МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет	*Кафедра*
вычислительной техники	«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки: «Программное обеспечение средств вычислительной техники и автоматизированных систем»

БАКАЛАВРСКАЯ РАБОТА

на тему

КОНТРОЛЛЕР СЦЕПИАЛИЗИРОВАННОГО ИНТЕРФЕЙСА «СТЫК-С1ФЛ»

Студент 31.05.22 Шерстобитов И.Д.
(подпись, дата)

Руководитель 31.05.22 Дубравин А.В.
(подпись, дата)

Нормоконтролёр 31.05.22 А.С. Бычков
(подпись, дата)

Работа допущена к защите (протокол заседания кафедры от 02.06.2022 г. №13)

Заведующий кафедрой _____ М.А. Митрохин
(подпись)

Работа защищена с отметкой хорошо (протокол заседания ГЭК от 20.06.22 № 14)

Секретарь ГЭК _____ Е.И. Гурин
(подпись)

Пенза 2022

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«26» апреля 2022г.

ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ

1. Студент Шерстобитов Иван Дмитриевич, гр.18ВВ2 факультета вычислительной техники.

2. Тема работы: «Контроллер специализированного интерфейса «СТЫК-С1ФЛ»

Тема утверждена приказом ПГУ №357/о от "26" апреля 2022 г.

3. Руководитель бакалаврской работы к.т.н., доцент каф. ВТ ПГУ Дубравин А.В.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Устройство предназначено для преобразование логических сигналов ПЭВМ, получаемые по локальной шине PCI ПЭВМ, в биимпульсный сигнал по ГОСТ 27232-87, пригодный для передачи по каналам связи и осуществление обратного преобразования при приеме данных из канала связи. В качестве контроллера использовать ПЛИС ALTERA семейства FLEX10K

5. Перечень подлежащих разработке вопросов

Требуется произвести анализ предметных областей: выбрать функциональный состав устройства, разработать принципы функционирования Контроллер специализированного интерфейса «СТЫК-С1ФЛ», разработать структурную схему устройства, осуществить выбор элементной базы, разработать принципиальную схему, разработать функциональную схему, разработать и отладить программную часть устройства, листинг и временные диаграммы. Устройство должно принимать и передавать данные по физическим линиям. Контроллер необходимо подключить к ПЭВМ через разъём PCI. Установить драйвер на ПЭВМ для работы с данным контроллером. Также выполнить задание по БЖД.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	24	26.04.2022 – 07.05.2022	
2.	Выбор функционального состава устройства	12	08.05.2022 – 11.05.2022	
3.	Разработка схемы электрической структурной	6	12.05.2022 – 13.05.2022	
4	Разработка схемы электрической функциональной	7	14.05.2022 – 15.05.2022	
5.	Разработка схемы электрической принципиальной	9	16.05.2022 – 18.04.2022	
6.	Разработка временных диаграмм	15	19.04.2022 – 23.05.2022	
7.	Отладка и тестирование программной части	6	24.05.2022 – 25.05.2022	
8.	Выполнение задания по БЖД	6	26.05.2022 – 27.05.2022	
9.	Оформление пояснительной записки	15	28.05.2022 - 31.05.2022	

Дата выдачи задания "26" апреля 2022 г.

Руководитель бакалаврской работы 26.04.22 Дубравин А.В.
(подпись, дата)

Задание к исполнению принял студент 26.04.22 Шерстобитов И.Д.
(подпись, дата)

Работу к защите допустить
Декан факультета 17.06.2022 Фионова Л.Р.
(подпись, дата)

Реферат

Работа содержит 109 листов, 8 рисунков, 2 таблицы.

ПЛИС, КОНТОЛЛЕР, ПЭВМ, ВВ, ПК, СЗУПВ, ИС, КАНАЛ СВЯЗИ, *PCI, ALTERA FLEX10K, QUARTUS II, DWORD, FIFO.*

Цель ВКР – разработка контроллера специализированного интерфейса «СТЫК-С1ФЛ».

Разработанное устройство предназначено для преобразования логических сигналов ПЭВМ, получаемые по локальной шине *PCI* ПЭВМ, в биимпульсный сигнал по ГОСТ 27232-87, пригодный для передачи по каналам связи и осуществление обратного преобразования при приеме данных из канала связи. В качестве микроконтроллера использован ПЛИС *ALTERA FLEX10K10LC84-4.*

В ходе выполнения ВКР были разработаны структурная, функциональная и принципиальная схемы контроллера специализированного интерфейса «СТЫК-С1ФЛ», разработаны и отлажены основные программные модули.

Графическая часть выпускной квалификационной работы содержит следующие чертежи:

- схема электрическая структурная – 1 лист формата А1;
- схема электрическая принципиальная – 1 лист формата А1;
- схемы электрическая функциональная – 1 лист формата А1;
- временные диаграммы – 1 лист формата А1.

Данное устройство позволяет передавать данные по каланам связи.

Содержание

Перечень сокращений и обозначений 5
Введение 6
1 Цели и задачи проектирования устройства 8
1.1 Постановка задачи 8
1.2 Обзор состояния рынка устройств передачи сигналов 8
1.3 Выбор вариантов реализации проекта 9
1.4 Обоснование технической реализации проекта 14
1.5 Разработка устройства 15
1.5.1 Выбор и обоснование схемы электрической структурной 15
1.5.1.1 Краткое описание шины PCI 15
1.5.1.2 Характеристики локальной шины PCI 19
1.5.2 Описание работы по схеме электрической структурной 38
1.5.3 Выбор и обоснование схемы электрической функциональной 42
1.5.4 Описание работы по схеме электрической функциональной 45
1.5.4.1 Описание работы тракта приёма информации 45
1.5.4.2 Описание работы тракта передачи информации 46
1.5.4.3 Описание работы интерфейсной части 48
1.5.5 Выбор и обоснование схемы электрической принципиальной 54
1.5.6 Описание работы по схеме электрической принципиальной 58
1.5.7 Расчёт потребляемой мощности. 61
2 Обеспечение безопасности жизнедеятельности 63
2.1 Анализ условий труда 63
2.2 Разработка рекомендаций по созданию безопасности условий труда 65
2.2.1 Нагрузка на органы зрения 65
2.2.2 Эргономика клавиатуры 68
2.2.3 Правильная посадка 69
2.2.4 Требования к освещению помещений и рабочих мест с ПЭВМ 69
2.3 Расчёт искусственного освещения помещения для рабочих мест 72

Заключение 77
Список используемых источников 78
Приложение А – Листинг программы 79
Приложение Б – Схема электрическая структурная 89
Приложение В – Схема электрическая функциональная 90
Приложение Г – Схема электрическая приципиальная 90
Приложение Д – Перечень элементов 90
Приложение Е – Временные диаграммы 90
Приложение Ж – Презентация 100

Перечень сокращений и обозначений

FIFO (First In – First Out) – первым вошел – первым вышел

LAB (logic-array block) – логические матрические блоки

LE (logic element) – логические элементы

PCI (Peripheral component interconnect) – взаимосвязь периферийных компонентов

БМК – базовый матричный кристалл

ВДТ – видео дисплейные терминалы

В/В – ввод и вывод

ГОСТ – региональный стандарт, принятый Межгосударственным советом по стандартизации, метрологии и сертификации

ЗУПВ – запоминающее устройство с произвольным доступом

ИС – интегральная схема

КС – канал связи

НС – нано секунды

ПЗУ – постоянное запоминающие устройство

ПК – персональный компьютер

ПЛИС – программируемая логическая интегральная схема

ППВМ – программируемые потребителем вентильные матрицы

ПЭВМ – персональная электронно-вычислительная машина

СЗУПВ – статическое запоминающее устройство с произвольной выборкой

СППЗУ – стираемые программируемые постоянные запоминающие устройства

ФЛ – физические линии

ЭППЗУ – электрически перепрограммируемое ПЗУ

ЭСППЗУ – электрически стираемое перепрограммируемое ПЗУ

Введение

Электронно-вычислительные машины применяются сегодня практически во всех областях человеческой деятельности. Широкое использование в выпускаемых сегодня электронных устройствах получили контроллеры, обеспечивающие решение задач автоматизации управления и передачи данных. Важную роль среди них производства играют автоматизированные системы сбора, обработки и передачи информации.

Последние годы характеризуются бурным развитием интегральной микроэлектроники и конструирования радиоэлектронной аппаратуры на основе интегральных микросхем. Внедрение в радио, а также и электронную аппаратуру интегральных микросхем в значительной мере изменило методы проектирования и производства различной радиоэлектронной аппаратуры, повысило ее надежность и экономичность при одновременном уменьшении габаритов и веса. Благодаря интегральным микросхемам значительно расширилось внедрение радиоэлектроники в различные области науки и техники. Применение современной элементной базы позволило не только усовершенствовать старые, но и создать новые методы проектирования, конструирования и производства бытовой радиоаппаратуры. Малые габариты, масса, потребляемая мощность, высокая надежность, долговечность и большое множество электронных приборов различного функционального назначения позволили создать широкую номенклатуру новейших устройств во всех сферах применения интегральных микросхем.

В настоящее время рынок услуг телекоммуникации развивается как никогда ранее. Современные компьютеры стали более производительными, эргономичными, приятными и удобными в работе. И сейчас очень остро встает проблема высокопроизводительного, дешевого соединения компьютеров на больших расстояниях.

В настоящий момент, существуют решения для данной проблемы:

- прокладка оптоволоконной линии связи;

- использование аналоговых модемов для выделенных линий;

- использование высокоскоростных цифровых модемов;

- применение устройств, использующих для передачи данных лазерный луч.

Но все вышеперечисленные решения могут не удовлетворять по тем или иным причинам. Например, прокладка оптоволокна стоит пока еще слишком дорого из-за сложного оконечного оборудования, кроме того, имеется очень большое количество уже проложенных медных линий, но зачастую не использованных. Использование обычных аналоговых модемов позволяет достичь больших расстояний с применением существующих телефонных линий, но скорость передачи у них уже довольна мала. Цифровые модемы – довольно дорогие изделия, работающие на достаточно небольшие расстояния по существующим телефонным линиям. Казалось бы, самым привлекательным решением является использование средств связи, применяющих лазерный луч для передачи данных – им не нужно никаких физических каналов связи, не нужны диапазоны радиоэфира. Но надежность данной технологии еще пока не слишком высока, так как сильный дождь, туман, снегопад, стая птиц, могут прервать связь между оконечными устройствами. К тому же данные устройства весьма сложно спозиционировать.

Поэтому является целесообразной разработка контроллера, осуществляющего связь на большие расстояния по существующим каналам связи и обладающего низкой стоимостью. Именно таким является контролер специализированного интерфейса «СТЫК-С1ФЛ» использующий биимпульсное кодирование сигнала и адаптированный для применения на физических каналах связи.

1 Цели и задачи проектирования устройства

1.1 Постановка задачи

Задачей данного дипломного проекта является разработка «Контроллера специализированного интерфейса «СТЫК-С1ФЛ» (далее – контроллера). Назначение этого устройства – преобразование логических сигналов ПЭВМ, получаемые по локальной шине PCI ПЭВМ, в биимпульсный сигнал по ГОСТ 27232-87, пригодный для передачи по каналам связи и осуществление обратного преобразования при приеме данных из канала связи. Данный алгоритм реализуется в устройстве под управлением ПЭВМ посредством стандартной логики локальной шины PCI.

1.2 Обзор состояния рынка устройств передачи сигналов

На сегодняшний день в России подобные устройства производятся, но большинство из них являются внешними устройствами, подключаемыми к ЭВМ при помощи интерфейса *RS-232C*.

Основным предназначением устройств данного типа является преобразование логических сигналов ПЭВМ, получаемых по интерфейсу *RS-232C*, в биимпульсный сигнал при передаче данных в канал связи и преобразование биимпульсного сигнала в логические сигналы, адаптированные для ПЭВМ, при приеме данных из канала связи. Данный алгоритм реализуется в устройстве под управлением от ПЭВМ с помощью служебных логических сигналов стандартного интерфейса *RS-232C*.

В современных условиях нередко возникает потребность в устройствах к внутреннему интерфейсу ЭВМ, например интерфейсу *PCI*.

Разрабатываемое устройство и является альтернативным вариантом, подключаемым к интерфейсу *PCI*.

1.3 Выбор вариантов реализации проекта

Одним из наиболее перспективных направлений в развитии современной микроэлектроники становится использование ПЛИС — программируемых логических интегральных схем. Значительное снижение цен и увеличение производительности обусловливают возможность использования ПЛИС во многих практических схемах.

В настоящее время существует два десятка фирм-изготовителей ПЛИС, производящие большое многообразие архитектур от простых ПЛИС с числом внешних выводов менее 44 и числом вентилей 1000. до сложных - с числом выводов более 44, числом вентилей более 1000 и с функцией программирования межсоединений.

Программируемые потребителем вентильные матрицы (ППВМ) обладают наивысшей функциональной гибкостью благодаря возможности их программирования и перепрограммирования в условиях эксплуатации. Сегодня ППВМ догнали по быстродействию и интеграции БМК (с минимальным размером 1,4-2,0 мкм). В арсенале разработчиков в настоящее время находятся десятки различных ППВМ - архитектур, построенных на базе статических запоминающих устройств произвольной выборки (СЗУПВ), а также на базе конфигурационных перемычек.

Существующие архитектуры ППВМ подразделяются на два класса: мелко - и крупнозернистые. Мелкозернистые кристаллы (например, ППВМ компаний *ACTEL, CROSSPOINT, QUICRLOGIC* с прожигаемыми перемычками и ППВМ на базе ЗУПВ компаний *PIKINTON CLASS*, выпускаемые *PLESSEY* и *TOSHIBA*) имеют в своей структуре большое число мелких макроэлементов вентильного уровня, из которых посредством коммутации можно получать широкую гамму функций от триггеров до сложных конечных автоматов. Мелкозернистые кристаллы весьма близки по конструкции к БМК и строятся на простых функциональных блоках, которые

могут состоять всего из нескольких n- и p-канальных транзисторов. Ведущая архитектура типа «море вентилей» обеспечивает при тех же минимальных размерах технологического процесса примерно в 10 раз более высокую плотность упаковки по сравнению с ППВМ. Обусловлено это тем, что элементы управления конфигурацией в ППВМ имеют достаточно большие размеры, а сетка межсоединений - фиксированную структуру.

В области ППВМ с мелкозернистыми архитектурами прочные позиции на рынке занимают семейства фирмы *ACTEl*. Базовый логический модуль в них выполнен на основе двухуровневого мультиплексора.

Крупнозернистые кристаллы содержат более сложные макроэлементы. В комбинационных секциях этих макроэлементов выполняются функции от 4 и более переменных. В последовательностных секциях триггеры с управляемой конфигурацией позволяют с минимальными усилиями строить счетчики, регистры, конечные автоматы. В кристаллах с крупнозернистой архитектурой преобладают элементы задания конфигурации на базе ЗУПВ (например, приборы компаний *XILINX, CONCURENT LOGIC* и семейство *FLEX* компании *ALTERA*). Важнейшая особенность ППВМ - кристаллов на базе СЗУПВ — это возможность внутрисистемного репрограммирования. Такая возможность многократного использования матричного кристалла, очевидно, удобна и выгодна для макетных и экспериментальных приложений. Кроме того, репрограммирование открыло целый ряд совершенно новых приложений, поскольку появилась возможность модификации системных функций путем простой загрузки новой программы конфигурации за 0,1 с. в работающей схеме.

В практике применения ППВМ достаточно часто возникает проблема радиационной стойкости кристалла. Для обеспечения устойчивости к облучению есть два пути: применить архитектуру с перемычками или возобновлять программу конфигурации с необходимой частотой. Оба пути имеют свои издержки, и разница между ними весьма значительна.

Семейство *Cli 600* ППВМ компаний *CONCURRENT LOGIC* содержит схемы сложностью до 10 тыс. вентилей. Компания *LATTICE SEMICONDUCTOR* выпускает семейства *pLSI* и *LSI* сложностью до 8 тыс. вентилей.

Существуют также плотноупакованные ПЛИС, такие как *ATV 5000 (ATMEL)*, серия *MACH (AMD)*, серия *MARL (NATIONAL SEMICONDUCTOR), MAX 5000 и MAX 7000 (ALTERA)*, которые по числу эквивалентных вентилей близки к некоторым ППВМ, по архитектуре предоставляют собой матрицы ПЛИС блоков.

Данный дипломный проект разрабатывается с использованием САПР ПЛИС *Quartus II* фирмы *ALTERA*. Поэтому в качестве элементной базы выбирается микросхема серии *FLEX 8000* фирмы *ALTERA.*

На сегодня - это самая емкая архитектура с высокой регистровой насыщенностью (от 282 до 1500 триггерных схем) и большим числом линий ввода/вывода. Высокая предсказуемость и определенность временных параметров обеспечивается глобальными каналами под названием *Fast Tracks*. Эти каналы позволяют для наилучшего случая получать задержки в соединениях через весь кристалл величиной всего 9 нс. Другие ППВМ - архитектуры делятся на два класса — «мелкозернистые», обеспечивающие повышенный уровень использования вентилей, и «крупнозернистые», предназначенные для получения максимального быстродействия. В матрицах *FLEX* сочетаются оба эти класса В основе каждой матрицы есть логические матрические блоки (*LAB - logic-array block*), которые образуют «крупные зерна». Вместе с тем каждый блок LAB можно разделить на 8 логических элементов (*LE — logic element*), при этом каждый такой элемент («мелкое зерно») имеет в своем составе триггерную и комбинационные схемы. Комбинационная логика состоит из 4-входового генератора любой логической функции с задержкой 5 нс и двух цепей ускоренного

распространения Триггер перестраивается в конфигурации *D-,T- ,YK- , RS-* типа.

Элементы ввода/вывода (ВВ), обслуживающие выходы кристалла, вводят внешние сигналы в схемы *FLEX* или выводят их во внешние цепи. Каждый элемент имеет в составе триггер и логику управления скоростью нарастания выходного напряжения, которая позволяет увеличивать фронты сигналов по сравнению с исходными значениями, чтобы снизить броски напряжения на земляной шине. Элементы ВВ могут быть использованы как входы, выходы или двунаправленные контакты. Триггер может быть использован как защелка для входных или выходных сигналов.

Элементы СЗУПВ, в которые записывается информация о конфигурации ИС, необходимо загружать сразу после включения *FLEX*-кристалла. Соответствующую информацию можно хранить в таких стандартных приборах памяти, как СППЗУ, ЭСППЗУ, или в системных ЗУПВ.

Фирма *ALTERA* предлагает шесть различных схем конфигурации памяти. Эти схемы делятся на два класса - с активным и пассивным режимами конфигурирования. В активном режиме *FLEX* - кристалл сам управляет процессом загрузки данных непосредственно из внешней памяти. В пассивном режиме *FLEX* - крислалл работает как ведомое устройство, а передачей данных управляет ведущая система.

Данные можно передавать по байтам или в виде последовательного битового потока данных. Для данного дипломного проекта выбирается активный режим загрузки с помощью последовательной ПЗУ серии *AT17C65*. Загрузка памяти выполняется менее чем за 100 мс.

Генерацию опорной частоты в активных схемах загрузки в диапазоне 2...6 МГц выполняет сам прибор *FLEX 8000*. Последовательность адресов генерируется до тех пор, пока *FLEX 8000* не завершит загрузку. Тогда

контакт *CONF-DONE* освободится и перейдет в состояние «1» за счет внешнего резистора, индицируя, что загрузка завершена.

Контакт *nCONFIC* может быть соединен с *Vcc* для автоматического вызова загрузки при включении питания. Если в процессе загрузки произошла ошибка, то сигнал *nSTATUS* принимает значение «О».

После выхода из режима загрузки микросхема внешней памяти отключается, а контакты, использовавшиеся как шины адреса и данных, принимают свое назначение как элементы ввода/вывода. В любое время работы схемы контакт *nCONFIG* может быть использован для рестарта загрузки. Семь контактов прибора предназначены только для процедуры загрузки, остальные имеют двойное назначение. Контакты двойного назначения можно не использовать, а перевести в высокоимпедансное состояние соответствующими установками, предлагаемыми в САПР *Quartus II*.

FLEX-архитектура обеспечивает очень высокий уровень надежности в условиях снижения напряжения питания.

Двумя ключевыми особенностями ИС семейства *FLEX* являются высокое быстродействие и малая мощность потребления. Например, 16-бит реверсивный счетчик работает на частоте 71 МГц. При напряжении питания 5В ток питания матрицы составляет менее 1мА в покое и до100 мА в активном режиме.

Приборы *FLEX 8000* имеют большое количество ячеек памяти, незаменимых для таких применений, как цифровые сигнальные сопроцессоры, передача и преобразование многоразрядных данных. Существуют три крупные сферы применения, которым лучше всего соответствуют характеристики ИС *FLEX*: видео и графические подсистемы, аппаратура связи и управляющие подсистемы.

Максимальное использование достижений архитектуры и технологии *FLEX* - кристаллов достигается благодаря удачным алгоритмам, заложенным в САПР *Quartus II* фирмы *ALTERA.*

1.4 Обоснование технической реализации проекта

В данном дипломном проекте разработан контроллер специализированного интерфейса «СТЫК С1-ФЛ» входящий в состав автоматизированной системы передачи данных. Назначение этого устройства – преобразование логических сигналов ПЭВМ, получаемые по локальной шине *PCI* ПЭВМ, в биимпульсный сигнал, пригодный для передачи по каналам связи и осуществление обратного преобразования при приеме данных из канала связи. Данный алгоритм реализуется в устройстве под управлением ПЭВМ посредством стандартной логики локальной шины *PCI*.

Широкое внедрение персональных компьютеров в различные отрасли промышленности обуславливает необходимость быстрого обмена информацией между персональными компьютерами и машинами типа IBM PC, которые используются для хранения больших массивов информации и обладают значительными вычислительными средствами. Данная разработка может использоваться для передачи пакетов информации между периферийными объектами, где работа ведется на персональных компьютерах, и центром, где информация хранится и обрабатывается на ЭВМ.

Основным предназначением устройств данного типа является преобразование логических сигналов ПЭВМ, получаемых по интерфейсу *RS-232C*, в биимпульсный сигнал при передаче данных в канал связи и преобразование биимпульсного сигнала в логические сигналы, адаптированные для ПЭВМ

Разрабатываемое устройство является альтернативным вариантом, подключаемым к интерфейсу *PCI*, что позволяет резко уменьшать

трудоемкость изготовления оборудования, добиваться высокой степени унификации узлов оборудования значительно снижать его стоимость, потребляемую энергию и габаритные размеры. Кроме того, существенно упрощается эксплуатация системы и повышается надежность оборудования.

1.5 Разработка устройства

1.5.1 Выбор и обоснование схемы электрической структурной

Темой данного дипломного проекта является «Контроллер специализированного интерфейса «СТЫК-С1ФЛ». Назначение этого устройства – преобразование логических сигналов ПЭВМ, получаемые по локальной шине *PCI* ПЭВМ, в биимпульсный сигнал, пригодный для передачи по каналам связи и осуществление обратного преобразования при приеме данных из канала связи. Данный алгоритм реализуется в устройстве под управлением ПЭВМ посредством стандартной логики локальной шины *PCI*.

1.5.1.1 Краткое описание шины *PCI*

Сегодня без шины *PCI* уже невозможно представить себе новое поколение не только офисных компьютеров общего назначения (персональные компьютеры, рабочие станции), но и новейшие разработки в области высокопроизводительных одноплатных компьютеров-контроллеров *VMEbus/ VXIbus/ MULTIbusII/ FUTUREbus+* промышленного/военного применения.

Сам факт принятия стандарта локальной шины *PCI* абсолютным большинством крупнейших мировых производителей компьютерной техники означает переход к новому поколению не только компьютеров как таковых, но и к новому поколению высокопроизводительных электронных (процессоры, сетевые и графические контроллеры и т.п.), механических и программных компонентов, к новым подходам в самой идеологии построения новейших компьютерных архитектур, к новым подходам в

области технической и маркетинговой политики. В некотором смысле стандартизация нового локального интерфейса *PCI* означает качественный шаг вперед для всей компьютерной индустрии в целом.

Для того чтобы сделать *PCI* действительно открытой стандартной шиной, была сформирована *PCI*-группа (*PCI Special Interest Group*), насчитывающая более 300 членов. В эту группу входят все крупнейшие мировые производители компьютеров и электронных компонентов. Среди них *Intel, IBM, DELL, HP* и многие другие. Эта группа работала над спецификациями *PCI*, добиваясь того, чтобы шина *PCI* была ещё более привлекательна для использования не только в ПК, в высокоуровневых серверах, но и в широком классе промышленных/военных встраиваемых приложений реального времени. Сегодня уже никому не придет в голову оспаривать необходимость стандартизации современного локального интерфейса. Практически прекращены глобальные диспуты о том, каким он должен быть. Выбор сделан - в пользу *PCI*.

Итак, *PCI* обладает всеми характеристиками открытой стандартной шины В/В:

- высокой пропускной способностью;
- стоимостной эффективностью;
- процессорной и программной независимостью;
- автоконфигурированием;
- масштабируемостью;
- широкой поддержкой независимых конкурирующих производителей.

Будучи определённой как локальная шина, *PCI* обладает процессорной независимостью. Она может работать со скоростью до 264 МБ/с параллельно с подсистемами процессора/памяти. Она может поддерживать требования таких приложений мультимедиа, как анимация и видеоконференции. Со спецификациями на 5В и на 3,3В *PCI* готова к практически

беспрепятственному переходу в промышленные/военные встроенные применения (при соответствующем механическом исполнении).

PCI - это стандартный интерфейс между подсистемой процессор/память и периферийными компонентами. В нем стандартизуются сигналы на контактах, электрические характеристики и протоколы взаимодействия компонентов, подключенных к этому интерфейсу.

Стандартом *PCI* определяется также адресное пространство памяти, адресное пространство В/В и специфичное для *PCI* пространство конфигурации. Любая *PCI*-плата может устанавливаться в любой слот расширения *PCI*. На *PCI*-плате может находиться более одного *PCI*-устройства, которые через шину *PCI* соединяются с *PCI*-мостом. Локальная шина *PCI* представляет собой параллельную мультиплексированную 32-разрядную шину для передачи адреса/данных с возможностью передачи 64-разрядных адресов/данных.

На рисунке 1.1 показана типичная система на базе шины *PCI*. В серверных системах высокого уровня для получения высокой пропускной способности В/В, требуемой главными (*host*) процессорами, используется несколько *PCI*-шин.

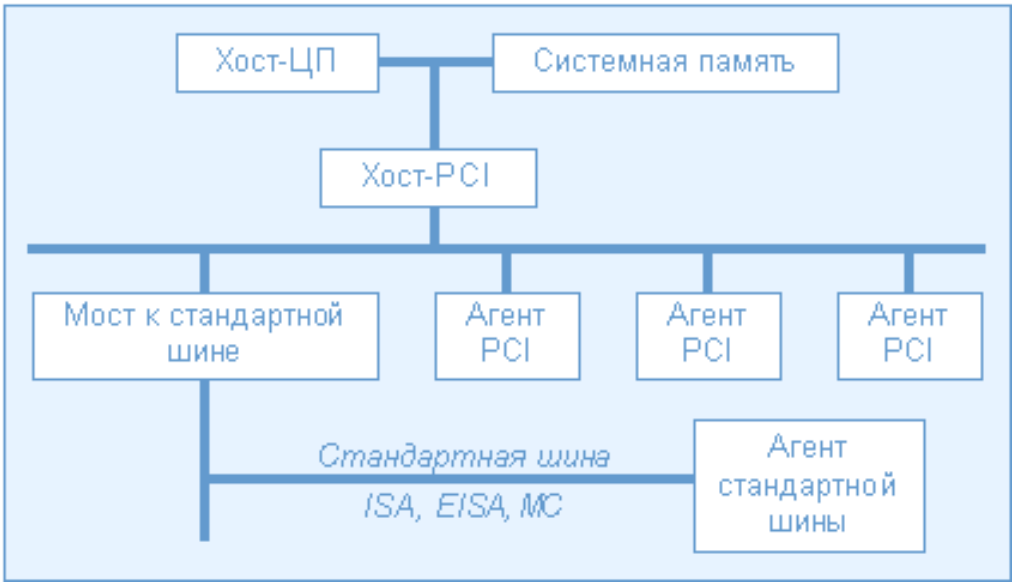


Рисунок 1.1 - Типовая система на базе *PCI*

Для создания систем с иерархией шин *PCI* могут применяться мосты *host-PCI* и *PCI-PCI*, как показано на рисунке 1.2

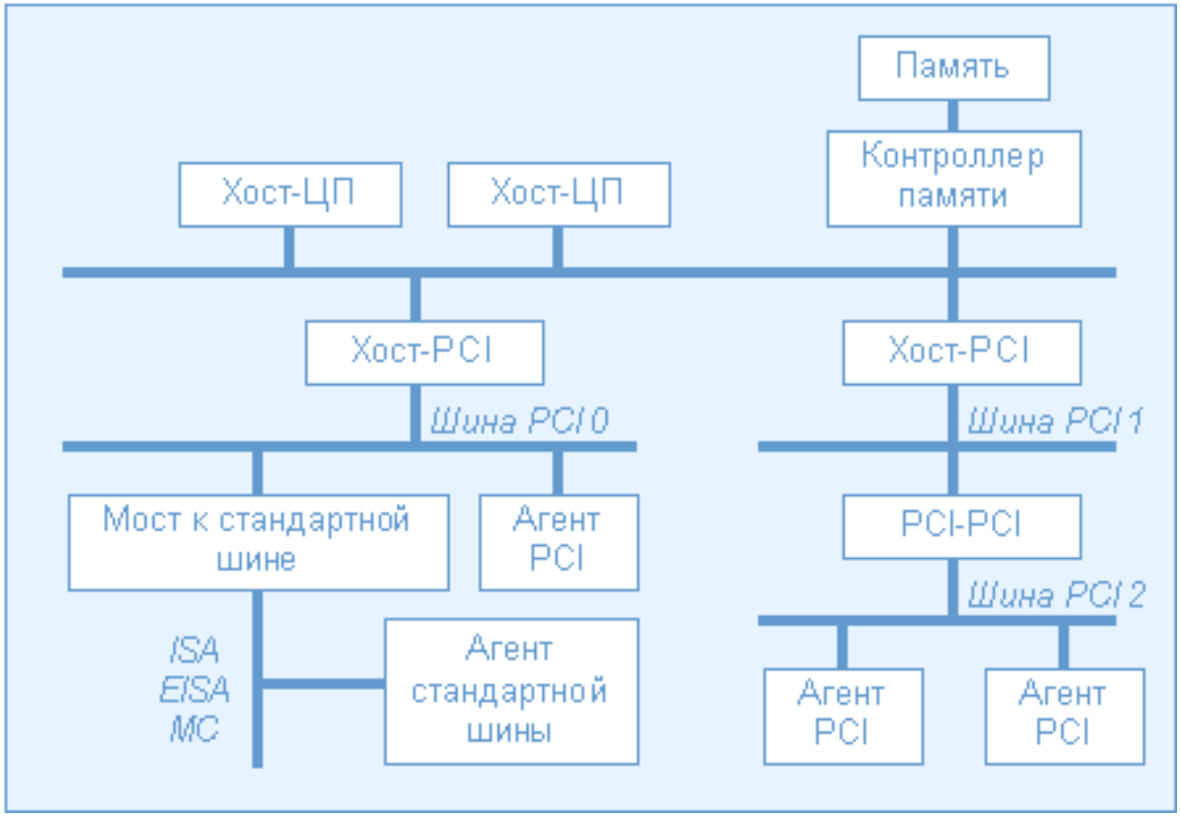


Рисунок 1.2 - Иерархичная структура шин *PCI*

Локальная шина *PCI* поддерживает до 10 нагрузок. Например, у нас может быть три разъёма для вставных плат (6 нагрузок) и 4 непосредственных нагрузки на материнской (системной) плате. Разъём представляет собой соединитель типа *MC (Micro Channel-style*). Стандартом *PCI* для офисных применений определяются также 2 размера вставных плат: стандартной и короткой длины (для жестких промышленных применений механическое исполнение модулей *PCI* иное). Для быстрого и простого перехода на другую технологию определяется два соединителя вставных плат - для сигнальной среды 5В и 3,3В.

1.5.1.2 Характеристики локальной шины *PCI*

PCI - это высокопроизводительная открытая стандартная локальная шина, предназначенная для использования в различных изделиях. Она обладает целым рядом привлекательных характеристик:

Высокая производительность:

- максимальная скорость передачи *PCI* составляет 132 МБ/с (при 32-разрядных передачах) и 264 МБ/с (при 64-разрядных) при такте 33 МГц. В перспективе при реализации такта в 66 Мгц соответственно в 2 раза выше;
- переменная длина пакета передаваемых данных - как при чтении, так и при записи;
- доступ с малым временем ожидания. Время доступа при операциях записи из ведущего устройства к ведомому составляет 60 нс;
- частота синхронизации шины - до 66 МГц;
- наличие схемы скрытого арбитража. То есть фаза арбитража может накладываться на циклы передачи данных.

Низкая стоимость:

- шина оптимизирована для непосредственного взаимодействия компонентов (микросхем процессоров, микросхем периферийных контроллеров ввода/вывода, мостовых микросхем для стандартных шин *VME, ISA* и т.д.) без использования связующих логических схем;
- шина мультиплексирована по адресам/данным, что уменьшает количество контактов и размеры компонентов *PCI*. Для ведомого *PCI*-устройства требуется только 47 сигнальных контактов и 49 - для ведущего *PCI*-устройства.

Перспективность:

- как локальная шина, *PCI* может использоваться со многими семействами процессоров, в том числе и будущих поколений. *PCI* особенно эффективна в качестве высокопроизводительной локальной шины;

- тщательно разработана для применения со многими поколениями процессоров с 64-разрядной адресацией;

- тщательно продуманы и определены две сигнальные среды - для поддержки стандартных промышленных сигналов 5В и для перехода в будущем на промышленные сигналы 3,3В.

Автоконфигурирование: стандарт *PCI* требует, чтобы *PCI*-устройства имели некоторый набор регистров конфигурации, в которых содержится информация об устройстве и которые поддерживают полное автоконфигурирование. Это позволяет пользователям, не заботясь о *DIP*-переключателях, работать с устройством сразу после его подключения (принцип *"plug and play"* - "подключи и работай").

Масштабируемость:

- совместимость в одной системе 32- и 64-разрядных компонентов;

- источники питания могут оптимизироваться по двум присутствующим на *PCI*-разъёме сигналам. Назначения этих двух сигналов:

- наличие вставной платы;

- диапазон потребляемой энергии;

- малые размеры вставных плат и применение соединителя типа MC (в офисных приложениях);

- рабочий протокол встроен в *PCI*-компоненты, поэтому никакой согласующей логики и буферов не требуется.

Программная совместимость: компоненты шины *PCI* могут быть полностью совместимы с существующими драйверами и прикладными программами. Драйверы устройств могут переноситься на разные классы платформ. Это одно из самых ключевых положений идеологии и стандартизации новой локальной шины. *PCI* обеспечивает одну из ключевых основ для переносимости программных продуктов с одной процессорной архитектуры на другую. Это касается в том числе и операционных систем реального времени, которые в целях оптимизации, как правило, всегда заново

портируются на конкретный тип промышленного вычислителя. В мире операционных систем реального времени стандарт *PCI* создает предпосылки для значительного сокращения инвестиций в перенос программного обеспечения, как это происходит в мире офисных приложений в рамках той же *MS Windows*.

Целостность данных: контроль целостности данных по чётности как для адресов, так и для данных.

Кроме вышеуказанных характеристик, *PCI* поддерживает:

- кэшируемую память;
- исключительный доступ;
- специальные *PCI*-циклы;
- *parking*-шины;
- быстрые передачи *Back to Back*.

Основным механизмом передачи данных на шине *PCI* является блочный механизм. Блок состоит из фазы адреса и одной или более фаз данных. *PCI* поддерживает передачу блоками как для адресного пространства памяти, так и для адресного пространства ввода - вывода. Основной интерфейс (между главным процессором и шиной *PCI*) может объединять запросы по записи в память в единую транзакцию, при условии, что не будет никаких побочных эффектов. В этом случае устройство устанавливает бит предварительной выборки в регистре базового адреса (чтобы разрешить чтение данных и объединение данных для записи в любом порядке). Интерфейс может различать, когда такое объединение разрешено, а когда - нет, при помощи анализа адресного диапазона, данные по которому должно предоставлять программное обеспечение при инициализации. Прекращение объединения данных в буфер (и сброс содержимого буфера) должно происходить во время следующей записи или чтения, которые заранее не предусмотрены (для любого диапазона). Транзакции записи после любого из этих двух событий могут быть объединены с последующими

транзакциями, но это не относится к ранее объединенным данным, если они попали в «предсказанный» диапазон.

Основной интерфейс может комбинировать последовательности двойных слов, записываемых в память данных (*DWORD*), которые генерируются процессором в блоки, при условии сохранения порядка следования адресов (ассоциированного с каждым двойным словом). Например, интерфейс может записывать следующую последовательность - *DWORD 0, DWORD 2 и DWORD 3*. Тогда последовательность на *PCI* может быть следующей: *DWORD 0, DWORD 1* (байты не допускаются), *DWORD 2* и в конце блока - *DWORD 3*. Такое объединение допускается во всех случаях, когда последующий адрес *DWORD* более значим, чем предыдущий. Интерфейс может преобразовывать одиночные запросы по чтению памяти, поступающие от процессора, в пакетный запрос (чтение производится с выборкой, до прихода данных в процессор), при условии, что при этом чтении не будут возникать побочные эффекты в адресуемом целевом устройстве.

Если запросы от процессора для ввода-вывода нельзя объединить, то они будут иметь единственную фазу - фазу данных. В настоящее время нет таких процессоров или «управителей» шины, которые бы генерировали пакетные запросы в пространстве ввода - вывода. Однако, если в будущем некоторое новое устройство будет генерировать значимые пакетные запросы на ввод-вывод, (например, для доступа к порту *FIFO*), то они не будут игнорироваться. Для пакетных запросов по вводу-выводу адресация не подразумевается. В этом случае адресация должна быть установлена целевым устройством и мастером после выполнения запросов по вводу-выводу. *PCI* - устройства, которые не работают с фазами данных при множественном вводе - выводе, должны разрывать связь после первой фазы данных. Чтобы убедиться, что устройства ввода - вывода функционируют правильно, интерфейсы не должны объединять или комбинировать в

одиночный запрос доступа к PCI или блок последовательности запросов ввода - вывода. Все запросы на ввод - вывод должны появляться на шине PCI точно так, как их сгенерировал процессор. (Если адресуется целевое устройство, для которого *Byte Enables* показывает, что объем передаваемых данных больше, чем оно поддерживает, то это устройство аварийно завершает свою работу).

Значение всех сигналов проверяется по фронту синхроимпульса Каждый сигнал имеет свою установочную апертуру и апертуру неактивного состояния относительно положительного фронта синхроимпульса, при которых не позволяется переход в другое состояние. Величины сигналов выше этой апертуры не имеют никакого значения. Эта апертура имеет значение только на «подходящих» фронтах синхроимпульсов для сигналов *AD[31::00], [63::32], PAR, PAR64* и *IDSEL* и на любом фронте синхроимпульса для *LOCK#, IRDY#, TRDY#, FRAME#, DEVSEL#, STOP#, REQ#, GNT#, REQ64#, ACK#64, SBO#, SDONE, SERR#* (только на срезе), а для сигналов *PERR#, C/BE[3::0]#, C/BE[7::4]#* (так как это команды шины) существует ограничение по положительному фронту синхроимпульса, при условии активности сигнала *FRAME#*. Сигналы *C/BE[3::0], C/BE[7::4]#* (с разрешением байта) ограничены на любом положительном фронте синхроимпульса, после завершения фазы адреса или фазы данных. *RST*, IRQA*, IRQB*, IRQC*, и IRQD** не ограничены и асинхронны.

Всего определено три физических адресных пространства. Пространства адресов памяти и ввода-вывода объединены. Адресное пространство конфигураций было введено для обеспечения аппаратной конфигурации *PCI*.

Дешифрирование адреса на шине *PCI* распределено; это означает, что оно выполняется на каждом устройстве. Это устраняет проблемы для центральной дешифрирующей логики, а также для сигналов выбора устройства, независимо от их использования для конфигурации. Каждый

агент отвечает только на свой дешифрированный адрес. *PCI* поддерживает два режима дешифрирования адреса: суммирующий и вычитающий. Суммирующее дешифрирование быстрее, так как устройство ищет запрос в своем адресном интервале. Вычитающее дешифрирование может выполняться только одним устройством на шине, так как это подразумевает запрос, который не может быть положительно дешифрирован каким - то другим агентом. Данный механизм дешифрирования медленнее, так как он должен давать всем остальным агентам на шине «первое право по отказу» при запросе.

Тем не менее, это может оказаться полезным для такого агента, как стандартная шина расширения, которая должна работать в высоко-фрагментированном адресном пространстве. Целевые устройства, которые выполняют как положительное, так и отрицательное дешифрирование, не должны реагировать на зарезервированные команды шины (путем установки сигнала *DEVSEL#*).

Информация, содержащаяся в двух младших битах адреса (*AD[1::0]*), определяется адресным пространством. Чтобы обеспечить полный адрес байта, в адресном пространстве ввода - вывода используются все 32 линии. Это позволяет агенту, требующему разрешение адресации на уровне байта, завершить дешифрирование адреса и начать свой цикл без дополнительного цикла ожидания разрешения адресации байта (таким образом осуществляется задержка всех циклов вычитающего дешифрирования дополнительным тактом). *AD[1::0]* используются только для генерирования сигнала *DEVSEL#*, указывающего при передаче последний допустимый байт. Например, если BE0# были активны, то *AD[1::0]* будет ″00″; если были активны только *BE3#*, то *AD[1::0]* будет ″11″. Как только целевое устройство установило запрос на ввод - вывод (используя *AD[1::0]*), оно может определить, можно ли завершить запрос, как это указано разрешением байта. Если все выбранные байты оказались вне выбранного целевым устройством адресного интервала,

то данный запрос не может быть завершен. В этом случае целевое устройство не может передать никакие данные, но прекращает свою работу с аварийным завершением. Таблица ниже подводит итог по дешифрирования *AD[1::0]*.

Всем целевым устройствам требуется проверять линии *AD[1::0]* в течение транзакции команды при обращении к памяти, а также обеспечивать требуемую последовательность передаваемого блока, либо прерывать связь с целевым устройством в первую фазу данных или по ее завершении. Реализация линейного порядка передаваемого блока требуется всеми устройствами, которые его не могут обеспечить. Выполнение переключения строки кэша при этом не требуется. В адресном пространстве памяти запросы дешифрируются в адрес *DWORD* с использованием линий *AD[31::02]*. В линейном режиме приращения инкрементирование адреса осуществляется путем инкрементирования двойного слова *DWORD* (четыре байта) после каждой фазы данных, до тех пор, пока не завершится транзакция.

Запросы для адресных пространств конфигураций дешифрируются в адрес *DWORD*, который используется на *AD[7::2]*. Когда команда дешифрирована, агент определяет целевое устройство запроса (активен сигнал *DEVSEL#*), активен *IDSEL* и на *AD[1::0]* присутствует "00". В противном случае агент игнорирует текущую транзакцию. Интерфейс определяет, что запрос конфигурации предназначен для устройства, находящегося перед ним, дешифрируя команду конфигурации и номер интерфейса, на линиях *AD[1::0]* - "01".

Для того, чтобы уменьшить время ожидания запроса, при арбитраже *PCI* используется такой подход: запрос должен завершаться за определенное время. Это означает, что мастер шины должен осуществлять арбитраж для каждого запроса, который выполняется на шине. *PCI* использует центральную схему арбитража, где каждый агент мастера имеет уникальный сигнал запроса (*REQ#*) и сигнал разрешения (*GNT#*). Для получения доступа к шине используется простой прием «рукопожатие», типа запрос - ответ.

Арбитраж на шине - "скрытый", это означает, что он осуществляется в течение предыдущего запроса таким образом, чтобы для арбитража не использовались никакие циклы шины *PCI*, за исключением случая, когда шина находится в состоянии ожидания.

Указанный алгоритм арбитража должен выполняться центральным арбитром, например, при вращении приоритета, точно и т.д. При определении алгоритма арбитража следует установить значение для гарантирования заданного времени ожидания в наихудшем случае. Тем не менее, так как алгоритм арбитража не является частью спецификации шины, то проектировщики системы могут по выбору модифицировать его, но при этом они должны обеспечить соблюдение требований для времени ожидания выбранных ими контроллеров ввода - вывода, а также для плат расширения. За информацией относительно значений для времени задержки обращайтесь к подпункту 1.5.4.3. Шина позволяет агенту осуществлять back-to-back транзакции, а также позволяет арбитру определенную гибкость в распределении запросов по приоритетам и весам. Арбитр может реализовывать схему любой сложности, при этом в любом такте должен быть активен только один сигнал *GNT#*.

Агент запрашивает шину путем установки сигнала *REQ#*. Агенты могут использовать сигнал *REQ#*, чтобы указать, что им действительно требуется использовать шину. Агент никогда не должен использовать сигнал *REQ#*, чтобы "закрепить" себя на шине. Если же закрепление выполнено, то арбитр должен определить мастера по умолчанию. Когда арбитр решает, что агент может использовать шину, то он устанавливает в активное состояние сигнал *GNT#* агента.

Арбитр может установить в неактивное состояние сигнал *GNT#* агента в любом такте. Если агент желает начать транзакцию, он должен убедиться, что *GNT#* активен по положительному фронту синхроимпульса. Если *GNT#* неактивный, то транзакция не должна продолжаться. Если *GNT#* был

установлен в активное состояние, то он может перейти в неактивное состояние в соответствии со следующими правилами.

Если *GNT#* - неактивный, и активен *FRAME#*, то транзакция шины допустима и продолжается.

Один сигнал *GNT#* может быть установлен в неактивное состояние, когда соответствующий сигнал *GNT#* активен, причем шина не должна находиться в состоянии ожидания. В противном случае, потребуется один такт задержки в период, когда *GNT#* устанавливается в неактивное состояние, а следующий сигнал *GNT#* переходит в активное состояние, либо если будет конкуренция на адресных линиях *AD* и линии *PAR*.

Пока сигнал *FRAME#* - неактивный, *GNT#* может быть установлен в неактивное состояние в любой момент времени в порядке обслуживания высокоприоритетным мастером, или при отклике на соответствующий сигнал *REQ#*, который перешел в неактивное состояние.

Используются два агента, чтобы показать, как арбитр может чередовать запросы шины. Основной арбитраж представлен на рисунке 1.3

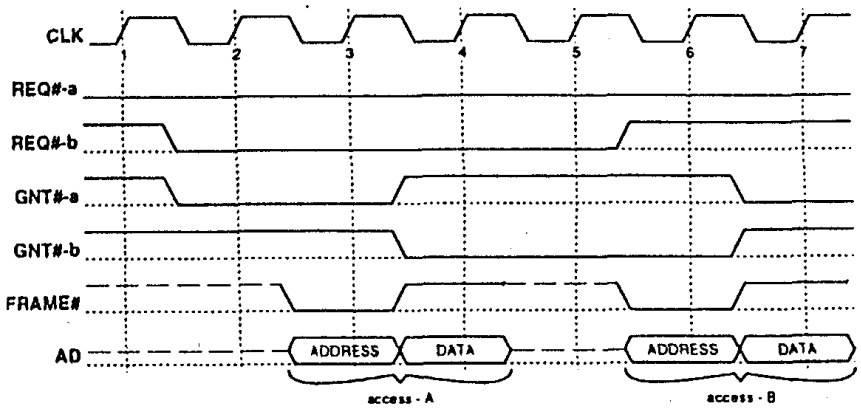


Рисунок 1.3 - Основной арбитраж

REQ# активен до такта 1 или в течение его, если поступил запрос на использование интерфейса. Агент А получает разрешение к шине, так как сигнал *GNT#*-а устанавливается в активное состояние в такте 2. Агент А может начать транзакцию в такте 2, так как сигналы *FRAME#* и *IRDY#* были

установлены в неактивное состояние, а сигнал *GNT#* установлен в активное состояние. Транзакция агента А начинается, когда *FRAME#* устанавливается в активное состояние в такте 3. Так как агент А решает выполнить другую транзакцию, то он оставляет сигнал запроса *REQ#-a* активным.

Когда сигнал *FRAME#* активен в такте 3, арбитр определяет, что далее следует агент В, и поэтому устанавливает в такте 4 сигнал *GNT#-b* - в активное состояние, а сигнал *GNT#-a* - в неактивное.

Шина освобождается, когда агент А завершает свою транзакцию в такте 4. Все агенты *PCI* могут определить конец текущей транзакции по неактивным сигналам *FRAME#* и *IRDY#*. Агент В захватывает шину в такте 5 (так как *FRAME#* и *IRDY#* - неактивные) и завершает свою транзакцию в такте 7.

Обратите внимание, что неактивный *REQ#-b* и активный *FRAME#* в такте 6 показывают, что агенту В требуется единственная транзакция. Арбитр разрешает следующую транзакцию агенту А, так как *REQ#* все еще активен.

Текущий мастер шины сохраняет *REQ#* активным, когда ему требуются дополнительные транзакции. Если больше нет никаких других активных запросы, или текущий мастер имеет самый высокий приоритет, арбитр продолжает предоставлять шину этому текущему мастеру.

Сигнал *GNT#* необходим агенту для запроса на шине одиночной транзакции. Если агент желает сделать другой запрос, он должен сохранять активным сигнал *REQ#*. Агент может установить сигнал *REQ#* в неактивное состояние в любой момент времени, но арбитр может интерпретировать это, как ситуацию, в которой агенту больше не требуется использования шины, и установить в неактивное состояние сигнал *GNT#*. Агент должен установить *REQ#* в неактивное состояние в том же самом такте, когда активен *FRAME#*, при условии, что агенту требуется выполнить одиночную транзакцию. Когда транзакция завершена целевым устройством (активен *STOP#*), мастер должен

установить в неактивное состояние *REQ#* менее, чем за два такта *PCI*, первый из них - когда шина переходит в состояние ожидания (в конце транзакции, когда становится активным сигнал *STOP#*), и еще один - на такт раньше или позже, после состояния ожидания. Если мастер предполагает завершать транзакцию, то должен еще раз установить сигнал *REQ#* после того, как *REQ#* перейдет в неактивное состояние, или может произойти потенциальное состояние "зависания". Если мастер не предполагает завершение (так как была предвыборка, или должен быть обслужен более приоритетный внутренний запрос), агент должен устанавливать *REQ#* всякий раз, когда ему может потребоваться использование интерфейса. Это позволяет другому агенту использовать интерфейс, пока предыдущее целевое устройство готовится к следующему запросу. Арбитр может принять решение, что текущий мастер «разрушен», если тот не начал запрос после того, как был установлен в активное состояние сигнал *GNT#* (*REQ#* также активен), и шина находилась в ожидании 16 тактов *PCI*. Тем не менее, арбитр может убрать сигнал *GNT#* в любое время, для обслуживания более приоритетного агента.

Необходимо согласовывать устройства, чувствительные как производительности, так и к задержке во времени. Например, во время связанной с ЦП модификации изображения на экране, трафик при передаче буфера изображения может занимать большую часть от ширины диапазона передаваемых по шине данных. Иногда возможны необычайно длинные запросы, если типичная задержка очень мала. Такие особенности *PCI*, как пересылка блоков данных с неограниченной длиной (широкий диапазон блочного ввода-вывода; *FDDI*) и закрепление шины за агентом (высокая пропускная способность; например, при передаче буфера изображения) отвечают требованиям устройств с высокой производительностью. Чтобы сохранить экономичные характеристики таких устройств (требования по оптимизации буферов), такие особенности *PCI*, как основанный на запросах

арбитраж и счетчики задержек мастер - устройства (*LT*), позволяют ограничить наихудший случай времени ожидания. Например, счетчик задержки может вынудить устройство с большим буфером *FDDI*, обычно реализующее за один запрос блок длиной в 128 двойных слов (*DWORD*), разбить передачу на отдельные малые части, осуществляя таким образом поверхностную буферизацию. Устройства, чувствительные ко времени ожидания, обращаются к шине наиболее часто. Высокая производительность и малое время ожидания часто дополняют друг друга. Соответственно, очень важно, чтобы системы с *PCI* компонентами разрабатывались очень тщательно, чтобы минимизировать риск, связанный с временем ожидания при взаимодействии систем, и облегчать приемлемые соотношения цена/производительность.

У большинства *PCI* - систем типичное время задержки очень мало (около 2 мкс), и оно легко определяется. Тем не менее, в наихудшем случае (который, в свою очередь, достаточно редок), время задержки не может быть слишком большим, но такие ситуации очень трудно предсказать. Например, задержка для стандартной платы расширения (*ISA/EISA/MC*) относительно интерфейса часто зависит от самого адаптера, а не от шины *PCI* (это особенно проблематично, так как некоторые существующие адаптеры не обеспечивают временные параметры задержки, определенные стандартом шины). Для того, чтобы это компенсировать, мастер - устройства, для которых надо гарантировать наихудший случай задержки, должны обеспечить адекватную буферизацию за 30 мкс. Это подразумевает буферизацию минимум около 50 байтов для *LAN* - адаптеров со скоростью 10 Мбит/сек, и около 500 байтов - для *LAN* - адаптера со скоростью 100 Мбит/сек (если буферы организованы в виде строк (например, с 16- или с 32-битным выравниванием) для наилучшего использования *PCI* и памяти целевого устройства, то минимальный размер буфера желательно увеличить). Несмотря на некоторую неопределенность в наихудшем случае, для

реализуемых системных разработок 30 микросекунд должны обеспечить достаточно широкое поле деятельности. Даже при самых жестких требованиях, 30 микросекунд должно хватить в наихудшем случае. Для снижения стоимости проектирования в наихудшем случае мастер - устройство может уменьшить задержку для наихудшего случая, если допускаются случайные нарушения величины задержки без создания ошибочных условий. Например, *LAN* - адаптер должен иметь соответствующую инфраструктуру (аппаратное и программное обеспечение), чтобы можно было определить, когда происходят ошибки при буферизации, и инициировать повторную передачу блока данных (как в случае, когда блоки искажаются из-за наводок в линиях или коллизий). Секторы на диске можно повторно считать или записать. Передача в реальном времени потока данных (например, аудио- данных) может допускать случайные типовые потери, используя при этом предыдущую выборку данных. Устройство, достаточно устойчивое к сбоям и нарушениям задержки, изменяет значение задержки из обычного состояния в более «благоприятное» для соотношения цена/производительность (то есть, в случае обработки звука мы получим шумы). Например, дешевый *LAN* - адаптер для клиентских станций (с небольшой загрузкой шины *PCI*) проектируют с задержкой 10 мкс в наихудшем случае и принимают, что частота повторения пакетов слишком мала, чтобы повлиять на снижение эффективности. Устройства, разрабатываемые для серверов, должны иметь большие буферы (при высокой стоимости) для уменьшения повторений в случае больших нагрузок (если принимать во внимание необходимость разработки очень устойчивых приложений, то устройства, спроектированные для задержки 30 мкс и более, должны самовосстанавливаться во всех случаях нарушения задержки). Некоторые приложения (например, встраиваемые контроллеры, мультимедиа и т. д.) могут принимать на себя более устойчивое управление задержками, чем то, которое обеспечивается обычной (общего назначения) главной шиной

PCI (такие условия могут быть реализованы на вторичной шине *PCI* и подсоединены к *host* - машине через интерфейс *PCI - PCI*). Этому типу приложений требуется, чтобы у целевых устройств было малое и предсказуемое время задержки. Для обеспечения такого проектирования, и, в общем случае, для поощрения хороших разработок, при разработке интерфейса целевых устройств рекомендуется пользоваться принципами, перечисленными ниже. Далее, термином «одноуровневый» (single-layer) обозначается целевое *PCI* - устройство, которое обеспечивает немедленный доступ к выбранному ресурсу. В качестве примера могут служить одно-портовая динамическая память (*DRAM*) и буфер изображения видеопамяти (*VRAM*). «Многоуровневым» называется целевое устройство, осуществляющее арбитраж ресурса, выбранного другим ресурсом, для которого арбитраж не осуществляется. В качестве примера можно указать интерфейсы «*PCI* - шина расширения» (например, *PCI - ISA/EISA/MC; PCI - PCI; PCI* - основная память и т. д.), а также двух-портовую *DRAM*.

Одноуровневое целевое устройство должно отсрочить первую фазу данных на 16 тактов. В противном случае, если временное внутреннее состояние (например, регенерация *DRAM*, полная очередь отложенных запросов на запись и т. д.) увеличивает задержку запроса более, чем на 16 тактов, то немедленно осуществляется повтор.

Многоуровневое целевое устройство должно повторять запрос, который находится в коллизии с занятым ресурсом. Например, если осуществляется запрос, подчиненный *EISA* - шине, которая захвачена другим мастером, то следует немедленно повторить этот запрос (в любом случае, был бы необходим цикл ожидания для исключения тупиковой ситуации, поэтому ожидание и происходит). Аналогично, при запросе регенерируемого буфера изображения *DRAM* следует повторить данный запрос.

«Многоуровневые» целевые устройства (особенно, шинные интерфейсы) должны соблюдать большую осторожность в стратегии записи

буфера. Буферы записи затрудняют определение задержки, так как обычно предъявляемые строгие требования гласят, что перед разрешением какого-то запроса, должна быть обслужена вся очередь записей. Рекомендуется, в частности, чтобы основной интерфейс ЦП - *PCI* мог откладывать записи в резидентный *PCI* - буфер, при этом подавляя отложенные записи на подчиненную шину *ISA/EISA/MC*.

Обычно, чтобы облегчить задачу создания систем, продавцы компонентов должны прилагать описание «поведения» для времени ожидания (обычно запрещено давать описание поведения адаптеров стандартных шин в реальном времени). Для мастера - устройства необходимо указывать как время ожидания, так и последствия его нарушения. Целевые устройства должны определить наихудший случай ответа, а также все события, которые могут вызывать повторение или разъединение. Если время ожидания адресата обуславливается внешними факторами, то это должно быть ясно установлено.

PCI обеспечивает исключительный механизм доступа, который разрешает завершить неисключительный доступ до начала исключительного доступа. Это называется блокировкой ресурса. Это позволяет будущим процессорам задержать аппаратную блокировку параллельно нескольким неисключительным доступам, передачу анимационных данных в реальном масштабе времени, например, видео. Механизм работает только при блокировке *PCI*- ресурса, для которого оригиналу использовался блокированный доступ. Этот механизм полностью совместим с существующим программным использованием исключения.

LOCK# требуется на любом устройстве, обеспечивающем системную память. Определенно, если устройство является выполнимой памятью, оно должно также выполнить *LOCK#* и гарантировать полное исключение доступа в память (то есть, если имеется хозяин, использующий эту память,

он должен также выполнить блокировку). Ведущие интерфейсы, которые имеют системную память ниже них, должны также выполнить *LOCK#*.

Сигнал *LOCK#* указывает, что исключительный доступ начал работать. Утверждение *GNT#* не гарантируют управление сигналом *LOCK#*. Управление *LOCK#* получено под собственный протокол вместе с *GNT#*. При использовании блокировки ресурса, агенты выполняющие неисключительный доступ могут продолжать свою работу, даже в то время, когда другой мастер использует сигнал *LOCK#*. Однако, из требований совместимости, арбитр может преобразовывать блокировку ресурса в блокировку "шины", предоставляя агенту, который выставил *LOCK#*, исключительный доступ к шине до выставления *LOCK#*.

Блокировка ресурса, гарантируется адресатом доступа, не исключая всех других агентов, обратившихся к шине. Степень детализации блокировки должна быть 16 выровненных байтов. Исключительный доступ к любому байту в блоке 16 байтов будет весь 16 байтовый блок. Мастер не может использовать любые адреса вне 16 байт, чтобы быть блокирован. Адресат может блокировать минимум 16 (выровненных) байтов и максимумом весь ресурс. Исходя из этого, следующие параграфы описывают поведение хозяина и адресата.

Правила *LOCK#* будут установлены и для хозяина и адресата. После правил, следует описание того, как запустить, продолжать и завершать исключительную операцию, и будет их обсуждение. Затем обсуждение того, как адресат ведет себя, когда он поддерживает блокировку ресурса и кэшированную память с обратной записью. В заключении раздела обсуждается, как выполнить полную блокировку шины на *PCI*.

Адресат, который поддерживает *LOCK#* на *PCI*, должен твердо придержаться следующих правил:

- адресат доступа блокирует себя самостоятельно, когда *LOCK#* является деактивированным в течение адресной фазы.

- если только блокировка установлена, адресат остается блокированным, пока не произведена выборка *FRAME#* и *LOCK#* одновременно или поступило сообщение об аварийном прекращении работы.

- исключительно гарантировать владельцу сигнала *LOCK#* (если только блокировка установлена) минимум l6 байтов (выровненных) ресурса. Это включает доступ, которого нет на *PCI* для многопортовых устройств.

Все *PCI* адресаты, которые поддерживают монопольные доступы, должны произвести выборку *LOCK#* с адресом. Если адресат доступа выполняет декодировку средне или медленно, нужно удерживать сигнал *LOCK#* в течение фазы адреса, чтобы определить, блокирован ли доступ, пока завершается декодирование. Адресат маркера транзакции блокирует сам себя, если *LOCK#* выключен в течение фазы адреса. Если адресат ожидает сигнала *LOCK#*, пока активирован *DEVSEL#*, он не может различить, блокирован ли текущий доступ или тот, который происходит одновременно с блокированным доступом. Исполнитель может сохранять "состояние", чтобы определить, блокирован ли доступ, но это требует защелки *LOCK#* на последовательных тактах и сравнения для определения блокирован ли доступ. Более простой способ для адресата - пометить себя блокированным при любом доступе. Это требуется там, где *LOCK#* является деактивированным в течение фазы адреса. Блокированный адресат остается в блокированном состоянии, пока *FRAME#* и *LOCK#* являются деактивированными. Для разрешения другого доступа на многопортовое устройство, адресат может производить выборку *LOCK#* на такте, следующим после фазы адреса, чтобы определить, блокировано ли устройство действительно. Когда *LOCK#* является выключенным в течение фазы адреса и включенным (такт после фазы адреса), многопортовое устройство блокировано и должно гарантировать доступ только *PCI* мастеру. Когда *LOCK#* является выключенным в течение фазы адреса и такта после фазы адреса, адресат свободен, чтобы ответить на другие запросы и не

блокирован. Блокированный адресат может только принимать запросы, когда *LOCK#* выключен в течение фазы адреса. Блокированный адресат ответит, выставив *STOP#* с выключенным *TRDY#* всем транзакциям, когда *LOCK#* включена в течение фазы адреса.

Итак, адресат доступа самостоятельно блокирует себя при любом доступе, это требуется, когда *LOCK#* выключен в течение фазы адреса. Адресат отпирает себя, когда *FRAME#* и *LOCK#* являются оба выключенными. Это небольшая путаница для адресата, который блокирует себя на транзакцию, которая не блокирована. Однако с точки зрения реализации, это простой механизм, который использует комбинаторную логику и всегда работает. Устройство отопрет себя непосредственно в конце транзакции, когда *FRAME#* и *LOCK#* выключены. Адресат может также помнить состояние (которое является пригодным для многопортового устройства), чтобы определить, правильно блокировано ли он или нет. Адресат правильно блокирован, когда *LOCK#* выключен в течение фазы адреса и включен на следующем такте.)

Обратите внимание: арбитр должен быть в алгоритме "равнодоступности", когда *LOCK#* включен; иначе может происходить *Livelock* (длительная блокировка).

Существующее программное обеспечение, которое не поддерживает правила использования блокировки *PCI*, неправильно работать. Для *PCI* резидентной памяти (прежде всего системной памяти), которая поддерживает *LOCK#,* требуется быть обратно совместимой к существующему программному обеспечению, рекомендуется выполнить полную блокировку ресурса.

Мастер, который использует *LOCK#* на *PCI*, должен твердо придержаться следующих правил:

Мастер может обращаться только к одиночному ресурсу в течение операции блокировки.

Блокировка не может колебаться между границей устройства.

Шестнадцать байтов (выровненные) - максимальный размер ресурса, который доступен только мастеру в течение операции блокировки. Доступ к любой части из этих 16 байтов блокирует все 16 байтов.

Первая транзакция операции блокировки должна быть транзакцией чтения.

LOCK# должен быть выставлен на такте, следующем после фазы адреса и оставаться таким до поддержки управления.

LOCK# должен быть снят, если повторение сообщается прежде, чем фаза данных завершилась, и блокировка не была установлена.

LOCK# должен быть всегда выключен, когда доступ завершен аварийным прекращением работы как со стороны ведомого, так и со стороны ведущего.

LOCK# должен быть выключен минимум для одного цикла ожидания (*IDLE*) между последовательными операциями блокировки.

В соответствии с вышеперечисленным материалом и поставленным техническим заданием, проектируемое устройство должно содержать следующие блоки:

- блок приемника;
- блок декодера биимпульсного сигнала;
- блок буферный (*FIFO*);
- блок буферной памяти приемника;
- блок прерываний;
- блок хранения загрузочной конфигурации;
- блок интерфейсной логики шины *PCI*;
- блок регистров управления и состояния;
- блок буферной памяти передатчика;
- блок генератора битовой частоты;
- делителя частоты;

- блок кодера биимпульсного сигнала;

- блок передатчика.

Так как проектируемое устройство является относительно медленным для шины PCI, является целесообразным разрабатывать блок интерфейсной логики с ориентацией на работу как Target (целевое) устройство.

1.5.2 Описание работы по схеме электрической структурной

Выбранная структурная схема проектируемого контроллера продемонстрирована в приложении Б. В структурную схему входят 13 блоков. Для того чтобы более точно понять принцип функционирования системы, необходимо рассмотреть каждый блок в отдельности, описать его назначение и взаимодействие с другими блоками устройства.

Блок линейного приемника и блок линейного передатчика служат для согласования контроллера с каналом связи: обеспечивает гальваническую развязку, детектирование и преобразование уровня сигнала линии передачи данных в сигнал, пригодный для дальнейшего цифрового преобразования. После вышеуказанных операций дискретный сигнал из КС преобразуется в сигнал уровней микросхем цифровой логики и передается в следующий блок.

Блок декодера биимпульсного сигнала служит для выделения из полученной закодированной последовательности импульсов полезной информации и преобразования ее из последовательного цифрового кода в параллельный восьмиразрядный код. Восьмиразрядный потому, что вся информация, передаваемая по каналу связи, кодируются в пакеты длиной 8 бит, а параллельный код более удобен для последующей обработки в контроллере. После этого полученные восьмиразрядные слова передаются в следующий блок.

Блок буферный *FIFO* (*“First In – First Out”* – первым вошел – первым вышел) служит для согласования относительно медленной части схемы (сопряжение с КС) с относительно быстрой (интерфейс с шиной *PCI*). Кроме того, этот блок отслеживает получение условной последовательности,

означающей отсутствие полезной информации в передаваемых данных. О ней более подробно будет рассказано далее по тексту. Блок буферный *FIFO* – это стековая память прямого магазинного типа. Как только она заполнится, она подает сигнал переполнения следующему блоку.

Блок буферной памяти приемника – это уже часть высокопроизводительного интерфейса шины *PCI*. Его назначение состоит в следующем: он по сигналу переполнения стека полученному из предыдущего блока в течении нескольких тактов шины считывает в себя содержимое стека и по окончании этой операции подает внутренний сигнал для активизации системы выдачи запроса на прерывание в блок прерываний. Блок хранит информацию до тех пор, пока подсистема интерфейсной логики не запросит ее. Это должно произойти до следующего переполнения блока *FIFO*, иначе данные, находящиеся в буферной памяти, будут затерты новой информацией и потеряются. После того, как блок интерфейсной логики запросит информацию, хранящуюся в нем, он должен незамедлительно, в течение одного цикла шины *PCI* передать всю полученную информацию драйверу. То есть этот блок служит для реализации пакетной (burst) передачи по шине, которая позволяет повысить производительность системы в целом, снижая при этом нагрузку на центральный процессор и уменьшая количество прерываний, вырабатываемых устройством.

Блок формирования прерывания служит для формирования запроса прерывания на шину PCI по внутренним сигналам контроллера, а также модификации регистров состояния устройства, то есть для индикации причины возникновения прерывания. Этот блок тесно взаимодействует с блоком интерфейсной логики, блоком регистров, а также блоками буферной памяти приемника и передатчика.

Блок хранения загрузочной конфигурации необходим для хранения информации о начально необходимых ресурсах для устройства (номер прерывания и адрес порта ввода/вывода), а также некоторой другой

информации, формат которой будет описан далее. Эта информация после сброса контроллера исходное состояние в определенном формате передается блоку интерфейсной логики, который по ней вместе с арбитром шины произведет конфигурирование устройства.

Блок интерфейсной логики шины *PCI* – это основной блок устройства. Он необходим в следующих целях:

- формирование циклов шины *PCI* с реализацией целевого (*Target*) устройства:
- цикл конфигурирования устройства;
- цикл чтения из порта ввода;
- цикл записи в порт вывода;
- цикл вычисления четности;
- формирование протокола общения с драйвером, по которому в один цикл шины будет выдаваться информация о состоянии устройства и полезная информация полученная из КС или требующая передачи;
- адаптация внутренней системы прерываний контроллера к системе прерываний шины *PCI*;
- взаимодействие с блоком регистров состояния и управления – выдачу регистров состояния драйверу и чтение/запись регистра управления.

Работа этого блока будет более подробно описана в следующих разделах.

Блок регистров управления и состояния отвечает за индикацию состояния устройства – сколько слов израсходовано в стековой памяти, произошло ли заполнение блока буферной памяти приемника, опустел ли блок буферной памяти передатчика; и за хранение управляющей информации – какой делитель битовой частоты используется, формировать ли прерывание при следующем опустошении блока буферной памяти передатчика. Блок регистров управления и состояния взаимодействует со следующими блоками

устройства: блоком формирования прерывания, блоком интерфейсной логики шины и блоком делителя битовой частоты.

Блок буферной памяти передатчика предназначен для достижения тех же целей, что и блок буферный *FIFO* приемника, то есть для согласования относительно медленной и относительно быстрой частей схемы. По окончании передачи ожидающих данных в блок кодера биимпульсного сигнала этот блок вырабатывает внутреннее прерывание и сбрасывает содержимое буфера в нуль. Это необходимо для того, чтобы драйвер на приемной стороне не продублировал одни и те же данные. А так как вся информация передается в сжатом виде, длительное (5 - 7 тактов) повторение одного и того же бита – самый простой и надежный способ показать, что полезные данные в передаваемых пакетах отсутствуют.

Блок генератора битовой частоты предназначен для формирования максимальной битовой частоты передаваемой информации. Этот блок вырабатывает постоянный меандр с частотой 18432 кГц. Такая частота необходима для получения основных битовых скоростей из ряда 115200, 57600, 38400, 19200, 14400, 9600, 7200, 4800, 2400, 1800, 1200, 600 и 300 бит/с.

Блок делителя битовой частоты необходим для получения основных битовых скоростей, перечисленных выше путем деления полученного с генератора меандра на делитель, полученный из блока регистров управления. Частотой полученной после деления, тактируется весь модуль передатчика, включающий в себя блок биимпульсного кодера и блок линейного передатчика.

Блок кодера биимпульсного сигнала предназначен для формирования биимпульсного кода. Биимпульсный код – это относительный линейный код. Относительный – потому, что следующее его состояние целиком и полностью зависит от предыдущего сигнального состояния. Кратко его можно описать так: фаза биимпульсного сигнала не меняется, пока не будет

встречен ноль в исходной цифровой последовательности – после этого фаза меняется на противоположную. Этот блок взаимодействует с блоком буферной памяти передатчика и делителем частоты. После кодирования информация попадает в блок линейного передатчика.

Таким образом, были рассмотрены все блоки, из которых будет состоять устройство, описано их назначение и кратко освещено их устройство.

1.5.3 Выбор и обоснование схемы электрической функциональной

Функциональная схема контроллера специализированного интерфейса приведена в приложении В. Она тесно связана со структурной схемой, рассмотренной ранее. Чтобы лучше понять структуру системы, необходимо рассмотреть состав всех блоков схемы.

Одним из требований технического задания на проектирование является обеспечение совместимости контроллера с предыдущими его реализациями (например, У8804С), поэтому блоки приемника и передатчика, блоки кодера и декодера биимпульсного сигнала используются из технической документации на У8804С и подробному описанию структуры и работы не подвергаются. Блоки приемника и передатчика обладают последовательными входом и выходом соответственно, также параллельными входами и выходами шириной 8 бит для принимаемой и передаваемой информации. Приемник имеет также линию требования на передачу принятых данных далее в схему. Передатчик также имеет еще три служебных линии – индикатор готовности к передаче данных, вход для осуществления записи в порт и вход для подачи битовой частоты f_{bit}.

Блок буферный *FIFO* состоит из детектора пустых посылок сигнала, собственно стековой памяти прямого магазинного типа, а также индикатора переполнения. Глубина стековой памяти берется в размере восьми восьмиразрядных слов.

Блок буферной памяти приемника необходимо составить из схемы ее загрузки содержимым стековой памяти, модулями оперативной памяти, а также схемой выдачи контроллеру содержимого оперативной памяти и переключающего мультиплексора для возможности передачи на 32-разрядную шину *PCI* 64 разрядов принятых данных. Кроме того, этот блок содержит формирователь внутреннего сигнала прерывания.

Блок прерываний фиксирует сигналы внутренних запросов на прерывание от приемника, передатчика и сбрасывать их после того, как с шины придет запрос на обмен данными с устройством. Поэтому в этом блоке содержится логика (2 элемента 2-ИЛИ и элемент 2-И) и *RS*-триггер. Триггер обладает выходом с открытым стоком транзистора (активный уровень - низкий) – так требует спецификация шины *PCI*.

Блок хранения загрузочной конфигурации устройства состоит из ПЗУ размером как минимум 64 байта и обладать шириной в 32 разряда.

Блок регистров состояния и управления необходимо составить из десяти разрядов: 3 разряда – на индикацию уровня заполнения стековой памяти; 4 разряда – на хранение значения делителя для блока делителя битовой частоты; по одному разряду на индикацию блока, вызвавшего прерывание (приемник или передатчик), и разряд на разрешение возбуждения передатчиком прерывания.

Блок буферной памяти передатчика необходимо составить из схемы записи данных в буферную память. Должна присутствовать память размером 32x2 бита с возможностью 32-разрядной записи и 8-разрядного чтения. Кроме того, необходимо предусмотреть схему обнуления буферной памяти по окончании передачи данных, содержащихся в ней. Данные передаются пакетами длиной 8 бит полезной информации и стартовый бит плюс стоп-бит. Исходя из вышеперечисленного, блок передатчика строится из восьми восьмиразрядных регистров, переключающего мультиплексора (8x8)x8, два счетчика, три элемента И и одного элемента ИЛИ.

Блок делителя частоты реализуется из делителя частоты и ключа-селектора на 13 разрядов. Информация для его работы извлекается из регистра управления.

В блоке интерфейсной логики шины *PCI* необходимо реализовать следующие модули:

- модуль формирования сигнала *DEVSEL#* - аналога сигналу *CS (chip select)* у ИМС. Он формируется из декодированного адреса устройства и команд шины PCI, передаваемых по шине *C/BE#* при помощи инвертора, дешифратора на 16 разрядов (срабатывает на конкретный адрес *100h*), логического блока 2-И и буферного элемента с третьим состоянием;

- модуль формирования сигнала *TRDY#* - индикатора готовности устройства к передаче данных, вырабатывается с помощью *D*-триггера и буферного элемента с третьим состоянием;

- модуль конфигурирования устройства состоящий из совокупности ПЗУ на 512 байт (блок хранения загрузочной конфигурации) и ОЗУ такой же емкости для возможности динамического конфигурирования устройства – реализации протокола *Plug&Play*. В данном модуле необходимо использование логических элементов 2-И и 2-ИЛИ, а также *RS*-триггера и двоичного счетчика от 00 до 10;

- модуль дешифрации команд шины и формирования внутренних сигналов управления, для его реализации используется дешифратор с 4 на 16, а также элементы 4-НЕ, 2-И-НЕ и 4-И;

- модуль вычисления четности и реакции на ошибку четности (установка сигнала *PERR#* в активное состояние). Для его реализации используется каскадное поразрядное сравнение с использованием функции «исключающее ИЛИ»;

- модуль формирователя внутреннего сигнала сброса (окончания транзакции). В нем используются элемент 4-И-НЕ и 2-И с одним инвертированным входом;

- модуль соблюдения протокола взаимодействия с драйвером (необходимо сначала выдать на шину двойное слово управления и состояния, и только потом два такта шины передаются данные – итого три такта). Для реализации функций модуля используется элемент задержки;

- мультиплексор шины адреса/данных *AD* (реакция на сигналы *FRAME#* и чтения из порта), используется переключающий мультиплексор 64х32 разряда с третьим состоянием.

1.5.4 Описание работы по схеме электрической функциональной

Электрическая структурная схема контроллера, которая находится в приложении Б. Чтобы лучше понять функционирование разрабатываемого контроллера, необходимо рассмотреть работу каждого блока, реализованного в системе, опираясь на абстрактную элементную базу.

1.5.4.1 Описание работы тракта приема информации

В тракт приемника включаются следующие блоки:

- блок приемника;
- блок декодера биимпульсного сигнала;
- блок буферный (*FIFO*);
- блок буферной памяти приемника.

Как уже упоминалось ранее, исходя из технического задания на проектирование, блок приемника и декодера биимпульсного сигнала подробно описываться не будут.

Итак, данные из линии связи обрабатываются в следующей последовательности. После того, как декодер *D1* примет из канала связи очередной байт данных (линия *SERIN* на схеме) и декодирует его, он выставляет на порте *RQ* высокий уровень сигнала, индицирующий наличие дешифрованных данных. В это же время логический элемент 8-ИЛИ проверяет входящие данные на наличие полезной информации: если в принятом байте данных присутствует хотя бы одна единица, то принятые данные считаются не пустыми и на следующий далее одновибратор,

построенный на триггере *D*10 и элементе 2-И, поступает пускающее напряжение высокого уровня. Одновибратор генерирует импульс длительностью в такт, совпадающий по фазе с сигналом *CLK* шины *PCI*. Этот импульс разрешает считывание данных из декодера в стековую память *STM*. Этот же импульс подается на асинхронный вход *ARD* декодера *D*1, сбрасывающий сигнал на выходе *RQ*.

После того, как заполнится стековая память, она активизирует порт переполнения *OF*, запуская тем самым счетчик со встроенным дешифратором *D*17 от 0 до 8 и переключая триггер *D*18 в высокий уровень. Высокий уровень с триггера поступает на порт *RD* (чтение) буфера. Триггер остается в активном состоянии пока счетчик не досчитает до 8. После этого триггер сбрасывается и устанавливается сигнал *RXIRQ*, индицирующий заполнение буфера памяти приемника *D*23. Эта память заполняется пока идет счет – импульсы, получаемые со счетчика, являются стробирующими для отдельных 8-разрядных модулей регистровой памяти.

По приходу на счетчик *D*27 сигнала *R*IO (чтение ввода/вывода), он запускается и переключает мультиплексор *D*26 с возможностью переключения выхода в состояние высокого импеданса, сначала на входы A, затем на входы B, тем самым отправляя два 32-разрядных слова собранных из восьми байтов на внутреннюю шину *DIO* имеющую 32 разрядных линии.

1.5.4.2 Описание работы тракта передачи информации

В тракт передатчика включаются следующие блоки:

- блок передатчика;
- блок кодера биимпульсного сигнала;
- блок буферной памяти передатчика
- блок генератора битовой частоты;
- делитель частоты.

Как уже упоминалось ранее, блок передатчика и кодера биимпульсного сигнала подробно описываться не будут.

Тракт передатчика работает следующим образом. По пришедшему сигналу *WIO* начинает работу счетчик *D*19, переключающий группы регистров *D*20. Одновременно с этим процессом на линии шины *DIO* выставляются активные данные. За 2 такта работы контроллера буферная память передатчика заполняется.

Сигналом *WIO* также устанавливается в активное состояние триггер *D*18, активизирующий подсистему передачи данных хранящихся в буферной памяти в 8-разрядный порт *ID* кодера биимпульсного сигнала *D*25. Эта подсистема состоит из счетчика *D*24 (ведет циклический счет от 0 до 8) и трех логических элементов И.

Подсистема загрузки передатчика работает следующим образом. Как только освобождается передатчик, он активизирует порт *RDY* (готов), по этому сигналу, в совокупности с логическим уровнем, полученным с триггера *D*21, запускается счетчик *D*24, осуществляющий счет только по присутствию активного уровня на порте +1 (разрешение счета). По тому же сигналу в совокупности с линией *TXRDY* активизируется линия *TXWR*, подаваемая на порт *W* (запись) передатчика.

Три разряда, полученные со счетчика, управляют переключающим мультиплексором *D*22. При помощи этого мультиплексора производится конвертирование 32-разрядных данных в 8-разрядные.

Как только вся информация, хранящаяся в буферных регистрах, передастся, активизируется сигнал *TXIRQ*, индицирующий опустошение передатчика. Поэтому же сигналу происходит очистка буферных элементов, и в канал связи попадают 8-разрядные слова, состоящие из нулей. Такое простое кодирование посылок, не содержащих полезной информации весьма эффективно, так как, передача информации осуществляется в компрессированном виде – в нем невозможно длительное (5-7 бит) повторение одного и того же бита (в данном случае - 0).

Передатчик *D*25 и счетчик *D*24 тактируются битовой частотой f_{bit}, полученной с делителя частоты с логическим управлением, построенного на селекторе *D*16 на 13 линий коммутации. Делитель для управляющего элемента берется из четырех младших бит пятиразрядного регистра управления.

После всех вышеперечисленных преобразований закодированные данные передаются в канал связи через линию *SEROUT*.

1.5.4.3 Описание работы интерфейсной части

Интерфейсная часть контроллера состоит из блоков:

- блок хранения загрузочной конфигурации;
- блок прерываний;
- блок регистров управления и состояния;
- блок интерфейсной логики шины *PCI*.

Для более полного восприятия, изложенного ниже рекомендуется иметь спецификацию на локальную шину *PCI*.

Основные временные диаграммы, иллюстрирующие рабочие циклы шины, представлены на чертеже.

Для того, чтобы осуществить полноценную работу на шине, необходимо реализовать все рабочие циклы, которые на шине происходят.

Цикл чтения/записи конфигурационного пространства происходит в следующей последовательности.

На шине арбитром выставляются сигналы *FRAME#* и *IDSEL* (на один такт) шина *AD* содержит конфигурационный адрес специального формата, показанного на рисунке 1.4.

Рисунок 1.4 - Формат адреса конфигурационного цикла

Шина *C/BE#* содержит команду конфигурационное чтение/запись (1010/1011). Сигнал *IDSEL* уникален для каждого устройства, подключенного к шине.

Далее, пока активен сигнал *FRAME#*, происходит активизация сигналов *IRDY#* (готовность инициатора), *TRDY#* (готовность цели), *DEVSEL#* (подтверждение выбора устройства) и по линиям шины *AD* передаются конфигурационные данные.

По окончании цикла передачи данных управляющие сигналы убираются с шины в следующей последовательности: *FRAME#, IRDY#, TRDY# и DEVSEL#.*

Реализация этого алгоритма осуществляется следующим образом.

По активизации сигнала *IDSEL* защелкивается дешифратор *D*5, выдающий на выходе сигнал *WCNF* или *RCNF*. По возникновению хотя бы одного из этих сигналов, запускается генератор *DEVSEL#* и *TRDY#*, выполненный на элементе задержки и триггерах *D11* и *D13* с третьим состоянием, так как линии *DEVSEL#* и *TRDY#* являются общими для всех устройств на шине.

Сигналы *CONF, RCNF* и *WCNF* защелкиваются на групповом *RS*-триггере *D*35 (имеет общий сброс). По сигналу *CONF* через 2 коммутатора на 4 линии (*D*3 и *D*8) разряды *AD*7..4 выставляются на адресный порт элемента оперативной памяти *D*9. В то же время по сигналам *WCNF* или *RCNF* (в зависимости от вида транзакции) либо осуществляется запись в ОЗУ по выбранному адресу, либо происходит чтение его содержимого через переключающий мультиплексор *D*12.

Формат содержимого конфигурационного пространства представлен на рисунке 1.5.

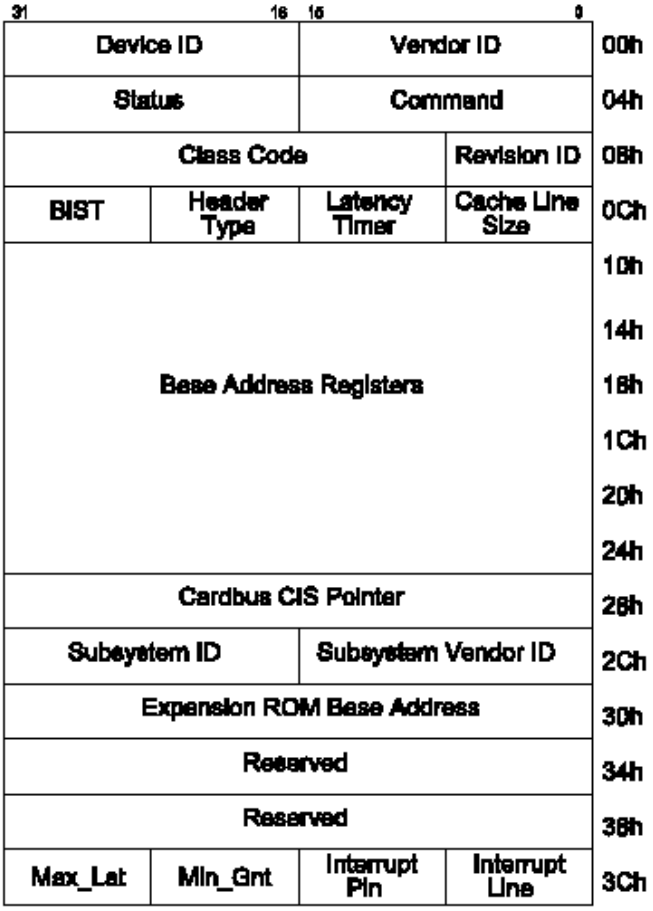


Рисунок 1.5 - Структура конфигурационного пространства

Для проектируемого устройства оно будет содержать следующую информацию:

```
(c)1997-1999 Altitech PCI Statisitcs V3.02 (5/26/99)
Local Report Generated 01:09:11 on 6/26/2002

_____________________________________________________

PCI Bus #0,  PCI Device #13,  Function #0
     10ECh  Vendor ID Leha PGTA diplom project
     8139h Device IDSpecial Communication Controlle
     0000h  Command
     0280h  Status
     0200h - Medium Devsel Speed
       01h  Revision ID
       00h  Prog Interface
       80h  Sub Class Other Communications Device
       07h  Base Class Simple Communication Devices
```

```
       00h  Cache Size
       00h  Latency Timer
       00h  BIST
       00h  Header Type     Single Function Device
 00000100h  Base Address 0  I/O Space (100h)
 00000000h  Base Address 1
 00000000h  Base Address 2
 00000000h  Base Address 3
 00000000h  Base Address 4
 00000000h  Base Address 5
 00000000h  ROM Address
     10ECh  Sub Vendor ID
     8139h  Sub system ID
Hex Dump (Standard PCI Header)
   00h : ECh 10h 39h 81h  00h 00h 80h 02h
   08h : 10h 00h 00h 02h  00h 20h 00h 00h
   10h : 01h 00h 00h 00h  00h 00h 00h DAh
   18h : 00h 00h 00h 00h  00h 00h 00h 00h
   20h : 00h 00h 00h 00h  00h 00h 00h 00h
   28h : 00h 00h 00h 00h  ECh 10h 39h 81h
   30h : 00h 00h 00h 00h  00h 00h 00h 00h
   38h : 00h 00h 00h 00h  05h 01h 20h 40h
```

Чтение/запись конфигурационного пространства осуществляется словами в 32 разряда.

По окончанию транзакции арбитр шины убирает сигналы *FRAME#* и *IRDY#* по этому признаку в контролере формируется внутренний сигнал *RESET*, сбрасывающий устройство в исходное состояние (все внутренние управляющие сигналы в неактивном состоянии).

Цикл чтения/записи порта ввода/вывода показан на чертеже. Он происходит в следующей последовательности.

На шине выставляется сигнал *FRAME#*, индицирующий начало фазы адреса. По нему защелкивается адрес контроллера, установленный на шине

AD, совпадающий со взятым из конфигурационного пространства контроллера. В это же время декодируется команда чтения или записи порта ввода/вывода сформированная на линиях *C/BE#*. Далее генерируются сигналы *DEVSEL#, IRDY#, TRDY#*. Передача данных осуществляется по 32 бита за цикл пока все 3 управляющих сигнала находятся в активном состоянии. Транзакция завершается тогда, когда сигналы *FRAME#* и *IRDY#* перейдут в неактивное состояние.

Реализация алгоритма чтения/записи информации по шине *PCI* в проектируемом устройстве осуществляется следующим образом.

По возникновению сигнала *FRAME#* разрешается сравнение адреса выставленного на шине *AD* (разряды 15..0) и адреса, содержащегося в конфигурационном ОЗУ по смещению 4*h*. Это достигается при помощи цифрового компаратора *D*4 и информации, считанной из ОЗУ *D*9. Смещение адреса 4*h* подается на нее через два коммутатора *D*3 и *D*8 вследствие того, что ни один из сигналов *CONF* и *PRE* не находится в активном состоянии. Если информация на входах A и B компаратора совпадает, он вырабатывает сигнал равенства, и, в совокупности с сигналами *PREWIO* и *PRERIO*, полученными с дешифратора *D*5 и зафиксированными на групповом триггере *D*35, этот импульс запускает генератор сигналов *DEVSEL#* и *TRDY#.*

По сигналу *PRERIO* или *PREWIO* начинает формироваться протокольный обмен информацией между приемником/передатчиком и драйвером ОС. Вначале создается 32 разряда данных, содержащих в себе статус и конфигурацию устройства, считанных, или записанных в блок регистров управления и состояния. Это происходит в следующем порядке.

Для чтения конфигурации - по сигналу *PRERIO*, пока еще не активен сигнал *RIO* (формируется из *PRERIO* при помощи элемента задержки на один такт *D*36), на шину *DIO* через переключающий мультиплексор с состоянием высокого импеданса *D*12, буферный элемент *D*37 и коммутатор *D*33

соединяются регистры *D*32 и *D*34, формирующие слово состояния устройства.

Для записи конфигурации – по сигналу *PREWIO* входы *D* регистра *D*32 коммутируются на шину *DIO*, с которой читается слово управления. Через такт, *D*28 отключает этот регистр от шины.

После вышеперечисленных действий (длительность – три такта шины *PCI*) активизируется сигнал *WIO* или *RIO*, передаваемый в тракт передатчика или приемника. В течение двух тактов на шине *DIO* формируются два 32-разрядных слова, передаваемых или принимаемых с шины *PCI*.

По завершению транзакции генерируется внутренний сигнал *RESET* (если происходила запись в порт, то по пропаданию сигналов *FRAME#* и *IRDY#*, а если происходило чтение из порта, то по активизации сигнала *EMP*), сбрасывающий все внутренние и внешние управляющие сигналы.

Таким образом, вся транзакция шины *PCI* со спроектированным устройством в пакетном (*burst*) режиме будет занимать максимум 5 тактов, что относит разработку к быстродействующим устройствам, использующим минимум процессорного времени.

Система внутренних прерываний реализована на базе *RS*-триггера *D*31, обладающем выходом с открытым стоком, и логических элементов, следящих за установкой и сбросом прерываний.

По окончанию любого из циклов передачи данных по шине *AD*, контролер отключает от нее внутреннюю шину *DIO* при помощи коммутатора *D*6.

В течение каждого цикла передачи данных по шине производится вычисление четности шины *AD* (сигнал *PAR*) при помощи иерархического поразрядного сравнения функциями *XOR*, также производится генерация *PERR#*, если передаваемая четность не совпадает с вычисленной.

Электрическая функциональная схема разработана с помощью САПР *Quartus II* и подтверждена моделированием.

1.5.5 Выбор и обоснование схемы электрической принципиальной

Прежде чем приступить к разработке схемы электрической принципиальной, необходимо выбрать элементную базу устройства. При выборе элементной базы учитываются электрические параметры используемых микросхем, в первую очередь такие как потребляемая мощность и быстродействие.

Рассмотрим, какой критерий является основным при выборе элементной базы для контроллера специализированного интерфейса. Поскольку данное устройство должно обрабатывать сигналы, следующие с высокой частотой, и обладающие логическими уровнями ТТЛ логики, то основными критериями в выборе элементной базы для данного устройства будут высокое быстродействие и соответствие входных и выходных уровней логическим уровням ТТЛ логики.

В настоящее время в основном применяются три типа цифровых ИМС:

- ИМС эмиттерно–связанной логики (ЭСЛ);
- ИМС транзисторно-транзисторной логики (ТТЛ);
- ИМС на комплементарных МОП структурах (КМОП).

Рассмотрим отличительные особенности каждого типа.

Микросхемы ЭСЛ получили широкое распространение в качестве элементной базы быстродействующей и сверхбыстродействующей вычислительной техники. Эти ИМС являются на данном этапе развития техники самыми быстродействующими. Однако быстродействие достигается за счёт большого потребления мощности (порядка 80 – 800 мВт), к тому же логические уровни ЭСЛ логики существенно отличаются от логических уровней ТТЛ логики и для взаимодействия их между собой необходимы специальные микросхемы преобразователи уровней, поэтому данная элементная база не подходит для реализации проекта.

Главным достоинством ИМС КМОП является низкая потребляемая мощность и высокое быстродействие. Существенным недостатком ИМС

КМОП является то, что они чувствительны к статическому электричеству. Следовательно, при сборке устройств с использованием микросхем КМОП требуется создать специальные условия труда, что приведёт к повышению стоимости изделия, поэтому использование микросхем КМОП для разработки данного устройства считаю не целесообразным.

Микросхемы ТТЛ получили самое широкое распространение в качестве элементной базы ЭВМ. Микросхемы ТТЛ делятся на две группы: ТТЛ и ТТЛШ, различающихся быстродействием и потребляемой мощностью.

Микросхемы ТТЛ обладают более высокой потребляемой мощностью, и меньшим быстродействием. Микросхемы ТТЛШ обладают меньшим потреблением мощности и более высоким быстродействием, что обусловлено применением в них транзисторов с переходом Шотки.

Существуют три основных серии ТТЛШ логики:

- К531 – сверхбыстродействующая серия;
- К555 – милливаттная серия с пониженным энергопотреблением;
- К1531 – компромиссный вариант между первыми двумя.

Исходя из вышеизложенного наиболее приемлемым вариантом элементной базы являются микросхемы ТТЛШ логики, серии К1531 обладающие следующими параметрами, представленными на рисунке 1.6.

			Серия			
Параметр	Обозн.	Единицы измер.	К155	К555	К531	К1531
Потребляемая мощность на логический элем-т	Рпот	мВт	10	2	19	4
Среднее время задержки распространения	tзд.р	нс	9	9.5	3	3
Энергия переключения	Эпот	пДЖ	90	19	57	12
Входной ток лог. 0	Iвх0	мА	1.6	0.4	2	0.8
Входной ток лог. 1	Iвх1	мА	0.04	0.02	0.05	0.04
Выходной ток лог. 0	Iвых0	мА	16	8	20	20
Выходной ток лог. 1	Iвых1	мА	0.4	0.4	1	1
Параметры нагрузки: сопротивление	Rн	кОм	0.4	2	0.28	0.28
емкость	Сн	пФ	15	15	15	15

Рисунок 1.6 - Характеристики элементной базы ТТЛШ логики

Так как разрабатываемое устройство должно устанавливаться в стандартный слот шины PCI, то к его габаритам предъявляются достаточно жёсткие требования, а с учётом того, что в данном устройстве осуществляется обработка 32 разрядных массивов информации техническая реализация данного устройства на микросхемах малой и средней степени интеграции представляется весьма затруднительной. Самым рациональным решением данной проблемы является применение программируемых логических интегральных схем (ПЛИС), и в частности ПЛИС фирмы *ALTERA*. Выбор микросхем данной фирмы обосновывается тем, что при примерно равных ценах с ПЛИС таких фирм как *Xilinx Inc*. и *Actel Corporation*, ПЛИС фирмы *ALTERA* позволяют применять в качестве конфигурационных ПЗУ, микросхемы фирмы *Atmel* серии 17С которые обладают на порядок более низкими ценами чем конфигурационные ПЗУ предлагаемыми фирмами изготовителями ПЛИС, кроме того, фирмой *ALTERA Corporation* распространяется бесплатное программное обеспечение *Quartus II* обладающего весьма широкими возможностями, и способного работать с весьма широкой номенклатурой устройств данной фирмы, обычно этой номенклатуры бывает достаточно, чтобы удовлетворить практически любые требования разработчика. Для данной разработки, после проведения анализа функциональной схемы, является ПЛИС *ALTERA FLEX10K (EPF10K10LC84-4)*, обладающая следующими характеристиками:

- число эквивалентных вентилей 10000;
- число логических элементов 576;
- встроенные блоки памяти 3;
- объем памяти, 6144 бит;
- максимальное число выводов пользователя 64.

Кроме того, в ПЛИС возможно создание произвольных логических элементов на произвольное число разрядов и имеются разработки кодеров/декодеров биимпульсного сигнала для реализации внутри ПЛИС.

На сегодня - это емкая архитектура с высокой регистровой насыщенностью (от 576 до 20000 триггерных схем) и большим числом линий ввода/вывода. Высокая предсказуемость и определенность временных параметров обеспечивается глобальными каналами под названием *Fast Tracks.* Эти каналы позволяют для наилучшего случая получать задержки в соединениях через весь кристалл величиной всего 9 нс. Другие ППВМ - архитектуры делятся на два класса — «мелкозернистые», обеспечивающие повышенный уровень использования вентилей, и «крупнозернистые», предназначенные для получения максимального быстродействия. В матрицах *FLEX* сочетаются оба эти класса В основе каждой матрицы есть логические матрические блоки *(LAB - logic-array block)*, которые образуют «крупные зерна». Вместе с тем каждый блок *LAB* можно разделить на 8 логических элементов *(LE — logic element)*, при этом каждый такой элемент («мелкое зерно») имеет в своем составе триггерную и комбинационные схемы. Комбинационная логика состоит из 4-входового генератора любой логической функции с задержкой 5 нс и двух цепей ускоренного распространения Триггер перестраивается в конфигурации *D- ,T- ,YK- , RS-* типа.

Элементы ввода/вывода (ВВ), обслуживающие выходы кристалла, вводят внешние сигналы в схемы *FLEX* или выводят их во внешние цепи. Каждый элемент имеет в составе триггер и логику управления скоростью нарастания выходного напряжения, которая позволяет увеличивать фронты сигналов по сравнению с исходными значениями, чтобы снизить броски напряжения на земляной шине. Элементы ВВ могут быть использованы как входы, выходы или двунаправленные контакты. Триггер может быть использован как защелка для входных или выходных сигналов.

Элементы СЗУПВ, в которые записывается информация о конфигурации ИС, необходимо загружать сразу после включения *FLEX*-кристалла. Соответствующую информацию можно хранить в таких

стандартных приборах памяти, как СППЗУ, ЭСППЗУ, или в системных ЗУПВ.

Во избежание воздействия высокого напряжения, которое может возникнуть в линии связи на электронные цепи устройства, вследствие которого данное устройство может выйти из строя, применяется гальваническая развязка. С этой целью в выходные и входные цепи линии связи устанавливаются линейные трансформаторы типа ТОТ18.

Для загрузки ПЛИС применяется последовательная ЭППЗУ фирмы *Atmel* типа *17C256* ёмкостью 256 Кбит, тактируемая от внешнего генератора тактовой частоты с *f* не более 10 МГц (9216 кГц) собранного на микросхемах КР1531ЛН1 и КР1531ТМ2. С целью загрузки конфигурационной области устройства применяется последовательная ЭППЗУ типа *24LC64* ёмкостью 64 Кбит, тактируемая внешним тактовым генератором.

Генератор тактовой частоты собран на микросхеме КР1531ЛН1 с использованием кварцевого резонатора 18,432 МГц.

С целью обеспечения необходимого уровня входного сигнала применяется операционный усилитель КР544УД2А. С целью преобразования уровня сигнала, приходящего из линии к уровню ТТЛ логики, и обеспечения совместимости с аналогичными устройствами предыдущего поколения.

Необходимо также предусмотреть возможность конфигурирования ПЛИС при помощи интерфейса *ByteBlaster* в отладочных целях.

1.5.6 Описание работы по схеме электрической принципиальной

На обосновании вышеизложенного была разработана схема электрическая принципиальная устройства, которая находится в приложении Г. Основная часть устройства, в которую входит вся структурная схема контроллера, кроме следующих блоков:

- блок приемника;
- блок хранения загрузочной конфигурации *PCI*;
- блок генератора битовой частоты;

- блок передатчика.

Спроектированное устройство функционирует следующим образом.

После включения питания происходит загрузка конфигурации ПЛИС. Элементы СЗУПВ, в которые записывается информация о конфигурации ИС, необходимо загружать в течении 200 мс сразу после включения *FLEX*-кристалла. Соответствующую информацию можно хранить в таких стандартных приборах памяти, как СППЗУ, ЭСППЗУ, или в системных ЗУПВ.

Фирма *ALTERA* предлагает шесть различных схем конфигурации памяти. Эти схемы делятся на два класса - с активным и пассивным режимами конфигурирования. В активном режиме *FLEX* - кристалл сам управляет процессом загрузки данных непосредственно из внешней памяти. В пассивном режиме *FLEX* - кристалл работает как ведомое устройство, а передачей данных управляет ведущая система. Для выбора режима конфигурации ПЛИС служит перемычка *X2*, позволяющая оператору выбрать, откуда будет загружаться ПЛИС: из последовательной ЭППЗУ *DD1*, или из разъема *X3*, представляющего собой интерфейс *ByteBlaster*.

Данные передаются в виде последовательного битового потока данных. Для данного дипломного проекта выбирается пассивный режим загрузки с помощью последовательной ПЗУ серии *AT17C256*. Загрузка памяти выполняется менее чем за 100 мс.

Опорная частота в пассивных схемах загрузки в должна находиться в диапазоне 2..10 МГц. Опорная частота в данной схеме (9,216 МГц) получена из сигнала генератора битовой частоты передатчика при помощи делителя частоты *DD5*. Последовательность адресов генерируется до тех пор, пока *FLEX10K* не завершит загрузку. Тогда контакт *CDONE* освободится и перейдет в состояние «1» за счет внешнего резистора, индицируя, что загрузка завершена.

Контакт *nCNF* может быть соединен с *Vcc* для автоматического вызова загрузки при включении питания. Если в процессе загрузки произошла ошибка, то сигнал *nST* принимает значение «О».

После выхода из режима загрузки микросхема внешней памяти отключается, а контакты, использовавшиеся как шины адреса и данных, принимают свое назначение как элементы ввода/вывода. В любое время работы схемы контакт *nCFG* может быть использован для рестарта загрузки. Семь контактов прибора предназначены только для процедуры загрузки, остальные имеют двойное назначение. Контакты двойного назначения можно не использовать, а перевести в высокоимпедансное состояние соответствующими установками, предлагаемыми в САПР *Quartus II.*

По прошествии загрузки ПЛИС происходит предварительная установка динамического конфигурационного пространства контроллера, как устройства *PCI.* Этот процесс осуществляется при помощи ПЗУ *DD4*. Загрузка конфигурационного пространства осуществляется через порт *CSDA DD2*, при этом процесс загрузки тактируется частотой 33 МГц, полученной с порта *CCLK DD2*. Порты *CA0..CA2* содержат адрес считываемого слова, а чтение начинается по высокому уровню на порту *CWP* ИМС *DD2.*

На элементах *DD3.1, DD3.2* и *DD3.5* собран высокостабильный тактовый генератор битовой частоты на 18432 кГц с использованием кварцевого резонатора. Битовая частота через порт *CLKB DD2* поступает на делитель и кодер биимпульсного сигнала, реализованный внутри *DD2*.

Данные, полученные с шины *PCI* по портам *AD{31..0} DD2* обрабатываются внутри ее, и на выходе *QDAT* формируется последовательный биимпульсный код уровня ТТЛ.

На элементах *DD3.2* и *DD3.3* реализован буферный формирователь-усилитель. Выходной трансформатор блока линейного передатчика Т1 обеспечивает гальваническую развязку и симметрию выхода передатчика биимпульсного сигнала, а резистор *R*12 - выходное сопротивление на

передаче. После вышеуказанных преобразований сигнал через разъем *X4* попадает в канал связи.

Принятый из канала биимпульсный сигнал через разъем *X4* и входной трансформатор блока линейного приемника *T*2, обеспечивающий гальваническую развязку и симметрию цепей приемника, поступает на операционный усилитель *DA1*. С выхода усилителя сигнал, увеличенный до необходимого уровня, проходит на детектор импульсных сигналов, выполненный на *DD6*. После таких преобразований нормализованный под логические уровни сигнал попадает на порт *INDAT DD2*, и обработанный внутри контроллера – через шину *PCI* к пользователю.

Питание контроллера осуществляется через стандартный интерфейсный разъем шины *PCI X1* напряжениями +5В, +12В, -12В. Конденсаторы С1,…,С6,С8,…,С15 выполняют функцию фильтров по питанию.

1.5.7 Расчёт потребляемой мощности

Одной из основных характеристик при разработке электронных устройств является потребляемая мощность.

Расчет потребляемой мощности производится для всего устройства. Мощность, потребляемая всем устройством равна сумме мощностей, потребляемой микросхемами.

Потребляемая мощность интегральной микросхемой *Pcc* (*P*пот) – значение мощности, потребляемой одной микросхемой от источника питания

Расчет потребляемой мощности, Р, Вт, производится по формуле:

$$P=Pcc\cdot N, \quad (1.1)$$

где *Pcc* – мощность, потребляемая одной микросхемой данной серии и данного вида, Вт;

N – количество микросхем данной серии и данного вида, шт.

Мощность, потребляемая одной микросхемой, находится по формуле:

$$Pcc=Icc\cdot Ucc, \quad (1.2)$$

где *Icc* – ток потребления;

Ucc – напряжение источника питания.

Мощность, потребляемая устройством, составляет P=0,6725 Вт. Расчет потребляемой мощности и данные для него представлены в таблице 1.1

Таблица 1.1 - Расчет потребляемой мощности

Наименование	Кол-во шт.	*Icc*, мА	*Ucc*, В	*Pcc*, Вт	*P*, Вт
КР544УД2А	1	5	±12В	0,12	0,12
КР1531ЛН1	1	15,3	+5В	0,0765	0,0765
КР1531ТМ2	1	16	+5В	0,080	0,080
24LC64-I/P	1	10	+5В	0,050	0,050
AT17C256	1	16	+5В	0,080	0,080
EPF10K10LC84-4	1	100	+5В	0,05	0,050
MC1489	1	9	±12В	0,216	0,216
Итого					0,6725

Таким образом, разработанное устройство полностью соответствует спецификации на потребляемую мощность из стандарта шины PCI, которая требует, чтобы мощность, потребляемая картой расширения, не превышала 57,28 Вт.

2 Безопасность жизнедеятельности

2.1 Анализ условий труда

Контроллер специализированного интерфейса «СТЫК С1-ФЛ» предназначен для передачи данных в биимпульсном коде. Данный контроллер управляется с помощью ПЭВМ, установленной в помещении.

Работа инженера связана с непосредственным взаимодействием с видео дисплейными терминалами (ВДТ) и персональными электронно-вычислительными машинами (ПЭВМ). Электрические установки, в том числе и компьютеры, работающие от напряжения электропитания 220 В и с которыми приходится иметь дело, представляют для человека большую потенциальную опасность, которая усугубляется тем, что органы чувств человека не могут на расстоянии обнаружить наличие электрического напряжения на оборудовании. Проходя через тело человека, электрический ток оказывает на него сложное воздействие, являющееся совокупностью термического (нагрев тканей и биологических сред), электролитического (разложение крови и плазмы) и биологического (раздражение и возбуждение нервных волокон и других органов тканей организма) воздействий. Любое из этих воздействий может привести к электрической травме, т.е. к повреждению организма, вызванному воздействием электрического тока или электрической дуги. Электробезопасность в данном случае будет обеспечиваться соответствующей конструкцией ВДТ и ПЭВМ, изоляцией токоведущих частей, электрическим разделением сетей, защитным заземлением и отключением, правильной эксплуатацией оборудования и соблюдением мер электробезопасности работающим персоналом.

Использование электрической энергии также может стать и источником пожара, причиной которого может быть прежде всего короткое замыкание, которое сопровождается большим тепловыделением и образованием в зоне замыкания дуги с разбрызгиванием металла опасна

также и перегрузка сети и устройств, которая влечет за собой сильный разогрев токоведущих проводников и загорание изоляции. Плохой электрический контакт в местах соединений проводников приводит к возникновению больших переходных сопротивлений и повышенному выделению теплоты, что также может стать причиной пожара. Для предупреждения пожаров и взрывов от короткого замыкания, перегрузок и других причин, связанных с эксплуатацией электрооборудования, необходимы правильный выбор, монтаж и соблюдение установленного режима эксплуатации электрических сетей, ПЭВМ и ВДТ, а также электрических средств автоматизации и связи.

При выполнении ряда работ по настройке и отработке аппаратуры, также, как и в общем, при эксплуатации ПЭВМ и ВДТ, оператору неизбежно приходится находиться в зоне электромагнитных излучений (ЭМИ). Длительное воздействие ЭМИ небольшой интенсивности приводит к различным нервным и сердечно-сосудистым расстройствам (головные боли, утомляемость, нарушение сна, боли в области сердца и т.п.). Для защиты от ЭМИ надо обязательно устанавливать экраны, использовать поглотители мощности, устанавливать рациональный режим работы оборудования и обслуживающего персонала. Все вышеупомянутые опасные для жизнедеятельности человека факторы, для работы инженера-диспетчера, при соблюдении описанных правил и мер безопасности, носят характер потенциально опасных.

Вместе с применением компьютеров появились и правила их эксплуатации, которые описывают методы и правила использования имеющихся электронных средств (ЭВМ в частности). При исследовании влияния компьютера на человека и возникло направление, называемое «компьютерной эргономикой». Эргономика - это наука о удобстве использования, в данном случае компьютеров. Ниже освещены некоторые вопросы, связанные с эргономикой и безопасностью использования ЭВМ.

2.2 Разработка рекомендаций по созданию безопасности условий труда

2.2.1 Нагрузка на органы зрения

Наиболее же значимым фактором, обуславливающим тяжесть и напряженность труда, является большая нагрузка на зрение. Это вызвано постоянным мерцанием экрана ПЭВМ, быстрой сменой изображений. Кроме того, усложняет положение и то, что работа инженера-диспетчера связана не только с обслуживанием ПЭВМ и ВДТ, но и с документами. И в этом случае происходит постоянная смена фокусировки зрения то на вертикальной поверхности (экран), то на горизонтальной (клавиатура и документы).

Для того, чтобы свести к минимуму нагрузку на зрение, вызванную профессиональными особенностями работы, необходимо обеспечить нормальную освещенность рабочего места инженера. Кроме того, для обеспечения надежного считывания информации при соответствующей степени комфортности ее восприятия должны быть определены оптимальные и допустимые диапазоны визуальных эргономических параметров. Визуальные эргономические параметры ВДТ и ПЭВМ, соответствующие нормам и пределы их изменений, должны быть установлены оптимальные и допустимые значения которые указаны в таблице 2.1

Таблица 2.1 – Визуальные и эргономические параметры ВДТ и ПЭВМ и пределы их изменений

Наименование параметров	Пределы значений	
	минимальное	максимальное
Яркость знака (яркость фона), кд/кв.м	35	120
Внешняя освещенность экрана, лк	100	250
Угловой размер знака, угл.мин	16	60

Таким образом, визуальные эргономические параметры ВДТ ПЭВМ и освещенность помещения являются параметрами безопасности и их неправильный выбор приводит к большой нагрузке на органы зрения, к ухудшению здоровья пользователя и снижению качества и производительности труда. Хорошее освещение действует тонизирующе, создает хорошее настроение, улучшает протекание основных процессов высшей нервной деятельности. Увеличение освещенности способствует улучшению работоспособности даже в тех случаях, когда процесс труда практически не зависит от зрительного восприятия. При плохом освещении человек быстро устает. Работает менее продуктивно, возрастает потенциальная опасность ошибочных действием и несчастных случаев. Наконец, плохое освещение может привести к профессиональным заболеваниям, например, таким, как рабочая миопия (близорукость), спазм аккомодации и др.

Гигиенические требования к производственному освещению, основанные на психофизических особенностях восприятия света и его влияния на организм человека, могут быть сведены к следующему:

- спектральный состав света, создаваемого искусственными источниками, должен приближаться к солнечному;

- уровень освещенности должен быть достаточным и соответствовать гигиеническим нормам, учитывающим условия зрительной работы;

- должна быть обеспечена равномерность и устойчивость уровня освещенности в помещении во избежание частой «переадаптации" и утомления зрения;

- освещение не должно создавать «блестков» как самих источников света, так и других предметов в пределах рабочей зоны.

Понимание проблемы воздействия компьютеров на человека на государственном уровне приводит к созданию норм и правил по

использованию средств вычислительной техники. В нашей стране 3 июня 2003 года Роспотребнадзором был утвержден документ.

Этот документ регламентирует нормы и правила работы с компьютерами. Следуя ему можно полностью исключить вредное воздействие компьютера на человека. Ниже приведены выдержки из данного документа.

Требования к видео дисплейным терминалам:

- визуальные эргономические параметры ВДТ являются параметрами безопасности, и их неправильный выбор приводит к ухудшению здоровья пользователей. Все ВДТ должны иметь гигиенический сертификат, включающий оценку визуальных параметров;

- конструкция ВДТ, его дизайн и совокупность эргономических параметров должны обеспечивать надежное и комфортное считывание отображаемой информации. Конструкция ВДТ должна обеспечивать возможность фронтального наблюдения экрана путем поворота корпуса в горизонтальной плоскости вокруг вертикальной оси и в вертикальной плоскости вокруг горизонтальной оси в пределах 30 градусов с фиксацией в заданном положении. Дизайн ВДТ должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света. Корпус ВДТ и ПЭВМ, клавиатура и другие блоки и устройства ПЭВМ должны иметь матовую поверхность одного цвета с коэффициентом отражения 0.4 – 0.6 и не иметь блестящих деталей, способных создавать блики. На лицевой стороне корпуса ВДТ не рекомендуется располагать органы управления, маркировку, какие-либо вспомогательные надписи и обозначения. При необходимости расположения органов управления на лицевой панели они должны закрываться крышкой или быть утоплены в корпусе;

- при проектировании и разработке ВДТ сочетания визуальных эргономических параметров и их значения, соответствующие оптимальным и допустимым диапазонам, полученные в результате испытаний в

специализированных лабораториях, аккредитованных в установленном порядке, и подтвержденные соответствующими протоколами, должны быть внесены в техническую документацию на ВДТ;

- конструкция ВДТ должна предусматривать наличие ручек регулировки яркости и контраста, обеспечивающих возможность регулировки этих параметров от минимальных до максимальных значений.

- в целях обеспечения требований защиты от электромагнитных и электростатических полей допускается применение экранных фильтров, специальных экранов и других средств индивидуальной защиты, прошедших испытания в аккредитованных лабораториях и имеющих соответствующий гигиенический сертификат;

- конструкция ВДТ и ПЭВМ должна обеспечивать мощность экспозиционной дозы рентгеновского излучения в любой точке на расстоянии 0,05 м от экрана и корпуса ВДТ, при любых положениях регулировочных устройств не превышающей 7,74 х 10 А/кг, что соответствует эквивалентной дозе, равной 0,1 мбэр/ч (100 мкР/ч).

2.2.2 Эргономика клавиатуры

Клавиатура, как и монитор, является важной частью компьютерного терминала. Хорошая клавиатура позволяет избежать различных недомоганий в областях кистей рук. Эти недомогания являются профессиональной болезнью людей многих профессий, в частности операторов ЭВМ.

Одним из таких заболеваний является туннельный синдром запястья.

Туннельный синдром запястья (медицинское название: карпальный туннельный синдром - КТС) часто встречается у пользователей компьютеров. Любой может стать жертвой этого недуга, а, следовательно, и его неприятных последствий. Чтобы предотвратить развитие КТС, надо знать причины, вызывающие его. Правильный режим труда и эргономичное рабочее место, ранняя диагностика и лечение являются ключом к вашему здоровью и благополучию. Важна также забота вашего руководителя о

дружелюбной, ровной атмосфере в коллективе, так как стрессы также способствуют развитию КТС.

Как и любое другое заболевание костно-мышечной системы, КТС развивается под воздействием ряда факторов:

- генетическая предрасположенность;
- состояние здоровья и образ жизни;
- повторяющиеся движения;
- травма;
- профилактика КТС.

Пользователи ПК должны воспользоваться следующими советами, чтобы снизить риск развития КТС. Некоторые из них связаны с правильной организацией рабочего места (или эргономикой), другие направлены на улучшение режима труда.

2.2.3 Правильная посадка

Высота стула и положение спинки должны регулироваться. Желательно наличие колёсиков, так как без них свободное перемещение по комнате будет сопровождаться неоправданной затратой сил и времени. Полезны также подлокотники. Чтобы правильно отрегулировать стул, необходимо прежде всего обратить внимание на высоту, на которой расположена клавиатура. В среднем рекомендуется 65-75см над полом. Если высота стола регулируется, сядьте так, чтобы поясница была расположена к бедрам под углом 90 градусов, затем отрегулируйте высоту стола так, чтобы угол между плечом и предплечьем также составлял 90 градусов, при этом кисти должны лежать на клавиатуре.

2.2.4 Требования к освещению помещений и рабочих мест с ПЭВМ

Искусственное освещение в помещениях эксплуатации ВДТ и ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, допускается применение системы

комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов). Требования к освещению помещений и рабочих мест с ВДТ и ПЭВМ регламентируются нормами.

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк. Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Следует ограничивать отраженную блесткость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ВДТ и ПЭВМ не должна превышать 40 кд/кв.м и яркость потолка, при применении системы отраженного освещения, не должна превышать 200 кд/кв.м.

Показатель ослеплённости для источников общего искусственного освещения в производственных помещениях должен быть не более 20, показатель дискомфорта в административно-общественных помещениях не более 40, в дошкольных и учебных помещениях не более 25.

Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ВДТ и ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1 - 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

В качестве источников света при искусственном освещении должны применяться преимущественно люминесцентные лампы типа ЛБ. При устройстве отраженного освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп мощностью до 250 Вт. Допускается применение ламп накаливания в светильниках местного освещения.

Общее освещение следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении ВДТ и ПЭВМ. При периметральном расположении компьютеров линии светильников должны располагаться локализованно над рабочим столом ближе к его переднему краю, обращенному к оператору.

Для освещения помещений с ВДТ и ПЭВМ следует применять светильники серии ЛПО36 с зеркализованными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами (ВЧ ПРА). Допускается применять светильники серии ЛПО36 без ВЧ ПРА только в модификации "Кососвет", а также светильники прямого света - П, преимущественно прямого света - Н, преимущественно отраженного света – В. Применение светильников без рассеивателей и экранирующих решеток не допускается.

Яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/кв.м, защитный угол светильников должен быть не менее 40 градусов.

Светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов.

Коэффициент запаса (Кз) для осветительных установок общего освещения должен приниматься равным 1.4.

Коэффициент пульсации не должен превышать 5%, что должно обеспечиваться применением газоразрядных ламп в светильниках общего и местного освещения с высокочастотными пускорегулирующими аппаратами (ВЧ ПРА) для любых типов светильников. При отсутствии светильников с ВЧ ПРА лампы многоламповых светильников или рядом расположенные светильники общего освещения следует включать на разные фазы трехфазной сети.

Для обеспечения нормируемых значений освещенности в помещениях использования ВДТ и ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

2.3 Расчет искусственного освещения помещения для рабочих мест

Искусственное освещение применяется в часы суток, когда естественный свет недостаточен и в помещениях, где он отсутствует.

Искусственное освещение должно обеспечивать освещенность на рабочих местах в соответствии с требованиями освещенность от 300 до 500 лк.

Расчет искусственного освещения производился методом коэффициента использования светового потока. Этот метод наиболее применим для расчета общего равномерного освещения при горизонтальной рабочей поверхности с учетом света, отраженного от стен и потолка. Также этот метод дает возможность определить световой поток ламп, необходимый для создания нормированной освещенности или найти освещенность по заданному потоку.

Расчет производится по формуле:

$$F_{Ë} = \frac{E_{í} * S_{ï} * Z * k}{N * \eta * n}, \qquad (2.1)$$

где *Fл* – световой поток одной лампы, лм;

Ен – наименьшая нормируемая освещенность, в нашем случае Ен=300 лк;

k – коэффициент запаса, учитывающий старение ламп, запыление и загрязнение светильников, примем его значение равным 1.3;

Sn – площадь помещения м2, площадь помещения в котором работает оператор равна 55 м2;

Z – коэффициент неравномерности освещения, показывает отношение средней освещенности к максимальной (обычно значения его принимаются в пределах 1.1 – 1.2), примем его значение равным 1.2;

N – число светильников;

η - коэффициент использования светового потока, который характеризует отношение потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и находится в зависимости от величины индекса помещений (i) и коэффициента отражения потолка и стен (ρ=0.5);

n – количество ламп в светильнике.

Найдем индекс помещения. Он находился по формуле:

$$i = A*B / [h*(A+B)], \quad (2.2)$$

где h – расчетная высота (высота подвеса светильника над полом);

A,B – соответственно длина и ширина помещения, м (11 х 6).

Найдем высоту расположения светильника над освещаемой поверхностью по формуле:

$$h = H – hc – hp, \quad (2.3)$$

где H – общая высота помещения, м (3.2м);

hc – высота от потолка до нижней части светильника (0.2м);

hp – высота от пола до освещаемой поверхности (1м).

$$h=3.2-0.2-1=2м.$$

Величина L/h=1.3 выбирается по справочнику, где L – расстояние между светильниками.

Рассчитываем индекс помещения:

$$i=11*6/[2*(11+6)]=1.94.$$

Определяем коэффициент использования светового потока η = 43%.

Выбираем оптимальное расположение светильников. Светильники с люминесцентными лампами в основном располагаются рядами.

Рассчитываем расстояние между рядами светильников:

$$L=2*1.3=2.6\text{м}.$$

Определяем число светильников N исходя из формулы (2.1) при количестве ламп n=2 в каждом светильнике. Используем светильники ЛСО02 с люминесцентными лампами типа ЛБ.

$$FлN = (300*66*1.2*1.3)/(0.43)=25740/0.43=71832.56 \text{ лм}.$$

Для ламп мощностью 40вт $Fл$=2480 лм.

$$N=71832.56 /2480=29.8.$$

То есть 15 светильников по 2 лампы.

Для ламп мощностью 80вт $Fл$ = 4320 лм.

$$N=71832.56 /4320=17.85.$$

То есть 9 светильников по 2 лампы.

Так как светильники предполагается расположить в два ряда, возьмем четное число светильников (16 и 10 соответственно): Определяем расчетное значение освещенности по формуле:

$$E_Ь = \frac{F_1 * N * \eta * n}{Sn * Z * k}, \qquad (2.4)$$

где F_1 – световой поток одной лампы типа ЛБ;

k – коэффициент запаса равный 1.3;

Sn – площадь помещения м2, площадь помещения в котором работает оператор равна 55 м2;

Z – коэффициент неравномерности освещения равный 1.2;

N – число светильников;

η - коэффициент использования светового потока равный 0.43

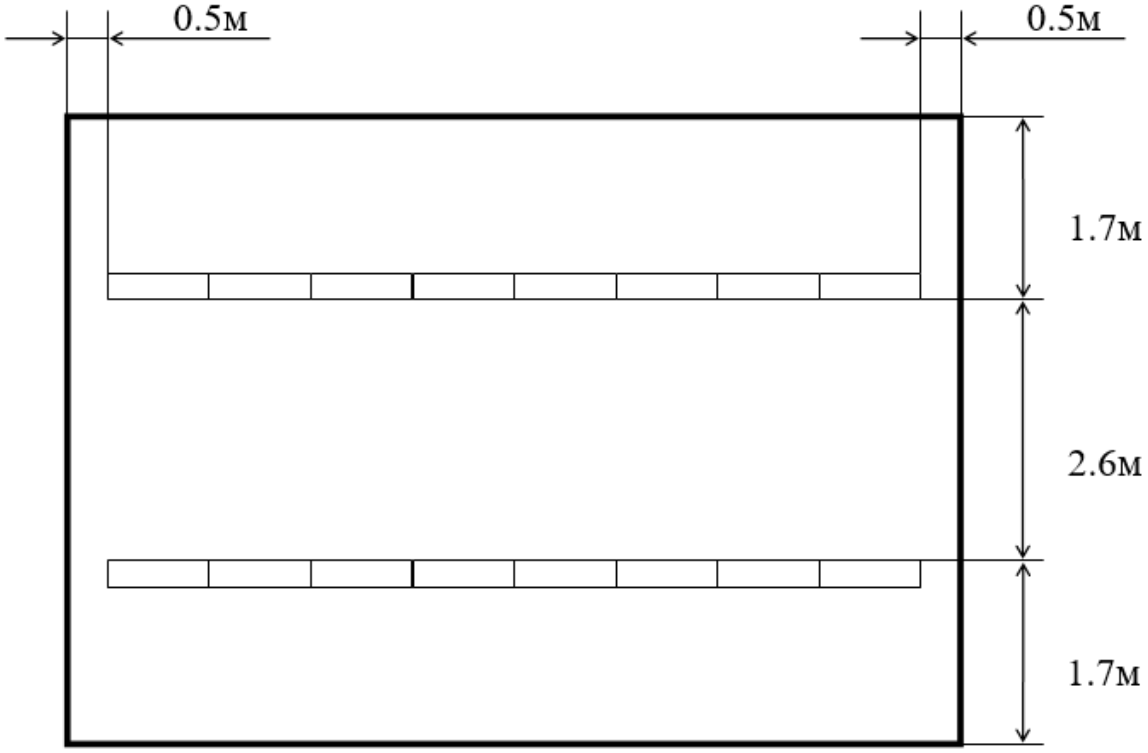


Рисунок 2.1 - Расположение светильников с лампами мощностью 40вт

$Ep40=(2480*2*16*0.43)/(1.2*1.3*55)=331.44$лк,

$Ep80=(4320*2*10*0.43)/(1.2*1.3*55)=360.84$лк,

$Kp40=(Ep–En)/En*100\%=(331.4-300)/300*100\%=10.48\%$,

$Kp80=(Ep–En)/En*100\%=(360.84-300)/300*100\%=20.28\%$.

Светильники с лампами 40вт имеют длину 1250мм, значит, их можно разместить в два ряда по 8 светильников, причем каждый ряд будет начинаться на расстоянии 0.5м от стены, представлены рисунке 2.1.

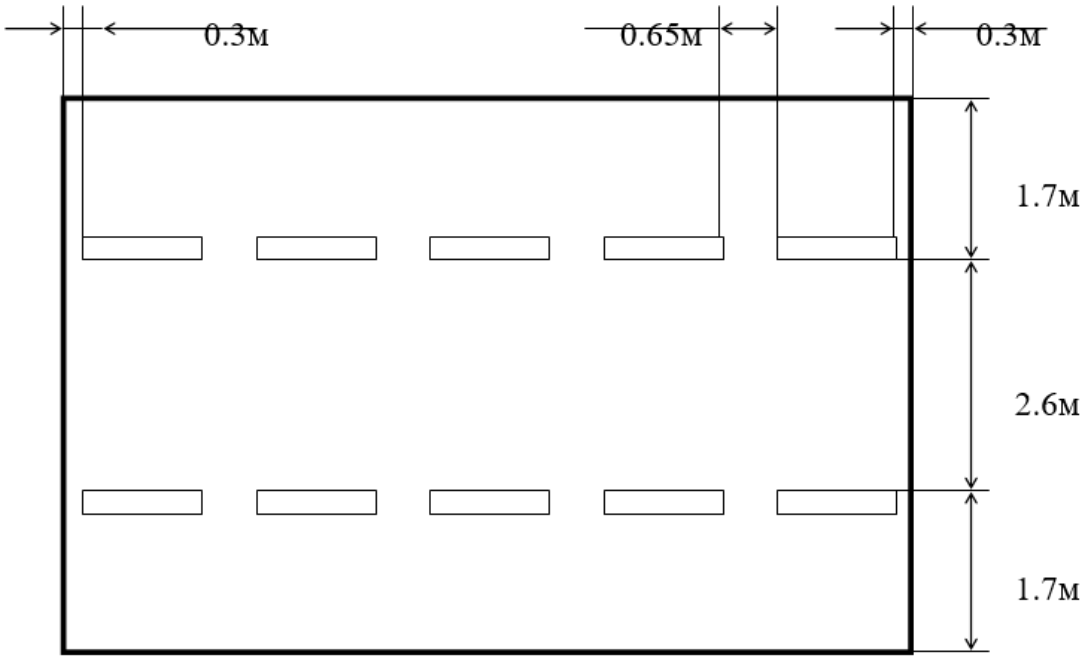


Рисунок 2.2 - Расположение светильников с лампами мощностью 80вт

Светильники с лампами 80вт имеют длину 1550мм, их не удастся разместить в сплошные ряды поэтому их можно равномерно распределить по длине аудитории через 0.65м в два ряда по 5 светильника, причем каждый ряд будет начинаться на расстоянии 0.3м от стены, представлены рисунке 2.2.

После проведенных исследований по безопасности жизнедеятельности можно сделать следующие выводы: на основании произведенных расчетов искусственного освещения помещения было найдено оптимальное распределение светильников. Освещенность рабочей поверхности в помещении при этом удовлетворяет нормативным требованиям.

Заключение

В процессе выполнения дипломного проекта был разработан контроллер, предназначенный для организации интерфейса между источником и потребителем информации, а также специализированными устройствами, использующими интерфейс С1-ФЛ для подключения к ПЭВМ – например, радиорелейные станции. Разработанный контроллер обладает высокими скоростями передачи, работает на больших расстояниях и требует мало процессорного времени, так как для подключения его к ПЭВМ используется прогрессивная технология PCI.

В ходе выполнения выпускной квалификационной работы были разработаны структурная и принципиальная схемы микроконтроллерного измерителя расхода топлива, а также полностью разработано и отлажено программное обеспечение.

Графическая часть выпускной квалификационной работы содержит следующие чертежи:

- схема электрическая структурная – 1 лист формата А1;
- схема электрическая функциональная – 1 лист формата А1.
- схема электрическая принципиальная – 1 лист формата А1;
- временные диаграммы – 1 лист формата А1;

Список использованных источников

1. Бойко В.И. и др. Схемотехника электронных систем. Цифровые устройства. – СПб.: БХВ-Петербург, 2004.
2. Белов А.В. Конструирование устройств на микроконтроллерах. – СПб.: Наука и Техника, 2005.
3. Бородин В.Б. Системы на микроконтроллерах и БИС программируемой логики / Бродин В.Б., Калинин А.В. - М. : ЭКОМ, 2002.
4. Виншип П. Преобразователь частота-напряжение на двух микросхемах/ Радиолоцман, №6, июль 2013. Режим доступа: https://www.rlocman.ru/shem/schematics.html?di=151548.
5. Калабеков Б.А. Цифровые устройства и микропроцессорные системы : учебник / Б. А. Калабеков. - 2-е изд. перераб. и доп. - М. : Горячая линия - Телеком, 2007.
6. Костров Б.В. Микропроцессорные системы : учебное пособие / Б. В. Костров, В. Н. Ручкин. - М. : Десс, 2006.
7. Шило В.Л. Популярные цифровые микросхемы. - М: Радио и связь 1988.
8. Плис фирмы "ALTERA". Элементная база, система проектирования и языки описания аппаратуры. - Книги Электротехника и связь В.Б. Стешенко, 2002
9. ГОСТ 27232-87 Стык аппаратуры передачи данных с физическими линиями/ Издательство стандартов, 1987. Режим доступа: https://docs.cntd.ru/document/1200021927
10. Системы автоматизированного проектирования фирмы Altera MAX+plus II и Quartus II. – РадиоСофтДата, Д. А. Комолов, Р. А. Мяльк, А. А. Зобенко 2002

Приложение А

(Обязательное)

Контроллер специализированного интерфейса «СТЫК-С1ФЛ»

Листинг программы

```
LIBRARY ieee;
 USE ieee.std_logic_1164.all;
 ENTITY  altfp_compare0_altfp_compare_doa IS
      PORT
      (
           aeb :    OUT  STD_LOGIC;
           clock    :    IN  STD_LOGIC;
           dataa    :    IN  STD_LOGIC_VECTOR (31 DOWNTO 0);
           datab    :    IN  STD_LOGIC_VECTOR (31 DOWNTO 0)
      );
 END altfp_compare0_altfp_compare_doa;
      SIGNAL  wire_cmpr1_aeb  :    STD_LOGIC;
      SIGNAL  wire_cmpr1_agb  :    STD_LOGIC;
      SIGNAL  wire_cmpr2_aeb  :    STD_LOGIC;
      SIGNAL  wire_cmpr2_agb  :    STD_LOGIC;
      SIGNAL  wire_cmpr3_aeb  :    STD_LOGIC;
      SIGNAL  wire_cmpr3_agb  :    STD_LOGIC;
      SIGNAL  wire_cmpr4_aeb  :    STD_LOGIC;
      SIGNAL  wire_cmpr4_agb  :    STD_LOGIC;
      SIGNAL  wire_w_lg_w305w306w  :    STD_LOGIC_VECTOR  (0  DOWNTO
0);
     STD_LOGIC_VECTOR (0 DOWNTO 0);
      SIGNAL  wire_w_lg_input_dataa_zero_w297w    :
     STD_LOGIC_VECTOR (0 DOWNTO 0);
      SIGNAL  wire_w_lg_input_datab_zero_w299w    :
     STD_LOGIC_VECTOR (0 DOWNTO 0);
      SIGNAL  wire_w_lg_w_datab_range208w210w    :
     STD_LOGIC_VECTOR (0 DOWNTO 0);
     STD_LOGIC_VECTOR (0 DOWNTO 0);
      SIGNAL  wire_w_lg_w_exp_agb_tmp_w_range265w268w :
     STD_LOGIC_VECTOR (0 DOWNTO 0);
      SIGNAL  wire_w_lg_w_exp_agb_tmp_w_range267w270w :
     STD_LOGIC_VECTOR (0 DOWNTO 0);
```

```
 SIGNAL  wire_w_lg_w_exp_agb_tmp_w_range269w272w :
STD_LOGIC_VECTOR (0 DOWNTO 0);
 SIGNAL  wire_w_lg_w_man_a_not_zero_dffe1_wo_range286w288w :
STD_LOGIC_VECTOR (0 DOWNTO 0);
 SIGNAL  wire_w_lg_w_man_b_not_zero_dffe1_wo_range289w291w :
STD_LOGIC_VECTOR (0 DOWNTO 0);
 SIGNAL  wire_w_lg_aligned_dataa_sign_adjusted_dffe2_wo304w:
STD_LOGIC_VECTOR (0 DOWNTO 0);
 SIGNAL  aclr  :    STD_LOGIC;
 SIGNAL  aligned_dataa_sign_adjusted_dffe2_wi :  STD_LOGIC;
 SIGNAL  aligned_dataa_sign_adjusted_dffe2_wo :  STD_LOGIC;
 SIGNAL  aligned_dataa_sign_adjusted_w :    STD_LOGIC;
 SIGNAL  aligned_dataa_sign_dffe1_wi :  STD_LOGIC;
 SIGNAL  aligned_dataa_sign_dffe1_wo :  STD_LOGIC;
 SIGNAL  aligned_dataa_sign_w :   STD_LOGIC;
 SIGNAL  aligned_dataa_w :   STD_LOGIC_VECTOR (30 DOWNTO 0);
 SIGNAL  aligned_datab_sign_adjusted_dffe2_wi :  STD_LOGIC;
 SIGNAL  aligned_datab_sign_adjusted_dffe2_wo :  STD_LOGIC;
 SIGNAL  aligned_datab_sign_adjusted_w :    STD_LOGIC;
 SIGNAL  aligned_datab_sign_dffe1_wi :  STD_LOGIC;
 SIGNAL  aligned_datab_sign_dffe1_wo :  STD_LOGIC;
 SIGNAL  aligned_datab_sign_w :   STD_LOGIC;
 SIGNAL  aligned_datab_w :   STD_LOGIC_VECTOR (30 DOWNTO 0);
 SIGNAL  both_inputs_zero :   STD_LOGIC;
 SIGNAL  both_inputs_zero_dffe2_wi :   STD_LOGIC;
 SIGNAL  both_inputs_zero_dffe2_wo :   STD_LOGIC;
 SIGNAL  clk_en    :    STD_LOGIC;
 SIGNAL  exp_a_all_one_dffe1_wi :  STD_LOGIC;
 SIGNAL  exp_a_all_one_dffe1_wo :  STD_LOGIC;
 SIGNAL  exp_a_all_one_w :   STD_LOGIC_VECTOR (7 DOWNTO 0);
 SIGNAL  exp_a_not_zero_dffe1_wi : STD_LOGIC;
 SIGNAL  exp_a_not_zero_dffe1_wo : STD_LOGIC;
 SIGNAL  exp_a_not_zero_w :   STD_LOGIC_VECTOR (7 DOWNTO 0);
 SIGNAL  exp_aeb :  STD_LOGIC_VECTOR (3 DOWNTO 0);
```

```
        SIGNAL  exp_aeb_tmp_w : STD_LOGIC_VECTOR (3 DOWNTO 0);
        SIGNAL  exp_aeb_w :     STD_LOGIC;
        SIGNAL  exp_aeb_w_dffe2_wi : STD_LOGIC;
        SIGNAL  exp_aeb_w_dffe2_wo : STD_LOGIC;
        SIGNAL  exp_agb :  STD_LOGIC_VECTOR (3 DOWNTO 0);
        SIGNAL  exp_agb_tmp_w : STD_LOGIC_VECTOR (3 DOWNTO 0);
        SIGNAL  exp_b_all_one_dffe1_wi :  STD_LOGIC;
        SIGNAL  exp_b_all_one_dffe1_wo :  STD_LOGIC;
        SIGNAL  exp_b_all_one_w :    STD_LOGIC_VECTOR (7 DOWNTO 0);
        SIGNAL  exp_b_not_zero_dffe1_wi : STD_LOGIC;
        SIGNAL  exp_b_not_zero_dffe1_wo : STD_LOGIC;
        SIGNAL  exp_b_not_zero_w :   STD_LOGIC_VECTOR (7 DOWNTO 0);
        SIGNAL  exp_eq_grp :    STD_LOGIC_VECTOR (2 DOWNTO 0);
        SIGNAL  exp_eq_gt_grp : STD_LOGIC_VECTOR (3 DOWNTO 0);
        SIGNAL  input_dataa_nan_dffe2_wi :    STD_LOGIC;
        SIGNAL  input_dataa_nan_dffe2_wo :    STD_LOGIC;
        SIGNAL  input_dataa_nan_w :  STD_LOGIC;
        SIGNAL  input_dataa_zero_w : STD_LOGIC;
        SIGNAL  input_datab_nan_dffe2_wi :    STD_LOGIC;
        SIGNAL  input_datab_nan_dffe2_wo :    STD_LOGIC;
        SIGNAL  input_datab_nan_w :  STD_LOGIC;
        SIGNAL  input_datab_zero_w : STD_LOGIC;
    SIGNAL  man_a_not_zero_dffe1_wi :  STD_LOGIC_VECTOR (1 DOWNTO 0);
    SIGNAL  man_a_not_zero_dffe1_wo :  STD_LOGIC_VECTOR (1 DOWNTO 0);
    SIGNAL  man_a_not_zero_merge_w :   STD_LOGIC_VECTOR (1 DOWNTO 0);
        SIGNAL  man_a_not_zero_w :   STD_LOGIC_VECTOR (22 DOWNTO 0);
    SIGNAL  man_b_not_zero_dffe1_wi :  STD_LOGIC_VECTOR (1 DOWNTO 0);
    SIGNAL  man_b_not_zero_dffe1_wo :  STD_LOGIC_VECTOR (1 DOWNTO 0);
    SIGNAL  man_b_not_zero_merge_w :   STD_LOGIC_VECTOR (1 DOWNTO 0);
        SIGNAL  man_b_not_zero_w :   STD_LOGIC_VECTOR (22 DOWNTO 0);
        SIGNAL  out_aeb_dffe3_wi :   STD_LOGIC;
        SIGNAL  out_aeb_dffe3_wo :   STD_LOGIC;
        SIGNAL  out_aeb_w :     STD_LOGIC;
        SIGNAL  out_unordered_w :    STD_LOGIC;
```

```
      SIGNAL  wire_w_dataa_range141w   :    STD_LOGIC_VECTOR     (0
DOWNTO 0);
     STD_LOGIC_VECTOR (0 DOWNTO 0);
      COMPONENT  lpm_compare
      GENERIC
      (
          LPM_PIPELINE   :    NATURAL := 0;
          LPM_REPRESENTATION  :    STRING := "UNSIGNED";
          LPM_WIDTH :    NATURAL;
          lpm_hint  :    STRING := "UNUSED";
          lpm_type  :    STRING := "lpm_compare"
      );
      PORT
      (
          aclr :    IN STD_LOGIC := '0';
          aeb  :    OUT STD_LOGIC;
          agb  :    OUT STD_LOGIC;
          ageb :    OUT STD_LOGIC;
          alb  :    OUT STD_LOGIC;
          aleb :    OUT STD_LOGIC;
          aneb :    OUT STD_LOGIC;
          clken     :    IN STD_LOGIC := '1';
          clock     :    IN STD_LOGIC := '0';
          dataa     :    IN  STD_LOGIC_VECTOR(LPM_WIDTH-1  DOWNTO  0)
:= (OTHERS => '0');
          datab     :    IN  STD_LOGIC_VECTOR(LPM_WIDTH-1  DOWNTO  0)
:= (OTHERS => '0')
      );
      END COMPONENT;
     PROCESS (clock, aclr)
     BEGIN
          IF (aclr = '1') THEN aligned_dataa_sign_adjusted_w_dffe2
<= '0';
          ELSIF (clock = '1' AND clock'event) THEN
```

```
IF (clk_en = '1') THEN aligned_dataa_sign_adjusted_w_dffe2 <=
aligned_dataa_sign_adjusted_dffe2_wi;
                END IF;
            END IF;
        END PROCESS;
        PROCESS (clock, aclr)
        BEGIN
            IF (aclr = '1') THEN aligned_dataa_sign_dffe1 <= '0';
            ELSIF (clock = '1' AND clock'event) THEN
IF   (clk_en   =   '1')   THEN   aligned_dataa_sign_dffe1   <=
aligned_dataa_sign_dffe1_wi;
                END IF;
            END IF;
        END PROCESS;
        PROCESS (clock, aclr)
        BEGIN
IF (aclr = '1') THEN aligned_datab_sign_adjusted_w_dffe2 <= '0';
            ELSIF (clock = '1' AND clock'event) THEN
                IF        (clk_en        =        '1')        THEN
aligned_datab_sign_adjusted_w_dffe2                                <=
aligned_datab_sign_adjusted_dffe2_wi;
                END IF;
            END IF;
        END PROCESS;
        PROCESS (clock, aclr)
        BEGIN
            IF (aclr = '1') THEN aligned_datab_sign_dffe1 <= '0';
            ELSIF (clock = '1' AND clock'event) THEN
IF   (clk_en   =   '1')   THEN   aligned_datab_sign_dffe1   <=
aligned_datab_sign_dffe1_wi;
                END IF;
            END IF;
        END PROCESS;
        PROCESS (clock, aclr)
```

```
    BEGIN
        IF (aclr = '1') THEN both_inputs_zero_dffe2 <= '0';
        ELSIF (clock = '1' AND clock'event) THEN
    IF  (clk_en  =  '1')  THEN  both_inputs_zero_dffe2  <=
both_inputs_zero_dffe2_wi;
            END IF;
        END IF;
    END PROCESS;
    PROCESS (clock, aclr)
    BEGIN
        IF (aclr = '1') THEN exp_a_all_one_w_dffe1 <= '0';
        ELSIF (clock = '1' AND clock'event) THEN
    IF  (clk_en  =  '1')  THEN  exp_a_all_one_w_dffe1  <=
exp_a_all_one_dffe1_wi;
            END IF;
        END IF;
    END PROCESS;
    PROCESS (clock, aclr)
    BEGIN
        IF (aclr = '1') THEN exp_a_not_zero_w_dffe1 <= '0';
        ELSIF (clock = '1' AND clock'event) THEN
    IF  (clk_en  =  '1')  THEN  exp_a_not_zero_w_dffe1  <=
exp_a_not_zero_dffe1_wi;
            END IF;
        END IF;
    END PROCESS;
    PROCESS (clock, aclr)
    BEGIN
        IF (aclr = '1') THEN exp_aeb_w_dffe2 <= '0';
        ELSIF (clock = '1' AND clock'event) THEN
    IF (clk_en = '1') THEN exp_aeb_w_dffe2 <= exp_aeb_w_dffe2_wi;
            END IF;
        END IF;
    END PROCESS;
```

```
    PROCESS (clock, aclr)
    BEGIN
        IF (aclr = '1') THEN exp_b_all_one_w_dffe1 <= '0';
        ELSIF (clock = '1' AND clock'event) THEN
    IF  (clk_en  =  '1')  THEN  exp_b_all_one_w_dffe1  <=
exp_b_all_one_dffe1_wi;
            END IF;
        END IF;
    END PROCESS;
    PROCESS (clock, aclr)
    BEGIN
        IF (aclr = '1') THEN exp_b_not_zero_w_dffe1 <= '0';
        ELSIF (clock = '1' AND clock'event) THEN
    IF  (clk_en  =  '1')  THEN  exp_b_not_zero_w_dffe1  <=
exp_b_not_zero_dffe1_wi;
            END IF;
        END IF;
    END PROCESS;
    PROCESS (clock, aclr)
    BEGIN
        IF (aclr = '1') THEN input_dataa_nan_dffe2 <= '0';
        ELSIF (clock = '1' AND clock'event) THEN
    IF  (clk_en  =  '1')  THEN  input_dataa_nan_dffe2  <=
input_dataa_nan_dffe2_wi;
            END IF;
        END IF;
    END PROCESS;
    PROCESS (clock, aclr)
    BEGIN
        IF (aclr = '1') THEN input_datab_nan_dffe2 <= '0';
        ELSIF (clock = '1' AND clock'event) THEN
    IF  (clk_en  =  '1')  THEN  input_datab_nan_dffe2  <=
input_datab_nan_dffe2_wi;
            END IF;
```

```
          END IF;
     END PROCESS;
     PROCESS (clock, aclr)
     BEGIN
     IF  (aclr  =  '1')  THEN  man_a_not_zero_w_dffe1  <=  (OTHERS  =>
'0');
          ELSIF (clock = '1' AND clock'event) THEN
     IF    (clk_en    =    '1')    THEN    man_a_not_zero_w_dffe1    <=
man_a_not_zero_dffe1_wi;
               END IF;
          END IF;
     END PROCESS;
     PROCESS (clock, aclr)
     BEGIN
     IF  (aclr  =  '1')  THEN  man_b_not_zero_w_dffe1  <=  (OTHERS  =>
'0');
     ELSIF (clock = '1' AND clock'event) THEN
     IF    (clk_en    =    '1')    THEN    man_b_not_zero_w_dffe1    <=
man_b_not_zero_dffe1_wi;
               END IF;
          END IF;
     END PROCESS;
     PROCESS (clock, aclr)
     BEGIN
          IF (aclr = '1') THEN out_aeb_w_dffe3 <= '0';
          ELSIF (clock = '1' AND clock'event) THEN
          IF    (clk_en    =    '1')    THEN    out_aeb_w_dffe3    <=
out_aeb_dffe3_wi;
               END IF;
          END IF;
     END PROCESS;
     cmpr1 :  lpm_compare
       GENERIC MAP (
          LPM_PIPELINE => 1,
```

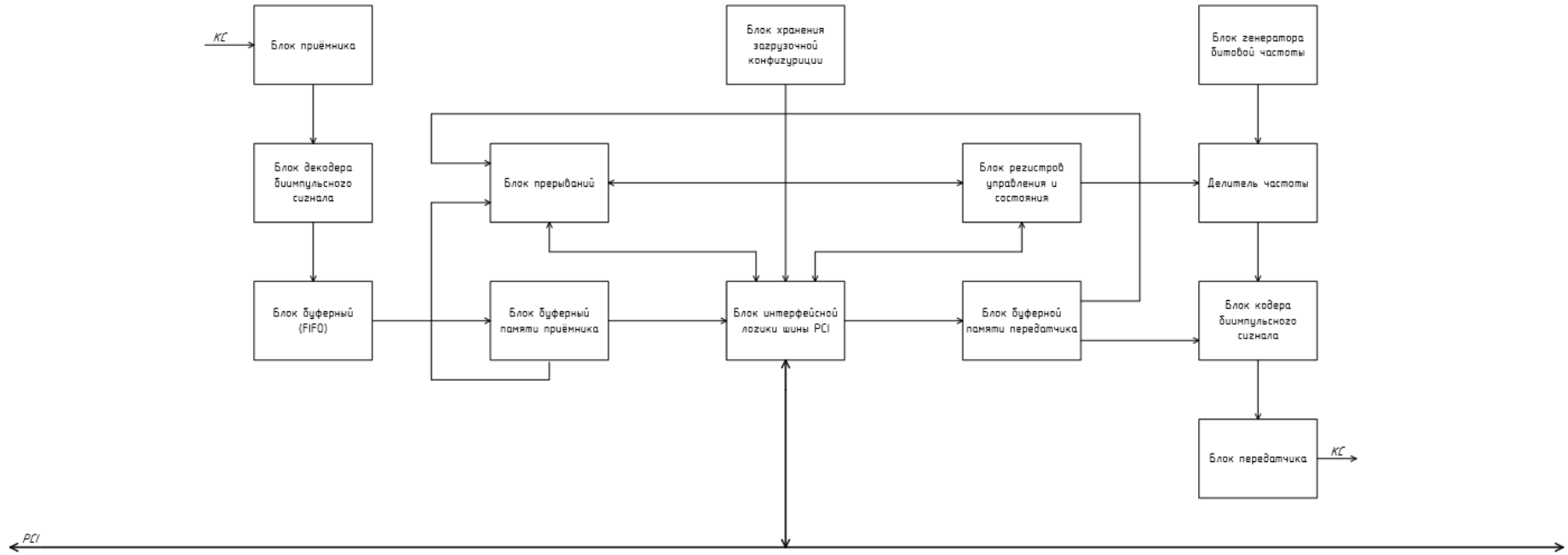
```
          LPM_REPRESENTATION => "UNSIGNED",
          LPM_WIDTH => 8
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY altfp_compare0 IS
     PORT
     (
          clock          : IN STD_LOGIC ;
          dataa          : IN STD_LOGIC_VECTOR (15 DOWNTO 0);
          datab          : IN STD_LOGIC_VECTOR (15 DOWNTO 0);
          aeb       : OUT STD_LOGIC
     );
END altfp_compare0;
ARCHITECTURE RTL OF altfp_compare0 IS
     SIGNAL sub_wire0    : STD_LOGIC ;
     COMPONENT altfp_compare0_altfp_compare_doa
     PORT (
               dataa     : IN STD_LOGIC_VECTOR (15 DOWNTO 0);
               datab     : IN STD_LOGIC_VECTOR (15 DOWNTO 0);
               clock     : IN STD_LOGIC ;
               aeb  : OUT STD_LOGIC
     );
     END COMPONENT;
BEGIN
     aeb    <= sub_wire0;
     altfp_compare0_altfp_compare_doa_component:
altfp_compare0_altfp_compare_doa
     PORT MAP (
          dataa => dataa,
          datab => datab,
          clock => clock,
          aeb => sub_wire0
     );
END RTL;
```

Приложение Б

(Обязательное)

Контроллер специализированного интерфейса «СТЫК-С1ФЛ»

Схема электрическая структурная



КС – Канал Связи

PCI – Peripheral component interconnect «взаимосвязь периферийных компонентов»

Изм.	Лист	№ докум.	Подпись	Дата	ПГУ 1.090301.44.002 Э1
Разраб.	Шереметьев А.Д.				Контроллер специализированного интерфейса «СТЫК-СМИЛ» Схема электрическая структурная
Пров.	Дубравин А.Б.				
Т. контр.					
Н. контр.	Бычков А.С.				
Утв.	Матрасов Н.А.				

Лит. | Масса | Масштаб

Лист 1 | Листов 1

ПГУ, каф.ВТ, хр. 18882

Изм. | Лист | № докум. | Подпись | Дата

ПГУ 1.090301.44.001 ПЗ

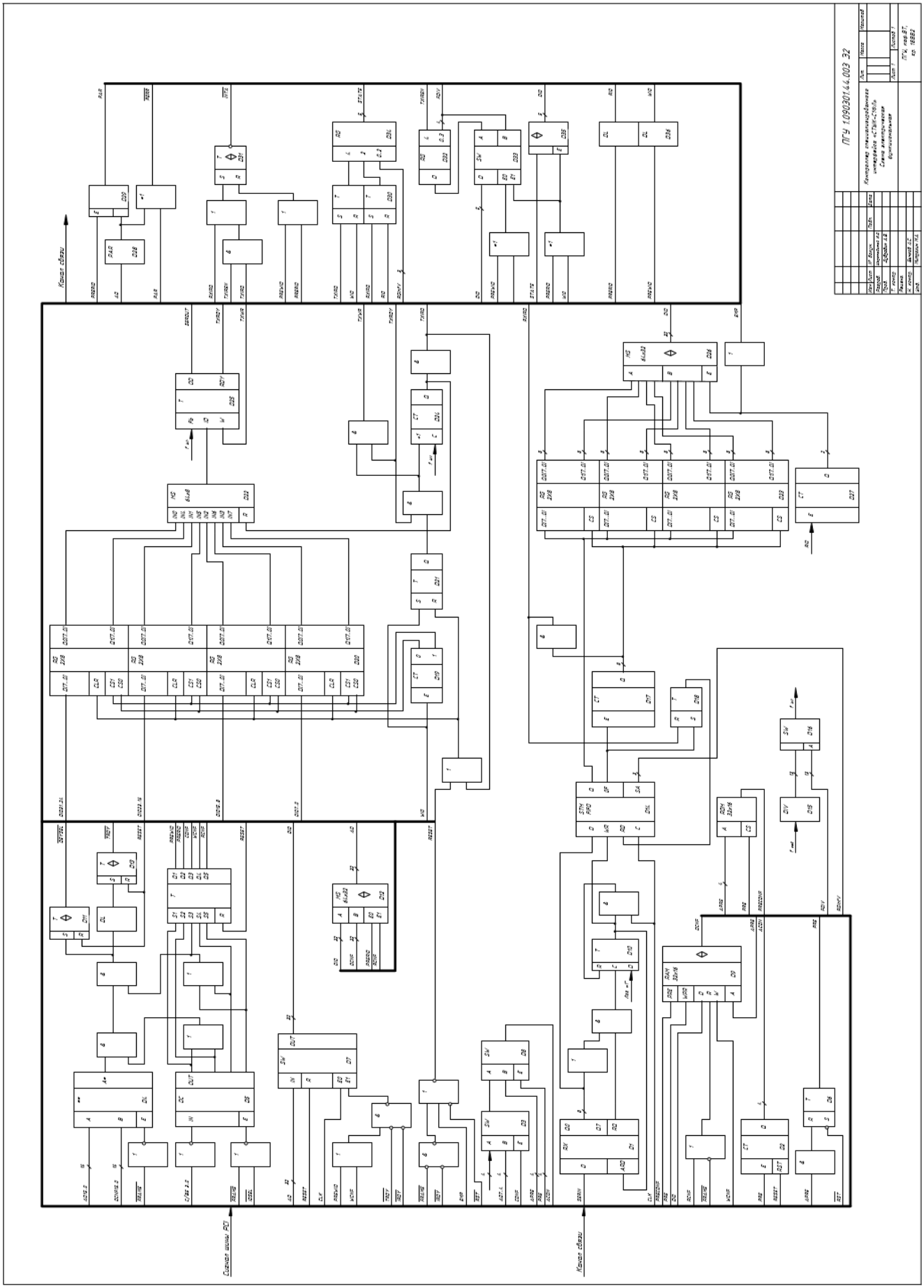
Лист 90

Приложение В

(обязательное)

Контроллер специализированного интерфейса «СТЫК-С1ФЛ»

Схема электрическая функциональная



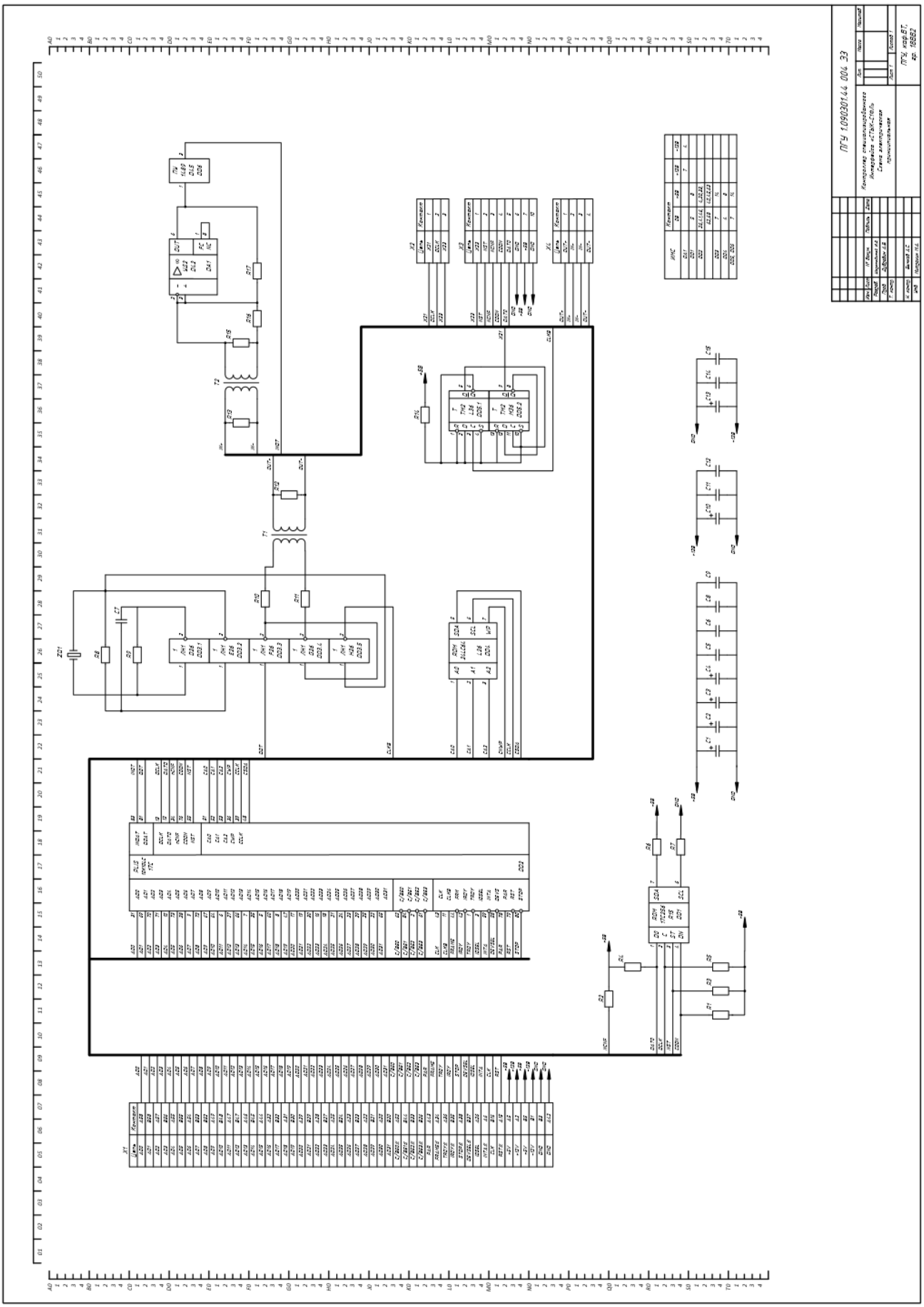
Изм.	Лист	№ докум.	Подпись	Дата	ПГУ 1.090301.44.001 ПЗ	Лист 92

Приложение Г

(обязательное)

Контроллер специализированного интерфейса «СТЫК-С1ФЛ»

Схема электрическая принципиальная



Изм.	Лист	№ докум.	Подпись	Дата	ПГУ 1.090301.44.001 ПЗ	Лист 94

Приложение Д

(обязательное)

Контроллер специализированного интерфейса «СТЫК-С1ФЛ»

Перечень элементов

Зона	Поз. обозн.	Наименование	Кол.	Примечание
		Конденсаторы		
	C1..C4	К50-35 47мкФ 16В ОЖО.464.156 ТУ	4	
	C5,C6	68 пФ 50В 0805X7R	2	Импорт
	C7	470пФ ±5% 50В 0805NP0	1	Импорт
	C8,C9	68 пФ 50В 0805X7R	2	Импорт
	C10	К50-35 47мкФ 16В ОЖО.464.156 ТУ	1	
	C11,C12	68 пФ 50В 0805X7R	2	Импорт
	C13	К50-35 47мкФ 16В ОЖО.464.156 ТУ	1	
	C11,C12	68 пФ 50В 0805X7R	2	Импорт
		Микросхемы		
	DA1	КР544УД2А бКО.347.064 ТУ2	1	
	DD1	AT17C256	1	Импорт
	DD2	EPF10K10LC84-4	1	Импорт
	DD3	КР1531ЛН1 бКО.347.064 ТУ7	1	
	DD4	24LC64-I/P	1	Импорт
	DD5	КР1531ТМ2 бКО.347.064 ТУ7	1	
	DD6	MC1489	1	Импорт
		Резисторы		
	R1..R7	Р1-12 0,125W 1 кОм ±5% 150В ШКАБ.434110.002 ТУ	7	
	R8,R9	Р1-12 0,125W 470 Ом ±5% 150В ШКАБ.434110.002 ТУ	2	
	R10,R11	Р1-12 0,125W 56 Ом ±5% 150В ШКАБ.434110.002 ТУ	2	
	R12,R13	Р1-12 0,125W 150 Ом ±5% 150В ШКАБ.434110.002 ТУ	2	
	R14,R15	Р1-12 0,125W 47 кОм ±5% 150В ШКАБ.434110.002 ТУ	2	
	R16,R17	Р1-12 0,125W 1 кОм ±5% 150В ШКАБ.434110.002 ТУ	2	
	R18,R19	Р1-12 0,125W 1 мОм ±5% 150В ШКАБ.434110.002 ТУ	2	

ПГУ 1.090301.44.005 ПЭ3

Изм.	Лист	№ докум.	Подпись	Дата
Разраб.		Шерстобитов И.Д.		31.05.22
Пров.		Дубравин А.В.		31.05.22
Т. контр.				
Н. контр.		Бычков А.С.		31.05.22
Утв.		Митрохин М.А.		31.05.22

Контроллер специализированного Интерфейса «СТЫК-С1ФЛ» Перечень элементов

Лит.	Лист	Листов
	1	2

ПГУ, каф.ВТ, гр. 18ВВ2

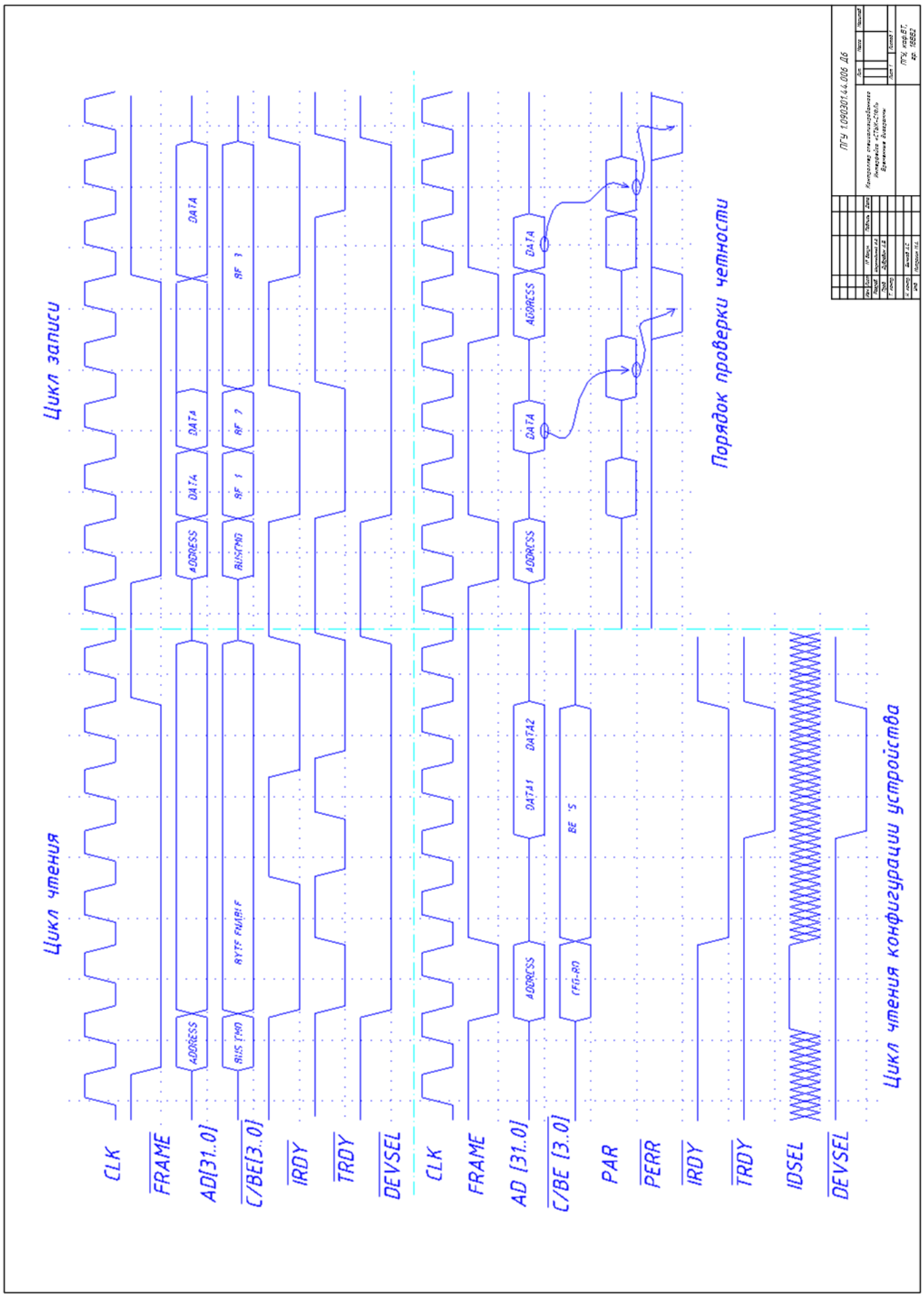
Зона	Поз. обозн.	Наименование	Кол.	Примечание
		Трансформаторы		
	T1,T2	ТОТ18 ОЖ0.493.095	2	
		Разъемы		
	X2	BLS 1X03	1	Импорт
	X3	BLS 1X05	1	Импорт
	X4	TJ6-6P4C-2	1	Импорт
		Резонатор кварцевый		
	ZQ1	HC-49/U 18432 КГц 16 Пф	1	Импорт

Приложение Е

(Обязательное)

Контроллер специализированного интерфейса «СТЫК-С1ФЛ»

Временные диаграммы



Изм.	Лист	№ докум.	Подпись	Дата	ПГУ 1.090301.44.001 ПЗ	Лист 99

Приложение Ж

(Обязательное)

Контроллер специализированного интерфейса «СТЫК-С1ФЛ»

Презентация

Тема работы: «Контроллер специализированного интерфейса «СТЫК-С1ФЛ»

Автор работы: студент группы 18ВВ2 Шерстобитов И.Д.

Научный руководитель: к.т.н., доцент каф. ВТ ПГУ Дубравин А.В.

Пенза 2022

Слайд 1

Цель проектирования устройства

Цель работы

разработка контроллера, осуществляющего связь на большие расстояния по существующим каналам связи и обладающего низкой стоимостью.

Слайд 2

Постановка задачи

Задачей данного дипломного проекта является разработка «Контроллера специализированного интерфейса «СТЫК-С1ФЛ» Назначение этого устройства – преобразование логических сигналов ПЭВМ, получаемые по локальной шине PCI ПЭВМ, в биимпульсный сигнал по ГОСТ 27232-87, пригодный для передачи по каналам связи и осуществление обратного преобразования при приеме данных из канала связи.

Слайд 3

Обзор состояния рынка устройств передачи сигналов

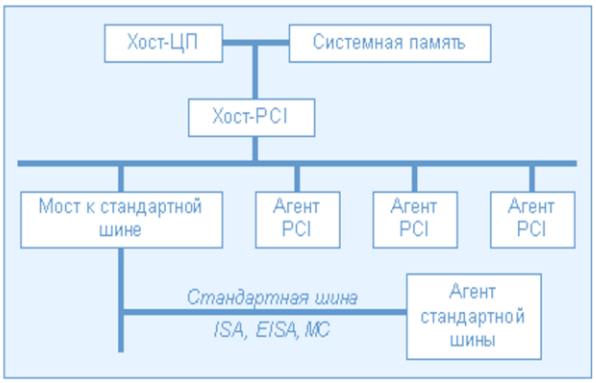
На сегодняшний день производятся подобные устройства, но большинство из них являются внешними устройствами, подключаемыми к ЭВМ при помощи интерфейса RS-232C

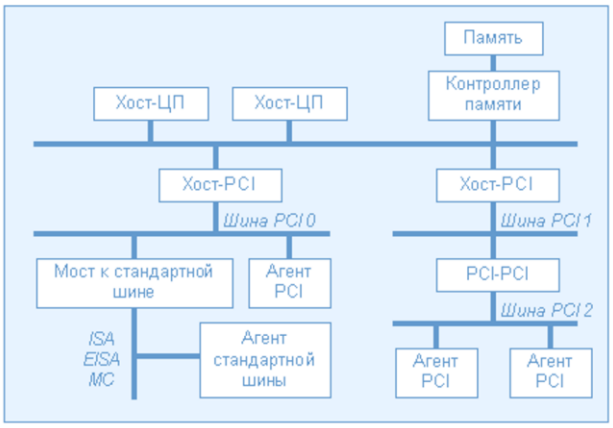
Слайд 4

Выбор вариантов реализации проекта

Слайд 5

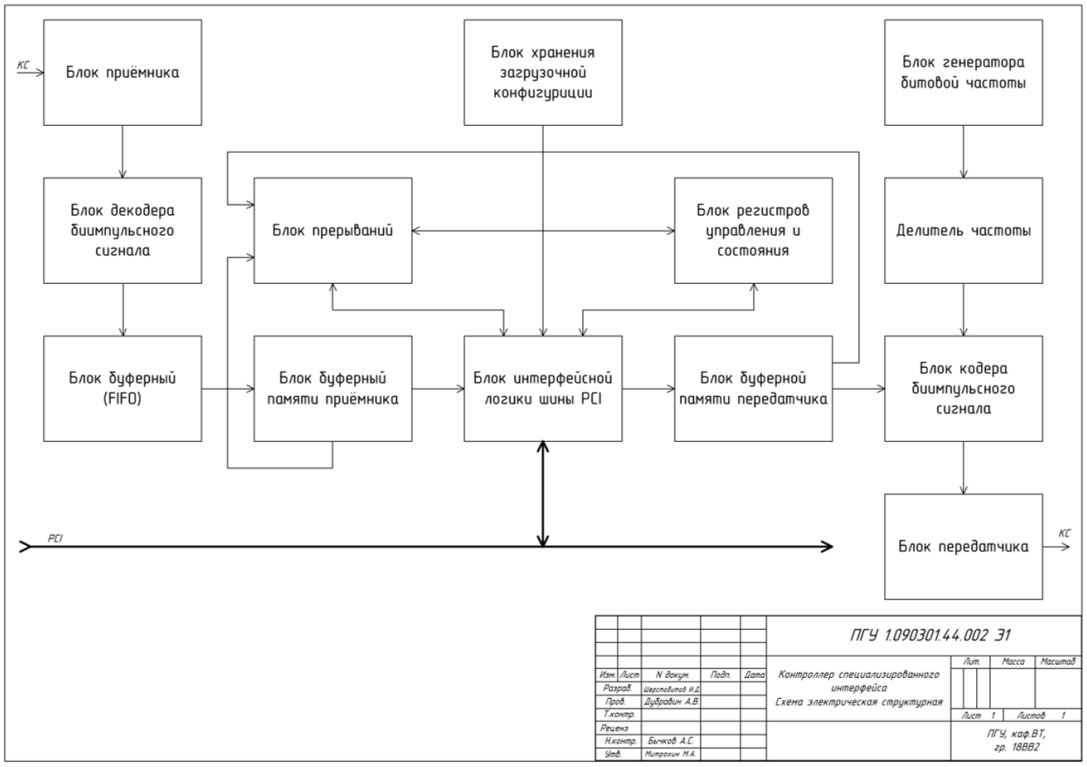
Краткое описание шины PCI





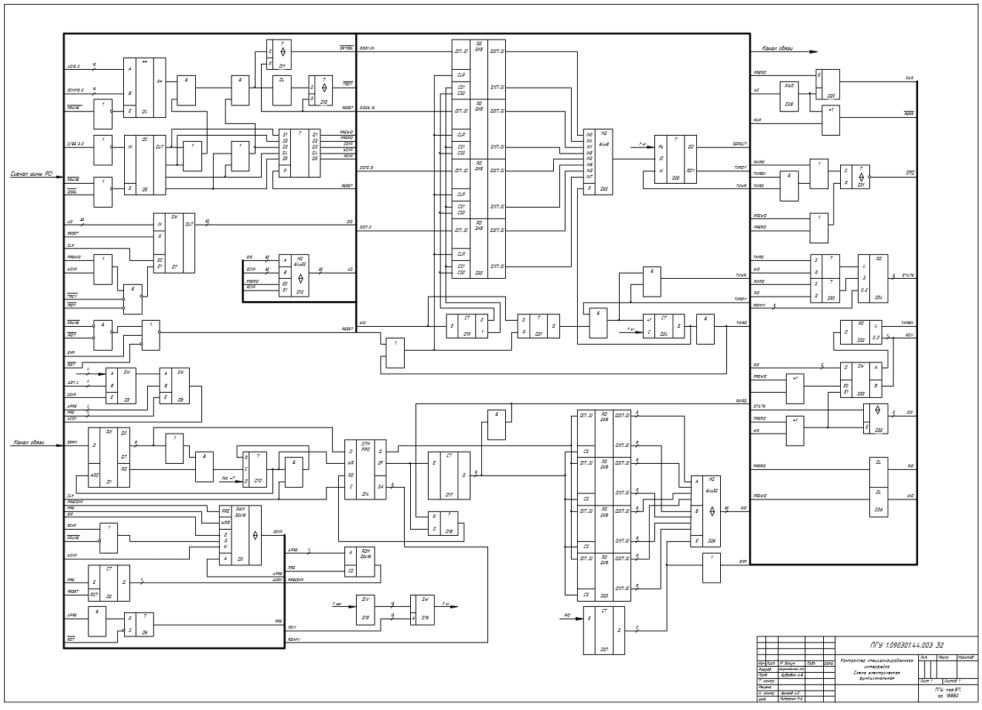
Слайд 6

Схема электрическая структурная



Слайд 7

Схема электрическая функциональная



Слайд 8

Выбор элементной базы

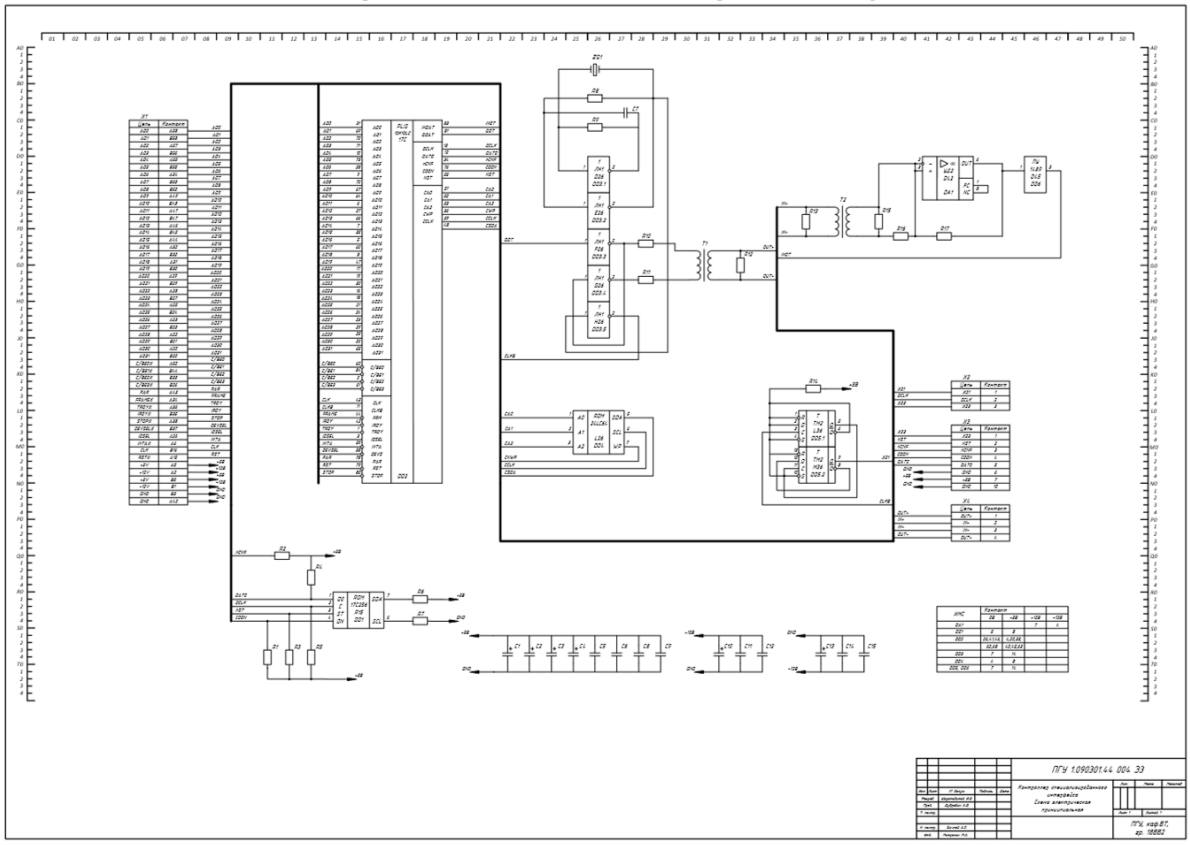
Существуют три основных серии ТТЛШ логики:

- К531 – сверхбыстродействующая серия;
- К555 – милливаттная серия с пониженным энергопотреблением;
- К1531 – компромиссный вариант между первыми двумя.

			Серия			
Параметр	Обозн.	Единицы измер.	К155	К555	К531	К1531
Потребляемая мощность на логический элем-т	Рпот	мВт	10	2	19	4
Среднее время задержки распространения	tзд.р	нс	9	9.5	3	3
Энергия переключения	Эпот	пДЖ	90	19	57	12
Входной ток лог. 0	Iвх0	мА	1.6	0.4	2	0.8
Входной ток лог. 1	Iвх1	мА	0.04	0.02	0.05	0.04
Выходной ток лог. 0	Iвых0	мА	16	8	20	20
Выходной ток лог. 1	Iвых1	мА	0.4	0.4	1	1
Параметры нагрузки: сопротивление	Rн	кОм	0.4	2	0.28	0.28
емкость	Сн	пФ	15	15	15	15

Слайд 9

Схема электрическая принципиальная



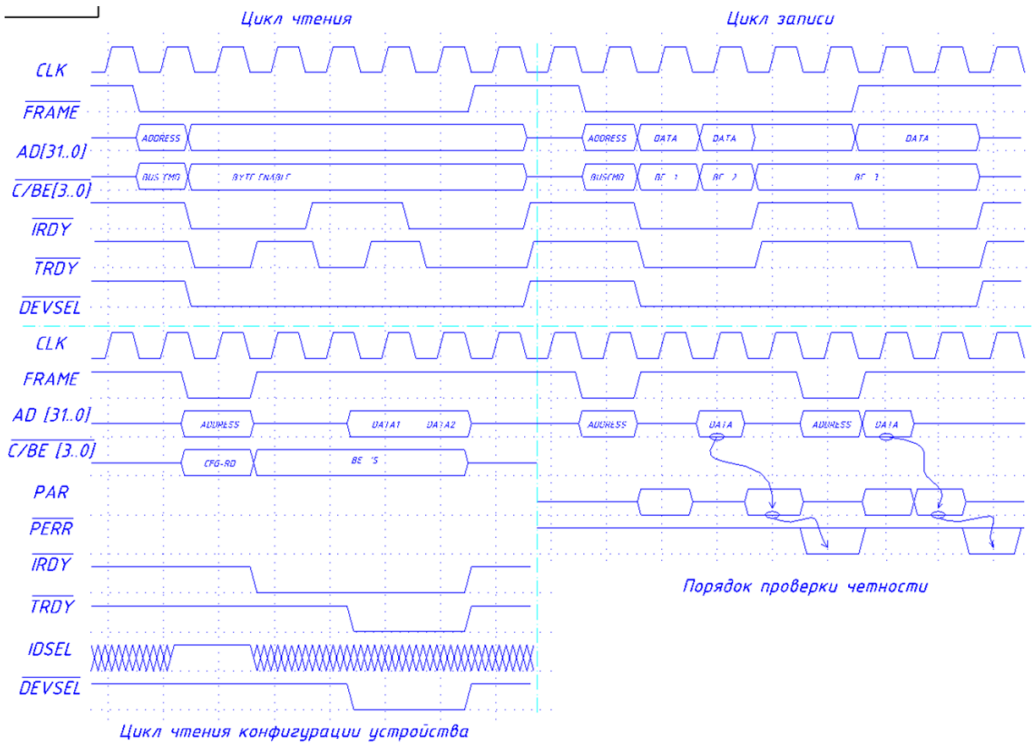
Слайд 10

Расчёт потребляемой мощности

Наименование	Кол-во шт.	Icc, мА	Ucc, В	Pcc, Вт	P, Вт
КР544УД2А	1	5	±12В	0,12	0,12
КР1531ЛН1	1	15,3	+5В	0,0765	0,0765
КР1531ТМ2	1	16	+5В	0,080	0,080
24LC64-I/P	1	10	+5В	0,050	0,050
AT17C256	1	16	+5В	0,080	0,080
EPF10K10LC84-4	1	100	+5В	0,05	0,050
MC1489	1	9	±12В	0,216	0,216
Итого					0,6725

Слайд 11

Временные диаграммы



Слайд 12

Заключение

В процессе выполнения дипломного проекта был разработан контроллер, предназначенный для организации интерфейса между источником и потребителем информации, а также специализированными устройствами, использующими интерфейс С1-ФЛ для подключения к ПЭВМ. Разработанный контроллер обладает высокими скоростями передачи, работает на больших расстояниях и требует мало процессорного времени, так как для подключения его к ПЭВМ используется прогрессивная технология PCI.

Слайд 13

Спасибо за внимание!

Слайд14