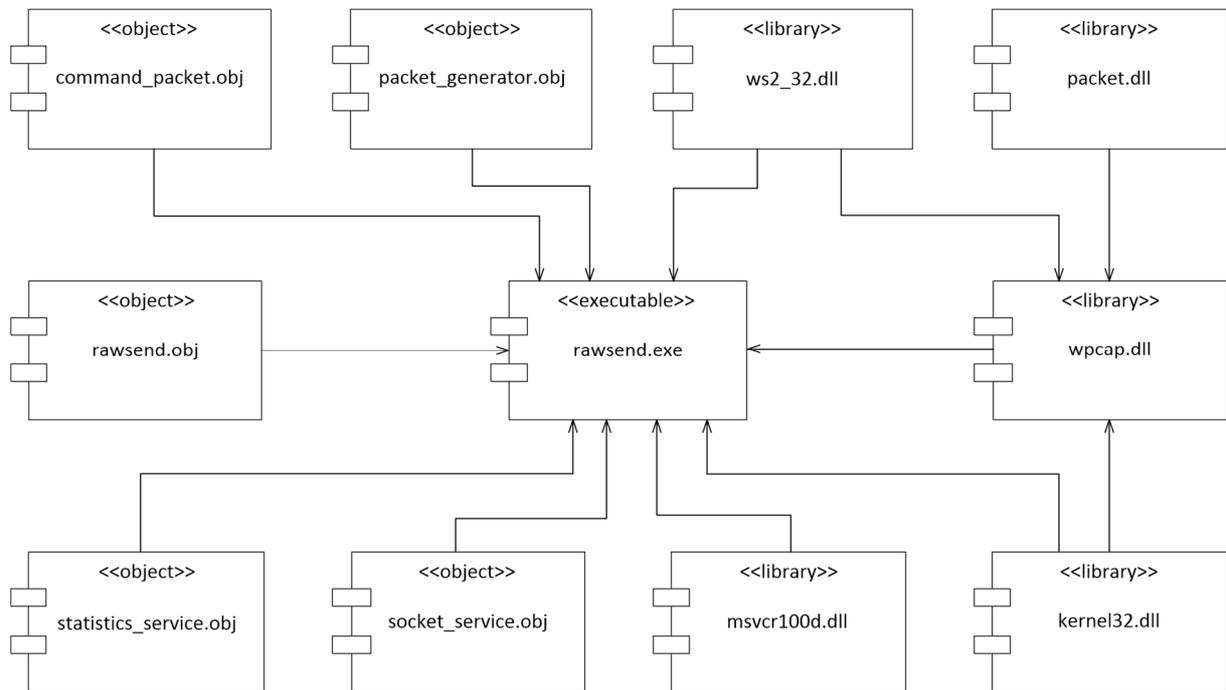


Рис. 34. Диаграмма компонентов программы

Библиотека WinPcap не единственная, позволяющая работать с сокетами, но она примечательно тем, что позволяет записывать принятые файлы на диск, используя технологию Zero-copy.

Zero copy (досл. ноль копирований) — описывает операции, в ходе которых процессор не выполняет задачу копирования данных из одной области памяти в другую. Термин применяется для описания технологий, которые помогли уменьшить количество копирований между промежуточными буферами.[17] Применительно к поставленной задаче это означает, что программа не копирует дополнительно пакеты, а использует буфер памяти, в котором уже есть необходимые данные для записи в файл.

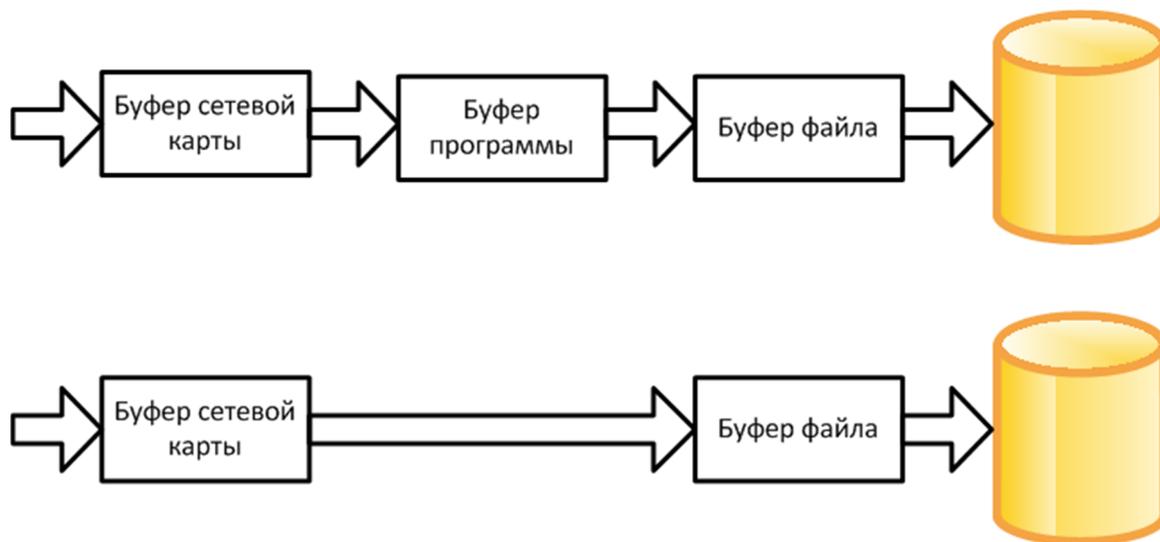


Рис. 35. Технология Zero-copy

6.2. Реализация программной части подсистемы передачи данных.

В отличие от обычных цифровых микросхем, логика работы ПЛИС не определяется при изготовлении, а задаётся посредством программирования, что позволяет задать желаемую структуру цифрового устройства в программы на специальных языках Verilog. На этапе рабочего проектирования были написаны программы, реализующие модули, структура, интерфейсы и поведение которых было разработано на техническом этапе проектирования.

FpgaToPcLink — модуль верхнего уровня, согласующий интерфейсы между основным модулем ПЛИС ячейки БОС и драйвером физического уровня. Модуль также обрамляет UDP пакеты Ethernet преамбулой и CRC.

Структура и интерфейс были разработаны на техническом этапе проектирования. Функциональная схема вынесена на лист 7 графических материалов.

PC_COM — модуль, реализующий стек UDP, без согласования его с GМII, структура и интерфейс которого были разработаны на техническом этапе проектирования.

Функциональная схема изображена на Рис. 36

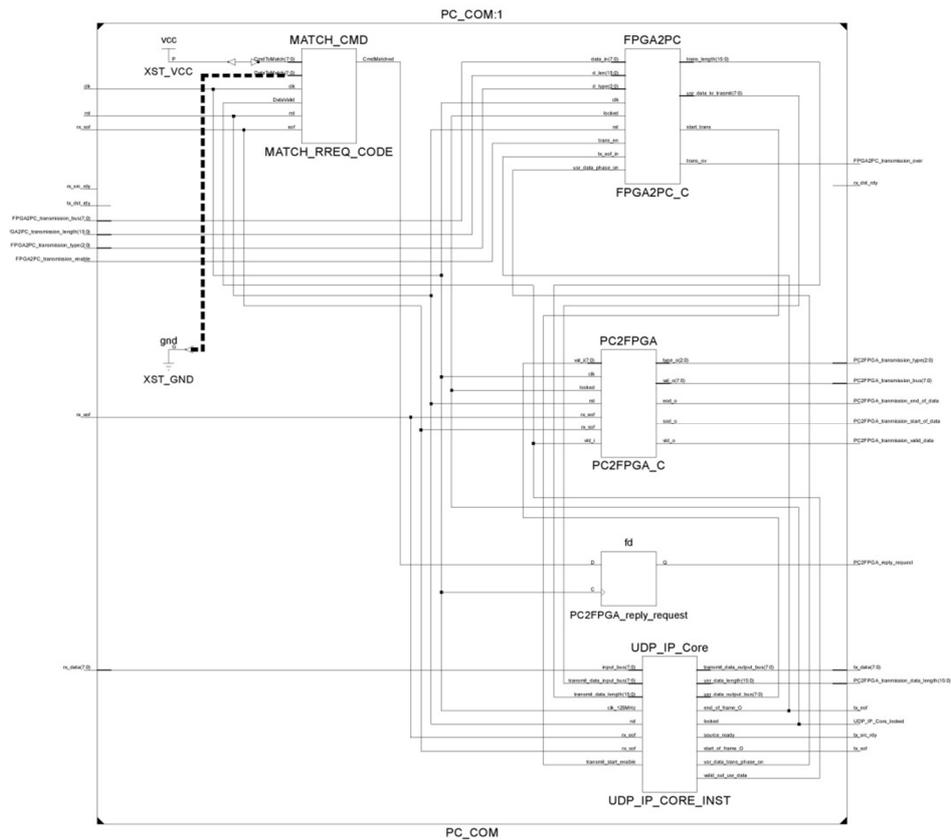


Рис. 36. Функциональная схема модуля PC_COM

UDP_IP_Core — модуль, реализующий большинство особенностей стека UDP, структура и интерфейс которого были разработаны на техническом этапе проектирования.

Функциональная схема изображена на Рис. 37.

IPv4_PACKET_RECEIVER — модуль, реализующий приемную часть сетевого уровня IP в модуле UDP_IP_Core, структура и интерфейс которого были разработаны на техническом этапе проектирования.

Функциональная схема изображена на Рис. 38.

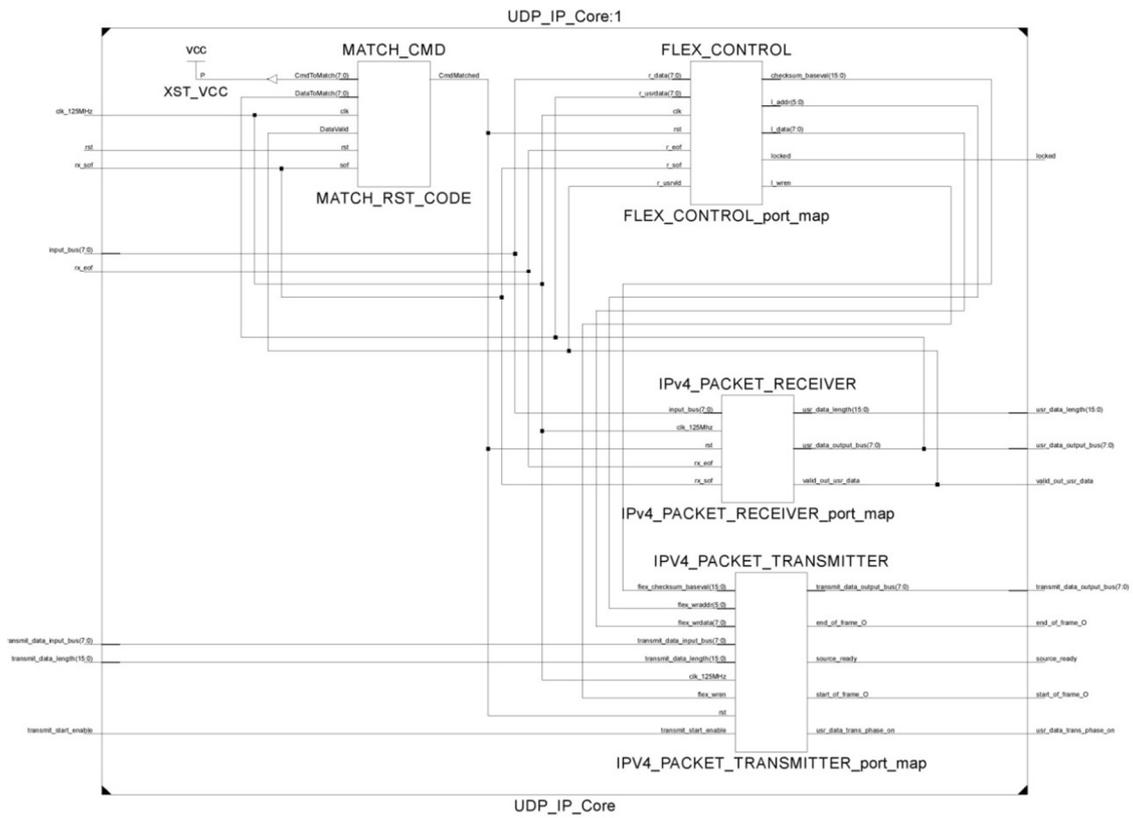


Рис. 37. Функциональная схема модуля UDP_IP_Core

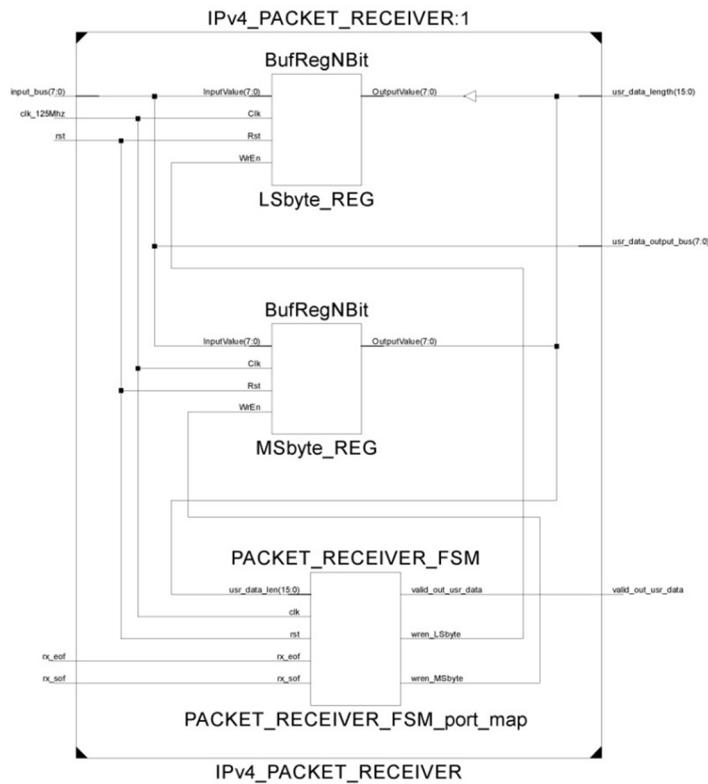


Рис. 38. Функциональная схема модуля IPv4_PACKET_RECEIVER

PACKET_RECEIVER_FSM — модуль, используемый в модуле IPv4_PACKET_RECEIVER для определения длины принимаемого пакета с последующей проверкой. Структура и интерфейс этого модуля были разработаны на этапе технического проектирования, а функциональная схема приведена в Приложении 2.

IPV4_PACKET_TRANSMITTER — модуль, реализующий передающую часть сетевого уровня IP в модуле UDP_IP_Core, структура и интерфейс которого были разработаны на этапе технического проектирования.

Функциональная схема приведена на Рис. 39.

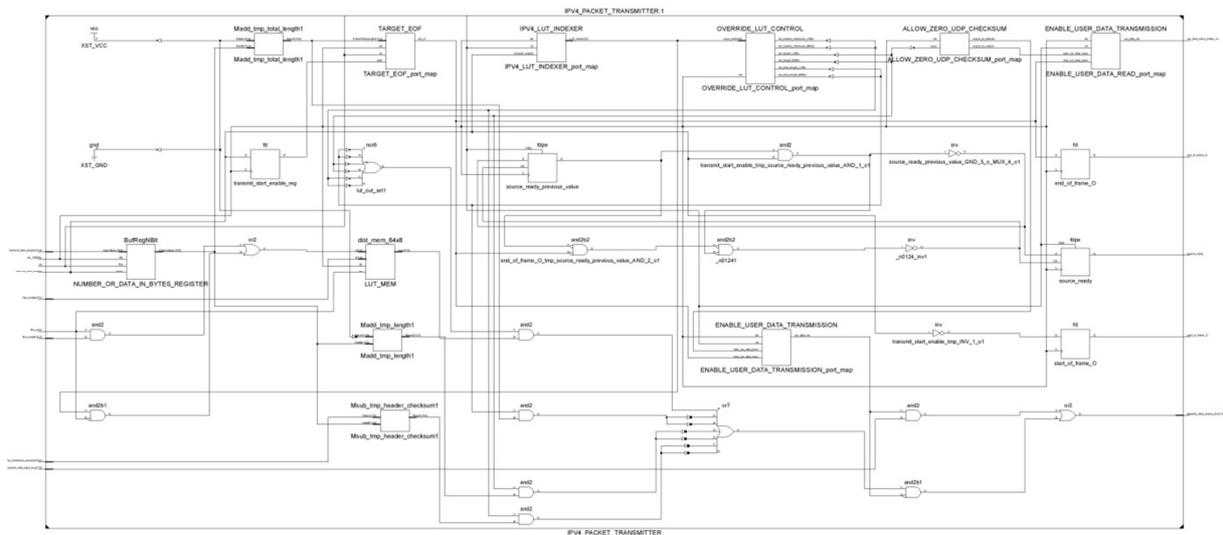


Рис. 39. Функциональная схема модуля IPV4_PACKET_TRANSMITTER

MATCH_CMD — модуль, разработанный, для отслеживания в потоке принятых от компьютера данных команд управления, структура и интерфейс которого были разработаны на этапе технического проектирования.

Функциональная схема приведена на Рис. 41

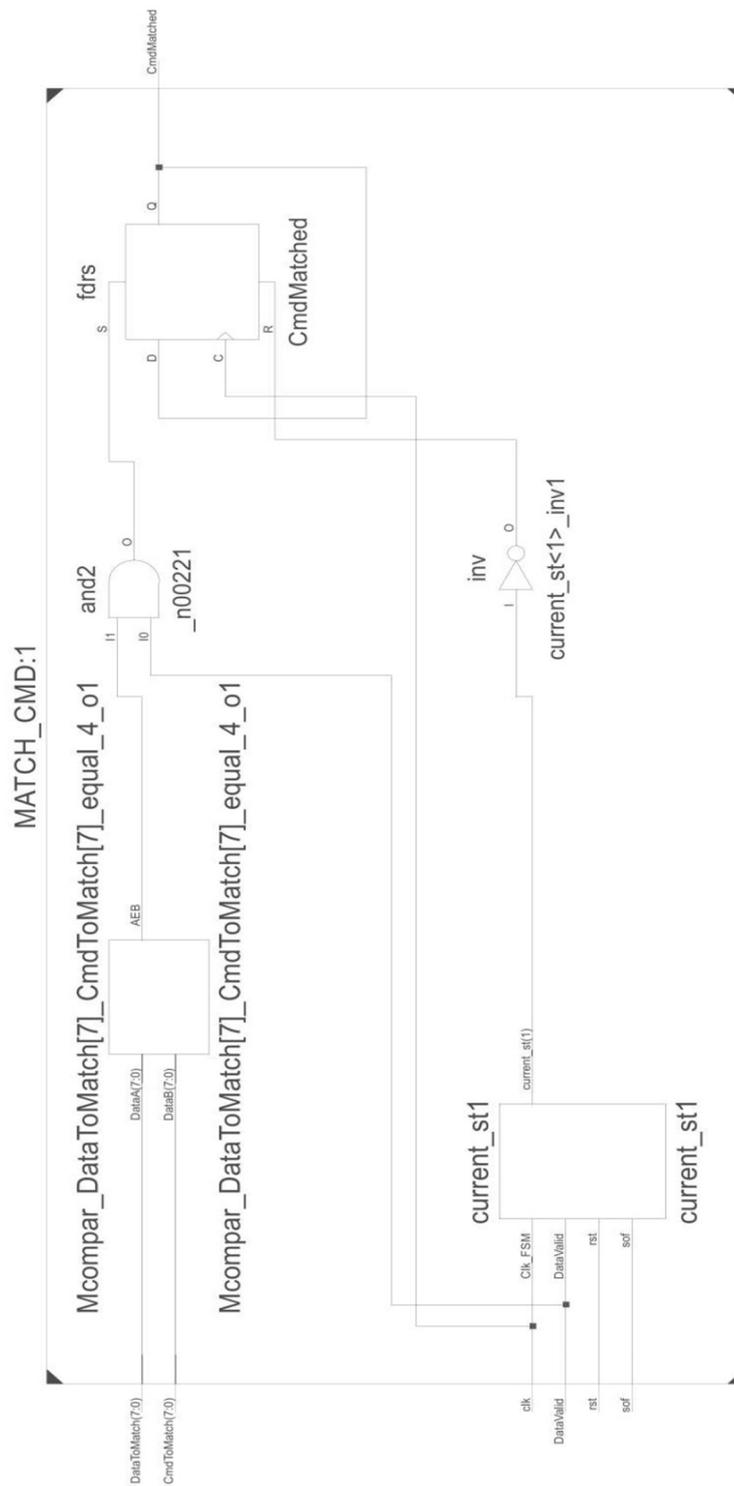


Рис. 40. Функциональная схема модуля MATCH_CMD

Диаграмма состояния конечного автомата **current_st1** была разработана на техническом этапе проектирования и изображена на Рис. 21.

CONFIG_CONTROL — модуль, отвечающий за конфигурацию подсистемы передачи данных, структура и интерфейс которого были разработаны на техническом этапе проектирования.

Функциональная схема модуля:

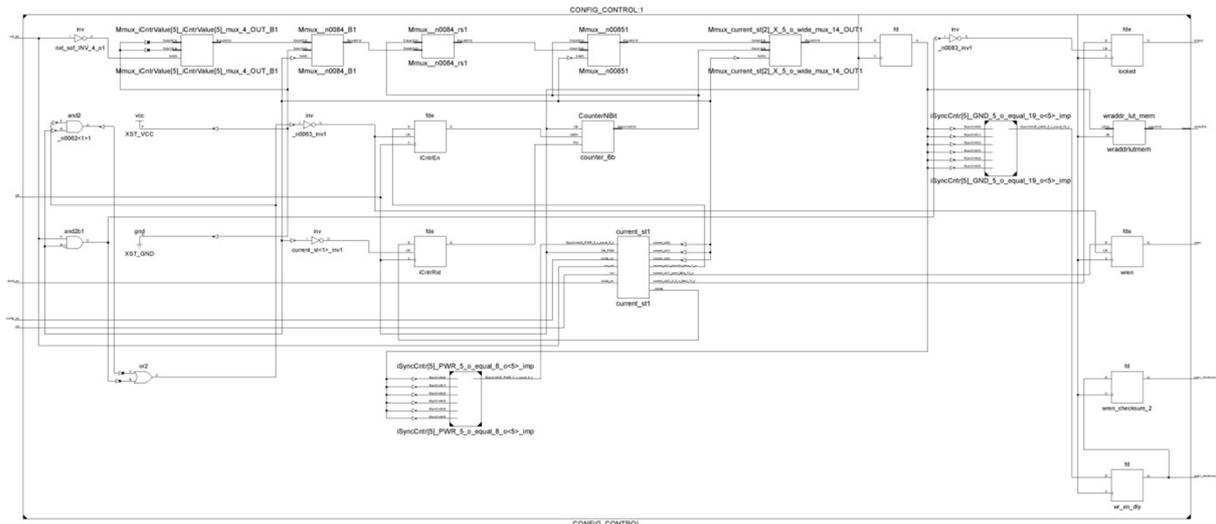


Рис. 41. Функциональная схема модуля CONFIG_CONTROL

Диаграмма состояния конечного автомата **current_st1** была разработана на техническом этапе проектирования и изображена на Рис. 24.

FSM_SEL_HEADER — модуль, управляющий оформлением данных заголовком при передаче, структура и интерфейс которого были разработаны на техническом этапе проектирования.

Функциональная схема:

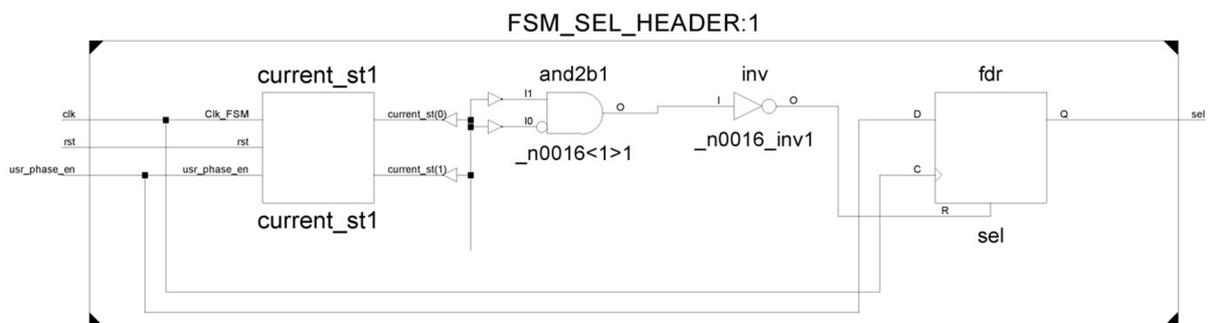


Рис. 42. Функциональная схема модуля FSM_SEL_HEADER

Диаграмма состояния конечного автомата **current_st1** была разработана на техническом этапе проектирования и изображена на Рис. 26.

PC2FPGA — Модуль, реализующий передающую часть модуля **PC_COM**, структура и интерфейс которого были разработаны на техническом этапе проектирования.

Функциональная схема:

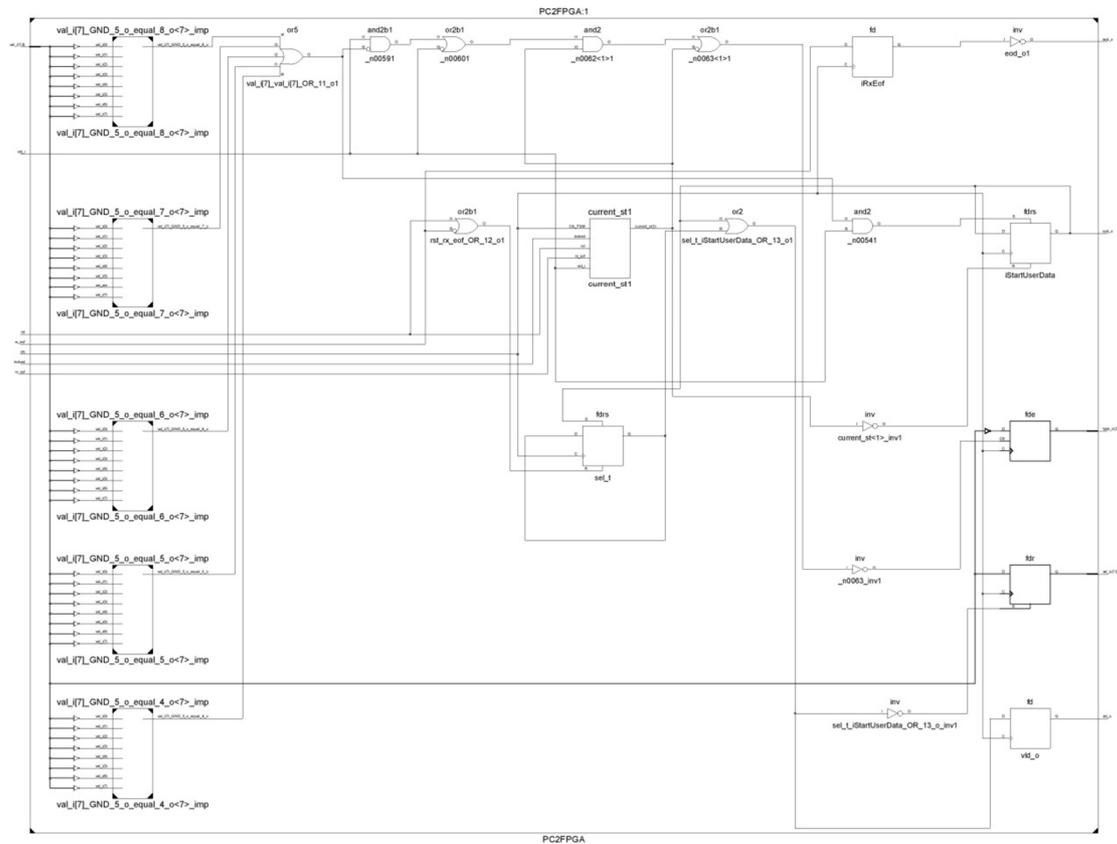


Рис. 43. Функциональная схема модуля PC2FPGA

7. Исследовательская часть.

Исходя из задачи, поставленной на проектирование, возникает требование — достижение максимальной возможной от системы скорости.

В качестве канала передачи данных в интерфейсе GМП используется восьмибитная шина, тактируема частотой 125 МГц. Итого получается 1000 Мб/с.

Это отнюдь не означает, что по этому каналу можно передать 1000 Мб/с полезных данных. Структура пакета была представлена ранее см. Таблица 12.

Состав потока на этом уровне:

- Непосредственно данные
- Заголовок пакета и контрольная сумма
- Задержка между пакетами

Полезная скорость передачи данных при этом рассчитывается по формуле.

$$V_d = \frac{D}{D+H+L} \cdot V_m \quad (1), \text{ где}$$

V_d — Полезная скорость.

D — Количество тактов, во время которых передается полезная информации, за один пакет.

H — Количество тактов, требуемых для передачи заголовка и другой необходимой информации на один пакет.

L — Задержка в тактах между пакетами.

V_m — Максимальная скорость.

Очевидно, что чем больше значение D , тем выше фактическая скорость. Величина D ограничена значением 1432 байтов в пакете, учитывая восьмибитную шину данных, получим такое же количество тактов на пакет и примем это значение для расчетов.

H и V_m — постоянные величины.

Требуется исследовать влияние задержки между пакетами на скорость передачи данных.

7.1. Исследование влияния задержки между пакетами на скорость передачи данных.

Необходимость исследования объясняется тем, что обычно системы взаимодействуют с драйверами физического уровня на канальном уровне модели OSI, а в разработанной системе этот уровень реализован с помощью программных средств и непосредственно взаимодействие с физическим драйвером происходит на уровне так называемых «сырых данных».

По формуле (1) очевидно, что чем меньше задержка, тем больше фактическая скорость передачи данных. Естественно, не все так просто, и драйверы физического уровня имеют минимальную задержку, которую необходимо выдержать при передаче потока, иначе данные могут и будут теряться.

Цель исследования — выбор оптимальной задержки. Критерием оптимальности является максимальная скорость при полной сохранности данных.

Теоретическая полезная скорость в зависимости от задержки рассчитывается по формуле (1)

График функции $V_d^t(L)$ представлен на Рис. 44.

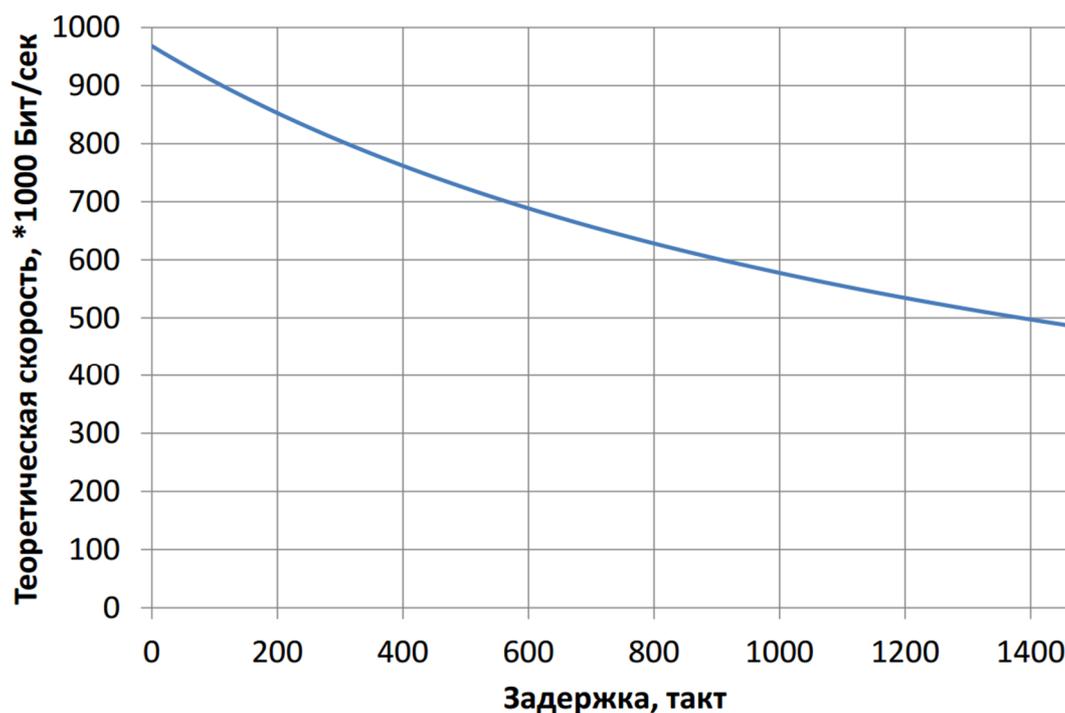


Рис. 44. Теоретическая скорость в зависимости от задержки

Для проведения исследования был поставлен ряд экспериментов, суть которых заключатся в замере времени передачи 50000 пакетов данных при разном значении задержки. Каждый эксперимент состоит из 10 опытов. Опытная выборка представлена ниже (см. Таблица 13).

Фактическая полезная скорость в зависимости от задержки рассчитывается по формуле

$$V_d^f = \frac{50000 \cdot D}{t}, \text{ где}$$

t — среднее время приема 50000 пакетов из опытной выборки

Таблица 13. Опытная выборка

Задержка \ № опыта	1	5	20	40	45	60	70	100
1	1184	1187	1199	1215	1219	615	620	632
2	1183	1187	1200	1215	1219	615	620	632
3	1184	1188	1200	1215	1219	615	620	631
4	1184	1187	1199	1215	1220	616	620	632
5	1184	1187	1199	1215	1219	615	620	632
6	1183	1188	1200	1215	1219	616	619	632
7	1184	1187	1200	1215	1219	615	620	631
8	1184	1187	1200	1215	1219	615	620	632
9	1184	1187	1200	1215	1219	615	620	632
10	1183	1187	1200	1215	1219	615	620	632
Среднее	1183,7	1187,2	1199,7	1215	1219,1	615,2	619,9	631,8

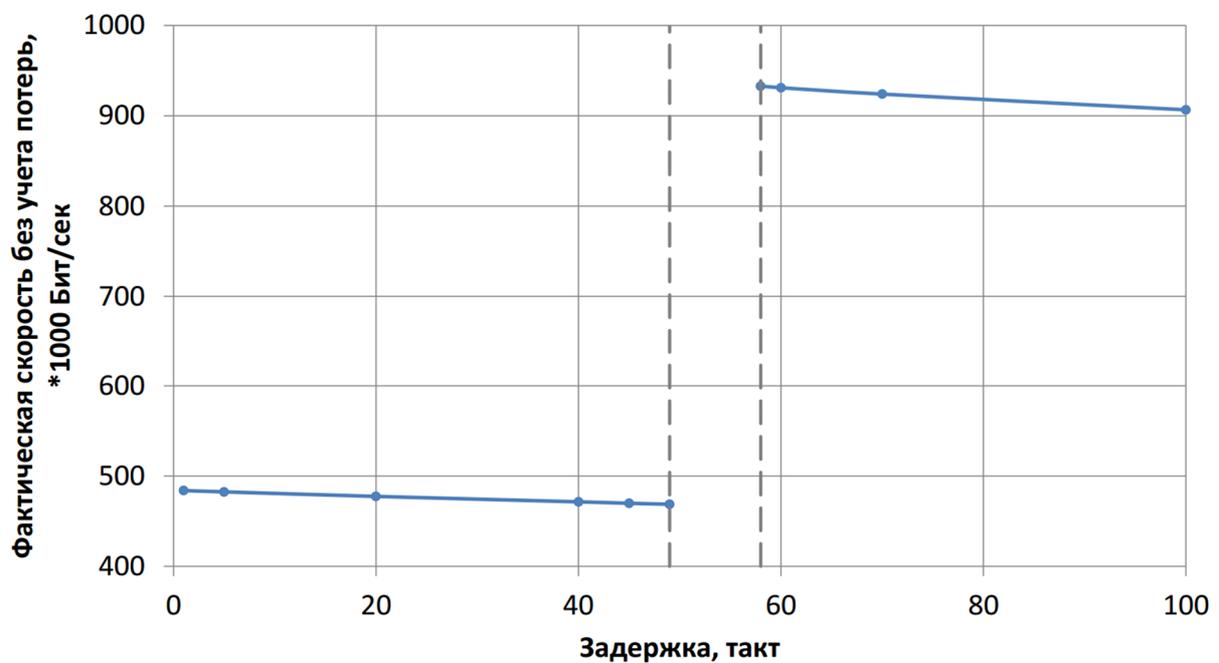


Рис. 45. Фактическая скорость в зависимости от задержки

Полученные значения в левой части графика составляют около половины от теоретических значений и примерно равны теоретической скорости при задержке, равной длине полного пакета. Это наталкивает на мысль, что при такой маленькой задержке драйвер физического уровня просто не успевает подготовиться к передаче каждого второго пакета.

Для проверки потерь данных была разработана программа, написанная на Python, которая проверяет номера пакетов и ведет статистику потерянных. Наличие отдельной программы необходимо для того, чтобы не вносить погрешности в замер скорости передачи данных. Листинг программы представлен в Приложении 4.

График, отображающий процентное соотношение количества потерянных пакетов к общему количеству, изображен на Рис. 46.

Он подтверждает предыдущую догадку о потере каждого второго пакета при передаче с маленькой задержкой. Левая часть графика имеет значение 0,499995, что соответствует потере 49999 пакетов (каждый второй при приеме 50000).

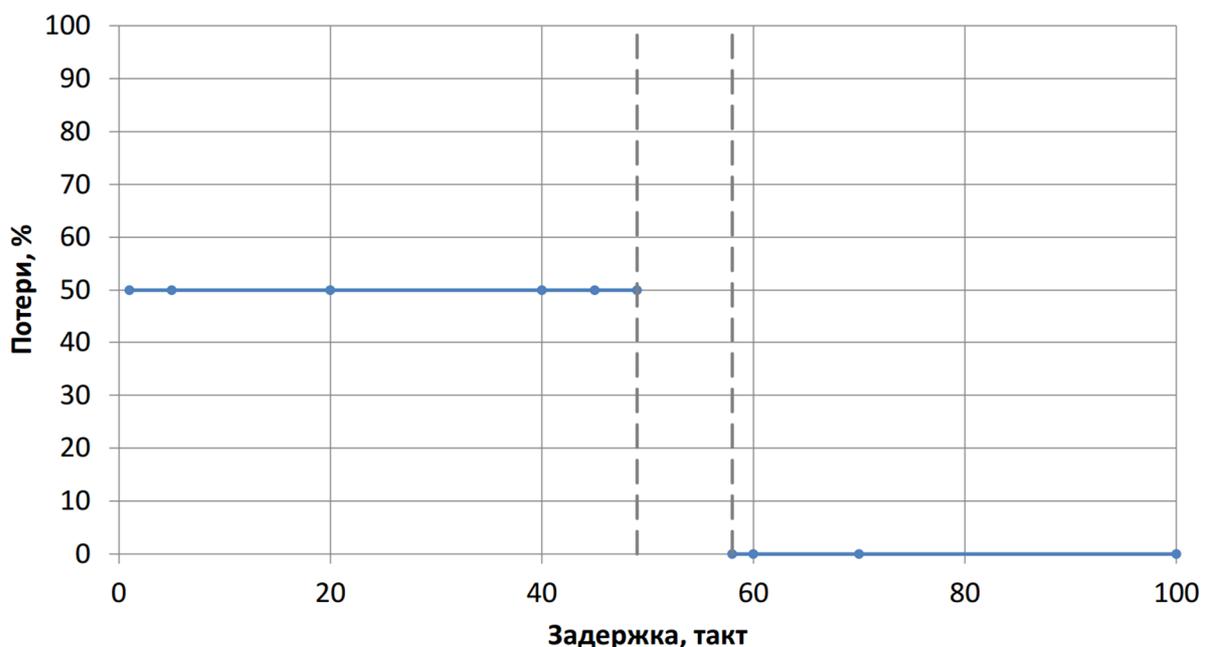


Рис. 46. Процент потерь пакетов в зависимости от задержки

На графиках отсутствует область [49,58], так как при этой задержке результаты опытов имеют выбивающиеся значения, которые нельзя считать достоверными. Объясняется это тем, что при задержке со значением из этой области передающий драйвер физического уровня уже успевает передавать данные, а вот принимающее устройство не успевает подготовиться к приему, что влечет за собой разрыв соединения. Обратная установка соединения — длительный процесс по сравнению с передачей данных, тем более при передаче 50000 пакетов повторяется он довольно часто.

Соотнесем полученные результаты с теоретическими выкладками, для этого построим график приведенной фактической скорости.

$$V_d^{pf} = \frac{V_d^f}{(1-\delta)}, \text{ где}$$

δ — отношение количества потерянных к общему количеству переданных пакетов.

На рисунке изображен график приведенной фактической скорости, наложенный на график теоретической скорости.

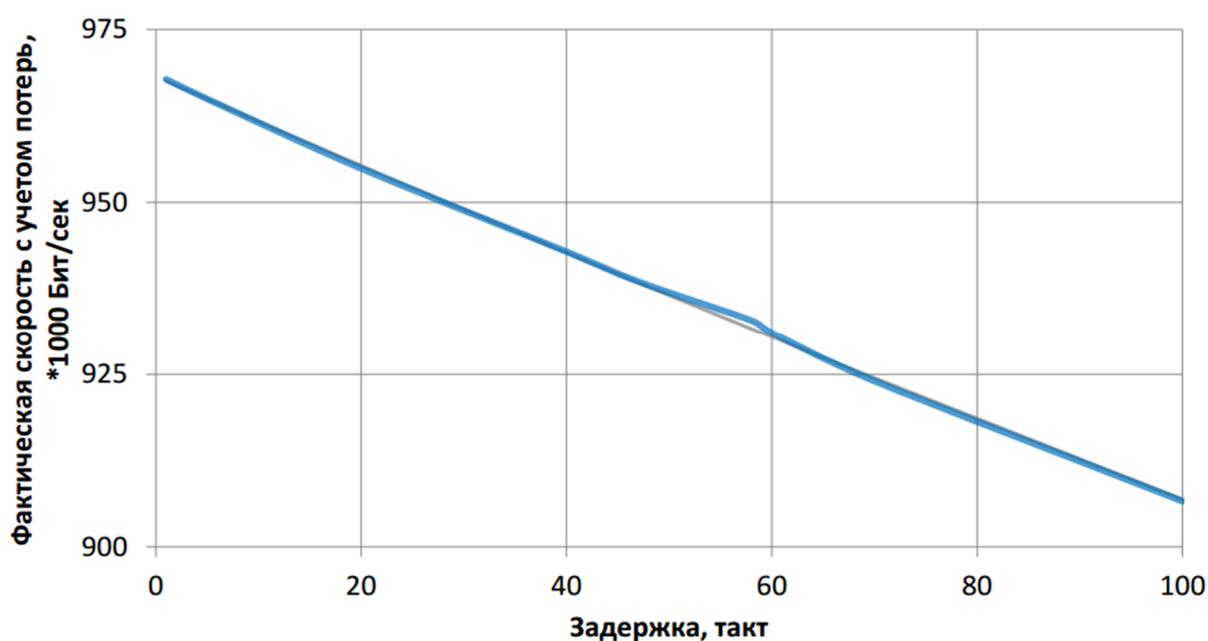


Рис. 47. Фактическая скорость в зависимости от задержки

На Рис. 47 видно, что графики фактической и теоретической скорости практически совпадают. Не нулевое отклонение объясняется тем, что время выполнения передачи 50000 пакетов измерялась с точностью до миллисекунды, а для расчета бралось среднее значение для каждого эксперимента. На графике отклонения фактической скорости от теоретической серой областью отображено отклонение, рассчитанное для верхней и нижней границ значения t .

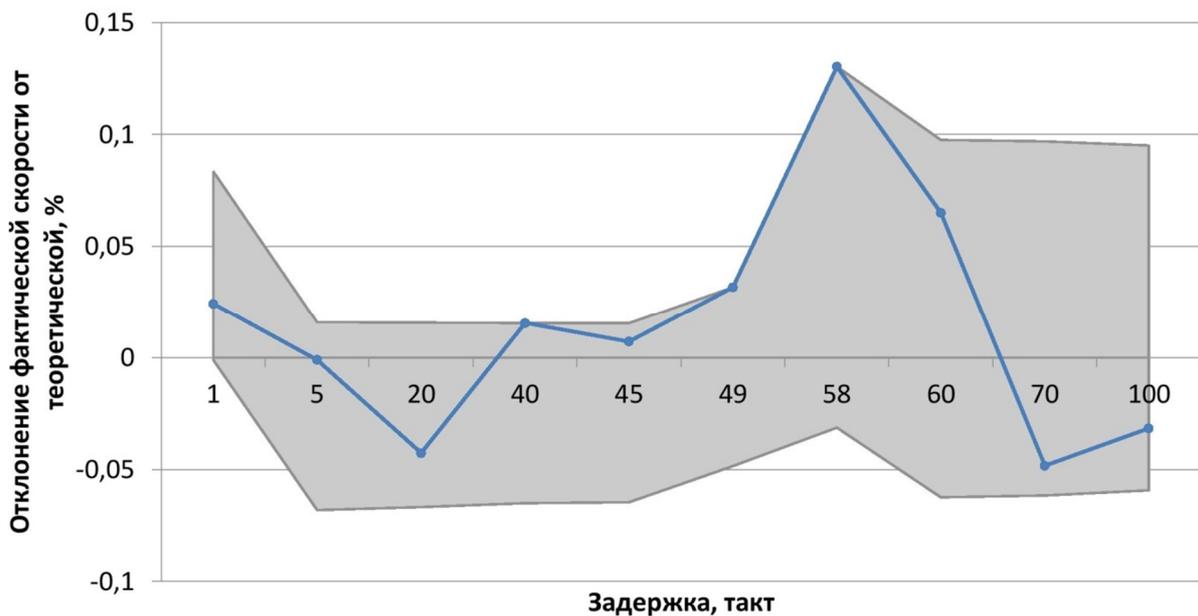


Рис. 48. Отклонение фактической скорости от теоретической.

По итогам исследования выбирается L — значение задержки между пакетами в тактах.

$$L = 60$$

$$V_d^f = 931079324 \text{ бит/с}$$

8. Апробирование.

Апробирование — опробование чего-либо с целью определения его практической пригодности, применимости или для подтверждения в действительности предполагаемых свойств, качеств, характеристик. [18]

Разработанная автоматизированная система регистрации потока данных радиолокационной станции была апробирована на различных этапах разработки с применением различных устройств в качестве подсистемы передачи и подсистемы приема и хранения данных, что подтверждает её гибкость. Система прошла приемо-сдаточные и предварительные испытания на тестовых стендах ОАО «ВНИИРТ» и ожидает на пыльных полках своей опытной эксплуатации.

Кроме того, система была апробирована на этапе исследовательской части дипломного проектирования, во время которой работа системы была проверена в работе, составляя опытную выборку. Результатом исследовательской части стал выбор оптимальной задержки между пакетами, а также подтверждение работы системы со скоростью 931079324 бит/с, что с точностью до 0,1 % совпадает с теоретической максимальной скоростью.

В Приложении 5 представлен акт о рассмотрении и проверке работоспособности программного продукта, подтверждающий выполнение возложенных на программную часть системы функций:

- Установка соединения с ПЛИС на плате
- Передача данных с ПЛИС со скоростью 1 Гбит/с
- Прием и запись на диск данных с ПЛИС со скоростью 1 Гбит/с

9. Организационно-экономическая часть.

9.1. Введение.

В организационно-экономической части определяется трудоемкость и затраты на разработку автоматизированной системы регистрации потока данных радиолокационной станции. Основной её частью является разработка программной части, а именно, программирование логики интегральной схемы платы, отвечающей за передачу данных, и разработка программы для компьютера, отвечающей за прием данных. Поэтому для организационно-экономической части дипломного проекта было получено задание рассчитать затраты на создание программного комплекса.

9.2. Трудоемкость и затраты на разработку программного комплекса автоматизированной системы регистрации потока данных радиолокационной станции.

При традиционном методе планирования организация и планирование процесса разработки программного комплекса предусматривает выполнение следующих работ:

- формирование состава выполняемых работ и группировка их по стадиям разработки
- расчет трудоемкости выполнения работ
- установление профессионального состава и расчет количества исполнителей
- определение продолжительности выполнения отдельных этапов разработки
- построение календарного графика выполнения разработки
- контроль выполнения календарного графика

Следует отметить, что процесс разработки программного продукта характеризуется совместной работой разработчиков постановки задач и разработчиков программного обеспечения.

Укрупненный состав работ по стадиям разработки программного продукта по ГОСТ 19.004-80 и 19.102-77 приведен в таблице [19].

Таблица 14. Укрупненный состав работ по стадиям разработки программного продукта

Стадия разработки программного продукта	Состав выполняемых работ
1	2
Техническое задание	Постановка задач, выбор критериев эффективности. Разработка технико-экономического обоснования разработки. Определение состава пакета прикладных программ, состава и структуры информационной базы. Выбор языков программирования. Предварительный выбор методов выполнения работы. Разработка календарного плана выполнения работ.
Эскизный проект	Предварительная разработка структуры входных и выходных данных. Разработка общего описания алгоритмов реализации решения задач. Разработка пояснительной записки. Консультации разработчиков постановки задач. Согласование и утверждение эскизного проекта.

Продолжение таблицы 14

1	2
Технический проект	<p>Разработка алгоритмов решения задач. Разработка пояснительной записки. Согласование и утверждение технического проекта. Разработка структуры программы. Разработка программной документации и передача ее для включения в технический проект. Уточнение структуры, анализ и определение формы представления входных и выходных данных. Выбор конфигурации технических средств.</p>
Рабочий проект	<p>Комплексная отладка задач и сдача в опытную эксплуатацию. Разработка проектной документации. Программирование и отладка программ. Описание контрольного примера. Разработка программной документации. Разработка, согласование программы и методики испытаний. Предварительное проведение всех видов испытаний.</p>
Внедрение	<p>Подготовка и передача программной документации для сопровождения с оформлением соответствующего Акта. Передача программной продукции в фонд алгоритмов и программ. Проверка алгоритмов и программ решения задач, корректировка документации после опытной эксплуатации программного продукта.</p>

Трудоемкость разработки программной продукции зависит от ряда факторов, основными из которых являются следующие: степень новизны разрабатываемого программного комплекса, сложность алгоритма его функционирования, объем используемой информации, вид ее представления и способ обработки, а также уровень используемого алгоритмического языка программирования. Чем выше уровень языка, тем трудоемкость меньше.

Таблица 15. Факторы трудоемкости программного продукта

Степень новизны	Группа В - Программная продукция, имеющая аналоги
Степени сложности алгоритма функционирования программной продукции	Группа 2 - Программная продукция, реализующая учетно-статистические алгоритмы
Вид представления исходной информации и способы ее контроля	Группа 12 - исходная информация представлена в форме документов, имеющих одинаковый формат и структуру. Требуется формальный контроль информации Группа 22 - требуется вывод на печать одинаковых документов, вывод информационных массивов на машинные носители

Трудоемкость разработки программной продукции

τ_{PP} может быть определена как сумма величин трудоемкости выполнения отдельных стадий разработки программного продукта из выражения:

$$\tau_{PP} = \tau_{TZ} + \tau_{EP} + \tau_{TP} + \tau_{RP} + \tau_V, \text{ где}$$

τ_{TZ} — трудоемкость разработки технического задания на создание программного продукта;

τ_{EP} — трудоемкость разработки эскизного проекта программного продукта;

τ_{TP} — трудоемкость разработки технического проекта программного продукта;

τ_{RP} — трудоемкость разработки рабочего проекта программного продукта;

τ_V — трудоемкость внедрения разработанного программного продукта.

9.2.1. Трудоемкость разработки технического задания

$$\tau_{TZ} = T_{RZ}^Z + T_{RP}^Z, \text{ где}$$

T_{RZ}^Z — затраты времени разработчика постановки задачи на разработку ТЗ, [чел.-дни];

T_{RP}^Z — затраты времени разработчика программного обеспечения на разработку ТЗ, [чел.-дни].

Их значения рассчитываются по формулам:

$$T_{RZ}^Z = t_Z \cdot K_{RZ}^Z$$

$$T_{RP}^Z = t_Z \cdot K_{RP}^Z, \text{ где}$$

t_Z — норма времени на разработку ТЗ на программный продукт в зависимости от функционального назначения и степени новизны разрабатываемого программного продукта.

$$t_Z = 61 \text{ [чел.-дни]} \text{ (Статистические задачи, группа новизны В)}$$

K_{RZ}^Z — коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки задачи на стадии ТЗ. В нашем случае разработчик постановки задач и разработчик программного обеспечения являются одним лицом, поэтому здесь и далее считаем, что

разработка проводится совместно разработчиком постановки задач и разработчиком программного обеспечения.

$$K_{RZ}^Z = 0,65$$

K_{RP}^Z — коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком программного обеспечения на стадии ТЗ.

$$K_{RP}^Z = 0,35$$

Тогда:

$$\tau_{TZ} = 61 \cdot (0,35 + 0,65) = 61 \text{ [чел.-дни]}$$

9.2.2. Трудоемкость разработки эскизного проекта

τ_{EP} рассчитывается по формуле:

$$\tau_{EP} = T_{RZ}^E + T_{RP}^E, \text{ где}$$

T_{RZ}^E — затраты времени разработчика постановки задачи на разработку эскизного проекта (ЭП), [чел.-дни];

T_{RP}^E — затраты времени разработчика программного обеспечения на разработку ЭП, [чел.-дни].

Их значения рассчитываются по формулам:

$$T_{RZ}^E = t_E \cdot K_{RZ}^E$$

$$T_{RP}^E = t_E \cdot K_{RP}^E, \text{ где}$$

t_E — норма времени на разработку ЭП на программный продукт в зависимости от функционального назначения и степени новизны разрабатываемого программного продукта.

$$t_E = 72 \text{ [чел.-дни]} \text{ [19, Табл. 3]}$$

K_{RZ}^E — коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки задачи на стадии ЭП.

$$K_{RZ}^E = 0,7$$

K_{RP}^E — коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком программного обеспечения на стадии ЭП.

$$K_{RP}^E = 0,3$$

Тогда:

$$\tau_{EP} = 72 \cdot (0,3 + 0,7) = 72 \text{ [чел.-дни]}$$

9.2.3. Трудоемкость разработки технического проекта

τ_{TP} рассчитывается по формуле:

$$\tau_{TP} = (t_{RZ}^T + t_{RP}^T) \cdot K_V \cdot K_R, \text{ где}$$

t_{RZ}^T — норма времени, затрачиваемого на разработку технического проекта (ТП) разработчиком постановки задач;

t_{RP}^T — норма времени, затрачиваемого на разработку ТП разработчиком ПО.

В зависимости от количества разновидностей форм выходной информации, количества разновидностей форм входной информации и функционального назначения:

$$t_{RZ}^T = 23 \text{ [чел.-дни]}$$

$$t_{RP}^T = 10 \text{ [чел.-дни]} \text{ [19, Табл. 15]}$$

K_R — коэффициент учета режима обработки информации. Информацию необходимо обрабатывать в реальном времени:

$$K_R = 1,26 \text{ [19, Табл. 17]}$$

K_V — коэффициент учета вида используемой информации, определяется по формуле:

$$K_V = \frac{K_P \cdot n_P + K_{NS} \cdot n_{NS} + K_B \cdot n_B}{n_P + n_{NS} + n_B}, \text{ где}$$

K_P — коэффициент учета вида используемой информации для переменной информации;

K_{NS} — коэффициент учета вида используемой информации для нормативно-справочной информации;

K_B — коэффициент учета вида используемой информации для баз данных.

Группа новизны - В:

$$K_P = 1,00$$

$$K_{NS} = 0,72$$

$$K_B = 2,08$$

n_P — количество наборов данных переменной информации;

n_{NS} — количество наборов данных нормативно-справочной информации;

n_B - количество наборов баз данных.

В нашем случае:

$$n_P = 1$$

$$n_{NS} = 0$$

$$n_B = 0$$

Находим значение K_V :

$$K_V = \frac{1,00 \cdot 1 + 1 \cdot 0 + 2,08 \cdot 0}{1 + 0 + 0} = 1,0$$

Тогда:

$$\tau_{TP} = (23+10) \cdot 1,0 \cdot 1,26 = 41,58 \text{ [чел.-дни]}$$

9.2.4. Трудоемкость разработки рабочего проекта

τ_{RP} зависит от функционального назначения программного продукта, количества разновидностей форм входной и выходной информации, сложности алгоритма функционирования, сложности контроля информации, степени использования готовых программных модулей, уровня алгоритмического языка программирования и определяется по формуле:

$$\tau_{RP} = K_K \cdot K_R \cdot K_Y \cdot K_Z \cdot K_{IA} \cdot (t_{RZ}^R + t_{RP}^R), \text{ где}$$

K_K - коэффициент учета сложности контроля информации.

$$K_K = 1,00$$

K_R - коэффициент учета режима обработки информации

$$K_R = 1,32 \text{ [19, Табл. 17]}$$

K_Y — коэффициент учета уровня используемого алгоритмического языка программирования. Принимаем значение:

$$K_Y = 1,00 \text{ (язык программирования высокого уровня). [19, Табл. 20]}$$

K_Z — коэффициент учета степени использования готовых программных модулей.

$K_Z = 0,7$ (использование готовых программных модулей составляет около 40% (60% для программы приемника, 20% для программы отправителя)). [19, Табл. 21]

K_{IA} — коэффициент учета вида используемой информации и сложности алгоритма программного продукта, его значение определяется по формуле:

$$K_{IA} = \frac{K_P' \cdot n_P + K_{NS}' \cdot n_{NS} + K_B' \cdot n_B}{n_P + n_{NS} + n_B}$$

K_P' — коэффициент учета сложности алгоритма ПП и вида используемой информации для переменной информации;

K_{NS}' — коэффициент учета сложности алгоритма ПП и вида используемой информации для нормативно-справочной информации;

K_B' — коэффициент учета сложности алгоритма ПП и вида используемой информации для баз данных.

$$K_P' = 1,10 \text{ [19, Табл. 22]}$$

$$K_{NS}' = 0,58$$

$$K_B' = 0,48$$

$$n_P = 1$$

$$n_{NS} = 0$$

$$n_B = 0$$

$$K_{IA} = \frac{1,10 \cdot 1 + 0,58 \cdot 0 + 0,48 \cdot 0}{1 + 0 + 0} = 1,10$$

t_{RZ}^R — норма времени, затраченного на разработку рабочего проекта при единственной форме входной информации.

$$t_{RZ}^R = 8 \text{ [чел.-дни] [19, Табл. 34]}$$

t_{RP}^R — норма времени, затраченного на разработку рабочего проекта при единственной форме входной информации.

$$t_{RP}^R = 54 \text{ [чел.-дни] [19, Табл. 34]}$$

Аналогично предыдущему пункту, принимаем суммарный норматив времени

$$\tau_{RP} = 1,00 \cdot 1,32 \cdot 1,00 \cdot 0,7 \cdot 1,10 \cdot (54 + 8) = 63,0168 \text{ [чел.-дни]}$$

9.2.5. Трудоемкость выполнения стадии “Внедрение”

Рассчитывается по формуле:

$$\tau_V = (t_{RZ}^V + t_{RP}^V) \cdot K_K \cdot K_R \cdot K_Z, \text{ где:}$$

t_{RZ}^V — норма времени, затрачиваемого разработчиком постановки задач на выполнение процедур внедрения программного продукта.

t_{RP}^V — норма времени, затрачиваемого разработчиком программного обеспечения на выполнение процедур внедрения программного продукта.

$$t_{RZ}^V = 8 \text{ [чел.-дни]}$$

$$t_{RP}^V = 8 \text{ [чел.-дни] [19, Табл. 37]}$$

Коэффициенты K_K , K_Z были найдены выше:

$$K_K = 1,16$$

$$K_Z = 0,7$$

K_R — коэффициент учета режима обработки информации. Для группы новизны — В, режима обработки информации в реальном масштабе времени:

$$K_R = 1,21$$

$$\tau_V = (8+8) \cdot 1,16 \cdot 0,7 \cdot 1,21 = 15,72032 \text{ [чел.-дни]}$$

Общая трудоемкость разработки ПП:

$$\tau_{PP} = 61 + 72 + 41,58 + 63,0168 + 15,72032 = 253,31712 \text{ [чел.-дни]}$$

Примем $\tau_{PP} = 253$ [чел.-дни]

9.3. Определение цены программной продукции.

Для определения стоимости работ необходимо на основании плановых сроков выполнения работ и численности исполнителей рассчитать общую сумму затрат на разработку программного продукта.

Если ПП рассматривается и создается как продукция производственно-технического назначения, допускающая многократное тиражирование и отчуждение от непосредственных разработчиков, то ее цена определяется по формуле:

$$C = K \times C + Pr, \text{ где}$$

C — затраты на разработку ПП (сметная себестоимость)

K — коэффициент учёта затрат на изготовление опытного образца ПП как продукции производственно-технического назначения ($K=1,1$).

В нашем случае ПП изготавливается для внутренних нужд предприятия, поэтому $Pr = 0$.

Затраты, образующие себестоимость продукции (работ, услуг), группируются в соответствии с их экономическим содержанием по следующим элементам:

- амортизация нематериальных активов и оборудования;
- затраты на оплату труда;
- отчисления на социальные нужды;
- прочие затраты.

9.3.1. Расчет амортизации нематериальных активов и оборудования.

В данной статье учитываются суммарные затраты на приобретение/амортизацию оборудования и нематериальных активов, требуемых для разработки данного программного продукта. Цены указаны по состоянию на июнь 2014 года.

$$C_{co} = \sum_i \frac{C_{6i} \alpha_i}{100 F_d} t_i,$$

C_{6i} — балансовая цена i -ого вида оборудования, руб.;

α_i — норма годовых амортизационных отчислений для оборудования i -го вида, %;

F_d — действительный годовой фонд времени, ч;

t_i — время использования i -ого вида оборудования при выполнении данной разработки, ч.

Годовой фонд рабочего времени на ПК (5-ти дневная неделя, 8-и часовой рабочий день) — 2080 ч.

Исходные данные для расчетов см. Таблица 16

Затраты на оборудование:

$$C_{co1} = (51000 + 15000 + 250 + 790) \cdot 20 \cdot 253 \cdot 8 / (100 \cdot 2080) = 13047 \text{ [руб.]}$$

Затраты на ПО:

Технологическая платформа используется все время при работе разрабатываемого ПО.

$$C_{co2} = (35820 + 50364) \cdot 20 \cdot 253,31712 / (100 \cdot 2080) = 16772,7 \text{ [руб.]}$$

Таблица 16. Стоимость оборудования

Наименование	Ед. изм.	Кол.	Цена за единицу, руб	Сумма, руб
ПК (Поставляется с ОС Microsoft Windows 7 Pro)	Шт	1	51000	51000
Монитор	Шт	2	7500	15000
Компьютерная мышь	Шт	1	250	250
Клавиатура	Шт	1	790	790
Годовая лицензия Xilinx Vivado System Edition	Шт	1	35820	35820
Годовая лицензия Synopsys Identify RTL Debugger	Шт	1	50364	50364

Тогда суммарные затраты на оборудование и нематериальные активы составят:

$$C_{co} = 13047 + 16772,7 = 29819,7 \text{ [руб.]}$$

9.3.2. Расчет основной заработной платы.

В данную статью включаются основная заработная плата всех исполнителей, непосредственно занятых разработкой данного программного продукта с учетом их должностных окладов и времени участия. Расчет проводится по формуле:

$$C_{zo} = \sum Z_i \cdot \tau_i / d, \text{ где}$$

Z_i — месячный оклад i -го исполнителя, [руб.]

τ_i — трудоемкость работ, выполняемых i -м исполнителем, [чел.-дни]

- определяются из календарного плана-графика;

d — среднее количество рабочих дней в месяце. Принимаем:

$$d = 21 \text{ день}$$

Расчет затраты на оплату труда исполнителя:

$$C_{zo1} = 41000 \cdot 253 / 21 = 493952 \text{ [руб.]}$$

Суммарная заработная плата равна:

$$C_{zo} = 493952 \text{ [руб.]}$$

9.3.3. Расчет дополнительной заработной платы.

В данной статье также учитываются выплаты непосредственным исполнителям за время, не проработанное на производстве, в том числе: оплата очередных отпусков, компенсация за недоиспользованный отпуск, оплата льготных часов подросткам и др. Дополнительная заработная плата рассчитывается по формуле:

$$C_{zd} = C_{zo} \cdot A_{\partial},$$

где A_{∂} - коэффициент отчислений на дополнительную заработную плату.

$$A_{\partial} = 0.2$$

$$C_{zd} = 493952 \cdot 0,2 = 98790,4 \text{ [руб.]}$$

9.3.4. Отчисления на социальные нужды.

В статье учитываются отчисления в бюджет социального страхования по установленному законодательством тарифу от суммы основной и дополнительной заработной платы. Расчет производится следующим образом:

$$C_{cc} = A_{cc} \cdot (C_{zo} + C_{zd}),$$

где A_{cc} — коэффициент отчислений на социальное страхование.

0,22 — отчисления в пенсионный фонд;

0,029 — в фонд социального страхования;

0,051 — в фонд медицинского страхования;

0,004 — страхование от несчастных случаев.

$$A_{cc} = 0,304$$

$$C_{cc} = 0,304 \cdot (493952 + 98790,4) = 180194 \text{ [руб.]}$$

9.3.5. Накладные расходы.

В данную статью входят другие затраты, входящие в состав себестоимости продукции (работ, услуг), но не относящиеся к ранее перечисленным элементам затрат.

$$C_n = A_n \cdot C_{зо},$$

где:

A_n — коэффициент накладных расходов.

$$A_n = 1,8$$

$$C_n = 1.8 \cdot 493952 = 889114 \text{ [руб.]}$$

9.4. Итоговые результаты.

Результаты расчетов затрат на разработку программного продукта приведены в табл.

Таблица 17. Результаты расчетов

Наименование статьи	Сметная стоимость, руб.
Затраты на нематериальные активы и оборудование	29 819,7
Затраты на оплату труда	493 952
Дополнительная заработная плата	98 790,4
Отчисления в ФСС	180 194
Накладные расходы	889 114
Итого затраты на разработку ПП.	1 691 870

Цена создания:

$$Ц = K \cdot C + Пр$$

где C — затраты на разработку ПП.

K — коэффициент учета затрат на изготовление опытного образца ПП как продукции производственно-технического назначения ($K=1,1$).

Пр — нормативная прибыль. Примем её равную нулю, так как разработка ведется по заказу военных.

$$Ц = 1,1 \cdot 1691870 + 0 = 1861057 \text{ [руб.]}$$

10. Охрана труда и экология.

10.1. Анализ опасных и вредных факторов, воздействующих на инженера при работе с регистратором данных.

Рассмотрим опасные и вредные факторы, действующие на инженера ОАО «ВНИИРТ» при работе по регистрации данных радиолокационного потока на РЛС Фуркэ-Э на корвете проекта 20380. В таблице 18 представлены опасные и вредные факторы [20, 21]

Таблица 18. Опасные и вредные факторы согласно ГОСТ 12.0.003-74

№	Фактор	Опасный	Вредный
	Физические		
1	Движущиеся машины и механизмы	+	-
2	Повышенная или пониженная температура воздуха рабочей зоны	-	+
3	Повышенная или пониженная подвижность воздуха	-	+
4	Повышенная или пониженная влажность воздуха	-	+
5	Повышенный уровень вибрации	-	+
6	Недостаточная освещенность рабочей зоны	-	+
7	Отсутствие или недостаток естественного света	-	+
8	Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека	+	-
9	Повышенный уровень электромагнитных излучений	-	+
	Психофизиологические		
10	Монотонность труда	-	+
11	Эмоциональные перегрузки	-	+

10.1.1. Движущиеся машины и механизмы.

Причины возникновения:

- Ремонтные и погрузочные работы
- Поворотные механизмы антенного полотна

Влияние на человека — возможность травмы, т.е. нарушения анатомической целостности или физиологических функций тканей или органов человека, вызванное внезапным внешним воздействием, вследствие получения удара или захвата движущимися частями.[22]

Предлагаемые меры по обеспечению безопасности:

- Проведение инструктажа, объясняющего возможные источники опасности
- Проведение обучения навыкам безопасного труда и поведения
- Размещение различных плакатов, предупредительных сигналов, маркировки и окраски, а также знаков безопасности
- Размещение коллективных устройств защиты оградительных и сигнальных
- Оснащение индивидуальными средствами защиты: касками

10.1.2. Повышенная или пониженная температура воздуха рабочей зоны.

Причина возникновения:

Работа по регистрации радиолокационных данных проводится в плохо отапливаемом помещении военного корвета. Температура на рабочем месте зависит от температуры окружающей среды. Металлическая обшивка корпуса корвета способствует повышению температуры в солнечную погоду в теплое время года.

Влияние на человека:

- Работа в условиях пониженной температуры опасна обморожениями и простудными заболеваниями, требует больших затрат энергии человека, подрывает иммунитет. Пониженная температура воздуха также отрицательно влияет на работу мышц, снижает быстроту и точность движений, следовательно, человек более подвержен механическим опасным факторам.
- Работа в условиях повышенной температуры воздуха быстрее утомляет человека, снижает его внимание, увеличивает время реакции, плохо влияет на состояние кровеносно-сосудистой системы человека

Согласно СанПиН 2.2.2/2.4.1340-03:

- Категория работ — Легкая — 1а
- Рабочее место — не постоянное

Таблица 19. Оптимальные и допустимые нормы температуры воздуха в рабочей зоне производственных помещений по ГОСТ 12.1.005-88.

Период года	Температура		
	Оптимальная	Допустимая на непостоянных рабочих местах	
		Верхняя граница	Нижняя граница
Холодный	22-24	25	21
Теплый	23-25	28	22

Предлагаемые меры по обеспечению безопасности:

- Оборудование рабочего места средствами измерения температуры.
- Проведение инструктажей по защите труда.

- Создание и донесение до сотрудников регламента о порядке действий в случае понижения температуры.
- Обеспечение средствами личной защиты: утепленная одежда в холодный период года.
- Обеспечение средствами групповой защиты: размещение отопительных элементов в рабочем месте в холодный период, средств кондиционирования и вентиляции в теплый.

10.1.3. Повышенная или пониженная подвижность воздуха.

Причиной возникновения является то, что естественная вентиляция при открытых люках помещения с рабочим местом сильно зависит от метеобстановки за бортом.

Согласно СанПиН 2.2.2/2.4.1340-03:

- Категория работ — Легкая — 1а
- Рабочее место — не постоянное

Таблица 20. Скорость движения воздуха

Период года	Скорость движения, м/с	
	Оптимальная, не более	Допустимая, не более
Холодный	0,1	0,1
Теплый	0,1	0,2

Влияние на человека:

Скорость движения потоков воздуха оказывает большое влияние на теплорегулирование организма. Человек начинает ощущать воздушные потоки при скорости движения воздуха 0,25 м/с. Увеличение скорости движения воздуха значительно увеличивает теплоотдачу с поверхности тела, тем самым усиливая вредное влияние пониженной температуры, а также снижая «ощущаемую температуру» рабочего пространства.

Пониженная скорость воздуха не регламентирована, однако, свидетельствует о возможном отсутствии вентиляции помещения.

Предлагаемы меры по обеспечению безопасности:

- Проведение обучающих мероприятий или инструктажа, посвященного защите труда
- Коллективные меры защиты: Установка системы искусственной вентиляции воздуха

10.1.4. Повышенная или пониженная влажность воздуха.

Причина возникновения:

Микроклимат рабочего места сильно зависит от метеосостояния за бортом в условиях водного судна — влажный соленый воздух.

Согласно СанПиН 2.2.2/2.4.1340-03:

- Категория работ — Легкая — 1а
- Рабочее место — не постоянное

Таблица 21. Влажность воздуха

Период года	Относительная влажность, %	
	оптимальная	Допустимая, не более
Холодный	40-60	75
Теплый	40-60	75, 55 (при 30°C)

Влияние на человека:

Большая относительная влажность воздуха при высокой температуре может быть причиной перегрева тела человека. В воздухе, насыщенном водяными парами, теплоотдача путем испарения невозможна или затруднена. Высокая влажность воздуха оказывает неблагоприятное

воздействие на организм человека также и при низкой температуре воздуха из-за повышенной теплоотдачи с поверхности тела.

Предлагаемы меры по обеспечению безопасности:

- Коллективные меры защиты: Установка системы кондиционирования воздуха

10.1.5. Повышенный уровень вибрации

Источники вибрации:

- Гребные винты
- Двигатель, механизмы проворачивания
- Удары волн
- Поворотные и подъёмные механизмы
- Ремонтные работы

Влияние на человека:

В результате действия вибрации развивается профессиональное заболевание - вибрационная болезнь.

Особенно опасно совпадение частоты вибрации с собственной частотой колебания тела человека или отдельных органов.

Для стоящего человека резонансными являются частоты 5-15 Гц, для сидящего — 4-6 Гц, собственная частота желудка составляет 2 Гц, сердца и печени — 4 Гц, мозга — 6-7 Гц.

Вредность фактора зависит от частоты воздействующей вибрации.

Низкочастотные (до 35 Гц). При этом поражаются нервы, мышцы, костный аппарат.

Высокочастотные (100 - 150 - 250 Гц). Поражаются кровеносные сосуды. Как следствие: Боль в органах (печень, селезенка), тошнота, рвота, нарушение сна, головокружение из-за нарушения работы вестибулярного аппарата.[25]

Предлагаемы меры по обеспечению безопасности:

- Проведение инструктажа и обучения
- Обеспечение медикаментозными средствами защиты
- Обеспечение групповой защиты: успокоители качки, центровка и уравнивание двигателей, виброизоляция
- Обеспечение косвенной групповой защиты: усиленная вентиляция, питание, физическая активность
- Обеспечение средствами индивидуальной защиты: специальная обувь, виброизоляционные кресла

10.1.6. Недостаточная освещенность рабочей зоны.

Причины возникновения:

Рабочие места при проектировании не подразумевало оснащение ПЭВМ, оформлялись без проверки на соответствие стандарту.

Регламентирующий документ:

СанПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы"

Требования к освещению на рабочих местах, оборудованных ПЭВМ:

Освещенность на рабочем столе:	300-500 лк
Освещенность на экране ПЭВМ:	не выше 300лк
Блики на экране:	не выше 40 кд/м ²
Прямая блесккость источника света:	200 кд/м ²
Показатель ослепленности:	не более 20
Показатель дискомфорта:	не более 15
Отношение яркости	
- между рабочими поверхностями:	3:1-5:1
- между поверхностями стен и оборудования:	10:1
Коэффициент пульсации:	не более 5%.

Влияние на человека:

Недостаток освещенности влияет на функционирование зрительного аппарата человека вплоть до нарушения зрительной функции, оптико-вегетативную систему, на психику, его эмоциональное состояние, утомляет центральную нервную систему, из-за чего человек становится раздражительным. [26**Ошибка! Источник ссылки не найден.**]

Предлагаемые меры по обеспечению безопасности:

- Произвести расчет освещения в помещении офиса, скорректировать существующее освещение в соответствии с результатами.
- Провести инструктаж и обучение персонала

10.1.7. Отсутствие или недостаток естественного света.

Причины возникновения:

Специфика военного водного судна ограничивает естественный свет в зоне рабочего места по регистрации данных радиолокационного потока.

Влияние на человека:

Отсутствие естественного света влияет на психику, его эмоциональное состояние, человеческий глаз в ходе эволюции привык к солнечному свету, освещённость помещения, значительно отличающаяся от естественного, плохо влияет на функционирование зрительного аппарата человека вплоть до нарушения зрительной функции.[27]

Предлагаемые меры по обеспечению безопасности:

- Оборудование помещения искусственным освещением.
- Провести инструктаж и обучение персонала

10.1.8. Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.

Причины возникновения:

- В процессе работы по регистрации данных радиолокационного потока используются приборы, требующие электропитания от сети переменного тока с номинальным напряжением 220В 50МГц.
- В непосредственной близости от рабочего места используются приборы с номинальным напряжением 380В 400МГц
- Приборы имеют металлические кожухи, которые являются потенциально опасными местами, в результате соприкосновения с которыми человек может попасть в электрическую цепь.

Влияние на человека:

Степень опасного и вредного воздействия на человека электрического тока зависит от силы тока, напряжения, рода тока, частоты электрического тока и пути прохождения через тело человека, продолжительности воздействия и условий внешней среды. [28]

Воздействие электрического тока на человека:

- Тепловое (термическое) действие проявляется в виде ожогов участка кожи, перегрева различных органов, а также возникающих в результате перегрева разрывов кровеносных сосудов и нервных волокон.
- Химическое (электролитическое) действие ведет к электролизу крови и других, содержащихся в организме человека растворов, что приводит к изменению их физико-химических составов, а значит к нарушению нормального функционирования организма.
- Биологическое действие проявляется в опасном возбуждении живых клеток и тканей организма, в результате чего они могут погибнуть.

Степень опасного воздействия на человека электрического тока зависит от нескольких параметров:

- Параметры электрического тока, протекающего через тело человека (величины напряжения, частоты, рода тока приложенного к телу)
- Путь тока через тело человека (рука-рука, рука-нога, нога-нога, шея-ноги и др.)
- Продолжительность воздействия тока на тело человека
- Условия внешней среды (влажность и температура)
- Состояние организма человека (толщина и влажность кожного покрова, состояние здоровья и возраст)

Опасное воздействие на людей электрического тока проявляется в виде электрических ударов и электротравм. [29**Ошибка! Источник ссылки не найден.**]

Предлагаемые меры по обеспечению безопасности:

- Инструктаж сотрудников по организации безопасной работы с электроприборами
- Создание регламента по обслуживанию электроприборов с целью своевременного определения ненадежных соединений и возможных мест замыкания на корпус
- Оборудование защитного заземления/зануления
- Установка устройств защитного отключения
- Размещение различных плакатов, предупредительных сигналов, маркировки и окраски, а также знаков безопасности

10.1.9. Повышенный уровень электромагнитных излучений.

Причина возникновения:

- Рабочие циклы РЛС судна
- Электропроводка и приборы с большой потребляемой мощностью.

Регламентирующие документы:

- СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

Таблица 22. Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах

Наименование параметров		ВДУ
Напряженность электрического поля	в диапазоне частот 5 Гц - 2 кГц	25 В/м
	в диапазоне частот 2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл
	в диапазоне частот 2 кГц - 400 кГц	25 нТл
Напряженность электростатического поля		15 кВ/м

- СанПиН 2.2.4.1191-03 "Электромагнитные поля в производственных условиях"

Требования к проведению контроля уровней электромагнитного поля диапазона радиочастот ≥ 10 кГц - 300 ГГц

ПДУ энергетических экспозиций ЭМП диапазона частот 300 МГц - 300 ГГц

$$ЭЭ_{ППЭ}, (\text{мкВт}/\text{см}^2) * \text{ч} — 200$$

Максимальные допустимые уровни напряженности электрического и магнитного полей, плотности потока энергии ЭМП не должны превышать значений:

Максимальные ПДУ напряженности и плотности потока энергии ЭМП диапазона частот 300 МГц - 300 ГГц

ППЭ, мкВт/см² — 1000

Влияние на человека:

Воздействие мощных электромагнитных полей на человека приводит к определенным сдвигам в нервно-психической и физиологической деятельности.

В электролитах, которыми являются жидкие составляющие тканей, крови, межклеточной жидкости и т. п., после приложения внешнего поля появляются ионные токи. Переменное электрическое поле вызывает нагрев тканей тела человека, как за счет переменной поляризации диэлектриков, так и за счет появления токов проводимости. [31]

Предлагаемые меры по обеспечению безопасности:

- Проведение инструктажа и обучения
- Размещение различных плакатов, предупредительных сигналов, маркировки и окраски, а также знаков безопасности
- Рациональное расположение излучающих устройств и, установление безопасных секторов работы
- Обеспечение групповой защиты: установка экранирующих защитных конструкций, ограничение опасной зоны

10.1.10. Монотонность труда.

Причина появления - однообразные, повторяющиеся монотонные действия, входящие в обязанности инженера при регистрации данных.

Влияние на человека:

Установлено, что монотонный труд вызывает, прежде всего, изменения в функциональном состоянии ЦНС, что проявляется в удлинении латентного периода простой и сложной зрительно-моторной реакции, увеличении процента расторможенных дифференцировок, замедлении способности к переключению внимания, снижению подвижности основных нервных процессов.[32]

Предлагаемые меры по обеспечению безопасности:

- Регламентированные перерывы в работе
- Проведение инструктажа
- Поощрение выполнения профильной работы, не входящей в обязанности, по договоренности с непосредственным начальником

10.1.11. Эмоциональные перегрузки.

Причины появления:

- Режим секретности
- Дефицит времени
- Большой объем и плотность информации
- Ответственность работы на военном объекте
- Эмоциональный груз других вредных факторов влияющих на эмоциональный фон человека

Влияние на человека:

Ускоряет и обостряет восприимчивость к воздействию вредных факторов окружающей и производственной среды, нарушает обмен веществ, ведет к изнашиванию и старению организма.

Меры по обеспечению безопасности:

- Введение регламентированных перерывов в работе
- Создание комфортного и эргономичного рабочего места
- Поощрение занятия спортом, йогой и другими занятиями, способствующими снятию эмоционального напряжения.

10.2. Расчет освещенности в помещении.

Задачей расчета является определение потребной мощности электрической осветительной установки для создания в помещении с

рабочим местом заданной освещенности или, при известном числе и мощности ламп, определение ожидаемой освещенности на рабочей поверхности.

Параметры помещения:

- Длина $A = 8$ м, Ширина $B = 5$ м, Высота $H = 2,75$ м
- Высота рабочей поверхности над полом: $h = 0,8$ м;
- Высота подвеса светильников: $T = 0,07$ м.

Требования стандарта СанПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы":

Таблица 23. Требования к освещению на рабочих местах, оборудованных ПЭВМ

Освещенность на рабочем столе:	300-500 лк
Освещенность на экране ПЭВМ:	не выше 300лк
Блики на экране:	не выше 40 кд/м ²
Прямая блескость источника света:	200 кд/м ²
Показатель ослепленности:	не более 20
Показатель дискомфорта:	не более 15
Отношение яркости	
- между рабочими поверхностями:	3:1-5:1
- между поверхностями стен и оборудования:	10:1
Коэффициент пульсации:	не более 5%.

При выборе типа светильников предпочтение было отдано люминесцентным лампам, при учете следующих факторов:

- Люминесцентные лампы обеспечивают высокий уровень освещенности при низком (по сравнению с лампами накаливания) энергопотреблении.

- Люминесцентные лампы имеют больший срок службы (по сравнению с лампами накаливания).

Для снижения пульсаций используются светильники с установленными электронными пускорегулирующими аппаратами (ЭПРА).

Для расчета общего равномерного освещения при горизонтальной рабочей поверхности основным является метод светового потока (коэффициента использования), учитывающий световой поток, отраженный от потолка и стен. Световой поток лампы F_l (лм) группы ламп светильника при люминесцентных лампах рассчитывают по формуле:

$$F_l = \frac{E_n S z k}{N \eta}, \text{ где}$$

E_n — нормированная минимальная освещенность, лк

S — площадь освещаемого помещения, м²

Z — коэффициент минимальной освещенности, равный отношению $E_{сп}/E_{min}$, значения которого обычно находятся в пределах 1,1-1,5 (в среднем 1,2), для люминесцентных ламп принимаем 1,1

k — коэффициент запаса, принимаемый в соответствии с [33 Табл. 3], для решаемой задачи равный 1,5

N — число светильников в помещении

η — коэффициент использования светового потока ламп. Значения коэффициента η определяют по таблицам, в зависимости от коэффициентов отражения светового потока от потолка и стен и показателя помещения i , определяемого из отношения

$$i = \frac{AB}{H_p(A+B)}, \text{ где}$$

A и B — два характерных размера помещения

H_p — высота светильников над расчетной поверхностью.

Для решаемой задачи $H_p = H - h - T = 2.75 - 0.8 - 0.07 = 1.88$, тогда

$$i = \frac{5 * 8}{1.88 * (5 + 8)} = 1.637$$

Потолок и стены помещения металлические, принимаем коэффициент отражения 40%. Тогда для светильника типа ЛСП-01 коэффициент использования составит 52% [33 Табл. 4].

Примем количество светильников 4, однако, учитывая, что в каждом светильнике расположены две лампы, рассчитаем световой поток лампы как

$$F_{л} = \frac{300 * 5 * 8 * 1,1 * 1,5}{4 * 2 * 0,52} = 4760 \text{ лм}$$

По [33 Табл. 5]. подбираем лампу ЛБ80 со световым потоком в 5220 лм. Отклонение от рассчитанного значения может составлять от -10% до +20%, в нашем случае эта величина составляет 9,66%, что является допустимым.

Суммарная потребляемая осветительной системой мощность составляет $4 * 2 * 80 = 640$ Вт. Учитывая оборудование светильников ЭПРА с КПД 95% получим результирующую мощность в 673 Вт.

Результаты расчета:

- 4 светильника типа ЛСП-01
- Лампы ЛБ80
- Оборудование светильников ЭПРА
- Потребляемая мощность 673 Вт

10.3. Выбор рационального способа утилизации печатных плат.

При разработке мероприятий по охране окружающей среды большое внимание уделяется малоотходному производству. При выходе из строя печатных плат или их модернизации встает вопрос их рациональной утилизации.

При утилизации печатных плат производится демонтаж чипов (микроконтроллеров, ПЛИС, преобразователей и др). Это позволяет использовать их повторно, что не только снижает затраты при

производстве опытных образцов, но защищает окружающую среду от загрязнения свинцом, ведь бессвинцовые микросхемы чаще всего не проходят военную приемку.

Однако, демонтаж не всех чипов имеет смысл, ввиду сложности процесса или невозможности их повторного использования.

Таблица 24. Примерный состав печатных плат после демонтажа

Материал	Содержание
Стеклополимер	70%
Медь	16%
Олово	4%
Свинец	4 %
Железо	3%
Никель	2%
Серебро	0,05%
Золото	0,05%
Палладий	0,01%
Прочие (висмут, сурьма, тантал)	<0,01%

Несмотря на большую сложность состава, он обладает большой разнородностью, поэтому при утилизации удастся разделить фракции и восстановить материалы для их последующего повторного использования.

В основе разделения материалов лежат их различия по плотности, магнитных, электропроводных и термических свойств, а также свойств химической активности.

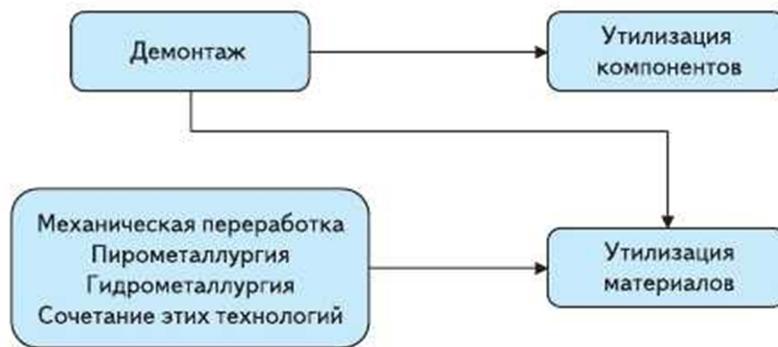


Рис. 49. Схема утилизации печатных плат

В настоящее время серийно выпускаются системы механической очистки для переработки разнообразных материалов электронных отходов, включая платы с компонентами и пустые платы. Механическая переработка, включает в себя следующие этапы:

- первичное измельчение крупных фракций с помощью измельчителя с вращающимися ножами разнообразного применения
- отделение крупных фракций черных металлов с помощью сильных магнитов, расположенных над вибрирующим конвейером
- измельчение в порошок - в этом процессе платы превращаются в порошок в шаровой мельнице, в которой используются шары, устойчивые к стиранию
- просеивание с использованием самоочищающихся сит
- электростатическое разделение, позволяющее фактически завершить разделение металлических фракций путем рециркуляции фракций частиц среднего размера

- дальнейшее уменьшение размера, представляющее собой вторичное измельчение в порошок для уменьшения размера крупных частиц.

Гидрометаллургические методы традиционно применяются для восстановления золота с контактных поверхностей разъемов. Золото высвобождалось либо в виде металлических чешуек посредством растворения в кислоте медных подложек, либо посредством растворения золота в растворах на основе цианида или тиомочевины, с последующим электролитическим осаждением или химическим замещением с использованием цинка.

Таким образом, выбран рациональный способ утилизации печатных плат при их выходе из строя или модернизации.

Заключение.

В рамках дипломного проектирования была разработана компактная, гибкая, автоматизированная система, пригодная для широкого диапазона задач регистрации данных, в тех случаях, когда необходимо передать большой объем данных с ПЛИС на компьютер в любой области.

В процессе выполнения дипломного проекта получены следующие результаты:

На этапе предпроектного исследования поставлена задача на разработку и произведен анализ существующих решений.

На этапе концептуального проектирования разработана укрупненная архитектура системы и её состав, а так же функциональная структура системы.

Сформировано техническое задание на разработку автоматизированной системы регистрации потока данных радиолокационной станции.

На этапе технического проектирования разработаны алгоритм программной части подсистемы приема и хранения данных, структура программной части подсистемы передачи данных и интерфейсы её модулей, а также диаграмма состояния модулей, реализующих конечный автомат. Так же на этом этапе была разработана схема принципиальная электрическая, отражающая требуемые изменения в плате ячейки БОС.

На этапе рабочего проектирования был разработан программный комплекс системы. Для подсистемы приема и хранения данных разработана программа на C++ и соответствующие диаграммы классов и компонентов, для подсистемы передачи данных модуль для прошивки ПЛИС на языке описания аппаратуры интегральных схем Verilog и приведены функциональные схемы модулей, полученных при синтезе прошивки.

Была проведена исследовательская работа, в результате которой была выбрана оптимальная задержка между передачей пакетов.

В организационно-экономической части дипломного проекта были определены трудоемкость и затраты на разработку программного комплекса автоматизированной системы регистрации потока данных радиолокационной станции.

В части, посвященной охране труда и экологии, проведён анализ опасных и вредных факторов, воздействующих на инженера при работе с регистратором данных, выполнен расчет освещения на рабочем месте оператора и выбран рациональный способ утилизации печатных плат.

Таким образом, задание на выполнение дипломного проекта выполнено полностью, поставленная цель дипломного проекта достигнута в полном объеме.

Список литературы

1. Корляков, В.В. ВНИИРТ. СТРАНИЦЫ ИСТОРИИ. / В.В. Корляков.- М.: Оружие и технологии, 2006.- 256 с.
2. ГОСТ 23282-91. Решетки антенные. Термины и определения. — Введ. 1992-01-01.- М: Изд-во стандартов, 1991.- 7 с.
3. Яшкардин, В. USB. Universal Serial Bus Specification. Универсальная последовательная шина [Электронный ресурс]. / В. Яшкардин.- Режим доступа: <http://www.softelectro.ru/usb.html>
4. PCI Express. [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: https://ru.wikipedia.org/wiki/PCI_Express
5. Thunderbolt. [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: <https://ru.wikipedia.org/wiki/Thunderbolt>
6. SATA. [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: <https://ru.wikipedia.org/wiki/SATAEsata>
7. IEEE 1394. [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: https://ru.wikipedia.org/wiki/IEEE_1394
8. Смирнова, Е.В. Технологии современных сетей Ethernet. Методы коммутации и управления потоками данных / Е.В. Смирнова, П.В. Козик; под ред. Б.В. Кострова- С-Пб: БХВ-Петербург, 2012 — 272 с.
9. Ядро Linux. [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: https://ru.wikipedia.org/wiki/Ядро_Linux

10. GМII [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: <https://ru.wikipedia.org/wiki/GMII>
11. ГОСТ Р ИСО/МЭК 7498-1-99. Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая модель— Введ. 1999-03-18.- М: Изд-во стандартов, 1999.- 62 с.
12. TCP/IP [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: <https://ru.wikipedia.org/wiki/TCP/IP>
13. UDP [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: <https://ru.wikipedia.org/wiki/UDP>
14. SCTP [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: <https://ru.wikipedia.org/wiki/SCTP>
15. SPX [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: <https://ru.wikipedia.org/wiki/SPX>
16. Pcap [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: <https://ru.wikipedia.org/wiki/Pcap>
17. Zero copy [Электронный ресурс].- Электронная свободная энциклопедия Википедия.- Режим доступа: <https://ru.wikipedia.org/wiki/Zero-copy>
18. Апробирование [Электронный ресурс].- Электронный многофункциональный многоязычный словарь и тезаурус.- Режим доступа: <https://ru.wiktionary.org/wiki/апробирование>
19. Арсеньев, В.В. Методические указания к выполнению организационно-экономической части дипломных проектов по созданию программной продукции. / Ю.Б Сажин, В.В Арсеньев.- М.: Изд-во МГТУ им. Баумана, 1994. - 52 с.

20. ГОСТ 12.0.003-74. Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация. — Введ. 1976-01-01.- М: Изд-во стандартов, 1974.- 4 с.
21. ТОИ Р-31-001-96. Типовая инструкция по охране труда для матроса транспортного судна.- Введ. 1997-03-01 — 1996 г. — 36 с.
22. Юмашев, Г.С. Травматология и ортопедия. / Г.С. Юмашев, С.З. Горшков, Л.Л. Силин и др. - Под ред. Г.С. Юмашева.- М.: Медицина, 1990. - 576 с.
23. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.- Введ. 2003-06-30 — 2003 г. — 54 с.
24. ГОСТ 12.1.005-88. Общие. Санитарно-гигиенические требования к воздуху рабочей зоны. — Введ. 1989-01-01.- М: Изд-во стандартов, 1988.- 49 с.
25. СН 2.2.4/2.1.8.566-96. Производственная вибрация, вибрация в помещениях жилых и общественных зданий. - Введ. 2003-06-30 — М: Минздрав России, 1997 г. — 12 с.
26. Влияние освещения на человека [Электронный ресурс] - Режим доступа: <http://www.ing-seti.ru/?p=1688>
27. Беянина, А.В. Влияние освещенности на физиологическое состояние организма человека [Электронный ресурс] / А.В Беянина - Режим доступа: <http://www.12sanepid.ru/press/publications/187.html>
28. ГОСТ Р 12.1.019-2009 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты. — Введ. 2011-01-01.- М: Стандартиформ, 2010.- 32 с.
29. Полтев, М.К. Охрана труда в машиностроении. / М.К. Полтев — М.: Высш. школа, 1980. — 294 с.
30. СанПин 2.2.4.1191-03. Электромагнитные поля в производственных условиях. - Введ. 2003-02-19 — М: Минздрав России, 2003 г. — 24 с.

31. ГОСТ 12.1.006-84. Система стандартов безопасности труда. Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля. — Введ. 1986-01-01.- М: Изд-во стандартов, 1984.- 5 с.
32. МОНОТОННОСТЬ [Электронный ресурс] - Режим доступа: <http://glossword.info/index.php/term/29-spravochnik-po-bezopasnosti,1836-monotonnost-.html>
33. Юдин, Е.Я. Охрана труда в машиностроении. / Е.Я. Юдин, С.В. Белов, С.К. Баланцев и др.; Под ред. Е.Я. Юдина, С.В. Белова. - М.: Машиностроение, 1983 г. - 432 с.

Список используемого программного обеспечения

Aldec, Inc., Active-HDL™ 9.3

ArgoUml v0.34

Mentor Graphics Corporation, DxDesigner Expedition Enterprise Flow

Microsoft Corporation, Microsoft Excel 2010

Microsoft Corporation, Microsoft Visio 2010

Microsoft Corporation, Microsoft Word 2010

Synopsys Inc., Synplify® Premier

Xilinx, ISE Design Suite 14.3

Приложение 1. Листинг программного кода, реализующего класс SocketService.

```
#include "socket_service.h"

SocketService::SocketService()
    : pg()
{
    initDriver();
}

bool SocketService::initDriver()
{
    pcap_if_t *allDevs;
    int numberOfDevices = 0;
    int deviceId = 0;
    char errbuf[PCAP_ERRBUF_SIZE];
    if(pcap_findalldevs(&allDevs, errbuf) == -1)
    {
        fprintf(stderr, "Error in pcap_findalldevs: %s\n",
errbuf);
        exit(1);
    }

    for(pcap_if_t *device = allDevs; device; device=device-
>next)
    {
        printf("%d. %s", ++numberOfDevices, device->name);
        if (device->description)
            printf(" (%s)\n", device->description);
        else
            printf(" (No description available)\n");
    }
    if(numberOfDevices == 0)
    {
        printf("\nNo interfaces found! Make sure WinPcap is
installed.\n");
        return false;
    }
    printf("Enter the interface number (1-
%d):", numberOfDevices);
    scanf("%d", &deviceId);

    if(deviceId < 1 || deviceId > numberOfDevices)
    {
        printf("\nInterface number out of range.\n");
        pcap_freealldevs(allDevs);
        return false;
    }
}
```

```

    pcap_if_t* device = allDevs;
    for(int i=0; i< deviceId-1 ; i++)
    {
        device=device->next;
    }

    if ((fp = pcap_open_live(device->name,
65536, 1, 1000, errbuf)) == NULL)
    {
        fprintf(stderr, "\nUnable to open the adapter. %s is
not supported by WinPcap\n", device->name);
        pcap_freealldevs(allDevs);
        return false;
    }
    printf("\nlistening on %s...\n", device-
>description);

    pcap_freealldevs(allDevs);
    return true;
}

void SocketService::sendResetCommand()
{
    sendCommand(0xFF);
}

void SocketService::sendPreconfigCommand()
{
    sendCommand(0x0F);
}

void SocketService::sendConfigCommand()
{
    sendCommand(0xAD);
}

void SocketService::mainLoop()
{
    errno_t err = fopen_s(&pFile, "file.binary", "wb");
    if (err != 0)
    {
        printf("\nERROR: Can't open file for output. Error
code = %d\n", err);
        return;
    }
    statistics.timerStart();
    pcap_loop(fp, 0, packetHandler, NULL);
    statistics.timerStop();
}

void SocketService::finish()

```

```

{
    pcap_close(fp);
    WSACleanup();
    fclose(pFile);
    statistics.getStatistics();
    printf("Program is finished. Press any key to
terminate...");
    getchar();
    getchar();
}

void SocketService::sendCommand(u_char command)
{
    CommandPacket commandPacket = pg.generateCommand(command);
    if (pcap_sendpacket(fp, commandPacket.packet(),
commandPacket.size()) != 0)
    {
        fprintf(stderr, "\nError sending the packet: %s\n",
pcap_geterr(fp));
    }
}

```


Приложение 3. Листинг программного кода, реализующего класс FpgaToPcLink.

```
-----  
-- Company: VNIIRT  
-- Engineer: Chernov  
-- Module Name: FpgaToPcLink - Behavioral  
-- Project Name: FPGA_to_PC  
-- Description: FPGA to PC communication via IPv4 link level  
--               GMII to Core interface, CRC count, SOF insert  
-----  
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;  
library UNISIM;  
use UNISIM.VComponents.all;  
  
entity FpgaToPcLink is  
  port (  
    Clk           : in  std_logic;  
    Rst           : in  std_logic;  
-----Service-----  
    UdpIpCoreLocked : out std_logic;  
-----FPGA to PC Interface-----  
    FPGA2PC_StartTx : in  std_logic;  
    FPGA2PC_TxType   : in  std_logic_vector( 2 downto 0);  
    FPGA2PC_TxDataLen : in  std_logic_vector(15 downto 0);  
    FPGA2PC_TxData    : in  std_logic_vector( 7 downto 0);  
    FPGA2PC_TxOver    : out std_logic;  
-----PC to FPGA Interface-----  
    PC2FPGA_RxStartOfData : out std_logic;  
    PC2FPGA_RxEndOfData   : out std_logic;  
    PC2FPGA_RxDataValid   : out std_logic;  
    PC2FPGA_RxDataLen     : out std_logic_vector(15 downto 0);  
    PC2FPGA_RxType        : out std_logic_vector( 2 downto 0);  
    PC2FPGA_RxData        : out std_logic_vector( 7 downto 0);  
    PC2FPGA_ReplyReq      : out std_logic;  
-----PHY Interface-----  
    GMII_RXD           : in  std_logic_vector(7 downto 0);  
    GMII_RX_DV         : in  std_logic;  
    GMII_RX_ER         : in  std_logic;  
    GMII_TXD           : out std_logic_vector(7 downto 0);  
    GMII_TX_EN         : out std_logic;  
    GMII_TX_ER         : out std_logic  
  );  
end FpgaToPcLink;
```

architecture Behavior of FpgaToPcLink is

```
TYPE StateType is (
    Reset,
    Idle,
    SendStartOfFrame,
    SendFrameData,
    SendCRC
);

signal CurrentState      : StateType;

signal iRst              : std_logic := '0';

signal iGMII_RXD        : std_logic_vector(7 downto 0);
signal iGMII_RX_DV     : std_logic;
signal iGMII_RX_ER     : std_logic;

signal iGMII_TXD        : std_logic_vector(7 downto 0);
signal iGMII_TXD_Dly   : std_logic_vector(7 downto 0);
signal iGMII_TX_ER     : std_logic := '0';

signal iTxSof           : std_logic;
signal iTxEof          : std_logic;

signal iRxEof          : std_logic := '1';
signal iRxSofDly       : std_logic := '1';
signal iRxSof          : std_logic := '1';

signal iRxSofDlyBuf    : std_logic_vector( 6 downto 0);
signal iTxEofDlyBuf    : std_logic_vector( 6 downto 0);
signal iTxUserDataEnd  : std_logic := '0';

signal iSofNumOct       : std_logic_vector( 2 downto 0);

signal iCrcVal          : std_logic := '0';
signal iCrcIni          : std_logic := '0';
signal iCrcHld         : std_logic := '0';
signal iCrcReg          : std_logic_vector(31 downto 0);
signal iCrcOctNumb     : std_logic_vector( 1 downto 0);

signal fcsd3           : std_logic_vector( 7 downto 0);
signal fcsd2           : std_logic_vector( 7 downto 0);
signal fcsd1           : std_logic_vector( 7 downto 0);
signal fcsd0           : std_logic_vector( 7 downto 0);

constant HeaderLen : integer := 48;
```

```

begin

iRst      <= Rst;

SkipEthernetMACSof : process (Clk) begin
  if (rising_edge(Clk)) then
    iGMII_RX_DV <= GMII_RX_DV;
    iRxSof      <= not ((iGMII_RX_DV xor GMII_RX_DV) and
GMII_RX_DV);
    iRxSofDlyBuf <= iRxSofDlyBuf(iRxSofDlyBuf'high - 1 downto
0) & iRxSof;
    iRxSofDly   <= iRxSofDlyBuf(iRxSofDlyBuf'high);
  end if;
end process;

process (Clk) begin
  if (rising_edge(Clk)) then
    iTxEofDlyBuf <= iTxEofDlyBuf(iTxEofDlyBuf'high - 1
downto 0) & iTxEof;
    iTxUserDataEnd <= iTxEofDlyBuf(iTxEofDlyBuf'high);
  end if;
end process;

process(Clk) begin
  if (rising_edge(Clk)) then
    if (iRst = '1') then
      CurrentState <= Reset;
    end if;
    case CurrentState is
      when Reset =>
        GMII_TX_EN <= '0';
        GMII_TXD   <= x"00";
        iCrcVal    <= '0';
        iCrcIni    <= '0';
        iCrcHld    <= '0';
        iCrcOctNumb <= b"00";
        if (iRst = '0') then
          CurrentState <= Idle;
        end if;
      --end Reset

      when Idle =>
        GMII_TXD   <= x"00";
        GMII_TX_EN <= '0';
        iCrcVal    <= '0';
        iCrcIni    <= '0';
        iCrcHld    <= '0';
        if (iTxSof = '0') then
          CurrentState <= SendStartOfFrame;
        end if;
      --end Idle
    end case;
  end if;
end process;

```

```

when SendStartOfFrame =>
  GMII_TX_EN <= '1';
  iCrcIni    <= '1';
  iCrcHld   <= '0';
  iSofNumOct <= iSofNumOct + 1;
  if (iSofNumOct >= 0 and iSofNumOct <= 6) then
    GMII_TXD  <= x"55";
  else
    GMII_TXD  <= x"D5";
    iCrcVal   <= '1';
    iCrcIni   <= '0';
    iSofNumOct <= (others => '0');
    CurrentState <= SendFrameData;
  end if;
--end SendStartOfFrame

when SendFrameData =>
  GMII_TXD <= iGMII_TXD_Dly;
  if (iTxUserDataEnd = '0') then
    iCrcVal   <= '0';
    iCrcHld   <= '1';
    CurrentState <= SendCRC;
  end if;
--end SendFrameData

when SendCRC =>
  case iCrcOctNumb is
    when b"00" =>
      GMII_TXD    <= fcsd3;
      iCrcOctNumb <= b"01";
    --end "00"
    when b"01" =>
      GMII_TXD    <= fcsd2;
      iCrcOctNumb <= b"10";
    --end "01"
    when b"10" =>
      GMII_TXD    <= fcsd1;
      iCrcOctNumb <= b"11";
    --end "10"
    when b"11" =>
      GMII_TXD    <= fcsd0;
      iCrcOctNumb <= b"00";
      CurrentState <= Idle;
    --end "11"
    when others =>
      CurrentState <= Reset;
    --end others
  end case;
--end SendCRC
end case;

```

```

    end if;
end process;

process (Clk) begin
    if (rising_edge(Clk)) then
        iRxEof      <=  not ((iGMII_RX_DV xor GMII_RX_DV) and not
GMII_RX_DV);
        iGMII_RXD   <=  GMII_RXD;
        GMII_TX_ER  <=  iGMII_TX_ER;
        iGMII_RX_ER <=  GMII_RX_ER;
    end if;
end process;

TxBusDelayForSof : DataBusDelay
    generic map(
        BusWidth => 8,
        CycleDly => 9
    )
    port map(
        Rst          => iRst,
        Clk          => Clk,
        DataBusIn    => iGMII_TXD,
        DataBusOut   => iGMII_TXD_Dly
    );
--end DataBusDelay_8

PC_COM_inst : PC_COM
    port map (
        rst          => iRst,
        clk          => Clk,

        UDP_IP_Core_locked      => UdpIpCoreLocked,

        -- FPGA to PC
        FPGA2PC_transmission_enable  => FPGA2PC_StartTx,
        FPGA2PC_transmission_type    => FPGA2PC_TxType,
        FPGA2PC_transmission_length  => FPGA2PC_TxDataLen,

        FPGA2PC_transmission_bus     => FPGA2PC_TxData,
        FPGA2PC_transmission_over    => FPGA2PC_TxOver,

        -- PC to FPGA
        PC2FPGA_tranmission_start_of_data =>
PC2FPGA_RxStartOfUserData,
        PC2FPGA_tranmission_end_of_data   => PC2FPGA_RxEndOfData,
        PC2FPGA_tranmission_valid_data    => PC2FPGA_RxDataValid,
        PC2FPGA_tranmission_data_length   => PC2FPGA_RxDataLen,
        PC2FPGA_transmission_type        => PC2FPGA_RxType,
        PC2FPGA_transmission_bus         => PC2FPGA_RxData,
        PC2FPGA_reply_request            => PC2FPGA_ReplyReq,

```

```

-- TX INTERFACE
tx_sof          => iTxSof,
tx_eof          => iTxEof,
tx_src_rdy     => open,
tx_data        => iGMII_TXD,

-- RX INTERFACE
rx_sof          => iRxSofDly,
rx_eof          => iRxEof,
rx_data        => iGMII_RXD
);
--end PC_COM

pecrc_inst : pecrc
port map (
  cclk => Clk,
  crst => iRst,
  cdat => iGMII_TXD_Dly,
  cval => iCrcVal,
  cini => iCrcIni,
  chld => iCrcHld,
  xcen => '1',

  creg => iCrcReg,
  cerr => open
);
--end pecrc_inst

fcsd3 <= (not iCrcReg(24), not iCrcReg(25), not iCrcReg(26),
not iCrcReg(27),
        not iCrcReg(28), not iCrcReg(29), not iCrcReg(30),
not iCrcReg(31)
        );
fcsd2 <= (not iCrcReg(16), not iCrcReg(17), not iCrcReg(18),
not iCrcReg(19),
        not iCrcReg(20), not iCrcReg(21), not iCrcReg(22),
not iCrcReg(23)
        );
fcsd1 <= (not iCrcReg( 8), not iCrcReg( 9), not iCrcReg(10),
not iCrcReg(11),
        not iCrcReg(12), not iCrcReg(13), not iCrcReg(14),
not iCrcReg(15)
        );
fcsd0 <= (not iCrcReg( 0), not iCrcReg( 1), not iCrcReg( 2),
not iCrcReg( 3),
        not iCrcReg( 4), not iCrcReg( 5), not iCrcReg( 6),
not iCrcReg( 7)
        );

end Behavior;

```

Приложение 4. Листинг программы

```
import sys
import re
import array

for arg in range(1, len(sys.argv)):
    inFileName = sys.argv[arg]
    outFileName = inFileName[:inFileName.rindex('.')] +
'short' + inFileName[inFileName.rindex('.):]
    logFileName = inFileName[:inFileName.rindex('.')] + '.log'

    inFile = open(inFileName, 'rb')
    outFile = open(outFileName, 'wb')
    logFile = open(logFileName, 'w')

    fullText = inFile.read()

    dataArray = array.array('B')

    header = b'\x55\x54\xe9\xc5\x05\xa0\x20\x20\x01'

    headerLen = len(header)
    textLen = len(fullText)

    x = 0
    while x < textLen:
        if fullText[x:x+headerLen] == header:
            dataArray.append(fullText[x+100])
            x += 1432
            print (str(100*x/textLen) + " % of" + inFileName)
        else:
            x += 1

    lost = 0
    log = ""
    sizeOfArr = len(dataArray)

    for i in range(1, sizeOfArr):
        lostNow = 0
        if dataArray[i] > dataArray[i-1]:
            lostNow = dataArray[i] - dataArray[i-1] - 1
        else:
            lostNow = dataArray[i] - dataArray[i-1] + 255
        if lostNow != 0:
            lost += lostNow
            log += "line " + str(i) + ": " + str(dataArray[i-
1]) + ".." + str(dataArray[i]) + " Потеряно: " + str(lostNow)
+ "\n"

    statistics = "Всего принято: " + str(sizeOfArr) + "\n"
```

```
statistics += "Потеряно: " + str(lost) + " " +  
str(100*lost/(lost+sizeOfArr)) + "%\n"
```

```
logFile.write(statistics)  
logFile.write(log)  
outFile.write(DataArray)
```

```
inFile.close()  
outFile.close()  
logFile.close()
```

Приложение 5. Акт о рассмотрении и проверке работоспособности программного продукта