

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И.Ульянова (Ленина)
(СПбГЭТУ «ЛЭТИ»)

Направление (специальность)	11.03.01 – Радиотехника
Профиль	Аудиовизуальная техника
Факультет	РТ
Кафедра	ТВ

К защите допустить

Зав. кафедрой

Лысенко Н.В.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
БАКАЛАВРА**

**ТЕМА: ETHERNET-КОММУТАТОР УРОВНЯ L2 С РЕАЛИЗАЦИЕЙ
НА ПЛИС**

Студент	_____	Кривченко С.К.
	<i>подпись</i>	
Руководитель	_____	Козлов А.С.
(Уч. степень, уч. звание)	<i>подпись</i>	
Консультанты	_____	Грива Я.Г.
(Уч. степень, уч. звание)	<i>подпись</i>	
ст. преп.	_____	Житенева М.И.
(Уч. степень, уч. звание)	<i>подпись</i>	

Санкт-Петербург

2019

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Утверждаю
Зав. кафедрой ТВ
_____ Лысенко Н.В.
« ____ » _____ 2019 г.

Студент Кривченко Сергей Константинович Группа 5105

Тема работы: Ethernet-коммутатор уровня L2 с реализацией на ПЛИС

Место выполнения ВКР: ПАО «ИНТЕЛТЕХ»

Исходные данные (технические требования):

В соответствии с техническим заданием на разработку бюджетного коммутационного модуля для обеспечения взаимодействия с магистральной сетью с применением ПЛИС, устройство должно удовлетворять следующим требованиям:

- Технология доступа: Ethernet.
- Тип разъемов: RJ-45.
- Тип сетевого кабеля: Витая пара (UTP категория 5е).
- Количество портов: 4.
- Тип портов: 100Base-TX (100 Мбит/с).
- Режим работы портов: Дуплекс.
- Вместимость таблицы MAC-адресов: 128 записей.
- Емкость промежуточных буферов на каждый порт – не менее 5 кадров размером 1500 байт.
- Способ питания устройства: Сеть постоянного тока с напряжением (12 ± 0.25) В.
- Максимальный потребляемый ток: 2 А.
- Диапазон рабочих температур: От 0 °С до + 85 °С.

Содержание ВКР: Теоретические сведения о сетевой модели OSI. Формат кадра технологии Ethernet. Принципы работы коммутатора уровня L2. Особенности реализации архитектуры проектируемого Ethernet-коммутатора. Синтез проекта разработанного устройства и моделирование синтезированной схемы с учетом временных задержек в ПЛИС в среде для моделирования Modelsim SE 10.2с. Макетирование разработанного устройства.

Перечень отчетных материалов: текст ВКР, иллюстративный материал, презентация в Microsoft Office PowerPoint.

Дополнительные разделы: Экономическое обоснование ВКР.

Дата выдачи задания

«__» _____ 20__ г.

Дата представления ВКР к защите

«__» _____ 20__ г.

Студент

подпись

Кривченко С.К.

Руководитель

подпись

Козлов А.С.

Консультант

подпись

Грива Я.Г.

КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Утверждаю

Зав. кафедрой ТВ

_____ Лысенко Н.В.

« » 2019 г.

Студент Кривченко С.К.

Группа 5105

Тема работы: Ethernet-коммутатор уровня L2 с реализацией на ПЛИС

№ п/п	Наименование работ	Срок выполнения
1	Разработка ТЗ	09.01 – 13.01
2	Анализ ТЗ и обзор литературы по теме работы	14.01 – 03.02
3	Проектирование архитектуры коммутатора	04.02 – 10.03
4	Выбор производителя и модели ПЛИС	11.03 – 17.03
5	Разработка прошивки для ПЛИС	18.03 – 28.04
6	Тестирование и отладка устройства	29.04 – 26.05
7	Оформление пояснительной записки	27.05 – 02.06
8	Оформление иллюстративного материала	03.06 – 07.06

Студент

Кривченко С.К.

подпись

Руководитель

Козлов А.С.

подпись

Консультант

Грива Я.Г.

подпись

РЕФЕРАТ

50 стр., 19 рис., 9 табл., 16 ист., 0 прил.

ETHERNET-КОММУТАТОР, СЕТЕВАЯ МОДЕЛЬ OSI, КОММУТАЦИОННАЯ МАТРИЦА, ПЛИС, XILINX СЕМЕЙСТВО SPARTAN – 6.

Объектом разработки является Ethernet-коммутатор уровня L2 с реализацией на ПЛИС.

Цель работы – разработать легко масштабируемую архитектуру коммутатора, обеспечивающую необходимую пропускную способность коммутационного узла для работы в сетях по стандарту Fast Ethernet, и реализовать её в ПЛИС.

В тексте ВКР приведены теоретические сведения о модели сети, её структуре и основных принципах работы. Описываются основные проблемы, с которыми столкнулись на разных этапах разработки, и приводятся варианты путей их решения. Рассматриваются архитектурные особенности коммутаторов уровня L2 и различные технологии внутренней коммутации кадров. С учетом особенностей процесса обработки кадра и принятых решений разрабатывается архитектура коммутатора. Исходя из требований технического задания рассчитываются необходимые параметры, которым должна удовлетворять выбранная модель ПЛИС, приводятся технические характеристики ПЛИС различных производителей и аргументируется итоговый выбор. Разрабатывается прошивка для ПЛИС в программном пакете ISE Design Suite 14.7. Приводятся результаты моделирования с учетом временных задержек при работе устройства в среде для моделирования MODELSIM SE 10.2с. Последним этапом разработки является макетирование устройства и его отладка. Производится расчет себестоимости процесса разработки устройства и изготовления опытного образца.

ABSTRACT

The text of the final qualifying work provides theoretical information about the network model and its structure, architectural features of L2 layer Ethernet-switches. Based on the requirements of the technical specifications calculates the required parameters, which must be met by the chosen model and manufacturing company of the FPGA.

Developing switch architecture in the software package ISE Design Suite 14.7 and modeling in the environment for modeling MODELSIM 10.2c. The last stage of development is device layout and debugging. Calculates the cost of the process of developing the device and manufacturing the prototype.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1. СЕТЕВАЯ МОДЕЛЬ КОММУТАТОРА	8
1.1. Сетевая модель OSI	8
1.2. Структура кадра Ethernet	11
1.3. Принцип работы коммутатора L2 уровня	12
2. РАЗРАБОТКА АППАРАТНОЙ РЕАЛИЗАЦИИ КОММУТАТОРА	15
2.1. Особенности архитектуры коммутаторов.....	15
2.2. Реализация физического уровня разрабатываемого устройства	16
2.3. Описание интерфейса RMI	18
2.4. Реализация канального уровня разрабатываемого устройства.....	20
2.5. Программируемые логические интегральные схемы	21
2.6. Особенности архитектуры ПЛИС	23
2.7. Расчет необходимого объема внутренней памяти	25
2.8. Выбор модели ПЛИС	26
3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОММУТАТОРА	
27	
3.1. Архитектура разрабатываемого устройства	27
3.2. Выбор типа коммутационного узла устройства	28
3.3. Блок выделения MAC-адресов	30
3.4. Алгоритм работы таблицы соответствия	31
3.5. Временные задержки при обработке кадров	32
3.6. Организация внутренних буферов.....	33
3.7. Обработка широковещательного трафика	34
4. РЕЗУЛЬТАТЫ РАЗРАБОТКИ	36
4.1. Моделирование с учетом временных задержек	36
4.2. Компиляция проекта и отчеты	37
4.3. Макетирование устройства.....	40

5. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ВКР	41
5.1. Оценка затрат на разработку и отладку устройства	41
5.2. Определение затрат на изготовление опытного образца	45
5.3. Общие экономические показатели	47
ЗАКЛЮЧЕНИЕ	48
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	50

ВВЕДЕНИЕ

Важной задачей практически для все современных цифровых систем является передача данных, как исходных данных для последующей обработки в высокопроизводительные системы, так и результатов вычислений этих систем конечному пользователю. Поэтому одной из самых быстроразвивающихся областей является инфокоммуникационная сфера, а именно разработка коммутационного оборудования, работающего по технологии 10GbE (скорость соединения 10 Гбит/с), 40GbE (скорость соединения 40 Гбит/с) и 100GbE (скорость соединения 100 Гбит/с).

Для обработки данных, поступающих с такой скоростью, требуются дорогостоящие специализированные микросхемы, которые являются стандартным не масштабируемым решением, позволяющим разрабатывать на их базе оборудование с ограниченным функционалом и строго фиксированным числом портов, причем, как правило, они потребляют большую мощность во время работы, ввиду высокой рабочей частоты микросхем (порядка нескольких ГГц).

Второй сложностью при выборе оборудования является обеспечение надежности при работе в нестандартных условиях, таких как климатически неблагоприятные условия, повышенная радиация и т.д. Современные устройства, как правило, удовлетворяют только одному из критериев и могут обеспечить либо высокую скорость передачи данных, либо устойчивость к внешним факторам. На сегодняшний день заказать всепогодный коммутатор с поддержкой 40GbE практически не представляется возможным и цена таких устройств крайне высока, что делает их недоступными для обычных потребителей. Также важно отметить, что иностранные покупные изделия и комплектующие не удовлетворяют требованиям потенциальных заказчиков для использования в специализированной аппаратуре.

Одно из возможных решений этих проблем является использование в данной области ПЛИС. Программируемые логические интегральные схемы (ПЛИС) имеют большую популярность в области разработки цифровых устройств для высокоскоростной обработки данных. За счет параллельности выполняемых в кристалле ПЛИС процессов можно добиться значительного быстродействия при обработке кадров на такой скорости соединения.

Использование разных серий ПЛИС позволяет работать практически в любых условиях, вплоть до устройств, используемых в космическом приборостроении, подвергающихся космическому облучению.

Целью данной работы является разработка бюджетного коммутационного модуля с применением ПЛИС, работающего по технологии Fast Ethernet (скорость соединения 100 Мбит/с), для освоения необходимой теоретической базы принципов работы Ethernet-сетей, коммутирующего оборудования и его архитектурных особенностей, позволяющей в перспективе перейти к разработке коммутаторов, работающих по вышеупомянутым технологиям, для устранения сложившегося в данном направлении пробела между теоретическими достижениями и возможностями их аппаратной реализации на сегодняшний момент.

Данный шаг является первым этапом в разработке высокоскоростного оборудования с высокой стабильностью и возможностью работы в агрессивной внешней среде. Разработка программного обеспечения ведется с учетом возможности дальнейшего совершенствования и масштабирования этого проекта в устройство, которое будет актуальным на современном рынке.

1. СЕТЕВАЯ МОДЕЛЬ КОММУТАТОРА

1.1. Сетевая модель OSI

Сетевая модель OSI – модель стека сетевых протоколов, описывающая их взаимодействие (рисунок 1). Она состоит из 7 уровней, каждый из которых представляет собой отдельный уровень абстракции. Каждый протокол модели OSI должен взаимодействовать либо с протоколами своего уровня, либо с протоколами на единицу выше/ниже него и может выполнять функции только своего уровня [1].

Каждому уровню с некоторой долей условности соответствует свой логически неделимый элемент данных, которым можно оперировать на отдельном уровне в рамках модели и используемых протоколов: на физическом уровне – бит, на канальном уровне информация объединена в кадры, на сетевом – в пакеты (датаграммы), на транспортном – в сегменты, на сеансовом и выше представляет собой непосредственно данные, которые необходимо передать другому пользователю сети.

Посредством данной модели различные сетевые устройства, вне зависимости от его типа (сервер, маршрутизатор, коммутатор, концентратор) и производителя, могут взаимодействовать друг с другом в рамках интернет сети.

Модель OSI состоит из следующих уровней (в порядке убывания):

- Прикладной уровень обеспечивает взаимодействие пользовательских приложений с сетью. На этом уровне выполняются все высокоуровневые протоколы, такие как http, ftp и т.д. Данные протоколы не решают вопросы маршрутизации или гарантии доставки информации – этим, как правило, занимаются нижестоящие уровни. На 7 уровне необходима реализации конкретных действий, например, отправка email-сообщения конкретному адресату.
- Уровень представления обеспечивает преобразование протоколов, кодирование/декодирование и, при необходимости, шифрование/дешифрование данных. В основном занимается синтаксисом и семантикой передаваемой и принимаемой информации. Это делает возможным общение компьютеров с различными внутренними представлениями данных, передавая их по сети в стандартизованном виде.

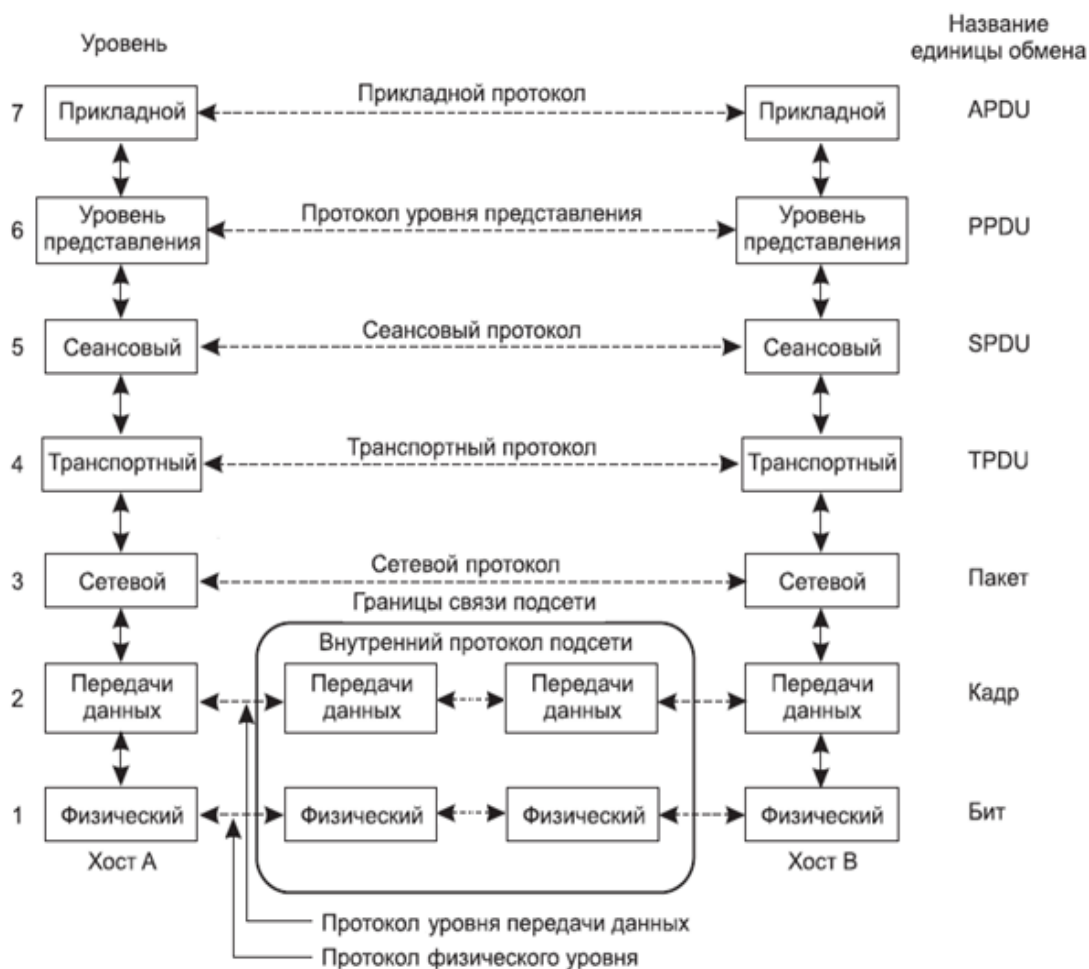


Рисунок 1 – Сетевая модель OSI

- Сеансовый уровень предоставляет различным сетевым узлам возможность устанавливать и обеспечивать поддержание сеанса связи с друг другом, позволяя приложениям взаимодействовать между собой длительное время. Уровень отвечает за управление созданием/завершением сеанса, синхронизацию задач, обмен информацией и поддержание сеанса в периоды неактивности приложений.
- Транспортный уровень отвечает за разбиение данных, полученных от сеансового уровня, на небольшие части при необходимости, их передачу сетевому уровню и гарантирование, что эти части придут к пункту назначения в правильном порядке и без ошибок. Также на этом уровне появляется понятия «порт», за счет чего обеспечивается многоканальная связь с различными удаленными узлами, позволяющая пользователю параллельно использовать несколько приложений.

- Сетевой уровень занимается управлением операциями подсети. Важнейшим моментом здесь является определение кратчайших маршрутов пересылки пакетов от источника к пункту назначения, маршрутизация и коммутация, отслеживание неполадок и «заторов» в сети.

- Канальный уровень обеспечивает взаимодействие сетей на физическом уровне, а также контроль и исправление ошибок, которые могут возникнуть при передаче. На этом уровне данные, полученные от вышестоящего уровня, делятся на кадры, размер которых колеблется от нескольких десятков до нескольких тысяч байт и передаются низ лежащему физическому уровню для непосредственной передачи. Полученные же с физического уровня данные в виде последовательности бит, собираются в кадры, проверяется их целостность и, если нужно, исправляются ошибки, после чего они передаются сетевому уровню [1]. На этом уровне появляется адресация узлов сети. Этим адресом является МАС-адрес устройств, который является их уникальным физическим адресом, присвоенным производителем. Адресация второго уровня работает только в пределах одного сетевого сегмента.

- Физический уровень определяет методы передачи полученных от канального уровня данных, представленных в двоичном виде, от одного сетевого узла (например, компьютера) к другому. На физическом уровне принципиальными моментами, которые должны быть согласованы со всеми устройствами, находящимися в пределах одного сегмента подсети, являются: методы кодирования цифровых сигналов, используемые для передачи электрических или оптических сигналов в кабель или радиоэфир и, соответственно, их приёма и обратного преобразования в последовательность бит данных; уровень напряжения, который будет использоваться для отображения логической единицы и нуля на приемной стороне; длительность одного бита, которая в свою очередь зависит от скорости соединения между узлами; возможность производить передачу данных одновременно в двух направлениях; условия определения момента начала и окончания процесса передачи, а также количество проводов в кабеле и функции каждого из них [1].

1.2. Структура кадра Ethernet

Кадр Ethernet II, называемый также кадром Ethernet DIX, имеет структуру, представленную на рисунке 2 [1]. Данный стандарт был опубликован в 1982 году, еще до появления официального стандарта IEEE 802.3, поэтому ко времени распространения оборудования, рассчитанного на новый стандарт, де-факто уже был стандартом для многих аппаратных и программных продуктов. В связи с этим в 1997 году IEEE был добавлен к официальному стандарту 802.3, и на данный момент, подавляющее большинство пакетов в Ethernet-сетях инкапсулированы согласно этому стандарту.

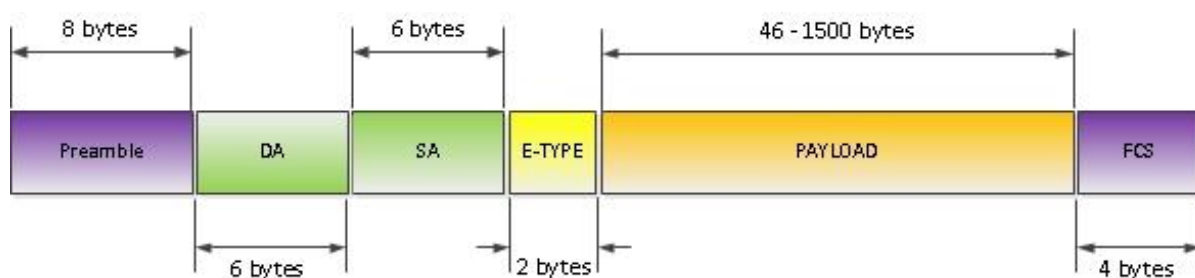


Рисунок 2 – Структура кадра Ethernet II

Кадр по стандарту Ethernet II содержит следующие поля [1]:

- **Преамбула (Preamble)** – поле длиной 8 байт, которое содержит последовательность бит, по сути, не являющихся частью заголовка второго уровня, а определяющих начало кадра. Согласно стандарту, в первых семи байтах содержится двоичная последовательность – 01010101 (0x55), в восьмом – 11010101 (0xD5). Данное поле необходимо для восстановления синхронизации станций сети перед приемом очередного кадра.
- **Адрес получателя (Destination Address)** – на это поле отводится 6 байт, и оно несет в себе MAC-адрес сетевого узла назначения. Самый старший передаваемый бит указывает на тип адреса: 0 – указывает на то, что кадр предназначен конкретной рабочей станции; в таком случае в остальных разрядах поля адреса назначения указывается её уникальный физический адрес; 1 – для групповых получателей, т.е. кадр предназначен группе рабочих станций, тогда в последующих разрядах указывается либо адрес этой группы, либо все остальные разряды могут быть равны 1. Такой кадр называется широковещательным и адресуется всем рабочим станциям в ЛВС.

- Адрес отправителя (Source Address) – на данное поле также отводится 6 байт, и оно несет в себе MAC-адрес сетевого узла отправителя. Старший бит первого байта всегда равен 0, что означает, что данное поле не может содержать широковещательный адрес.
- Тип протокола (Ether-TYPE) – поле длиной 2 байта, оно идентифицирует тип вышестоящего протокола сетевого уровня (к примеру, 0x0800 – IPv4, 0x86DD – IPv6). Наличие данного заголовка позволяет множеству протоколов высокого уровня разделять ЛВС без необходимости разбирать содержимое кадров друг друга.
- Полезная нагрузка (PAYLOAD) – на это поле может быть отведено до 1500 байт, оно содержит переносимые кадром данные. Однако важно отметить, что минимальная длина этого поля не может быть меньше, чем 46 байт. Это ограничение обусловлено технологией множественного доступа к общей передающей среде в ЛВС с контролем коллизий (технология CSMA/CD, определяемая стандартом IEEE 802.3). Кадры размером больше 1500 байт называются jumbo-кадрами, однако их формат не входит в стандарт IEEE 802.3, поэтому они не всегда поддерживаются оборудованием, так как каждый разработчик оборудования самостоятельно решает будет ли он поддерживать эту технологию. Большинство современных коммерческих коммутаторов поддерживают технологию передачи jumbo-кадров.
- Контрольная сумма (Frame Check Sequences) – поле длиной 4 байта, в нем содержится значение остатка избыточной циклической суммы, вычисленной с помощью полиномов типа CRC-32 для всех полей кадра за исключением преамбулы. Вычисляется отправляющей стороной непосредственно перед отправкой, а затем на приемной стороне вычисляется повторно уже по полученному кадру. Таким образом, сравнивая рассчитанное значение со значением поля принятого кадра, осуществляется контроль ошибок при передаче.

1.3. Принцип работы коммутатора L2 уровня

Коммутатор – устройство, предназначенное для соединения нескольких удаленных узлов сети в пределах одного или нескольких её сегментов. Он работает на канальном уровне сетевой модели OSI (рисунок 3) [2]. В отличие от концентратора, который работает на физическом уровне эталонной

модели OSI и весь трафик от одного подключённого устройства распространяет ко всем остальным в своем сегменте сети, коммутатор передаёт данные только непосредственно получателю (за исключением широковещательного трафика, предназначенного для всех узлов сети). Данная особенность повышает безопасность и производительность сети, избавляя остальных её пользователей от возможности и необходимости обрабатывать данные, которые им не предназначены.

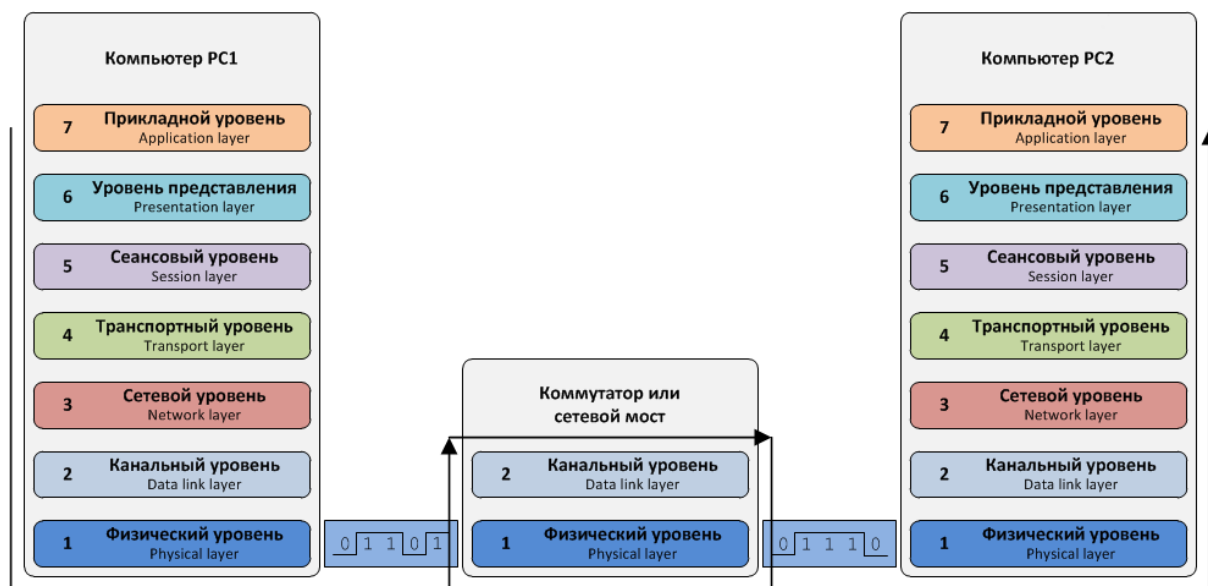


Рисунок 3 – Работа коммутатора в модели OSI

Коммутаторы идентифицируют пользователей ЛВС исключительно по их уникальным физическим адресам, т.е. в принятом кадре анализируются только 2 поля, содержащие MAC-адрес узла назначения и MAC-адрес узла отправителя [2].

Коммутатор заносит и хранит последний в ассоциативной памяти - так называемой «таблице соответствия», в которой каждому такому MAC-адресу узла отправителя ставится в соответствие конкретный порт устройства, на который следует отправлять весь трафик, предназначенный этому узлу. При включении устройства эта таблица пуста, и коммутатор работает в «режиме обучения». В этом режиме все кадры, поступающие с какого-либо порта, транслируются на все остальные, за исключением того, с которого он был принят. При этом, определив MAC-адрес отправителя кадра, коммутатор заносит его в вышеупомянутую таблицу на некоторый промежуток времени (как правило, определяемый производителем). Впоследствии, если на один из портов устройства поступит кадр, предназначенный сетевому узлу

с МАС-адресом, который содержится в таблице соответствия коммутатора, он будет передан только в соответствующий этому адресу порт. Однако, если МАС-адрес получателя до сих пор не ассоциирован ни с одним из портов, то кадр, как и в случае с режимом «обучения», также будет отправлен на все остальные порты, за исключением того, с которого был получен.

Таким образом с течением времени коммутатор составляет таблицу, содержащую МАС-адреса всех активных сетевых узлов в подсети, в результате чего трафик локализуется. Благодаря данному механизму другие сетевые устройства ЛВС ничего не знают о непредназначенных им кадрах. За счет такого подхода обеспечивается малая латентность (задержка) и высокая скорость пересылки на каждом порту коммутатора.

2. РАЗРАБОТКА АППАРАТНОЙ РЕАЛИЗАЦИИ КОММУТАТОРА

2.1. Особенности архитектуры коммутаторов

Коммутация внутри устройства может осуществляться тремя способами. Каждый из них можно считать комбинацией параметров надёжности передачи и задержки по отношению к принимаемому потоку:

- Коммутация с промежуточным хранением кадра. В этом режиме коммутатор принимает полностью весь кадр, проверяет его на отсутствие ошибок, и только после этого выбирает соответствующий ему выходной порт для коммутации и транслирует в него кадр. Является самым требовательным в отношении потребляемых ресурсов и накладывает высокие ограничения на объём и производительность внутренней памяти.

- Сквозная коммутация. В этом режиме коммутатор считывает из принимаемого кадра только адрес назначения, после чего сразу осуществляет коммутацию. Этот режим уменьшает задержки при передаче, однако из-за отсутствия контроля ошибок, данный метод обеспечивает наименьшую надёжность при передаче.

- Бесфрагментный или гибридный способ коммутации. Этот режим является модификацией предыдущего. Его смысл заключается в том, что коммутация осуществляется после проверки на наличие ошибок только первых 64 байт принимаемого кадра, после чего, в случае отсутствия ошибок, кадр обрабатывается в сквозном режиме. В этом режиме отсутствует возможность проверки всего кадра с помощью контрольной суммы, эта роль возлагается на оборудование узла получателя.

К задержке, требуемой для приема кадра и его отправки, добавляется также задержка, обусловленная временем принятия коммутатором решения о выборе канала коммутации, что в совокупности и определяет общую задержку устройства.

Понятие симметрии с точки зрения архитектуры коммутатора отражает равенство ширины полосы пропускания коммутируемого соединения между портами устройства. Другими словами, симметричный коммутатор обеспечивает соединение между своими портами только при условии, что все порты устройства имеют одинаковую скорость соединения, например, все порты имеют скорость соединения 10 Мбит/с или 100 Мбит/с.

Асимметричный же коммутатор обеспечивает коммутируемые соединения между портами с различной шириной полосы пропускания, например, в случаях комбинации портов с шириной полосы пропускания 10 Мбит/с и 100 Мбит/с. Однако, тогда возникает опасность переполнения первого порта при перенаправлении потока данных со второго, более быстрого. Для решения этой проблемы асимметричный коммутатор обязательно должен иметь большой объем промежуточной буферной памяти.

Буферизация может быть также необходима в том случае, когда порт пункта назначения занят. Поэтому, вне зависимости от типа симметрии архитектуры коммутатора, она необходима для временного хранения кадров и их последующей отправки в нужный порт.

Буфер памяти может использоваться для двух методов хранения и отправки кадров: буферизации по портам и буферизации с общей памятью.

В случае буферизации по портам, все принятые кадры хранятся в очередях, которые связаны с каждым портом отдельно. Кадр передаётся на выходной порт только тогда, когда все, находившиеся впереди него в очереди, были успешно переданы. При этом возможна ситуация, когда один кадр задерживает всю очередь из-за занятости порта его пункта назначения. Эта задержка может происходить даже в том случае, когда порты назначения остальных пакетов могут быть свободными и готовыми для передачи.

При буферизации кадров в общей памяти, они все хранятся в общем буфере памяти, который используется всеми портами коммутатора. Количество памяти, отводимой каждому порту, определяется в соответствии с интенсивностью их загрузки. После чего кадры, находящиеся в общем буфере памяти, динамически распределяются по выходным портам, что позволяет принять кадр на одном порту и отправить его с другого, не устанавливая в очередь [3].

2.2. Реализация физического уровня разрабатываемого устройства

Реализация физического уровня при разработке новых устройств, как правило, сводится к выбору готового решения. Простота реализации протоколов и их жесткая стандартизация позволяют выбрать внешнюю микросхему по требуемым параметрам и получить работоспособное оборудование. Данный подход позволяет существенно сократить время разработки. Протоколы обмена данными с такими микроконтроллерами также описаны

в различных стандартах, которые довольно легко реализуются: МП, RМП, RGMII и т.д.

В качестве модуля, обеспечивающего выполнение функций физического уровня в разрабатываемом устройстве, было решено использовать импортные высокопроизводительные Ethernet-трансиверы LAN8720 ETH Board, представленные на рисунке 4.



Рисунок 4 – Ethernet-трансивер LAN8720 ETH Board от WaveShare

Данный модуль имеет стандартизированный физический сетевой интерфейс RJ-45, поддерживает функцию HP Auto-MDIX, позволяющую в автоматическом режиме определять тип подключенного к порту кабеля (прямой или перекрестный), и функцию автосогласования, позволяющую в автоматическом режиме выбрать параметры канала связи между подключенными устройствами. Обмен данными с устройством MAC-уровня осуществляется через внешний интерфейс RМП. Для отслеживания текущего состояния модуля и его управления предусмотрен последовательный интерфейс SMI.

Таким образом, задача разработки коммутатора сводится только к разработке архитектуры MAC-уровня устройства и реализации протокола обмена данными с выбранным модулем физического уровня посредством интерфейса RМП.

2.3. Описание интерфейса RMII

Интерфейс RMII (Reduced Media Independent Interface – сокращенный независимый от среды передачи интерфейс) представляет собой стандартизованный интерфейс, использующий сокращенный набор сигналов интерфейса MII, и применяется для обеспечения взаимодействия между устройствами MAC- и PHY-уровней в сетях Fast Ethernet (рисунок 5). Он полностью совместим со стандартом IEEE 802.3u и позволяет работать устройству PHY-уровня в режимах дуплекса и полудуплекса со скоростью соединения между устройствами 10 и 100 Мбит/с [4].

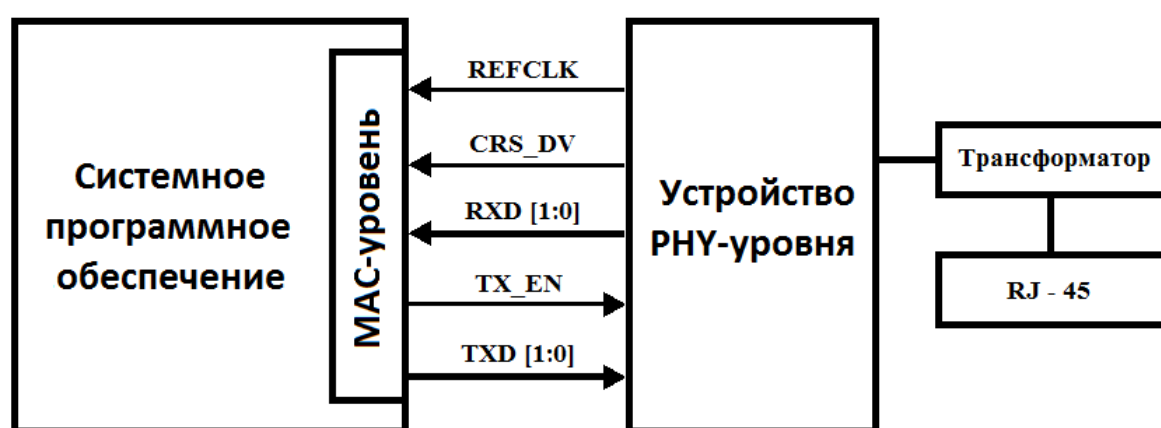


Рисунок 5 – Блок-схема подключения MAC- и PHY- устройств посредством интерфейса RMII

Согласно стандарту, в интерфейсе RMII есть следующие сигналы:

- Опорный тактовый сигнал (REFCLK). Его источником может являться как само устройство MAC-уровня, так и общий для MAC- и PHY-устройств внешний кварцевый генератор, работающий на частоте 50 МГц. Так как ширина передающей шины TXD [1:0] – 2 бита, а частота опорного тактового сигнала составляет 50 МГц, то скорость передачи данных по интерфейсу составляет 100 Мбит/с, что совпадает со скоростью передачи данных в сетях Fast Ethernet. Это обстоятельство позволяет в передающем тракте не использовать промежуточную буферизацию данных, а работать в сквозном режиме. Однако, в приемном тракте промежуточная буферизация необходима для компенсации сдвига фаз между фронтами опорного тактового сигнала внутри интерфейса и опорного тактового сигнала, восстановленного из принимаемого потока данных.

- Сигнал наличия несущей/достоверности принимаемых данных (Carrier Sense/Receive Data Valid). Этот сигнал переходит в активный уровень (логическая единица «1») немедленно, когда устройство РНУ-уровня обнаруживает наличие несущей в принимаемом потоке. При скорости соединения 100 Мбит/с несущая считается обнаруженной, когда в принимаемом потоке встречаются 2 несмежных нулевых бита. Потеря несущей выражается в переходе сигнала обратно в неактивный уровень (логический ноль «0»), однако он должен быть осуществлен синхронно относительно опорного тактового сигнала (REFCLK). Если в этот момент РНУ – устройство имеет в буфере непереданные MAC-уровню данные, то сигнал будет переходить из неактивного в активный уровень и обратно на длительность одного такта до тех пор, пока все данные не будут переданы, после чего сигнал останется в неактивном состоянии. Таким образом устройство MAC-уровня может легко разделить этот сигнал на два независимых: сигнал наличия несущей и сигнал достоверности принимаемых данных.

- Приемная шина данных (RXD [1:0]), по которой поступают данные извне на устройство. Данные на приемной шине всегда изменяются синхронно относительно опорного тактового сигнала (REF_CLK). На каждый период тактового сигнала в течение промежутка времени, когда сигнал CRS_DV находится в активном состоянии, приходится передача по интерфейсу 2 бит информации устройству MAC-уровня. Когда сигнал CRS_DV неактивен, для обозначения состояния простоя РНУ – устройства (отсутствия данных в принимаемом потоке), на шине должны быть установлены 2 нулевых бита. Любые другие комбинации данных на шине RXD [1:0] при неактивном сигнале CRS_DV устройству MAC-уровня следует игнорировать.

- Достоверность передаваемых данных (TX_EN) – сигнал TX_EN всегда изменяется синхронно относительно опорного тактового сигнала. Переход в активный уровень сигнала TX_EN означает, что устройство MAC-уровня готово к передаче данных по интерфейсу и на передающей шине TXD [1:0] представлены первые 2 бита преамбулы передаваемого кадра. Сигнал будет оставаться в активном состоянии до тех пор, пока все биты кадра не будут переданы. Переход в неактивное состояние сигнала TX_EN должен быть осуществлен синхронно с передачей последних 2 бит кадра.

- Передающая шина данных (TXD [1:0]) – данные на передающей шине всегда изменяются синхронно относительно опорного тактового сигнала (REF_CLK). На каждый период тактового сигнала в течение промежутка времени, когда сигнал TX_EN активен, приходится передача по интерфейсу 2 бит кадра РНУ-устройству. При отсутствии данных для передачи, для обозначения состояния простоя МАС-устройства, на шине должны быть установлены 2 нулевых бита. Любые другие комбинации данных на шине TXD [1:0] при неактивном сигнале TX_EN устройству РНУ-уровня следует игнорировать.

Передаваемые кадры обязательно должны иметь структуру, соответствующую стандартам Ethernet II или IEEE 802.3. Каждый октет кадра разбивается на группы по 2 бита и передается в порядке, обратном их следованию в исходном байте, как показано на рисунке 6, т.е. передача должна осуществляться в направлении от младшего бита к старшему [4].

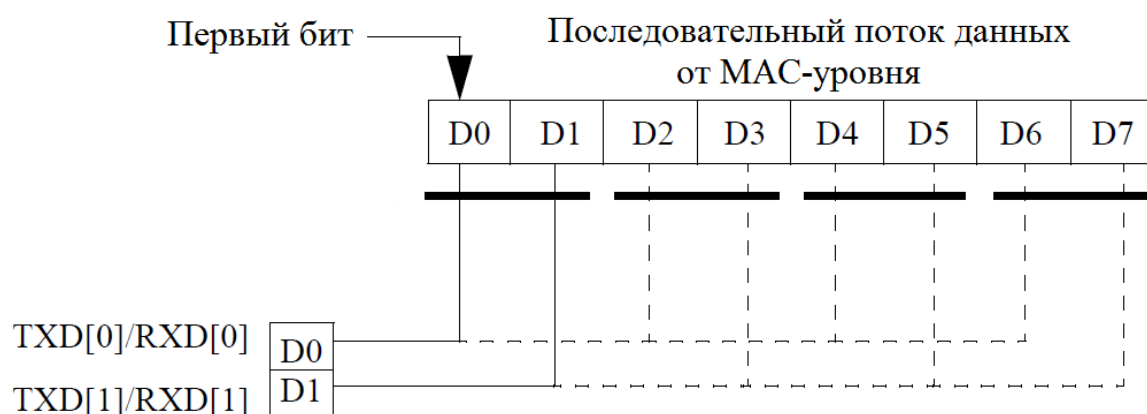


Рисунок 6 – Порядок передачи данных через интерфейс RMII

2.4. Реализация канального уровня разрабатываемого устройства

На МАС-уровне разрабатываемое устройство должно выполнять следующие обязательные функции:

- прием кадра от РНУ-трансивера по интерфейсу RMII;
- выделение в принятом кадре МАС-адреса отправителя и МАС-адреса получателя;
- осуществление поиска номера порта назначения, соответствующего МАС-адресу получателя, в ассоциативной таблице коммутатора;

- добавление в таблицу соответствия новой записи с номером текущего активного входного порта и MAC-адресом узла отправителя;
- непосредственно коммутация кадра в порт назначения с помощью внутреннего коммутационного узла устройства;
- передача кадра РНУ-трансиверу соответствующего порта через интерфейс RMII;

Исходя из названия и цели работы очевидно, что для выполнения всех вышеперечисленных функций устройства на MAC-уровне должна быть использована ПЛИС. Таким образом, упрощенная структурная схема разрабатываемого устройства имеет вид, представленный на рисунке 7.

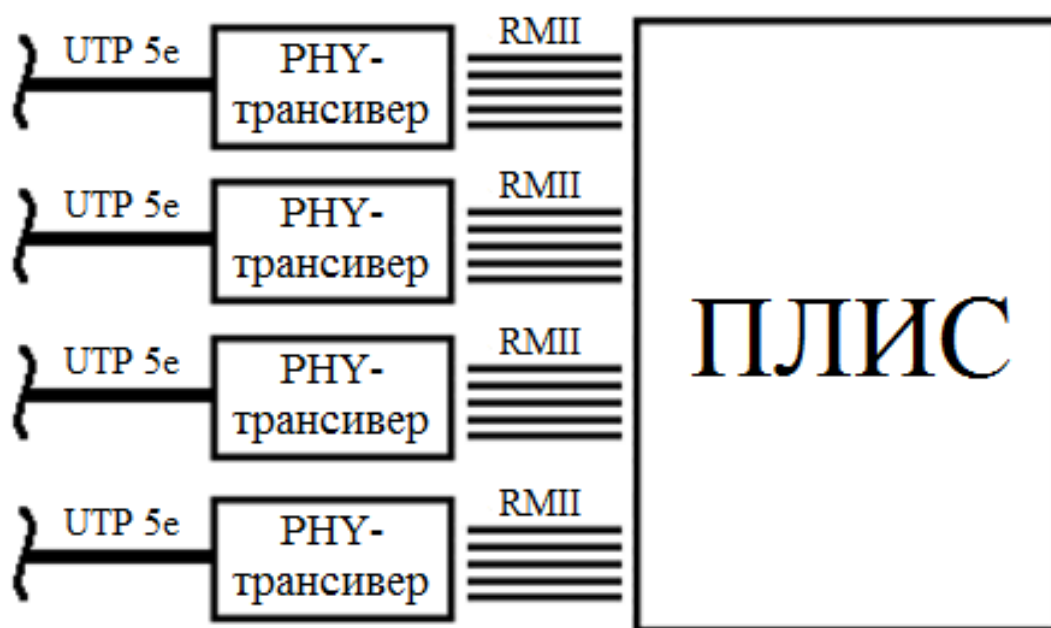


Рисунок 7 – Упрощенная структурная схема коммутатора

2.5. Программируемые логические интегральные схемы

Программируемая логическая интегральная схема (ПЛИС) в общем случае представляет собой кристалл, на котором расположено большое количество простых логических элементов, триггеров и "программируемых" перемычек из полевых транзисторов, служащих для их соединения. Затворы всех полевых транзисторов подключены к выходам триггеров одного длинного сдвигового регистра, который заполняется при программировании ПЛИС [5]. Таким образом, в отличие от обычных цифровых микросхем, ло-

гика работы ПЛИС определяется не при изготовлении, а задаётся посредством её программирования (конфигурирования), что дает возможность разработчику самому создавать архитектуру устройства из базовых логических элементов.

Желаемая структура цифрового устройства задается в виде его принципиальной электрической схемы или программы на специальных языках описания аппаратуры: Verilog, VHDL и др., после чего с помощью специализированной САПР создается программа-прошивка и, используя программатор, конфигурируется ПЛИС. Таким образом можно собрать цифровое устройство любой степени сложности, зависящей только от количества элементов, расположенных на кристалле, и их параметров.

Программа (прошивка) в FPGA хранится в распределённой памяти, которая может быть выполнена как на основе энергозависимых ячеек статического ОЗУ – в этом случае программа не сохраняется при исчезновении электропитания микросхемы, так и на основе энергонезависимых ячеек flash-памяти или перемычек antifuse – в этих случаях программа сохраняется при исчезновении электропитания.

- SRAM-Based ПЛИС одна из самых распространенных разновидностей. В этом варианте хранится ячейках статической памяти, изготовленной по стандартной технологии CMOS. Так как эти ячейки являются энергозависимыми, то при каждом включении питания микросхемы необходимо заново её конфигурировать, поэтому прошивка обычно хранится в ПЗУ, стоящем рядом с ПЛИС, а после включения питания или по сигналу сброса она автоматически переписывается в программирующий сдвиговый регистр.

- В Flash-based микросхемах хранение конфигурации происходит во внутренней FLASH памяти или памяти типа EEPROM. В таких ПЛИС при выключении питания прошивка не пропадает и после подачи питания микросхема опять готова к работе.

- Специальная технология Antifuse, по которой выполняются однократно программируемые ПЛИС. Программирование такой ПЛИС заключается в расплавлении в нужных местах чипа специальных перемычек для образования нужной схемы.

2.6. Особенности архитектуры ПЛИС

Как правило микросхема ПЛИС состоит из следующих составных частей, представленных на рисунке 8:

- конфигурируемых логических блоков (CLB), реализующих требуемую логическую функцию;
- программируемых электронных связей между конфигурируемыми логическими блоками (SB и CB);
- программируемых блоков ввода/вывода (I/O Block), обеспечивающих связь внешнего вывода микросхемы с внутренней логикой;

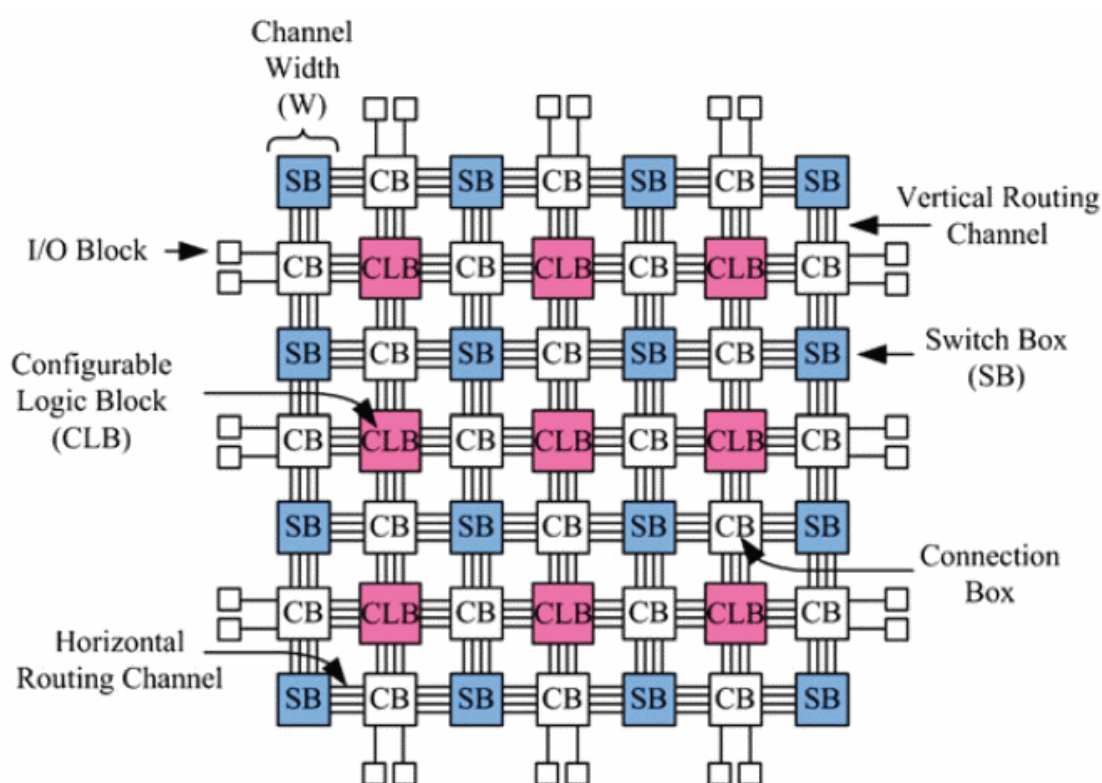


Рисунок 8 – Обобщенная архитектура ПЛИС

Конфигурируемый логический блок (далее - КЛБ) – это базовый элемент в ПЛИС, в котором может быть выполнена какая-то простая логическая функция или реализовано хранение результата этой функции в регистрах [5].

КЛБ может состоять из одного или нескольких базовых логических элементов. Пример простейшей структуры конфигурируемого блока приведен на рисунке 9.

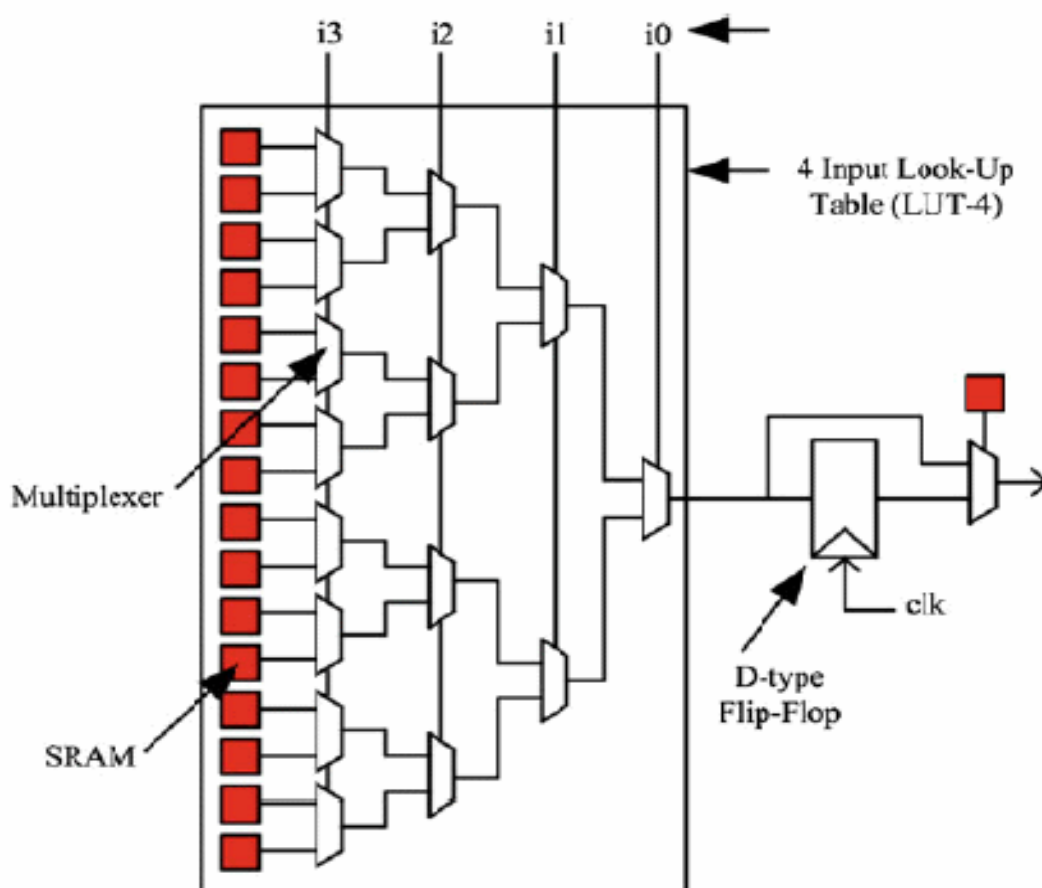


Рисунок 9 – Пример структуры простейшего КЛБ

В составе КЛБ показан четырехбитный LUT (Look-Up Table) - таблица преобразования, используемая в основном для реализации логических функций в ПЛИС. Здесь четырехбитному числу на входе логической функции ставится в соответствие однобитный результат. Видно, что для конфигурации 4-х битного LUT требуется 16 конфигурационных регистров (SRAM), содержимое которых определяет логическую функцию, реализованную внутри него. Также же видим опциональный регистр (D-type Flip-Flop) на выходе LUT, позволяющий сохранить в себе результат её преобразования, и мультиплексор, позволяющий получить на выходе КЛБ результат логической функции либо напрямую с LUT, либо с опционального регистра. Красными квадратиками на обозначены триггеры программируемого сдвигового регистра, в котором хранится прошивка ПЛИС.

В качестве устройства, способного выполнять поставленные на МАС-уровне задачи, также мог выступать микроконтроллер.

Однако, важно отметить, что главным отличием разработки на ПЛИС от микроконтроллеров является то, что разработчикам в микроконтроллерах

доступен лишь фиксированный набор решений и средств, присущих конкретному чипу, отступление от архитектуры которого невозможно. Все операции внутри него выполняются последовательно, по определенному алгоритму, в то время как блоки ПЛИС могут выполнять задачи параллельно и независимо друг от друга.

Таким образом, имея необходимость работать с четырьмя РНУ-трансиверами и осуществлять коммутацию четырех каналов внутри устройства одновременно, в случае с микроконтроллером, требовалась бы в четыре раза большая тактовая частота, чем в ПЛИС, что в разы увеличило бы потребляемую конечным устройством мощность и стоимость его изготовления.

Также к преимуществам использования при решении данной задачи ПЛИС следует отнести:

- возможность функционирования при гораздо более низких температурах окружающей среды (до -40°C);
- обеспечение минимальной задержки кадра при его обработке внутри устройства, за счет параллельности выполняемых процессов;
- возможность простой и быстрой дальнейшей масштабируемости разработанной архитектуры, посредством выбора более производительной ПЛИС с большим числом логических элементов на кристалле;

2.7. Расчет необходимого объема внутренней памяти

Важным параметром при выборе модели ПЛИС является доступный объем внутренней памяти, которая будет использоваться в качестве памяти входных и выходных буферов портов коммутатора. Исходя из требований ТЗ, вместимость каждого входного и выходного буфера должна равняться 5 кадрам максимального размера, разрешенного по стандарту Ethernet. Так как поддержка jumbo-кадров не требуется, максимальный размер передаваемого кадра с учетом заголовков второго уровня принимается равным 1 526 байт, следовательно, минимально-необходимый объем внутренней памяти в кристалле ПЛИС для одного порта составляет 7 630 байт, а для всех портов 61 040 байт. Однако, учитывая тот факт, что объем встроенных блоков памяти, как правило, всегда кратен степени 2, в качестве минимально-необходимого объема памяти для реализации входных и выходных буферов принимается значение 64 Кбайт (512 Кбит).

2.8. Выбор модели ПЛИС

Исходя из расчета, приведенного в подразделе 2.7 настоящей работы, основополагающим параметром при выборе модели ПЛИС для реализации разработанной архитектуры устройства является минимальный суммарный объем внутренних блоков памяти 64 Кбайт.

Приведем для сравнения и выбора несколько сопоставимых по техническим характеристикам моделей из бюджетной серии Spartan-6 и Cyclone IV, двух ведущих производителей на рынке в сфере ПЛИС – Xilinx и Altera, сведем их в таблицу 1.

Технические характеристики ПЛИС

Таблица 1

Производитель/модель ПЛИС	Xilinx Spartan-6 XC6SLX4	Altera Cyclone IV EP4CE6	Xilinx Spartan-6 XC6SLX9	Altera Cyclone IV EP4CE10
Число логических элементов	3 840	6 272	9 152	10 320
Объем внутренних блоков памяти, Кбит	216	270	576	414
Верхняя граница рабочего диапазон частот, МГц	200	200	200	200
Рабочий диапазон температур, °С	0 до 85	0 до 85	0 до 85	0 до 85
Приблизительная стоимость, руб.	746	777	1 070	1 459

Несмотря на то, что рабочий диапазон температур и верхняя граница рабочего диапазона частот совпадают у всех представленных моделей и с запасом превосходят требуемые показатели, выбор был сделан в пользу ПЛИС производителя Xilinx модели Spartan-6 XC6SLX9, так как остальные 3 модели не обеспечивают минимально-необходимый объем встроенной памяти, причем ПЛИС Altera Cyclone IV EP4CE10 имеет при этом в полтора раза большую стоимость, по сравнению с выбранной.

3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОММУТАТОРА

3.1. Архитектура разрабатываемого устройства

На рисунке 10 представлена архитектура разрабатываемого коммутатора в общем виде.

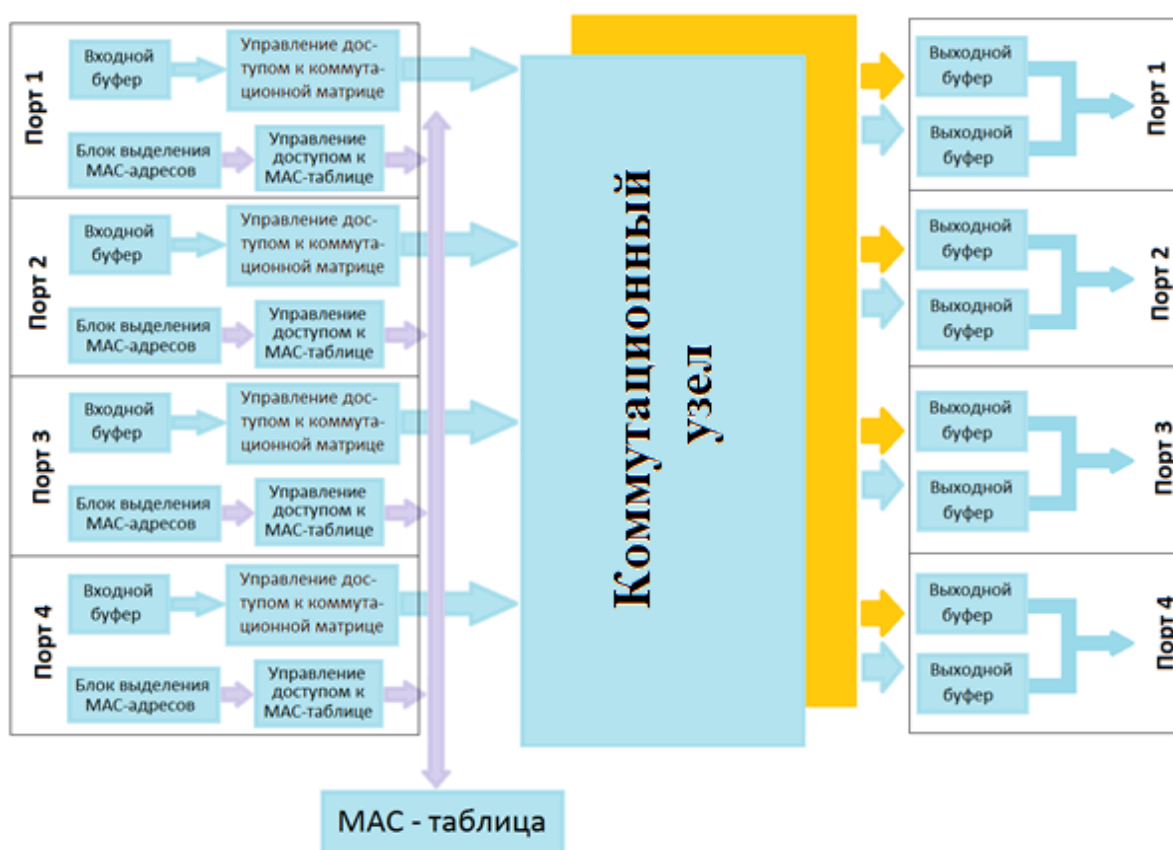


Рисунок 10 - Архитектура коммутатора

Следует отметить, что хоть на рисунке порты слева и справа нарисованы как отдельные устройства, физически это один и тот же порт. Но с точки зрения логических операций порт слева является источником информации, по нему устройство получает новые данные, а порт справа используется для отправки информации подсоединенному устройству. Поэтому мы можем один физический порт разделить на два независимых логических модуля.

У каждого физического порта есть своя независимая память во входной и выходной части для записи данных. Их объемы будут уточнены позже, после выбора типа коммутационного узла. Это устройство отвечает за передачу полученных данных внутри устройства. Для взаимодействия с ним

каждому порту нужен арбитр шины, который на рисунке обозначен, как «Управление доступом к коммутационному устройству». Для реализации адресации используется МАС-таблица и блок выделения МАС-адресов.

3.2. Выбор типа коммутационного узла устройства

Существует несколько возможных способов коммутации кадра внутри устройства. Каждое имеет свои преимущества и недостатки. Коммутационная матрица – самый быстрый способ взаимодействия между портами (рисунок 11). Однако, сложность её реализации возрастает пропорционально квадрату количества портов коммутатора, так как количество возможных узлов является квадратом числа портов.

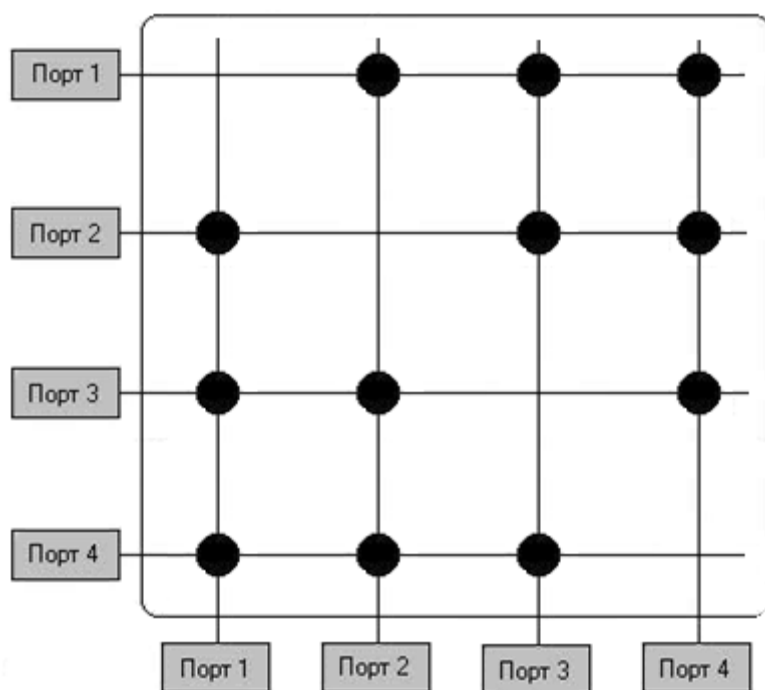


Рисунок 11– Коммутационная матрица

Матрица состоит из двух уровней двоичных переключателей, которые соединяют свой вход с одним из трех выходов в зависимости от номера порта назначения. Особенностью и преимуществом матриц этого типа является технология физической коммутации каналов. Внутри самой коммутационной матрицы буферизация не предусмотрена, поэтому необходимо иметь временное хранилище данных в самом источнике, т.е. во входном буфере порта.

Коммутаторы с общей шиной используют для связи портов высокоскоростную шину, используемую в режиме разделения по времени. Пример архитектуры с использованием общей шины приведен на рисунке 12.

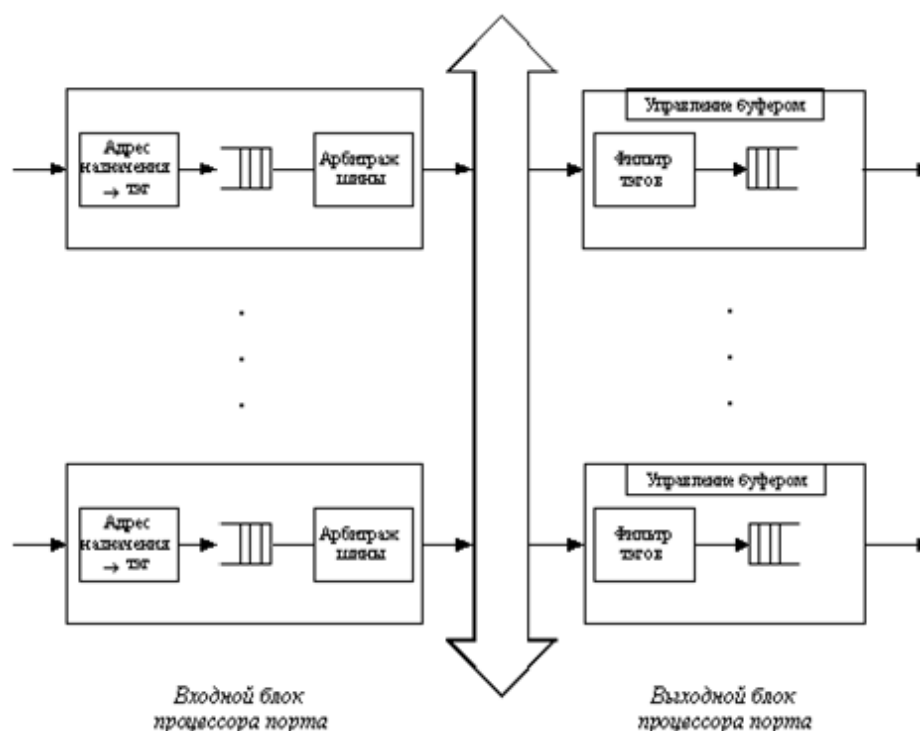


Рисунок 12 – Архитектура коммутатора с общей шиной

Основные требования в такие архитектуры накладываются на производительность общей шины. Так как передача кадров должна происходить в псевдопараллельном режиме между несколькими портами, она должна работать со скоростью в N -раз выше, чем скорость поступления данных во входные блоки. Время передачи от отправителя к получателю выше, чем в коммутационной матрице. Буферизация в данном случае, как и в коммутационной матрице не предусматривается, но при такой архитектуре требуются буферы данных на выходном порте. Процесс разработки более трудоемкий, так как необходимо синхронизировать потоки данных, значительно отличающиеся по скорости. Главным достоинством данного подхода является легкая масштабируемость проекта, если соблюдаются физические требования на скорость общей шины.

Таким образом, исходя из требований ТЗ к масштабируемости числа портов коммутатора и невысокой себестоимости конечного изделия, а также простоты реализации максимально удовлетворяет этим требованиям архитектура на основе коммутационной матрицы.

Данный выбор обусловлен тем, что для реализации коммутационной матрицы потребуется разработать только простейший алгоритм очереди доступа к каждому выходному буферу порта и независимые 12 коммутационных каналов, что вкупе с малыми требованиями по частоте работы матрицы позволит выбрать модель ПЛИС из малобюджетной линейки за сравнительно невысокую стоимость на рынке, а при масштабировании потребуется лишь увеличение числа каналов коммутации при неизменной частоте работы матрицы и добавлению однотипных блоков очереди доступа на каждый новый порт.

В случае архитектуры на основе общей шины при реализации 4-х портового коммутатора частота её работы должна быть не менее 50 МГц, чтобы обеспечить пропускную способность 100 Мбит/с на канал, что вполне приемлемо даже для малобюджетных моделей ПЛИС от ведущих производителей, однако при увеличении количества портов, например, до стандартных 24-х портов, это значение должно составлять уже не менее 300 МГц, что может обеспечить только высокопроизводительные модели ПЛИС с довольно высокой стоимостью. Также необходимо учитывать, что при выборе данного решения требуется разработка сложного алгоритма псевдопараллельного коммутирования кадров между портами, реализация которого также ведет к существенному повышению требований в отношении технических характеристик ПЛИС.

При разработке коммутаторов с большим числом портов, как правило, используется гибрид двух вышеописанных технологий. Все порты разбиваются на небольшие группы, которые объединяются в коммутационную матрицу, так как ее характеристики лучше, чем у архитектуры с использованием общей шины. В свою очередь коммутационные матрицы соединяются между собой с помощью общей шины. Таким образом, разрабатываемая коммутационная матрица, может быть использована для дальнейшего масштабирования проекта.

3.3. Блок выделения MAC-адресов

Так как коммутаторы идентифицируют пользователей ЛВС по их MAC-адресам, при разработке архитектуры необходимо предусмотреть специальный механизм, позволяющий параллельно с процессом приема кадра выделять «адрес получателя» и «адрес отправителя» из него.

Мы разрабатываем коммутатор L2 уровня, поэтому адресами для нас являются MAC-адреса. Так как эти 2 поля в кадре следуют сразу после преамбулы, то целесообразно немедленно после окончания процесса их выделения обращаться к таблице соответствий MAC-адресов. Таким образом данное решение позволит существенно снизить предъявляемые к быстродействию таблицы соответствия требования, так как поиск соответствующей записей в её памяти будет происходить параллельно процессу принятия потока данных.

Это же решение уменьшает временную задержку при обработке кадра, потому что в большинстве случаев к моменту окончания процесса приема ответ от таблицы соответствия с номером порта назначения уже будет получен, и кадр будет сразу транслироваться в выходной буфер соответствующего порта, а значит задержка внутри коммутатора будет определяться в основном длиной кадра. Эта задержка вызвана тем, что на текущем этапе разработки для упрощения логики устройства передача данных невозможна до полной записи пакета в промежуточную память. Соответственно, чем меньше длина пакета, тем меньше будет задержка.

3.4. Алгоритм работы таблицы соответствия

Отличительной особенностью коммутаторов от других устройств связи в сети является использование для коммутации трафика ассоциативной памяти, в которой хранятся данные с адресами всех устройств, подключенных к нему и соответствие их физическим портам устройства. Эта таблица должна быть актуальной, поэтому необходимо постоянно обновлять ее содержимое, а также удалять устаревшие данные.

Для разрабатываемого устройства был разработан алгоритм, который позволяет отказаться от постоянной перепроверки времени жизни записей. При получении нового пакета все записи в таблице последовательно сравниваются с адресом получателя и адресом отправителя. В процессе сравнения каждая запись проверяется по меткам времени последнего использования, является ли она устаревшей. Если это значение превышает порог, запись удаляется, как устаревшая.

Когда в таблице обнаруживается адрес отправителя, время его последнего использования обнуляется. В случае если в таблице такая запись не обнаружена, в таблицу вносится новая строка с данным MAC-адресом. Адрес

получается сравнивается параллельно с адресом отправителя с записями в таблице. Если в таблице обнаруживается данный адрес, извлекается номер ассоциированного физического порта, куда следует перенаправить трафик. В текущем устройстве не реализован полноценный ARP протокол, поэтому если адрес получателя в таблице не был найден, нет возможности сделать опрос всех подключенных устройств. Чтобы избежать потери данных соответствующий пакет рассылается на все подключенные порты.

Разработанный алгоритм позволяет избежать необходимости создавать отдельный поток обработки, отвечающий за проверку времени жизни всех записей. При этом он гарантирует, что в таблице соответствия всегда содержатся только записи с адресами активных в данный момент пользователей.

3.5. Временные задержки при обработке кадров

Основываясь на принятых в предыдущих подразделах решениях, целесообразно рассчитать временную задержку по основным этапам, вносимую устройством при обработке кадров, и свести их в таблицу 2.

Таблица 2

Временные задержки при обработке кадров разного размера

Размер кадра, байт	Этап приема кадра, мкс	Этап ком- мутации, мкс	Этап пере- дачи, мкс	Общая задержка, вно- симая коммутатором, мкс
46	5,76	1,44	5,76	12,96
300	26,08	6,52	26,08	58,68
600	50,08	12,52	50,08	112,68
900	74,08	18,52	74,08	166,68
1200	98,08	24,52	98,08	220,68
1500	122,08	30,52	122,08	274,68

Как видно из таблицы 2, вносимая коммутатором временная задержка при обработке кадра растет пропорционально длине обрабатываемого кадра. Так как максимальный размер кадра по стандарту IEEE 802.3 является 1500 байт, можно утверждать, что максимальная задержка на обработку кадра составляет не более 275 мкс.

3.6. Организация внутренних буферов

Одной из задач, с которой пришлось столкнуться в ходе разработки проектируемого устройства, стал способ разделения границ кадров внутри памяти принимающего и передающего буферов. Так как они организованны по типу FIFO, кадры хранятся в них непрерывно и отсутствует возможность доступа извне к отдельным ячейкам их памяти. Суть проблемы проиллюстрирована на рисунке 13.

0x000	55	55	55	55	55	55
0x006	55	D5
.
.
0x060	55	55	55
0x066	55	55	55	55	D5	...
.
.
0x17A	...	55	55	55	55	55
0x180	55	55	D5
.
.

Рисунок 13 – Разметка памяти буфера с данными

На рисунке разными цветами выделены «границы» кадров, но, так как используются буферы типа FIFO, нет возможности определить эти границы до момента чтения преамбулы следующего кадра, так как такая последовательность байт не может повторяться в остальных заголовках кадра, по ней можно точно определить границу начала следующего кадра. Для решения возникшей проблемы было решено использовать счетчик длины пакета, который осуществляет подсчет числа байт в принимаемом потоке, и отдельный блок памяти для хранения результатов его счета. Как только заканчи-

вается процесс приема очередного кадра, блок управления счетчика запишет в соответствующую память содержимое счетчика с информацией о длине кадра. Таким образом, данные могут храниться в буфере до момента коммутации, после чего необходимо считать из блока памяти информацию о длине коммутируемого кадра, загрузить в реверсивный счетчик и начать процесс обратного счета и считывания кадра из буфера одновременно. Когда реверсивный счетчик досчитает до нуля, следует немедленно прекратить процесс считывания кадра, так как это указывает, но что следующий байт уже будет относиться к следующему кадру.

3.7. Обработка широковещательного трафика

Еще одна техническая проблема, которую пришлось решить, возникла из-за требований ТЗ к минимальной задержке обрабатываемого трафика. Она связана с тем, что широковещательного кадра необходимо передать в выходные буферы всех портов устройства одновременно, однако любой из них может быть в текущий момент времени занят. При этом, так как буферы организованы по типу FIFO, единожды считанная из них информация не может быть считана повторно. Возможным решением проблемы может быть блокирование всех каналов коммутации по мере их освобождения и одновременная передача кадра сразу во все выходные буферы портов коммутатора. Однако такой подход подразумевает дополнительную задержку при коммутации широковещательного трафика, и, следовательно, не удовлетворяет требованиям ТЗ.

Другой подход к решению проблемы подразумевает наличие отдельного буфера для широковещательного трафика на каждый порт, тогда по отдельной, независимой от основной, матрице коммутации может быть осуществлена передача кадра сразу во все буферы выходных портов параллельно с остальными активными каналами коммутации, таким образом временно блокируется работа только тех портов, которые в данный момент хотят осуществить передачу широковещательного кадра, остальные же порты могут независимо работать и передавать трафик по основной матрице коммутации. Дополнительная матрица для коммутации широковещательного трафика может быть организована по тому же принципу, что и основная, за исключением того, что упрощается её архитектура, ввиду гораздо меньшего числа возможных вариантов для коммутации каналов (например, для 4-х

портового коммутатора всего 4 возможных варианта против 12 для коммутации по основной матрице).

В передающей части для того, чтобы решить вопрос очередности считывания пакетов из двух буферов должен быть предусмотрен простейший конечный автомат, получивший при разработке название «менеджер очередей». Он осуществляет поочередную передачу кадров при наличии кадров в каждом из них, либо отдает приоритет каналу с заполненным буфером, если буфер другого канала пуст. Выходная часть устройства имеет структуру, представленную на рисунке 14.



Рисунок 14 – Структура выходного буфера

станциям, что позволяет перейти к этапу подготовки прошивки и программировании ПЛИС.

4.2. Компиляция проекта и отчеты

После проведения моделирования с учетом временных задержек необходимо синтезировать разработанный проект устройства и провести анализ задействованных ресурсов, что позволит оценить правильность выбора производителя и модели ПЛИС. На рисунке 16 приведен отчет с результатами работы САПР ISE 14.7 от компании Xilinx, предназначенной для сквозного проектирования и отладки проектов на ПЛИС серии Spartan-6.

Анализируя представленный отчет можно заключить, что процент использования доступных КЛБ для синтеза схемы составляет 99 %, и 32 из 32 (100 %) доступных внутренних блоков памяти по 16 Кбит. Следовательно, выбранная модель ПЛИС максимально соответствует необходимым требованиям для реализации разработанной схемы, в чем также наглядно можно убедиться из схемы размещения на кристалле, приведенной на рисунке 17.

Device Utilization Summary			
Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	4,954	11,440	43%
Number used as Flip Flops	4,938		
Number used as Latches	16		
Number used as Latch-thrus	0		
Number used as AND/OR logics	0		
Number of Slice LUTs	4,017	5,720	70%
Number used as logic	2,746	5,720	48%
Number using O6 output only	1,333		
Number using O5 output only	470		
Number using O5 and O6	943		
Number used as ROM	0		
Number used as Memory	413	1,440	28%
Number used as Dual Port RAM	352		
Number using O6 output only	0		
Number using O5 output only	12		
Number using O5 and O6	340		
Number used as Single Port RAM	40		
Number using O6 output only	11		
Number using O5 output only	0		
Number using O5 and O6	29		
Number used as Shift Register	21		
Number using O6 output only	17		
Number using O5 output only	0		
Number using O5 and O6	4		
Number used exclusively as route-thrus	858		
Number with same-slice register load	825		
Number with same-slice carry load	33		
Number with other load	0		
Number of occupied Slices	1,421	1,430	99%
Number of RAMB16BWERS	32	32	100%

Рисунок 16 – Отчет из ISE 14.7 о количестве задействованных ресурсов

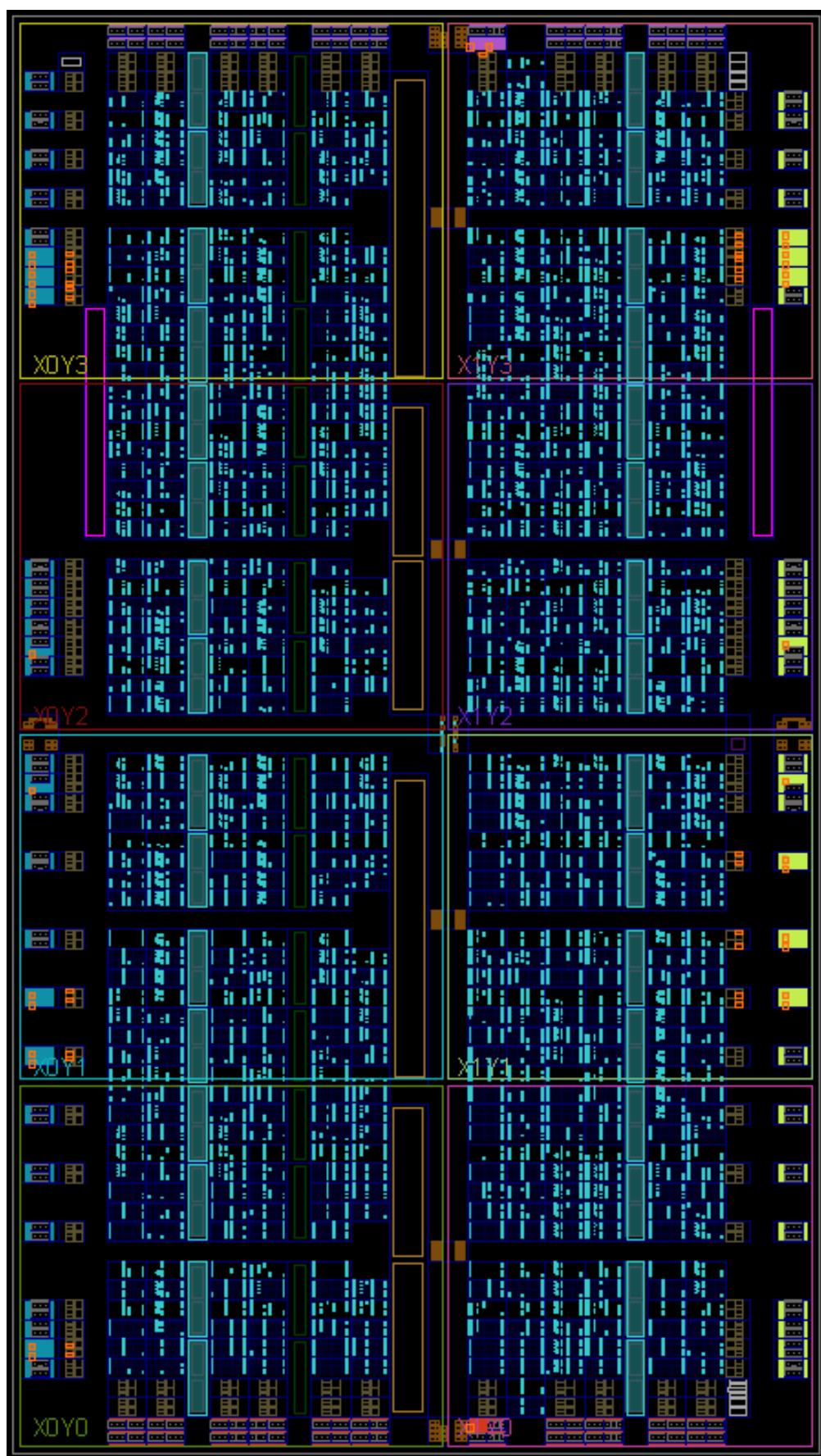


Рисунок 17 – Размещение схемы на кристалле

4.3. Макетирование устройства

Последним этапом разработки является макетирование спроектированного устройства. Корпус макета изготовлен с использованием 3D-принтера на предприятии. В качестве устройств, работающих на физическом уровне, были выбраны Ethernet-трансиверы LAN8720 ETH Board, описанные в подразделе 2.2 настоящей работы. Для реализации MAC-уровня была выбрана отладочная плата Mojo V3 от компании Alchitry с ПЛИС Xilinx Spartan-6 XC6SLX9, а также установленной внешней flash-памятью для хранения конфигурации ПЛИС (рисунок 18). Результат макетирования конечного устройства представлен на рисунке 19.

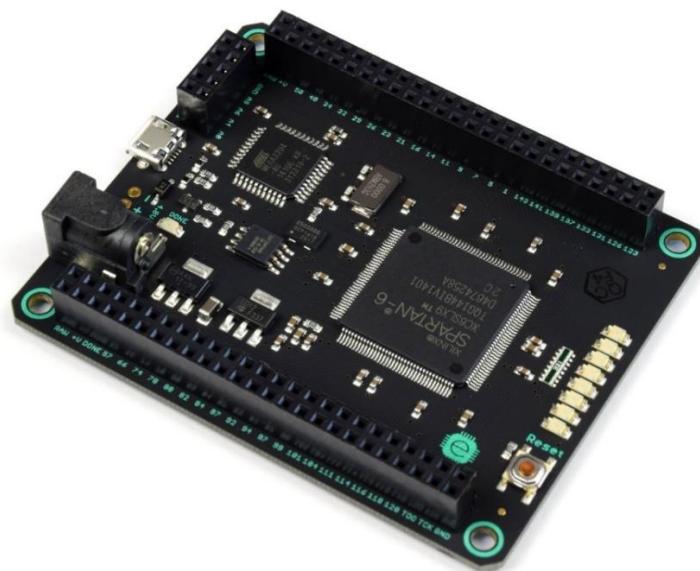


Рисунок 18 – Отладочная плата MOJO V3 от Alchitry



Рисунок 19 – Опытный образец разработанного устройства

5. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ВКР

5.1. Оценка затрат на разработку и отладку устройства

Для расчета затрат на оплату труда прежде всего необходимо составить детализированный план выполнения работ, а также оценить трудоемкость каждого из этапов разработки. Перечень этапов работы и трудоемкость их выполнения представлены в таблице 3.

Трудоемкость этапов разработки

Таблица 3

№	Наименование работ	Трудоемкость, чел./дни	
		Студент	Научный руководитель
1.	Разработка ТЗ	—	3
2.	Анализ ТЗ и обзор литературы по теме работы	15	—
3.	Проектирование архитектуры коммутатора	24	5
4.	Выбор производителя и модели ПЛИС	5	—
5.	Разработка прошивки для ПЛИС	30	—
6.	Тестирование и отладка устройства	10	2
7.	Оформление пояснительной записки	5	—
8.	Оформление иллюстративного материала	4	—
Итого:		93	10

Для оценки величины затрат на заработную плату и социальные отчисления участников разработки необходимо определить размер дневной ставки заработной платы для студента и научного руководителя. Для этого необходимо оклад за месяц разделить на количество рабочих дней в месяце (21 рабочий день).

В качестве месячной заработной платы студента принимается заработная плата инженера в размере 40 000 руб. Таким образом:

- дневная ставка студента составляет 1 904 руб./день;

- дневная ставка научного руководителя составляет 2 857 руб./день;
Расходы на основную заработную плату определяются по формуле:

$$З_{\text{осн.з/пл}} = \sum_{i=1}^k T_i \cdot C_i = 93 \cdot 1\,904 + 10 \cdot 2\,857 = 205\,642 \text{ (руб.)},$$

где $З_{\text{осн.з/пл}}$ – расходы на основную заработную плату исполнителей, k – количество исполнителей, T_i – время, затраченное i -м исполнителем на проведение разработки (дни), C_i – ставка i -го исполнителя (руб./день).

Расходы на дополнительную заработную плату исполнителей определяются по формуле:

$$З_{\text{доп.з/пл}} = З_{\text{осн.з/пл}} \cdot \frac{Н_{\text{доп}}}{100} = 205\,642 \cdot \frac{8.3}{100} = 17\,068.27 \text{ (руб.)},$$

где $З_{\text{доп.з/пл}}$ – расходы на дополнительную заработную плату исполнителей, $З_{\text{осн.з/пл}}$ – расходы на основную заработную плату исполнителей (руб.), $Н_{\text{доп}}$ – норматив дополнительной заработной платы (%), при выполнении расчетов принимается равным 8,3 %.

Расходы на страховые взносы на обязательное пенсионное, медицинское и социальное страхование в государственные внебюджетные фонды (Пенсионный фонд РФ, Фонд социального страхования РФ, Федеральный фонд обязательного медицинского страхования) определяются по формуле:

$$\begin{aligned} З_{\text{соц}} &= (З_{\text{осн.з/пл}} + З_{\text{доп.з/пл}}) \cdot \frac{Н_{\text{соц}}}{100} = (205\,642 + 17\,068.27) \cdot \frac{30.2}{100} = \\ &= 67\,258.5 \text{ (руб.)}, \end{aligned}$$

где $З_{\text{соц}}$ – отчисления на социальные нужды с заработной платы, $З_{\text{осн.з/пл}}$ – расходы на основную заработную плату исполнителей (руб.), $З_{\text{доп.з/пл}}$ – расходы на дополнительную заработную плату исполнителей (руб.), $Н_{\text{соц}}$ – норматив отчислений на единый социальный налог, при выполнении расчетов принимается равным 30.2 %.

Перечень комплектующих и материалов, необходимых для разработки и отладки устройства, представлен в таблице 4.

Перечень комплектующих и материалов

Таблица 4

№	Материал	Единица измерения	Количество	Цена, руб.	Сумма, руб.
1	Программатор ПЛИС Xilinx Platform Cable USB	шт.	1	1 635.00	1 635.00
2	Кабель USB 2.0, 1м	шт.	1	130.00	130.00
3	Сетевой патч-корд LAZSO WE-52U	шт.	4	110.00	440.00
4	Канцелярские товары	—	—	—	1 245.00
Итого затрат					3 450.00

Так как в ходе выполнения работы на компьютере использовалось бесплатное программное обеспечение, его стоимость в перечень не включена.

Оценка затрат на комплектующие и материалы производится по формуле:

$$\begin{aligned}
 Z_m &= \sum_{l=1}^L G_l \cdot C_l \left(1 + \frac{H_{т.з}}{100}\right) = (1 \cdot 1\,635 + 1 \cdot 130 + 4 \cdot 110) \left(1 + \frac{10}{100}\right) = \\
 &= 3\,795 \text{ (руб.)},
 \end{aligned}$$

где Z_m – затраты на материалы/комплектующие, G_l – количество единиц l -го материала/комплектующего (шт), C_l – цена приобретения единицы l -го материала/комплектующего (руб./шт.), $H_{т.з}$ – норма транспортно-заготовительных расходов (%), при выполнении расчетов принимается 10 %.

Расчет амортизационных отчислений будет определяться издержками на амортизацию ПК и спецоборудования. При расчете суммы амортизационных отчислений по объектам основных средств, используются данные о классификации в соответствии с Постановлением Правительства РФ от 01.01.2002 N 1 (ред. от 28.04.2018) "О Классификации основных средств, включаемых в амортизационные группы", согласно которому используемое оборудование имеет следующие сроки полезного использования:

- персональный компьютер, относится ко второй амортизационной группе, срок полезного использования от 2 до 3 лет;
- цифровой осциллограф, относится к третьей амортизационной группе, срок полезного использования от 3 до 5 лет;

- цифровой мультиметр, относится к третьей амортизационной группе, срок полезного использования от 3 до 5 лет;

Амортизационные отчисления по i -му основному средству за год определяются по формуле:

$$A_i = Ц_{п.н.i} \cdot \frac{H_{ai}}{100}, \quad (1)$$

где A_i – амортизационные отчисления за год по i -му основному средству; $Ц_{п.н.i}$ – первоначальная стоимость i -го основного средства (руб.); H_{ai} – годовая норма амортизации i -го основного средства (%).

Тогда для объектов основных средств величина амортизационных отчислений за год составит по формуле (1):

- $A_{ПК} = 30\,990 \cdot \frac{33.3}{100} = 10\,226.7$ (руб.);
- $A_{ЦО} = 205\,990 \cdot \frac{20}{100} = 41\,198$ (руб.);
- $A_{ЦМ} = 12\,520 \cdot \frac{20}{100} = 2\,504$ (руб.);

Для расчета величины амортизационных отчислений по основным средствам, используемым в ВКР необходимо определить время, в течение которого оборудование используется студентом. После чего величина амортизационных отчислений по основным средствам, используемым при работе над ВКР, определяется по формуле:

$$A_{iВКР} = A_i \cdot \frac{T_{iВКР}}{12}, \quad (2)$$

где $A_{iВКР}$ – амортизационные отчисления по i -му основному средству, используемому студентом в работе над ВКР; A_i – амортизационные отчисления за год по i -му основному средству; $T_{ВКР}$ – время потраченное на разработку и проектирование (мес.).

На разработку и отладку с использованием ПК было потрачено около трех месяцев, следовательно, амортизационные отчисления составляют:

$$A_{ПК ВКР} = 10\,226.7 \cdot \frac{3}{12} = 2\,556.68 \text{ (руб.)};$$

На отладку устройства с использованием спецоборудования был потрачен один месяц, следовательно, амортизационные отчисления составляют:

$$A_{ЦО ВКР} = 41\,198 \cdot \frac{1}{12} = 3\,433.17 \text{ (руб.)};$$

$$A_{ЦМ ВКР} = 2\,504 \cdot \frac{1}{12} = 208.67 \text{ (руб.)};$$

Величина накладных расходов рассчитывается по формуле:

$$З_{н.р.} = \alpha_{н.р.} \cdot (З_{осн.з./пл} + З_{доп.з./пл}) = 0.2 \cdot (205\,642 + 17\,038.27) = 45\,536.1 \text{ (руб.)},$$

где $\alpha_{н.р.}$ – коэффициент накладных расходов, при выполнении расчетов принимается равным 0.2; $З_{доп.з./пл}$ – расходы на дополнительную заработную плату исполнителей; $З_{осн.з./пл}$ – расходы на основную заработную плату исполнителей (руб.);

Расчёт себестоимости процесса разработки и отладки устройства представлен в таблице 5.

Смета затрат на разработку и отладку устройства Таблица 5

№	Статьи затрат	Сумма, руб.
1.	Расходы на оплату труда и отчисления на социальные нужды	289 968.77
2.	Материалы и комплектующие	3 795
3.	Амортизационные отчисления	6 198.52
4.	Накладные расходы	45 536.1
ИТОГО затрат		345 498.39

5.2. Определение затрат на изготовление опытного образца

Выполнение ВКР предполагает изготовление опытного образца устройства, поэтому целесообразно произвести оценку затрат на его изготовление.

Перечень материалов и комплектующих изделия, необходимых для его изготовления, представлен в таблице 6.

Таблица 6

Перечень материалов и комплектующих для изготовления

№	Материал	Единица измерения	Количество	Цена, руб.	Сумма, руб.
1	Отладочная плата Mojo V3 FPGA Development Board	шт.	1	1 990.00	1 990.00
2	Ethernet-модуль WaveShare LAN8720 ETH Board	шт.	4	290.00	1 160.00

Продолжение таблицы 6

№	Материал	Единица измерения	Количество	Цена, руб.	Сумма, руб.
3	Набор соединительных проводов (М-F), 10 см	шт.	1	590.00	590.00
4	Крепежный винт потайной, М3х10мм	шт.	12	3.00	36.00
5	Блок питания IP20	шт.	1	370.00	370.00
ИТОГО затрат					4 146.00

Величина затрат на покупку комплектующих и материалов для изготовления изделия составила по формуле (2):

$$З_{\text{п}} = 4\,146 \cdot \left(1 + \frac{10}{100}\right) = 4\,560.6 \text{ (руб.)};$$

Услуги сторонних организаций учитываются по их фактической стоимости за вычетом НДС. Перечень оказанных сторонними организациями услуг приведен в таблице 7.

Перечень услуг сторонней организаций Таблица 7

Наименование услуги	Фактическая стоимость за вычетом НДС, руб.
Изготовление пластикового корпуса устройства на 3D-принтере	820

Расчёт себестоимости изготовления опытного образца разработанного устройства представлен в таблице 8.

Смета затрат на изготовление опытного образца Таблица 8

№	Статьи затрат	Сумма, руб.
1.	Расходы на покупку комплектующих и материалов	4 560.6
2.	Услуги сторонней организации	820
ИТОГО затрат		5 380.6

Совокупные расходы на разработку, отладку и изготовление опытного образца устройства представлены в таблице 9.

Итоговая смета затрат на ВКР Таблица 9

№	Статьи затрат	Сумма, руб.
1.	Затраты на проектирование	345 498.39
2.	Расходы на изготовление образца	5 380.6
ИТОГО затрат		369 428.99

5.3. Общие экономические показатели

В данном разделе ВКР была рассчитана себестоимость разработки и изготовления опытного образца устройства. Итоговая себестоимость выполнения работ по ВКР составила 369 429 рублей, однако, большую часть этой суммы формируют затраты на саму разработку устройства, производство же опытного образца составило всего 5 380 рублей. Таким образом, невысокая себестоимость конечного изделия позволила бы конкурировать ему в своем ценовом сегменте рынка. Однако, в случае серийного производства, возможно еще снизить себестоимость устройства за счет более тщательного подбора комплектующих компонентов и их закупки по оптовым ценам у официальных поставщиков.

ЗАКЛЮЧЕНИЕ

В рамках ВКР была рассмотрена сетевая модель коммутатора в контексте модели OSI, на основе которой сформированы требования к программной и аппаратной составляющей разрабатываемого устройства. Произведен сравнительный анализ различных вариантов реализации внутренних модулей и сделан выбор, удовлетворяющий требованиям ТЗ. Было разработано программное обеспечение и проведено его временное моделирование, как в идеальных условиях, так и с учетом временных задержек.

В качестве макета спроектирован 4-х портовый коммутатор уровня L2 на ПЛИС XC6SLX9 семейства Spartan-6 фирмы Xilinx с применением отладочной платы Mojo V3 фирмы Alchitry и Ethernet-трансиверов LAN8720 ETH Board фирмы WaveShare, работающих по технологии Fast Ethernet (спецификация IEEE 802.3u) со скоростью соединения между конечными узлами 100 Мбит/с. Проведено тестирования макета в реальной локальной сети предприятия.

Принятые в ходе разработки архитектуры решения позволили добиться минимальной задержки при обработке кадра, несмотря на реализацию промежуточной буферизации данных. Такой результат был достигнут за счет распараллеливания процессов приема кадра и поиска номера порта назначения в таблице соответствия коммутатора. Таким образом, задержка на обработку кадра зависит в основном от его длины и увеличивается пропорционально её росту. Результаты теоретического расчета и экспериментальные данные показывают, что задержка будет пренебрежимо мала и не будет оказывать существенного влияния на качество работы сети и результирующую скорость соединения между конечными узлами.

Главным достоинством разработанной архитектуры устройства является её масштабируемость. Благодаря использованию модульного программного обеспечения, разработка новых версий микропрограммы для новых устройств, с большим количеством портов и более высокой поддерживаемой скоростью передачи данных, значительно упростится за счет использования уже написанных модулей. С аппаратной точки зрения будет необходима только замена ПЛИС на более ресурсоемкую, для удовлетворения новых технических требований.

Разработанное устройство полностью удовлетворяет поставленному техническому заданию. В результате тестирования макета не было выявлено критически важных недостатков. При этом был сформирован список задач и доработок, необходимых для дальнейшего совершенствования устройства с целью использования его в системах, актуальных на современном рынке.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. СПб.: Питер, 2012. 960 с.
- [2] Современные сетевые технологии. Сетевой коммутатор. URL: <https://14bytes.ru/razbiraemsja-kak-rabotaet-kommutator-seti-ethernet-switch/> (дата обращения 05.03.19)
- [3] Сетевой коммутатор. URL: https://ru.wikipedia.org/wiki/Сетевой_коммутатор (дата обращения 14.03.19)
- [4] RMII™ Specification. URL: http://ebook.pldworld.com/eBook/-Telecommunications,Networks-/TCPIP/RMII/rmii_rev12.pdf (дата обращения 01.04.19)
- [5] Методические указания по экономическому обоснованию выпускных квалификационных работ бакалавров / сост.: Алексеева О.Г. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2013. 15 с.
- [6] Клайв М. Проектирование на ПЛИС. Курс молодого бойца. М.: Издательский дом «Додэка-XXI», 2007. 408 с.
- [7] Sara L.H., David M.H. Digital design and computer architecture: ARM Edition. London: Morgan Kaufman, 2015. 1621 с.
- [8] Рональд Д.Т., Уидмер Н.С. Цифровые системы. Теория и практика. 8-е изд. М.: Издательский дом «Вильямс», 2004. 1024 с.
- [9] Бибило П.Н. Основы языка VHDL. М.: СОЛОН-Р, 2002. 224 с.
- [10] Суворова Е.А., Шейнин Ю.Е. Проектирование цифровых систем на VHDL. СПб.: БХВ-Петербург, 2003. 576 с.
- [11] Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем. 2-е изд. М.: Горячая линия-Телеком, 2007. 636 с.
- [12] Армстронг Дж.Р. Моделирование цифровых систем на языке VHDL: пер. с англ. М.: Мир, 1992. 175 с.
- [13] Поляков А.К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры. М.: СОЛОН-Пресс, 2003. 320 с.
- [14] Бибило П.Н., Авдеев Н.А. VHDL. Эффективное использование при проектировании цифровых систем. М.: СОЛОН-Пресс, 2006. 344 с.
- [15] Зотов Ю.В. Проектирование цифровых устройств на основе ПЛИС фирмы XILINX в САПР WebPack ISE. М.: Горячая линия-Телеком, 2003. 624 с.
- [16] Бибило П.Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum. М.: СОЛОН-Пресс, 2005. 384 с.