

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление 09.03.01 «Информатика и вычислительная техника»
подготовки:

Профиль подготовки: «Вычислительные машины, комплексы, системы и
сети

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
БАКАЛАВРА

на тему

УЗЕЛ ИНТЕРФЕЙСА FXS ДЛЯ УЧРЕЖДЕНЧЕСКИХ АВТОМАТИЧЕСКИХ
ТЕЛЕФОННЫХ СТАНЦИЙ

Студент

Малинин Эдуард Владимирович
(подпись, дата)

Малинин Эдуард Владимирович.
(ФИО полностью)

Руководитель

Карамышева Н.С.
(подпись, дата)

Карамышева Н.С.
(фамилия, инициалы)

Нормоконтролёр

Кучин А.В.
(подпись, дата)

Кучин А.В.
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 03.06.2020г. № 12)

Заведующий кафедрой

М.А. Митрохин
(подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от
22.06.2020 № 16)

Секретарь ГЭК

Е.И. Гурин
(подпись)

Е.И. Гурин
(фамилия, инициалы)

Пенза 2020

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

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
 М.А. Митрохин
«30» апреля 2020г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА

1. Студент Малинин Эдуард Владимирович гр.16ВВ1 факультета вычислительной техники.

2. Тема работы: «Узел интерфейса FXS для учрежденческих автоматических телефонных станций(УАТС)».

Тема утверждена приказом ПГУ №0335а/о от "30" апреля 2020г.

3. Руководитель бакалаврской работы к.т.н., доцент каф. ВТ ПГУ Карамышева Н.С.

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

Разработка модуля интерфейса FXS для учрежденческих автоматических телефонных станций.

Написать программу взаимодействия с модулем FXS.

Разработать принципиальную схему узла FXS.

5. Перечень подлежащих разработке вопросов.

Требуется провести анализ взаимодействия узла FXS с абонентской линией, рассмотреть существующие аналоги, разработать принципиальную и структурную схемы, осуществить выбор элементной базы, разработать программу взаимодействия с модулем FXS

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ взаимодействия узла FXS с абонентской линией	20	30.04.2020– 03.05.2020	
2	Анализ аналогов	7	04.05.2020– 10.05.2020	
3	Разработка принципиальной схемы	2	11.05.2020– 15.05.2020	
4	Разработка структурной схемы	1	16.05.2020– 18.05.2020	
5	Выбор компонентов для реализации принципиальной схемы	3	19.05.2020– 21.05.2020	
6	Разработка программного обеспечения	10	22.05.2020– 26.05.2020	
7	Оформление выпускной работы	20	27.05.2020– 29.05.2020	

Дата выдачи задания "30" апреля 2020г.

Руководитель бакалаврской работы  30.04.20 Карамышева Н.С.
(подпись, дата)

Задание к исполнению принял студент  30.04.20 Малинин Э.В.
(подпись, дата)

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

Реферат

Выпускная квалификационная работа бакалавра содержит: 94 листа, 17 рисунка, 4 таблиц, 8 приложений.

УЗЕЛ ИНТЕРФЕЙСА FXS, ПРОТОКОЛЫ ВЗАИМОДЕЙСТВИЯ, МИКРОСХЕМЫ SLIC, ДРАЙВЕР ИНТЕРФЕЙСА TAPI.

Целью ВКР является:

- Анализ взаимодействия существующих микросхем узла интерфейса FXS;
- Разработка программного обеспечения для взаимодействия с портом FXS.

Объектом разработки является проектирование принципиальной схемы порта модуля интерфейса FXS на современной электронной базе.

В результате проведенной работы были рассмотрены некоторые современные микросхемы для порта FXS, разработаны структурная и принципиальная схема модуля SLIC, а так же написана программа драйвера взаимодействия с портом FXS.

					ПГУ 2.090301.25.001 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Узел интерфейса FXS для учрежденческих автоматических телефонных станций Пояснительная записка	Литера	Лист	Листов
Разраб		Малинин Э.В.					2	85
Пров		Карамышева Н.С.						
Н. Контр.		Кучин А.В.				ФВТ, каф ВТ гр. 16ВВ1		
Утв		Митрохин М.А.						

Перечень принятых сокращений

FXO – (Foreign eXchange Office) – аналоговый интерфейс абонентских устройств телефонных станций).

FXS (Foreign eXchange Subscriber) – голосовой интерфейс или порт, эмулирующий расширение интерфейса АТС;

SLIC – (Subscriber Line Interface Circuit) – или схема интерфейса абонентской линии);

ВКР – выпускная квалификационная работа;

ТА – телефонный аппарат;

ТС – технические средства;

УЦАТС – учрежденческая цифровая автоматическая телефонная станция;

ЦТА – цифровой телефонный аппарат;

DTMF – (Dual-Tone Multi-Frequency) — или двухтональный многочастотный аналоговый сигнал;

SCI – (Scalable Coherent Interface) - или масштабируемая согласованная взаимосвязь;

PCI - (Motorola Serial Peripheral Interface) – или взаимосвязь периферийных компонентов;

BORSCHT – (Battery, Overload protection, Ringing, Supervision, Coding, Hybrid, Testing) – функции: батарея, защита напряжения, вызов, контроль, цифро-аналоговый преобразователь, сопряжение, тестирование;

ЦП – центральный процессор;

SIP – (Session Initiation Protocol) – или протокол установления сеанса;

RTP - (Real-time Transport Protocol) – или протокол передачи трафика реального времени;

RTCP - (Real – time Transport Control Protocol) – или протокол управления передачей в реальном времени;

VAD – (Voice Activity Detector) – или определение голосовой активности;

CELP – (Code Exited Linear Prediction) – или алгоритм кодирования речи;

ИКМ – импульсно-кодовая модуляция;

					ПГУ 2.090301.25.001 ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		3

ЦАП - цифро–аналогового преобразователь.

Содержание

Реферат	2
Введение	7
Актуальность работы	8
1.Анализ предметной области	9
1.1Обзор аналогов	9
1.2 Микросхема SLIC производства фирмы lantiq	9
1.2.1 Технические характеристики DuSLICxT PEF3201	9
1.2.2 Параметры микросхемы SLIC производства фирмы Lantiq.....	9
1.2.3 Поддерживаемые режимы сигнализации	10
1.2.4 Интерфейс связи микросхемы с АТС	10
1.2.5Дополнительные параметры	11
1.3 Микросхема SLIC производства фирмы Microsemi	11
1.3.1 Основные характеристики ZLR964321L:	11
1.3.2 Параметры микросхемы SLIC производства фирмы Microsemi .	11
1.4 Микросхема SLIC производства фирмы Nuvoton	12
1.4.1 Особенности.....	12
1.5 Сравнение микросхем модуля FSX	13
1.6 Алгоритм работы узла FXS.....	14
1.6.1 Установление исходящего соединения	15
1.6.2 Установление входящего соединения	15
1.6.3 Тестирование абонентского комплекта	16
1.7 Описание узла интерфейса FXS	16
1.8 Интерфейс TAPI	18
1.9 Протокол SiP.....	18
1.10 Общие сведения о протоколе RTP.....	20
1.10.1 Описание протокола RTP.....	20
1.11 Звуковые кодеки	22

1.11.1	Кодек G.711	26
1.11.2	Работа кодека	26
1.12	Интегральная микросхема PSB 21391H	31
1.13	Электронные схемы в интерфейсах абонентских телефонных линий	33
1.14	Интегральная микросхема PEF3201	37
1.15	Вывод	47
2	Реализация	48
2.1	Фрагмент структурной схемы разрабатываемого модуля FXS	48
2.2	Принципиальная схема разрабатываемого модуля FXS	49
3.	Разработка программного обеспечения	50
3.1	Программа управления модулем FXS	50
3.2	Вывод	59
4.	Охрана труда.....	60
4.1	Основные воздействия на программиста за ПЭВМ	60
4.2	Требования охраны труда перед началом работы	60
4.3	Требования охраны труда во время работы	61
4.4	Требования охраны труда по окончанию работы.....	62
5.	Заключение	63
	Список использованных источников	64
	Приложение А	65
	Приложение Б.....	4
	Приложение В	4
	Приложение Г	72
	Приложение Д	76
	Приложение Е.....	82
	Приложение Ж	84
	Приложение З	86

Введение

Важную роль в организации работы предприятий, фирм, различных форм собственности играют УЦАТС, предназначены для обеспечения голосовой связи работников. Особенностью УЦАТС является небольшое количество номеров, поддерживающих АТС до 128–256 номеров. Все абоненты находятся на небольшом расстоянии от УЦАТС. Для подключения телефонных аппаратов абонентов используется интерфейс FXS. Интерфейс FXS должен обеспечивать постоянное напряжение для питания ТА, генерировать звонки и тональные сигналы, воспринимать положение трубки (снята/положена) и набор номера от ТА. Также он позволяет использовать как новейшие ЦТА, так и аналоговые ТА, которые использовались еще с давних пор. Это позволяет не прокладывать новые телефонные линии, а использовать уже имеющиеся, что значительно уменьшает расходы на телефонию.

Целью работы: проанализировать микросхемы SLIC различных производителей, разработать модуль интерфейса FXS для учреждений АТС.

Задачи работы:

- Рассмотреть интерфейс TAPI;
- Рассмотреть протоколы взаимодействия SIP, RTR;
- Разработать структурную и принципиальную схему модуля FXS;
- Разработать фрагмент программного обеспечения модуля FXS.

Актуальность работы

Функциональность УЦАТС является важной для эффективной работы предприятия. При этом необходимо обеспечивать связь для каждого работника предприятия. У каждого работника на рабочем месте должен быть ТА. ТА включается в порт FXS, соответственно УЦАТС должно иметь необходимое количество портов.

ЦТА обеспечивают больше функций, более удобны. Но использование аналоговых ТА остается актуальным, поскольку позволяют использовать существующие ТА, проводку абонентских линий в здании, что экономит на организации ТС. Кроме того, ТА обычно ограничены длиной, витой пары 100м, а аналоговые позволяют длиннее. Поэтому для УЦАТС по-прежнему остается актуальным наличие портов FXS для аналоговых ТА.

Порты FXS УЦАТС позволяют связать все телефоны на предприятии. Одним из важных составляющих порта FXS является его набор функция, малые размеры корпуса и количество каналов. Отдельно следует выделить технологию работы и проектирование.

					ПГУ 2.090301.25.001 ПЗ	Лист
						8
Изм.	Лист	№ докум	Подпись	Дата		

1. Анализ предметной области

1.1 Обзор аналогов

Порт интерфейса FXS возможно реализовать на различных микросхемах легкой и средней степени интеграции как зарубежных, так и отечественных производителей.

В работе рассмотрим различия реализации порта интерфейса FXS на современных радиоэлектронных элементах.

Современный УЦАТС порт FXS строится с использованием микросхем SLIC .

Основными производителями SLIC являются:

- Lantiq;
- Microsemi;
- Nuvoton.

1.2 Микросхема SLIC производства фирмы lantiq

Компания Lantiq образовалась в ходе продажи подразделения проводной связи Golden Gate Capital фирмой Infineon Technologies в 2009 году. Компания производит изделия, использующиеся для сетей как домашних, так и общедоступных.

1.2.1 Технические характеристики DuSLICxT PEF3201

Основные Технические характеристики DuSLICxT PEF3201:

- полностью программируемый кодек и SLIC системное решение с расширенными возможностями обработки сигналов;
- минимизированное количество внешних компонентов для обеспечения высокой плотности конструкции платы;
- низкая стоимость материалов.

1.2.2 Параметры микросхемы SLIC производства фирмы Lantiq

Основные параметры микросхемы SLIC:

- рабочие параметры передачи переменного тока программируется в соответствии с ITU T Q.522 для использования во всем мире (специфическое для страны программирование для параметров таких как согласование импеданса переменного тока, гибридный баланс, передача и получение усиления, частотная характеристика) в соответствии с ITU T Q.522;
- интегрированная сбалансированная и несбалансированная возможность звона–программное обеспечение, программируемое до 144 в пиковый звон напряжение (в зависимости от внешних компонентов), диапазон частот от 15 до 50 Гц;
- сигнализация начала цикла;
- наземная сигнализация старта;
- перемена полярности;
- быстрое обнаружение отключения питания;
- автоматическое кольцевое регулирование тока;
- звон со смещением постоянного тока;
- поддержка индикации ожидания сообщения с лампой накаливания;
- ИКМ интерфейс G711 A-law.

1.2.3 Поддерживаемые режимы сигнализации

Режимы сигнализации:

- два встроенных программируемых генератора тонов, для генерации тестовых и DTMF тонов;
- встроенный генератор с поддержкой высокого уровня кадрирования;
- интегрируемая генерация измерительных импульсов;
- встроенный генератор звонка с программируемой амплитудой;
- приемник DTMF.

1.2.4 Интерфейс связи микросхемы с АТС

Последовательный интерфейс управления SCI, совместимый с PCI.

					ПГУ 2.090301.25.001 ПЗ	Лист
						10
Изм.	Лист	№ докум	Подпись	Дата		

1.2.5 Дополнительные параметры

Дополнительные параметры микросхемы SLIC включают в себя:

- DC/DC контроллер преобразования питания ТА;
- минимизация рассеиваемой мощности за счет регулирования требуемого напряжения питания линии;
- режим ожидания с низким энергопотреблением и возможностью обнаружения отключения;
- механизм перевода преобразователя для питания телефона в спящий режим.

1.3 Микросхема SLIC производства фирмы Microsemi

Американская корпорация являющееся одним из главных поставщиков полупроводников для оборонной промышленности и связи. В 2018 году компания Microchip приобрела компанию Microsemi.

Компания Microsemi выпускает одноканальный линейный модуль FXS ZLR964321L, предназначенный для взаимодействия с телефонией Microsemi.

1.3.1 Основные характеристики ZLR964321L:

- один порт FXS с доступными профилями для работы по всему миру;
- поддержка каденционного звонка для 85 Vpk;
- защита от перенапряжения/перегрузки по току;
- 12В входное напряжение – низкая стоимость батареи;
- датчик постоянного тока находится вне защиты, чтобы обеспечить поддержку для испытаний на внешнее напряжение;
- компактный размер примерно 6см² на канал FSX.

1.3.2 Параметры микросхемы SLIC производства фирмы Microsemi

Основные параметры микросхемы SLIC:

- ИКМ: G.711 u-law, A-law и 16-битная линейный;
- режим ИКМ/SPI ;

- программируемые характеристики питания постоянным током;
- сигнализация начала цикла;
- наземная сигнализация старта;
- комплект BORSCHT функций;
- сбалансированный и несбалансированный звонок до 85 V_{pk};
- два встроенных программируемых генератора тонов, для генерации тестовых и DTMF тонов.

1.4 Микросхема SLIC производства фирмы Nuvoton

Компания Nuvoton Technology, ведущий поставщик полупроводниковых компонентов для потребительских и вычислительных применений, сообщила о расширении серии устройств нового поколения Pro-X (от PROgrammable Extended CODEC/SLCC, т.е. расширенная программируемая микросхема CODEC/SLCC). Новым представителем серии стала интегральная схема для управления абонентскими линиями (SLCC) N681622. Она реализует порт FXS и способна напрямую работать с недавно представленными Nuvoton одно- и двухканальными микросхемами Pro-X CODEC/SLCC.

1.4.1 Особенности

- 1) Комплект BORSCHT функций;
- 2) сбалансированный и несбалансированный звонок до 90 V_{pk};
- 3) встроенные функции управлением питания:
 - а) интегрированный регулятор DC/DC регулирует напряжение батареи тока для того чтобы уменьшить диссипацию силы во всех режимах работы;
 - б) программируемое внешнее переключение батареи;
- 4) программируемые характеристики линейной подачи:
 - а) частота и амплитуда звонка;

- б) двухпроводное сопротивление переменного тока Постоянный ток питания (от 20 до 41) мА.
- с) обнаружение ключа замыкания;
- 5) программируемая генерация и обнаружения сигнала:
 - а) DTMF генерация;
- 6) поддержка тестирования и диагностики:
 - а) интегрированные режимы замыкания на себя;
 - б) мониторинг подачи строк в реальном времени;
 - с) встроенный датчик температуры;
- 7) цифровой интерфейс:
 - а) ИКМ: G.711 u-law, A-law и 16-битная линейный;
 - б) программируемое усиление звукового пути.

1.5 Сравнение микросхем модуля FSX

Сравнение микросхем использующихся в разработке портов интерфейса FXS разных производителей приведено в таблице 1.

Таблица 1 – Сравнение микросхем модуля FXS

Критерий сравнения	(PEF3201)Lantiq	(ZLR964321L)Microsemi	(N681622) Nuvoton
Звонок	144Vpk	85Vpk	90Vpk
Поддержка G711	A-law	u-law, A-law	u-law, A-law
Потребляемая мощность	1)активный режим <700 мВ 2)режим звонка <680 мВ (3REN) 3)на низкой мощности <80 мВ	1)активный режим <600 мВ 2)режим звонка <700 мВ (3REN) 3)на низкой мощности <140 мВ	1)активный режим <700мВ 2)режим звонка >800мВ (3REN) 3)на низкой мощности> 140 мВ.
Двухканальный	да	нет	нет
Индикации сообщения	есть	нет	Нет

Продолжение таблицы 1

Критерий сравнения	(PEF3201)Lantiq	(PEF3201)Lantiq	(N681622) Nuvoton
Установочные размеры	– однопакетное решение – 256 мм ²	– однопакетное решение – 200 мм ²	– мульти–чип решение – 350 мм ²
Температурный диапазон	от –50 до +90	от –40 до +85	от –40 до +85
Функционал BORSCHT	полный функционал	полный функционал	полный функционал
Рыночная позиция	– >47% рыночной доли	– 25% рыночной доли	– 14% рыночной доли
DTMF генерация	да	да	да

Сравнив характеристики в таблицы №1, для ВКР была выбрана микросхема SLIC PEF3201 фирмы Lantiq, так как он имеет больший функционал.

1.6 Алгоритм работы узла FXS

В работе рассмотрим микросхему DuSLICxT PEF3201. Микросхемы комплекта выполняют многие функции:

- Подача питания в абонентскую линию;
- обнаружением снятия микротелефона с рычажного переключателя ТА (замыкания шлейфа);
- прекращением посылки вызова на ТА при ответе абонента (замыкании шлейфа);
- регулировкой уровня речевого сигнала;
- ограничением тока питания и переходом с двухпроводной линии на четырехпроводную (дифсистема).

DuSLIC/SLIC контролирует и ограничивает подаваемое в линию постоянное напряжение и ток питания до достаточных для нормального действия значений. Микросхема также осуществляет наложение речевого сигнала от кодека на постоянное напряжение в линии, а также обеспечивает прием речевого сигнала из линии и подачу его на кодек. Номинальное напряжение питающей батареи составляет –60В относительно земли. Напряжение питающей батареи (–60В) подается на выходные буферы DuSLIC/SLIC. Выходные напряжения в абонентскую линию могут варьироваться в широких пределах от 5 до 75В. Микросхемы абонентских комплектов DuSLIC/SLIC используют протоколы SPI и PCM, имеют встроенные цепи защиты от повышенного тока, перегрева и перенапряжений. В качестве дополнительной защиты на каждый провод абонентской линии установлен терморезистор и защитные тиристоры, отключающие выходы микросхемы при превышении порога максимального напряжения (более 60В/48В).

1.6.1 Установление исходящего соединения

В исходном состоянии микросхема DuSLIC/SLIC находится в «спящем» режиме, т.е. выходные буферы отключены, но питание в линию подается через встроенные балластные резисторы. Снятие трубки ТА приводит к возникновению тока в абонентской цепи, что приводит к срабатыванию датчика снятия трубки и выходу микросхемы из «спящего» режима. Подключается разговорный тракт до платы ЦП, в котором начинает подаваться сигнал «Ответ станции». Набор номера производится процессором ЦАТС, который воспринимает как импульсный набор, так и набор DTMF (может отключаться программно).

1.6.2 Установление входящего соединения

При установлении входящего соединения по команде ЦП абоненту подается индукторный вызов путем перевода микросхемы DuSLIC/SLIC в состояние RING. Причем характер и длительность импульсов задаются ЦП, а

					ПГУ 2.090301.25.001 ПЗ	Лист
						15
Изм.	Лист	№ докум	Подпись	Дата		

вызывной сигнал генерируется микросхемой PEF3201. Микросхема автоматически определяет поднятие абонентом трубки и переходит в режим разговора.

1.6.3 Тестирование абонентского комплекта

Для измерения параметров абонентских линий и внутренних цепей в комплекте DuSLIC/SLIC предусмотрена измерительная функция. Причем, комплектом может измеряться как напряжение, так и ток. Управление измерителем осуществляется микроконтроллером по команде ЦП. Для измерения различных параметров линии используются разные методики (постороннее напряжение, сопротивление, сопротивление изоляции, емкость).

1.7 Описание узла интерфейса FXS

Данный узел является вспомогательный для работы АТС и сопряжения различных типов устройств.

FXS интерфейсы (модуль FXS на IP PBX) или FXS–порты (абонентские порты на PBX) служат для соединения аналоговых устройств, таких как аналоговые телефоны, факсы, модемы. Самым распространённым устройством с FXS портом можно считать АТС. АТС как правило имеет не только порт FXS, но и другие порты для соединения с разными сетями.

На рисунке 1 показано взаимодействие узла FXS в разных системах.

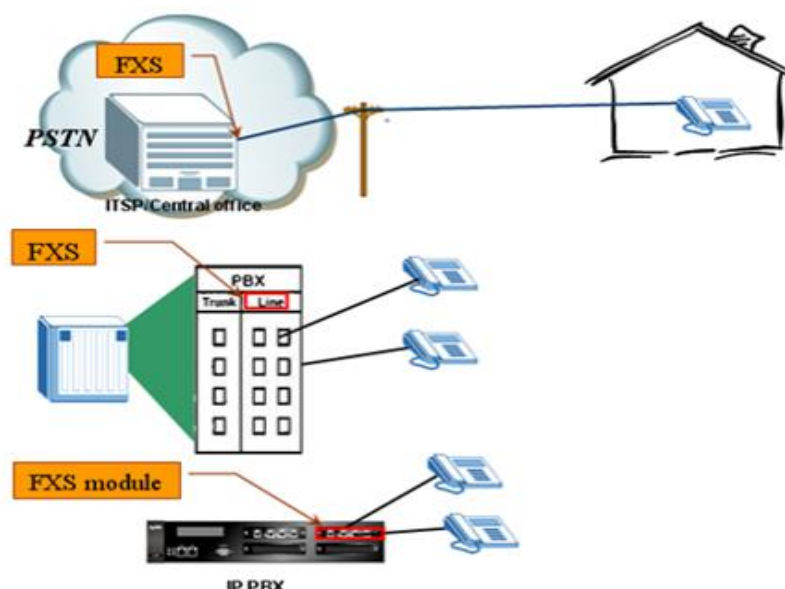


Рисунок 1 – взаимодействие узла FXS

Сам интерфейс FXS является эмуляцией АТС, к которой подключены телефонные аппараты. Устройства на FXS – портах выполняют:

- питание ТА;
- определения положения телефонной трубки;
- наличие детекторов цифр, набранных на ТА;
- формирование напряжения на ТА при приходе вызова с IP–сети.

Одной из проблем FXS–порта является то, что у него нет номеронабирателя. Решением этой проблемы является порт FXO, который может осуществить набор номера в телефонную линию, определение сигнала, приходящего от АТС, занятие и освобождение линии связи.

Исходящий вызов происходит в 2 этапа, в которых используются оба порта FXS и FXO:

- Абонент поднимает трубку (устройство FXO). Порт FXS устанавливает, что трубка поднята;
- в виде двухтанального сигнала набирается номер, который отправляется на порт FXS.

Входящий вызов происходит в 3 этапа в которых так же задействованы оба порта:

- Порт FXS приняв входящий вызов, посылает его с необходимым напряжением на сопряженный с ним порт FXO;
- ТА звонит;
- сняв трубку телефона можно разговаривать.

Прекращение разговора происходит портом FXS по сигналу от порта FXO.

1.8 Интерфейс TAPI

Драйвер устройства SLIC – это программная модель, позволяющая завершить SLIC с помощью Lantiq TAPI для продуктов CPE. Двоичный файл драйвера устройства включает в себя реализацию SLIC уровня TAPI Low Level.

В дополнение к поддержке уровня TAPI Low Level драйвер устройства SLIC предоставляет прямые интерфейсы для конкретных устройств, которые обеспечивают управление SLIC.

1.9 Протокол SIP

Протокол SIP, применяется для установки соединения между двумя аппаратами, а также реагирования на любые изменения в сети.

На рисунке 2 показан процесс соединения двух аппаратов. Соединение устанавливается, когда первый пользователь (Игорь) снимет трубку и наберет номер, в этом случае SIP клиент сгенерирует сигнал приглашения, у второго пользователя (Маша) зазвонит телефон, его клиент выдаст сообщение звонка, затем Маша берет трубку и клиент выдаст сообщения ОК, Игорь передаст сигнал подтверждения и начнется разговор. По завершению звонка один из клиентов посылает сообщения окончания.

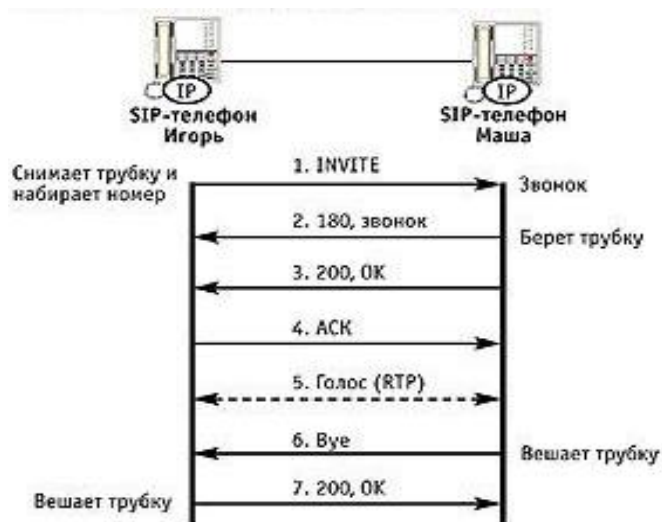


Рисунок 2 – соединение двух аппаратов

Такой подход не применим, когда в сети более двух клиентов. В этом случае SIP-сетям необходима инфраструктура, которую представляют сервера SIP. Сервер регистрации (registrar) отвечает за регистрацию и авторизацию пользователей. За определение местоположения отвечает сервер локализации (allocation). Redirect-сервер переадресации переводит звонки на фактическое место пребывания. Наиболее сложные функции ложатся на прокси-сервер (SIP Proxy), отвечающий за взаимодействие внутренних клиентов (например, учрежденческая связь) с внешним миром – именно он определяет все политики, правила общения и т. д. На рисунке 3 показан принцип работы протокола SIP в сети предприятия.

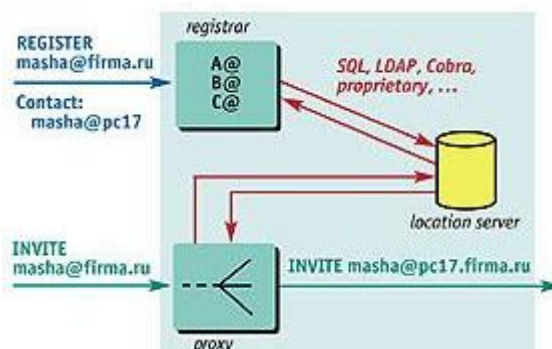


Рисунок 3 – взаимодействие в сети

Пользователь, придя на новое рабочее место включает в корпоративную сеть программный телефон, который автоматически регистрируется на

сервере. Сервер запрашивает информацию о пользователе в корпоративной базе данных и сообщает как с ним контактировать серверу локализации. После этого, когда кто-то позвонит пользователю, прокси-сервер запросив сервер локализации установит связь с рабочим местом пользователя. Протокол SIP – это потоки пакетов RTP, передающие голосовые и видео данные.

1.10 Общие сведения о протоколе RTP

Протокол RTP используется при передаче трафика реального времени. Протокол был разработан Audio–Video Transport Working Group в 1996 году.

Протокол RTP переносит в своём заголовке данные, необходимые для восстановления голоса или видеоизображения в приёмном узле, а также данные о типе кодирования информации (JPEG, MPEG и т. п.). В заголовке данного протокола, в частности, передаются временная метка и номер пакета. Эти параметры позволяют при минимальных задержках определить порядок и момент декодирования каждого пакета, а также интерполировать потерянные пакеты.

RTP не имеет стандартного зарезервированного номера порта. Единственное ограничение состоит в том, что соединение проходит с использованием чётного номера, а следующий нечётный номер используется для связи по протоколу RTCP.

1.10.1 Описание протокола RTP

RTP был разработан как протокол реального времени, из конца в конец (end-to-end), для передачи потоковых данных. Джиттер определяется разницей в суммарной задержке между сторонами. Передающая сторона помечает пакет временной меткой, принимающая получает пакет и вычисляет задержку. Таким образом, возможно снизить влияние джиттера установив выдачу пакетов с постоянной задержкой.

Еще одной функцией RTP является потеря пакетов при прохождении по сети. Это выражается в кратковременной тишине в трубке, что негативно влияет на разговор сторон. Поэтому RTP скрывает их «приятным шумом».

					ПГУ 2.090301.25.001 ПЗ	Лист
						20
Изм.	Лист	№ докум	Подпись	Дата		

RTP работает в паре с протоколом RTCP, который выполняет сбор статистики, определяет качество обслуживания, а также синхронизацию потоков RTP–сеанса.

Основной функцией RTCP является установление обратной связи с приложением для отчета о качестве получаемой информации. Участники RTCP сессии обмениваются сведениями о числе полученных и утраченных пакетов, значении джиттера, задержке и т.д. На основе анализа этой информации принимается решение об изменении параметров передачи, например, для уменьшения коэффициента сжатия информации с целью улучшения качества ее передачи.

Для выполнения этих функций RTCP передает специальные сообщения определенных типов:

- SR – Sender Report – отчет источника со статистической информацией о RTP сессии;
- RR – Receiver Report – отчет получателя со статистической информацией о RTP сессии;
- SDES – содержит описание параметров источника, включая name (имя пользователя);
- BYE – Иницирует завершение участия в группе;
- APP – Описание функций приложения.

RTP – это протокол однонаправленного действия, поэтому для организации двусторонней связи необходимо две RTP сессии.

RTP – сессия определяется IP адресами участников и парой не использующихся UDP портов из диапазона 16384 – 32767. Для организации обратной связи с приложением необходимо также установить двустороннюю RTCP сессию. Для RTCP сессии занимают порты с номером на единицу большим чем RTP. Так, например, если для RTP выбран 19556 порт, то RTCP сессия займет 19557 порт. Наглядно формирование RTP/RTCP сессии представлено на рисунке 4.



Рисунок 4 – формирование RTP/RTCP сессии

Сам протокол RTP не имеет возможности устанавливать или разрывать сессию, это выполняют протоколы сигнализации, такие как протокол SIP.

1.11 Звуковые кодеки

Важной чертой кодеков является на сколько сложен алгоритм кодирования, связан с вычислениями в реальном времени. Сложность определяет скорость обработки. От сложности обработки зависит мощность потребления, стоимость устройства, а также размеры устройства.

На качество воспроизводимой речи влияет размер кадра: чем длиннее кадр, тем более эффективно моделируется речь. Но также чем больше кадр, тем больше и задержка на его обработку. Как раз, размер кодека определяется компромиссом между этими требованиями.

Временная задержка увеличивается с увеличением размера кадра, а также с увеличением сложности алгоритма кодирования. При передаче речи допустимая задержка в одном направлении не может быть больше 250 мс.

В таблице 2 приведены характеристики кодеков аудио–кодирования серии G.

Таблица 2 – характеристики кодеков серии G

Кодек	Тип кодека	Скорость кодирования, кбит/с	Задержка при кодировании, мс	Размер кадра, мс	Сложность реализации
G.711	ИКМ		0.75	0.125	–
G.726	АДИКМ			0.125	8 MIPS
G.728	LD–CELP		3–5	0,625	40 MIPS
G.729	CS–ACELP				30 MIPS
G.729 a	CS–ACELP				20 MIPS

Продолжение таблицы 2

Кодек	Тип кодека	Скорость кодирования, кбит/с	Задержка при кодировании, мс	Размер кадра, мс	Сложность реализации
G.723.1	MP- ML0	6.3			16 MIPS
G.723 1	ACELP	5.3			16 MIPS

Качество кодирования голоса производится с помощью усредненного совокупного мнения по 5 – балльной шкале. Оценки характеризуются следующим образом:

- 4–5 – высокое качество;
- 4,5–4 – качество ТфОП;
- 3 –3.5 – удовлетворительное (ухудшение заметно на слух);
- 2,5–3 – речь разборчива, но требует концентрации для понимания.

В таблице 3 приведены результаты оценки кодеков серии G.

Таблица 3 –результаты оценки кодеков серии G

Кодек	Оценка
G.711	4.2
G.726	4.3
G.728	4.1
G.729	4.0

Для передачи пользователям дополнительной информации при активном подключении используется кодирование и передача сигналов DTMF. Для передачи неискаженного сигнала DTMF используется кодек G.711, который не накладывает ограничения на вид сигнала.

Искаженные сигналы DTMF должны подавляться во избежание возникновения эффектов слияния и раздвоения сигналов

Для систем IP–телефонии наиболее целесообразным и естественным является применение кодеков с переменной скоростью кодирования речевого сигнала. В основе кодека речи с переменной скоростью лежит классификатор входного сигнала, определяющий степень его информативности и задающий на основе этого метод кодирования и скорость передачи речевых данных. Наиболее простым классификатором

речевого сигнала является VAD, который выделяет во входном речевом сигнале активную речь и паузы. При этом, фрагменты сигнала, классифицируемые как активная речь, кодируются каким-либо из известных алгоритмов (как правило, на базе метода CELP с типичной скоростью 4–8 кбит/с. Фрагменты, классифицированные как паузы, кодируются и передаются с очень низкой скоростью (порядка 0.1–0.2 кбит/с), либо не передаются вообще. Передача минимальной информации о паузных фрагментах предпочтительна. Использование в голосовом шлюзе технологии VAD приводит к экономии полосы пропускания при некотором ухудшении разборчивости речи.

Схемы более эффективных классификаторов входного сигнала детальнее осуществляют классификацию фрагментов, соответствующих активной речи. Это позволяет оптимизировать выбор стратегии кодирования (скорости передачи данных), выделяя для особо ответственных за качество речи участков речевого сигнала большую скорость, для менее ответственных – меньшую скорость. При таком построении кодеков могут быть достигнуты низкие средние скорости (2–4 кбит/с) при высоком качестве синтезируемой речи.

На практике шлюзы IP – телефонии поддерживают, как правило, несколько кодеков. Шлюз определяет тип передаваемого сигнала (речь, факс или модем) и переключает его на соответствующий режим кодирования с целью обеспечения экономии ширины полосы пропускания.

В таблице 4 приведены значения коэффициентов сжатия и ширины полосы пропускания различных кодеков.

Таблица 4 – коэффициенты сжатия и ширина полосы пропускания кодеков серии G.

Кодек	Диапазон коэффициента сжатия	Рассматриваемый коэффициент сжатия	Полоса пропускания, кбит/с	Полоса пропускания с учетом подавления пауз, кбит/с
G. 711 μ- закон	1 – 4		84.80	46.59

Продолжение таблицы 4

Кодек	Диапазон коэффициента сжатия	Рассматриваемый коэффициент сжатия	Полоса пропускания, кбит/с	Полоса пропускания с учетом подавления пауз, кбит/с
G. 711 А-закон	1 – 4		84.80	69.52
G. 726	1 – 9		37.69	37.69
G. 729a	1 – 8		14.13	12.12

Недостатком кодеков серии G является то, что для корректного восстановления речи нужно знать предыдущие отчеты. В случае потери части пакетов, то восстановление займет большее время (зависит от количества потери пакетов), в течении чего голос не будет корректен изначальному. Поэтому качество передаваемой речи зависит от качества используемых каналов передачи. Качество речи остается относительно высоким при равномерно распределенных потерях пакетов (сплошные линии на рисунке 5), но резко ухудшается при увеличении потерь пакетов (штриховые линии).

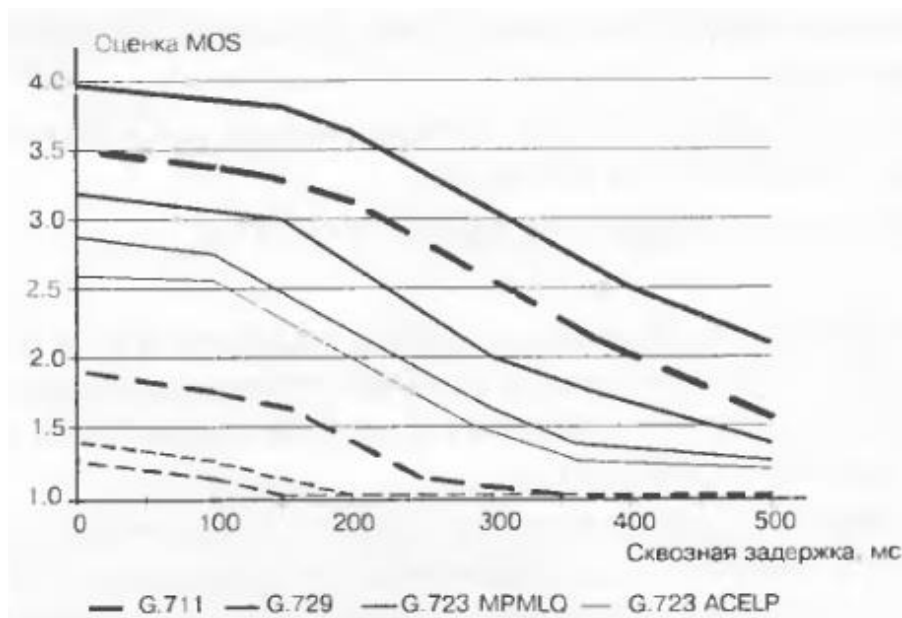


Рисунок 5 – сравнение кодеков (5 потерь пакетов)

1.11.1 Кодек G.711

Кодек G.711, использующий ИКМ–преобразование аналогового сигнала с точностью 8 бит, тактовой частотой 8 кГц и простейшей компрессией амплитуды сигнала. Скорость потока данных на выходе преобразователя составляет 64 кбит/с (8 бит, 8 кГц). Для снижения шума квантования и улучшения преобразования сигналов с небольшой амплитудой при кодировании используется нелинейное квантование по уровню согласно специальному псевдологарифмическому закону A или μ .

μ –закон кодирования предпочтительнее использовать при малой амплитуде сигнала и малом отношении сигнал/шум. При международной связи требуется преобразование μ –закона в A –закон, производимое на стороне применения μ –закона.

Кодек G.711 широко распространен в системах традиционной телефонии с коммутацией каналов. Кодек G.711 является минимально необходимым для оборудования VoIP.

Недостатком кодека являются высокие требования к полосе пропускания и задержки в канале передачи, вследствие чего в системах IP–телефонии он используется редко. Использование G.711 в системах IP–телефонии обосновано лишь в тех случаях, когда требуется обеспечить максимальное качество кодирования речевой информации при небольшом числе одновременных разговоров.

1.11.2 Работа кодека

В сущности, во всех кодеках используется принцип аппроксимирования, приведенный в иллюстративной форме на рисунке 6.

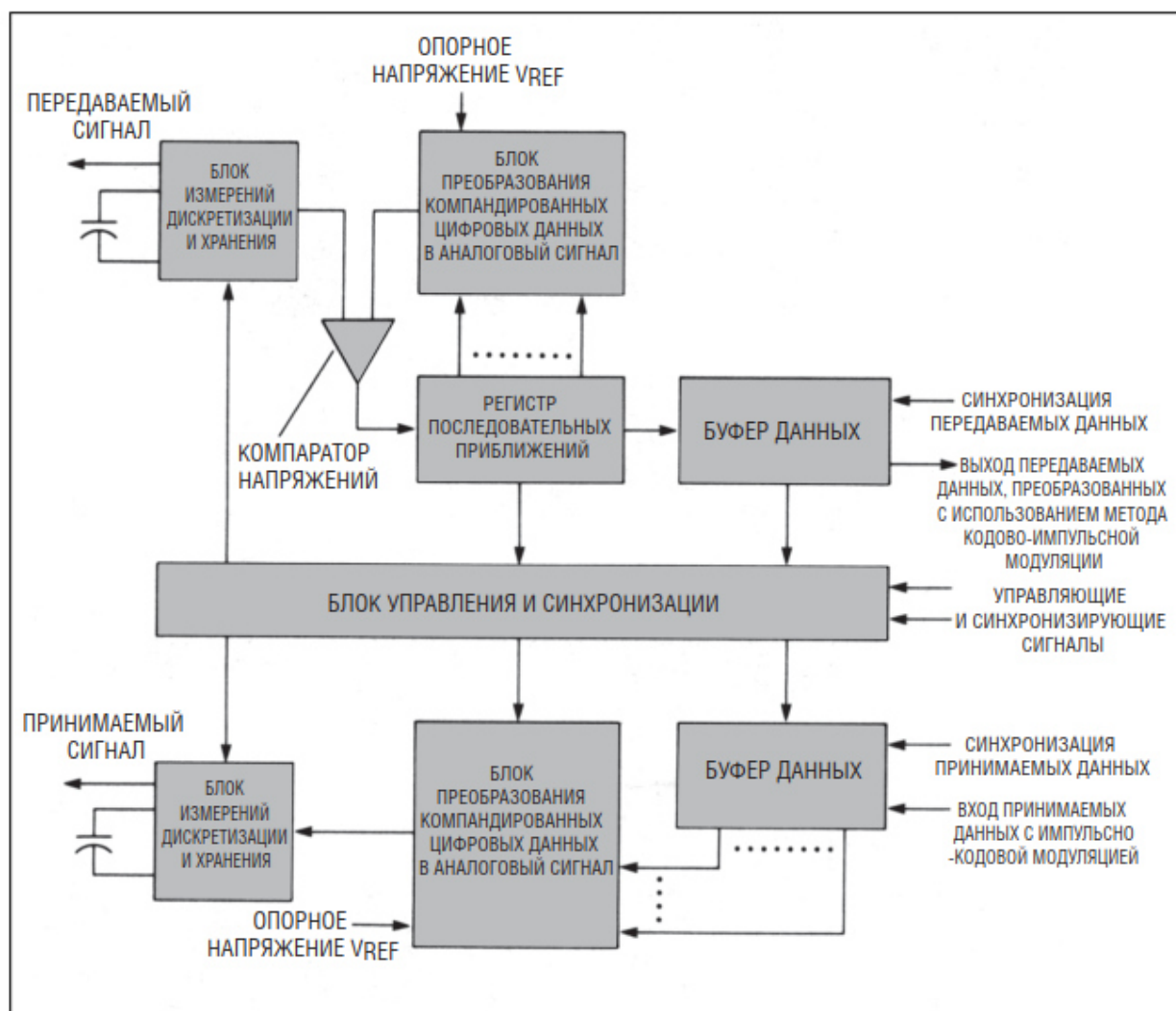


Рисунок 6— кодек ,обслуживающий отдельный канал связи

В направлении передачи сигнала сигнал абонента проходит дискретизацию в усилителе, также выполняющем функции дискретизатора—синхронизатора, кодируется с использованием компаратора напряжения и уплотняющего ЦАП, а также соответствующего аппроксимирующего регистра.

Результирующие сжатые данные в двоичной форме под управлением схемы синхронизации передаваемых данных загружаются в буфер данных, перемещаются в соответствующий временной ИКМ—слот. При приеме передаваемого сигнала информация в цифровом виде поступает по абонентской линии через другой буфер в расширяющий (для сжатых данных) ЦАП и синхронизируется в буфере синхронизирующим генератором принимаемых данных. Уплотняющий ЦАП, действующий как

аналого–цифровой преобразователь, вырабатывает напряжение, которое запоминается в приемном усилителе дискретизаторе–синхронизаторе, а затем поступает через восстанавливающий низкочастотный фильтр для сглаживания.

В базовой конфигурации кодека используется цифровая техника для выполнения уплотнения (командирования), цифро–аналогового и аналого–цифрового преобразования и т.д., и все это – при соблюдении условия точной синхронизации.

Кодирование с командированием, или нелинейное командирование, приведенное на рисунке 7, позволяет добавить ряд существенных деталей.

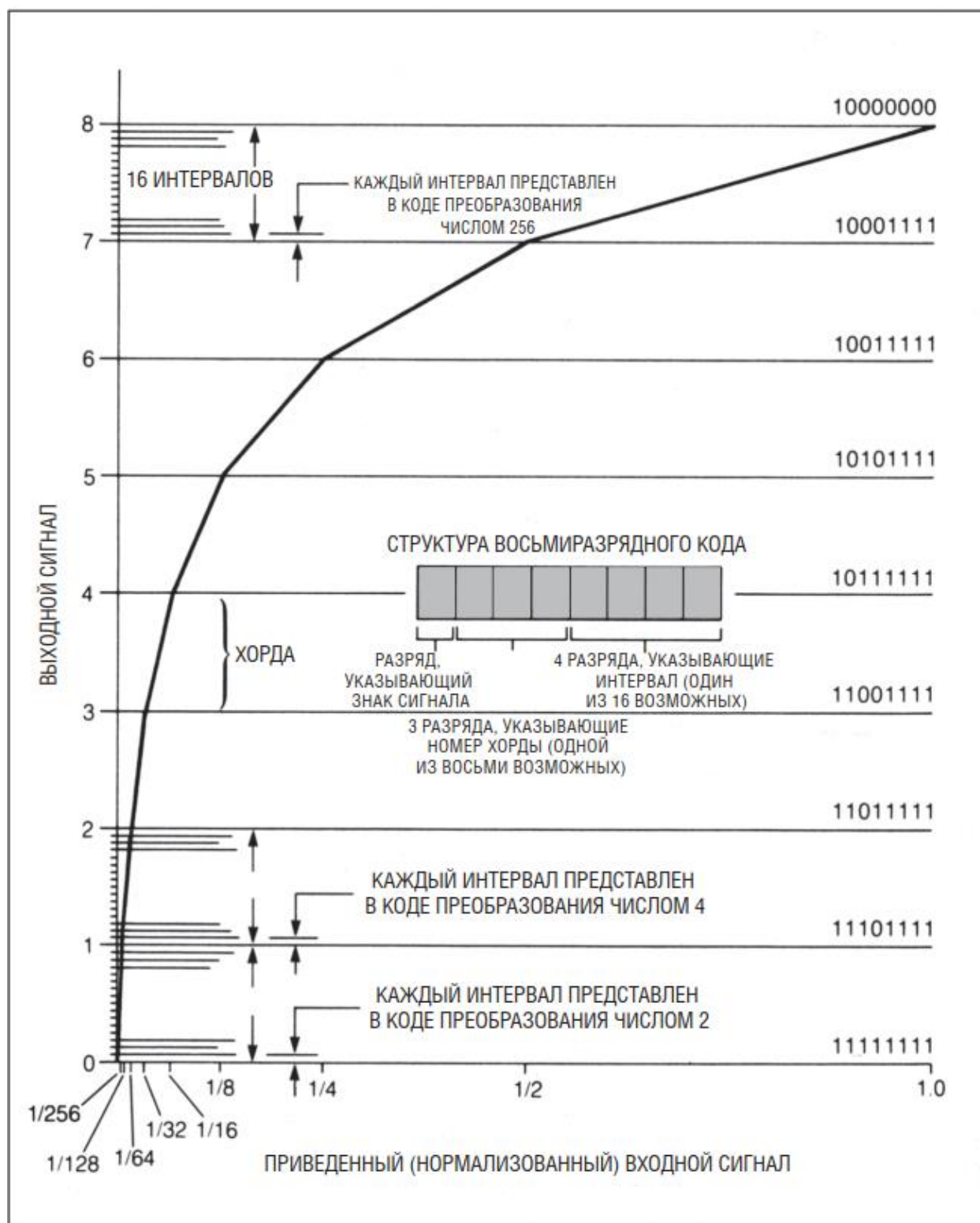


Рисунок 7 – процесс компандирования (нелинейного сжатия), выполняемого с использованием μ -характеристики

Кривая μ -характеристики приведенного (нормализованного) входного сигнала для положительных значений в зависимости от выходного напряжения используется в качестве примера потому, что является самой распространенной на территории России. Для использования на практике

характеристика в кодеке представляется в виде отдельных сегментов, представляющих линейно–кусочную аппроксимацию характеристики.

Весь диапазон значений выходного сигнала делится на 16 сегментов, называемых хордами, – 8 хорд для положительных значений сигнала и 8 хорд для отрицательных значений. Каждая хорда разбивается на 16 интервалов. Каждый интервал в любой конкретной хорде представлен в коде выходного сигнала одним и тем же количеством разрядов, однако при увеличении величины выходного сигнала, когда он располагается в хорде с более высоким номером, при кодировании интервалам присваивается более высокий номер. Как показано для второго сегмента выходного сигнала (хорда номер 2), код выходного сигнала возрастает в 4 раза для каждого интервала данного сегмента, тогда как для восьмого сегмента выходного сигнала для каждого интервала код выходного сигнала возрастает в 256 раз. Как показано на рисунке 7, полный код процесса компандирования представляет комбинацию (двоичных) цифр из 8 разрядов. Один разряд предназначен для обозначения знака сигнала (плюс или минус), 3 разряда предназначены идентифицировать хорду и 4 оставшихся разряда – для идентификации интервала.

Для выполнения операции кодирования с компандированием в кодеке при дискретизации сигнала используется μ –характеристика. Увеличение величины каждого интервала с уровнем сигнала происходит с увеличением числового значения кода выходного 8–разрядного сигнала. Пошаговое увеличение является линейным в пределах каждого сегмента (хорды) зависимости, однако величина шага возрастает по логарифмическому закону для каждого нового сегмента.

1.12 Интегральная микросхема PSB 21391H

Интегральная микросхема PSB 21391H компании Lantiq представляет законченный компандирующий кодек с μ -характеристикой. Она разрабатывалась в качестве кодека с импульсно-кодовой модуляцией и обладает полным набором функций, необходимых для создания интерфейса с четырехпроводной телефонной линией, предназначенной для работы в качестве стандартной цифровой линии передачи с временным разделением, такой, например, как линия T1. Каждая из присущих интегральной микросхеме PSB 21391H функций может быть обнаружена на блок-схеме, приведенной на рисунке 8.

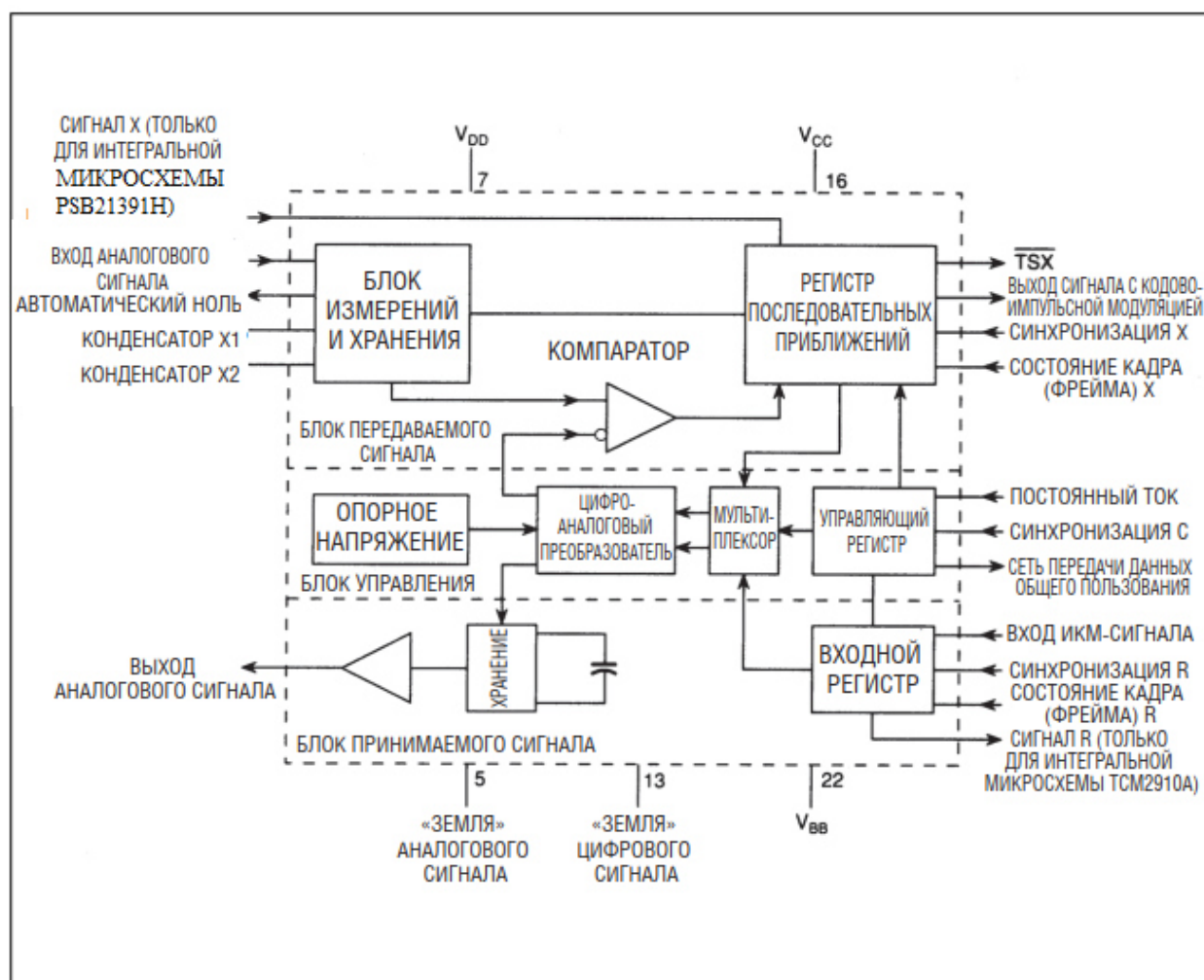


Рисунок 8 – блок-схема интегральной микросхемы PSB21391H

Данная интегральная микросхема совместима с акустическим фильтром PEF 24901H.

Микрокристалл интегральной микросхемы PSB 21391H может быть разбит на три отдельных области или блока: блок передаваемого сигнала, блок управления и блок принимаемого сигнала. Схема эталонного или опорного напряжения блока управления обеспечивает точное значение эталонного напряжения для схемы ЦАП. ЦАП выполняет две важнейшие функции в схеме кодека. Прежде всего, он преобразует и расширяет сжатые цифровые ИКМ данные в аналоговый выходной сигнал, который поступает на фильтр. Во-вторых, схема ЦАП также используется в качестве части схемы аналого-цифрового преобразователя, которая преобразует отфильтрованный аналоговый сигнал, поступающий из абонентской линии связи, в сжатые с использованием ИКМ данные, предназначенные для передачи. Другая схема управления переключает кодек из режима приема сигнала в режим передачи.

В интегральную микросхему PSB 21391H также встроены две дополнительные функции. Внешняя схема контроля энергопотребления позволяет отключаться кодеку, когда он не используется. В больших по размерам системах это может обеспечить существенную экономию электроэнергии. Также предусмотрен внешний контроль распределения временных слотов (интервалов). Такая гибкость схемы позволяет микропроцессору управлять синхронизацией принимаемых и передаваемых кодовых групп в кодеке. На рисунке 9 приводится схема соединений между интегральными микросхемами PSB 21391H и PEF 24901H.

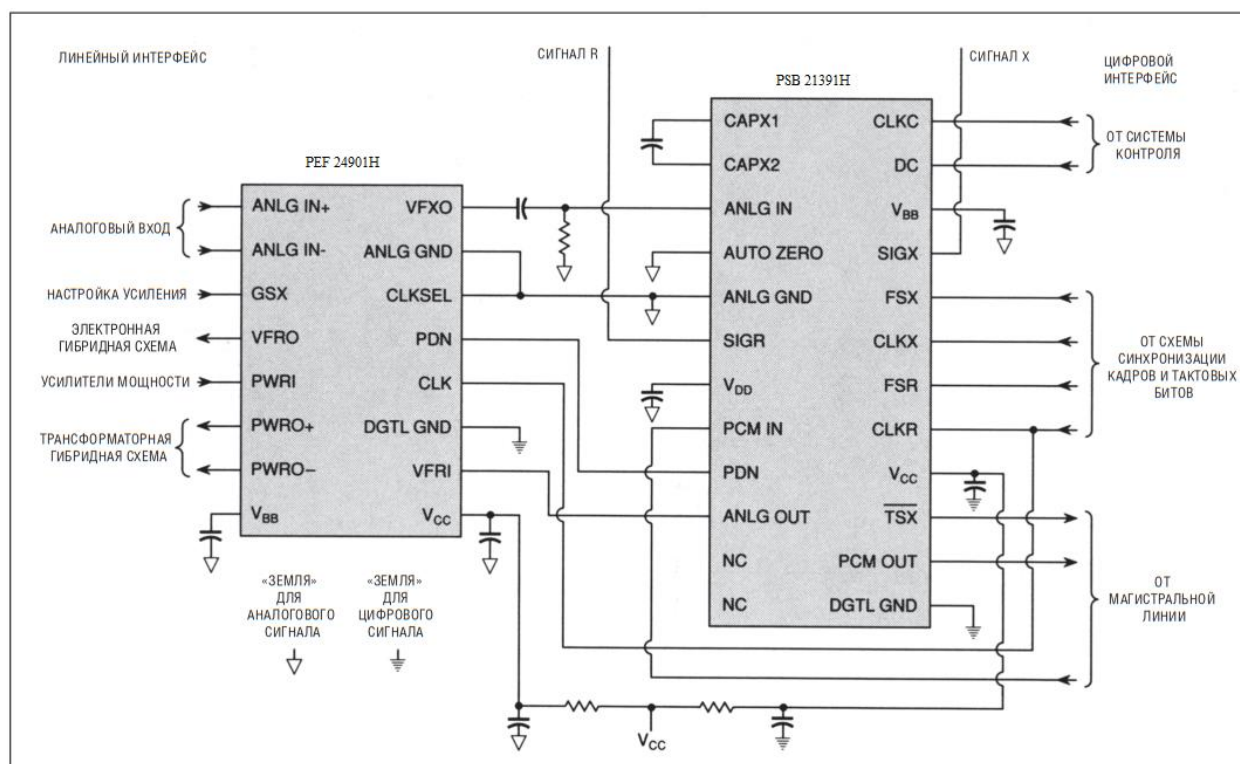


Рисунок 9 – взаимодействие кодека и фильтра

1.13 Электронные схемы в интерфейсах абонентских телефонных линий

Наибольший интерес в использовании телефонной станции представляют ее схемы (или платы) интерфейса абонентских линий связи (или абонентского интерфейса – в оригинале книги используется англоязычная аббревиатура SLIC) и электронные схемы коммутации. Коммутационная часть телефонной станции будет обсуждена в этой главе чуть позже, а сейчас для иллюстрации различных способов реализации функций абонентского интерфейса рассмотрим схему абонентского интерфейса станции, приведенную на рисунке 10.

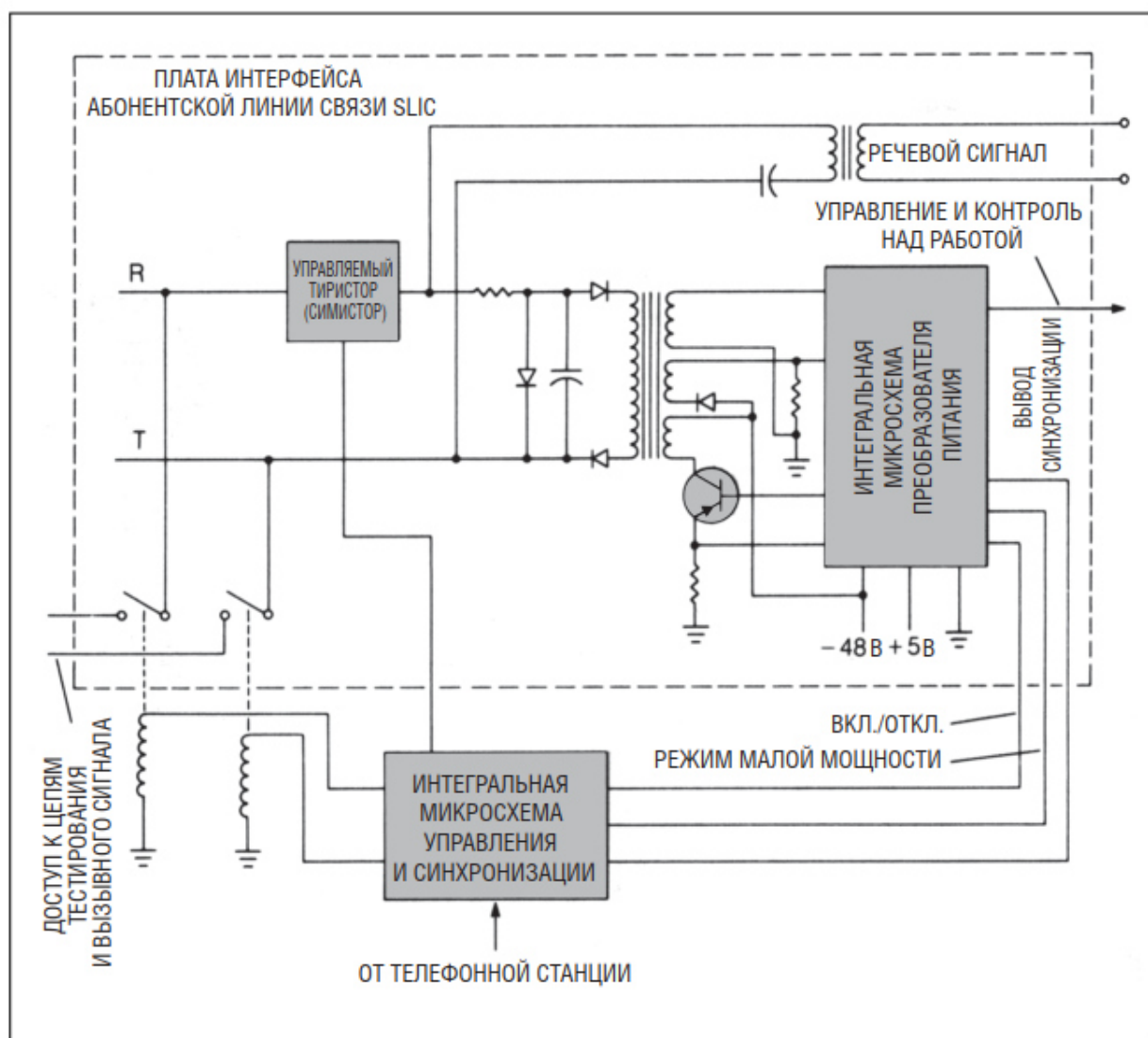


Рисунок 10 – схемы абонентского интерфейса (SLIC) телефонной станции с дистанционной системой коммутации .

Схема абонентского интерфейса телефонной станции занимает промежуточное положение между полностью законченной твердотельной интегральной микросхемой и схемой, в которой совершенно не используются электронные компоненты, поскольку в ней использована комбинация электронных элементов с трансформаторами, что позволяет данной схеме интерфейса обеспечить:

- Очень малое потребление мощности, составляющее 650 мВт из расчета на одну абонентскую линию;

- изоляцию схемы от прохождения синфазных сигналов абонентской линии за счет использования трансформаторной связи с преобразователями мощности (силовыми);
- защиту от высоковольтных скачков напряжения, возникающих при переходных процессах, путем совместного использования в схеме диодов и трансформаторов за счет низкочастотной характеристики трансформаторов;
- два режима работы: при высокой и малой мощности. В режиме высокой мощности рабочая частота стабилизатора мощности изменяется с 40 на 90 Гц в зависимости от сопротивления линии. Это позволяет изменять выходное напряжение в зависимости от длины абонентского шлейфа. В режиме малой мощности стабилизатор обеспечивает только питание, необходимое для протекания тока, компенсирующего потери в абонентской линии;
- использование напряжения вызывного сигнала;
- доступ к линии для проведения тестирования.

Схемы интерфейса абонентских телефонных линий SLIC и электронное коммутационное оборудование телефонных станций с электронной коммутацией обеспечивают наилучшие результаты от совместного применения элементов твердотельной электроники и более традиционных методов электротехники.

Во время прохождения вызывного сигнала и тестирования абонентской линии питающий стабилизатор мощности и источник питания для речевого сигнала должны быть отключены от линии. Это осуществляется установкой стабилизатора и размыканием отключающего симистора (триака). Напряжение вызывного сигнала с частотой 20 Гц подается с общей шины вызывного сигнала через вибрационные реле. Вибрационные реле также позволяют осуществлять тестирование. Работа этих реле управляется системой коммутации с удаленным доступом с использованием фиксаторов

и релейных драйверов интегральной микросхемы управления и синхронизации. Данная схема управления и синхронизации также осуществляет управление интегральной микросхемой стабилизатора, участвует в определении маршрута прохождения сигнала в системе коммутации, а также мультиплексирует сигналы контроля и управления, включая сигнал отключения вызывного сигнала.

Проектирование интегральной микросхемы, которая предназначалась бы, в первую очередь, для выполнения функций интерфейса абонентских телефонных линий, в реальности оказалось нацеленным не только на выполнение большинства функций, определяемых аббревиатурой BORSCHT, но и на все более явно намечающийся переход телефонных систем к методам цифровой передачи речевого сигнала. Более того, программная гибкость цифровых систем связи, позволяющая приспособливать ее к изменяющимся условиям практического применения либо вводить дополнительные возможности путем изменения программы, оказали большое влияние на разработку.

Для выполнения функций интерфейса абонентских телефонных линий были использованы интегральные микросхемы, так как они способны выполнять большую часть функций, определяемых аббревиатурой BORSCHT, совместимы с цифровой техникой связи и легко программируются.

Однако здесь существуют определенные ограничения. Полупроводниковые материалы и создаваемые на их основе структуры, используемые при создании интегральных микросхем, таких, например, как схема абонентского или линейного интерфейса SLIC, и предназначенных для обработки большого количества логических операций и усиления сигналов, имеют ограничения в своей применимости и, прежде всего, по значениям напряжения пробоя, которые они должны выдерживать. Следовательно, высоковольтные схемы и цепи защиты от высоковольтных бросков напряжения, возникающих при переходных процессах, не входят в

состав интегральной микросхемы, а выполняются в виде внешних ее элементов. Дифференциальные усилители, выполненные в составе интегральной микросхемы, имеют очень хорошие характеристики подавления синфазных сигналов, но имеют ограничения по управлению большими по величине токами, протекающими в абонентских линиях связи. Следовательно, использование изолирующих трансформаторов на входе абонентских линий связи становится оправданным при использовании большого количества плат интерфейса абонентских линий SLIC, выполненных на базе интегральных микросхем. Также становится более предпочтительным использование системы батарейного питания на районной АТС по сравнению с использованием питающих стабилизаторов, описанных чуть ранее. Когда платы интерфейса абонентских линий SLIC на базе интегральных микросхем используются совместно с батарейным питанием, наиболее общепринятым способом задать ток шлейфа является использование внешнего транзистора, позволяющего гарантированно обеспечить необходимый по величине ток, который должен поступать на абонентский телефонный аппарат для его нормального функционирования.

1.14 Интегральная микросхема PEF3201

Примером интегральной микросхемы, разработанной для обеспечения функций интерфейса для абонентской телефонной линии, может служить разработанная компанией Lantiq микросхема PEF3201. Как показано на рисунке 11, интегральная микросхема PEF3201 предназначена для создания интерфейса абонентской линии с цифровой телефонной станцией, использующей метод импульсно-кодовой модуляции, а также – для создания интерфейса с кодеком.

Интегральная микросхема PEF3201 разработана для создания интерфейса между абонентской телефонной линией и цифровым кодеком. Представляет отдельный модуль, обеспечивающий выполнение функций

АБОНЕНТСКАЯ ЛИНИЯ (ДВУХПРОВОДНАЯ)

УПРАВЛЯЮЩИЕ ЦЕПИ РЕЛЕ

ЦИФРОВОЙ ВЫХОДНОЙ СИГНАЛ (ДВУХПРОВОДНЫЙ)

ЗАЩИТА АБОНЕНТСКОЙ ЛИНИИ ОТ ПЕРЕНАПРЯЖЕНИЙ (ИНТЕГРАЛЬНАЯ МИКРОСХЕМА TSM4301)

РЕЛЕ ВЫЗЫВНОГО СИГНАЛА И ТЕСТИРОВАНИЯ

БАТАРЕЙНОЕ ПИТАНИЕ

2ВТ 4ВТ ГИБРИДНАЯ СХЕМА

ПОЛОСОВОЙ ФИЛЬТР

ИКМ КОДЕР

ЦИФРОВОЙ ПОСТУПАЮЩИЙ СИГНАЛ (ДВУХПРОВОДНЫЙ)

УПРАВЛЕНИЕ И КОНТРОЛЬ

ИНТЕГРАЛЬНАЯ МИКРОСХЕМА TSM4204

ВОССТАНАВЛИВАЮЩИЙ ФИЛЬТР ИНТЕГРАЛЬНОЙ МИКРОСХЕМЫ REF 3201

ИКМ -ДЕКОДЕР (ИНТЕГРАЛЬНАЯ МИКРОСХЕМА REF 3201)

К ГЕНЕРАТОРУ ВЫЗЫВНОГО СИГНАЛА

К ТЕСТИРУЕМОМУ ОБОРУДОВАНИЮ

СИНХРОНИЗАЦИЯ С ПОСТОЯННЫМ ТОКОМ

ОТ СХЕМЫ СИНХРОНИЗАЦИИ

УПРАВЛЕНИЕ ПЛАТОЙ ЛИНЕЙНОГО ИНТЕРФЕЙСА

РАСПРЕДЕЛЕНИЕ ВРЕМЕННЫХ СЛОТОВ

СИНХРОНИЗАЦИЯ М

ДАННЫЕ

СЧИТЫВАНИЕ /ЗАПИСЬ

Расшифровка функций аббревиатуры BORSCHT:

- B – Батарейное питание
- O – Защита от перенапряжений
- R – Вызывной сигнал
- S – Управление и контроль
- C – Кодирование сигнала
- H – Гибридная схема
- T – Тестирование

Она выполняет функции гибридной схемы, контроля и управления, а также контроля над вызывным сигналом и проведением тестирования. Так как в системе связи с использованием гибридной схемы осуществляется переход от четырехпроводной схемы к двухпроводной, данная функция передается от интерфейса магистральной линии к линейному интерфейсу районной АТС и выполняется для каждой абонентской линии. Батарейное питание сохраняется в том виде, в каком оно существует на районной АТС, а защита от высоковольтных перенапряжений, возникающих при

переходных процессах, обеспечивается самовосстанавливающимися предохранителями и динисторами.

Более подробная схема собственно интегральной микросхемы, используемой в качестве интерфейса абонентской линии, приведена на рисунке 12.

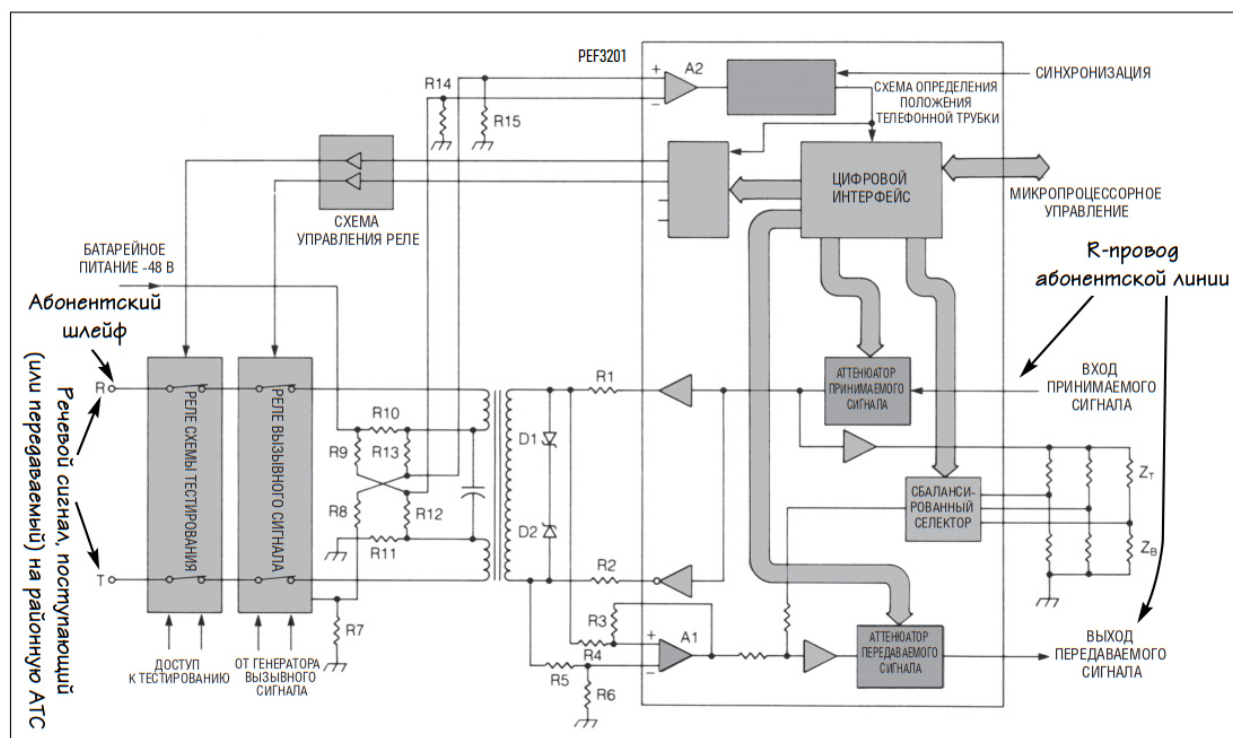


Рисунок 12 – схема управления абонентской линией

Речевой сигнал поступает в абонентскую линию с коммутатора районной телефонной станции через вход принимаемого сигнала (REC IN), аттенуатор поступающего сигнала (REC ATTN) и затем схему управления, обеспечивающую работу линейного трансформатора в двухтактном режиме с использованием внешних (относительно интегральной микросхемы или навесных) резисторов R1 и R2. Внешние стабилитроны D1 и D2 обеспечивают защиту от перенапряжений в телефонной линии. После прохождения трансформаторов связи на районной АТС речевые сигналы, поступающие от абонента, проходят на вход операционного усилителя интегральной микросхемы TCM4204, A1, а затем через аттенуатор передаваемого сигнала (XMIT ATTN) и выход двухпроводной

соединительной линии станции на схемы магистральных линий. Усиление операционных усилителей задается внешними резисторами R3 и R4.

Сигнал замыкания контактов телефонной трубки (состояние, соответствующее поднятой с рычагов трубки) детектируется схемой резистивного моста (резисторы R8–R13) со стороны выводов абонентской линии линейного трансформатора и подается через операционный усилитель A2 на фильтр нижних частот, а затем через цифровой интерфейс на шину управления контроллера. Фильтр нижних частот препятствует прохождению вызывного сигнала в момент его подачи, поэтому он не оказывает влияния на работу микропроцессора. Сигнал готовности набора номера в линию поступает через вход REC IN, а затем – в схему принимаемого сигнала. Тональный сигнал набора номера проходит через схему передаваемого сигнала и поступает на выход XMIT OUT.

Работа внешних схем управления реле вызывного сигнала и тестирования определяется встроенными в микрочип триггерами (схемами запуска), которые синхронизируются через шину управления микропроцессора с использованием цифрового интерфейса. При включении схема управления реле подает напряжение на реле вызывного сигнала, подающего вызывной сигнал в абонентскую линию. После ответа вызываемого абонента ток, соответствующий условию поднятой с рычагов трубки, фиксируется внешним резистором R7, чтобы обеспечить отключение вызывного сигнала через A2. Доступ к тестированию обеспечивается подачей питания на реле тестирования точно таким же способом, что и на реле вызывного сигнала. Когда на реле не подаются напряжения, подача батарейного питания –48В в абонентскую линию обеспечивается через нормально замкнутые контакты реле.

Наиболее примечательной особенностью такой схемы управления абонентской линией является цифровой интерфейс и те дополнительные возможности управления, которые он обеспечивает. Она позволяет внешнему микропроцессору (контроль которого распространяется на

					ПГУ 2.090301.25.001 ПЗ	Лист
						40
Изм.	Лист	№ докум	Подпись	Дата		

большое количество плат абонентского интерфейса SLIC) отдельно управлять аттенюаторами принимаемого и передаваемого сигналов, выбирать параметры внешней схемы согласования, обеспечивая увеличение или уменьшение значений Z_T и Z_B в соответствии с поступающими командами, а также проверять параметры аналоговой обратной (возвратной) петли, которые необходимы при проведении тестирования. Помимо этого, обеспечиваются оба режима подачи сигнала: подача «через землю» и подача по абонентской линии.

Цифровая схема управления работой абонентской линии позволяет удаленному микропроцессору управлять большим количеством функций, обеспечивающих ее работоспособность и тестирование.

Параметры управления интегральной микросхемой PEF 3201 сохраняются в 24-разрядной памяти в схеме цифрового интерфейса. Внешний микропроцессор устанавливает и поддерживает контроль над работой интегральной микросхемы последовательной записью в каждый разряд регистра. Каждый разряд также может считываться микропроцессором последовательно. Разряды нулевой и первый имеют фиксированное состояние, которое может только считываться, тогда как разряды со 2-го по 23-й предназначены для временного хранения данных, информация в которых может по мере необходимости как записываться, так и считываться. Точное назначение каждого разряда может быть найдено в паспортных данных интегральной микросхемы PEF 3201.

Для обслуживания интерфейса абонентской телефонной линии SLIC используются четыре отдельные линии управления: вход сигнала разрешения интегральной микросхемы (CE), вход ввода/вывода данных (I/O), вход записи/считывания (R/W), синхронизирующий вход (CLKM), управляющий синхронизацией и передачей данных на микропроцессор либо с него. Последовательная передача данных осуществляется через контакт I/O до тех пор, пока на входе CE присутствует низкий уровень

логического сигнала, а на контакте R/W – либо высокий, либо низкий уровень логического сигнала.

Каждый тактовый импульс на выводе CLKM будет перемещать указатель регистра на следующий доступный разряд линии ввода/вывода.

Два разряда регистра задают один из четырех режимов работы интерфейсной линейной платы. Режим 0 переводит плату интерфейса SLIM в режим ожидания. Этот режим очень напоминает режим пониженного энергопотребления, однако сигналы, характеризующие состояния на выходных выводах, отличаются. Режим 1 вызывает переход к режиму пониженного энергопотребления. В этом режиме питание на цепи управления интерфейсной платы подается, однако все цепи прохождения речевого сигнала оказываются недоступными. Режим 2 предназначен для передачи обычного речевого сигнала. В этом режиме на все внутренние цепи подается напряжение, и плата абонентского интерфейса будет работать в нормальном режиме. Режим 3 устанавливает условие обратной (возвратной) петли, при котором размыкается обычная схема согласования и любой передаваемый выходной сигнал должен следовать за принимаемым сигналом на приемник. Этот режим используется для проверки телефонного шлейфа.

При передаче к вызываемому абоненту речевого сигнала по системе связи, в которой используется дискретизация непрерывного аналогового сигнала, очень важно, чтобы никакие частоты, превышающие половину значения частоты измерений дискретизации, не поступали на вход схемы дискретизации кодера. При их попадании на вход сигналы с этой частотой не будут воспроизводиться должным образом, а на выходе появятся ложные сигналы, частота которых будет представлять разность между действительным значением частоты нежелательного сигнала и половиной значения частоты дискретизации. Это явление получило специальное название «искажения из-за наложения спектров при дискретизации», или, в более общем случае, эффекта наложения. Для того чтобы избежать данного

явления, входной сигнал перед дискретизацией должен быть отфильтрован. Наглядный, в прямом смысле этого слова, эффект наложения проявляется в фильмах–вестернах, в которых скорость кадров недостаточна для того, чтобы правильно записать быстрое вращение спиц колеса, поэтому вместо быстрого вращения колес вперед на экране они медленно вращаются назад либо вообще стоят на месте.

Для системы связи с дискретизированным речевым сигналом для того, чтобы предотвратить искажения, полоса пропускания голосового канала должна быть ограничена по частотам как сверху, так и снизу.

Когда сигнал приходит от вызываемого абонента, выходной сигнал декодера должен быть отфильтрован для того, чтобы удалить высокочастотные составляющие, вызванные процессом ступенчатости при восстановлении сигнала в декодере. Эти процессы требуют наличия в цифровых телефонных системах недорогих высококачественных фильтров.

Фильтры для передаваемого и принимаемого сигналов, изображенные на рисунке 11, представляют в своей основе фильтры нижних частот, которые обрезают частоты свыше 3 кГц. Помимо этого, фильтр передаваемого сигнала также должен ослаблять любую составляющую входного сигнала с частотой 60 Гц (которая может быть индуцирована в абонентской линии близкорасположенными линиями электропередачи). Следовательно, становится актуальным наличие полосового фильтра, полоса пропускания которого простирается от 300 до 3000 Гц. Как в большинстве случаев, технические характеристики фильтров представляются в виде полосы пропускания (от 300 до 3000 Гц) и ослабления на частотах области не пропуска (обрезания) – частоты менее 300 Гц и выше 3000 Гц. Однако в этом случае очень важное значение приобретают характеристики пульсации (изменения спектральной характеристики по полосе пропускания) из-за проблем, связанных с качеством передачи сигнала и возникающих при прохождении вызова через систему из нескольких фильтров, включенных последовательно, которые с

достаточно высокой вероятностью используются в системах связи с импульсно–кодовой модуляцией. Требования, предъявляемые к области непропускания, препятствуют возникновению эффекта наложения на частотах свыше 4 кГц.

Фильтры также вносят вклад в перекрестные помехи, а также в шум незанятой линии. Если в состав одной монолитной интегральной микросхемы входит более чем один фильтр, то особое внимание должно быть уделено для существенного ослабления перекрестных помех между ними и принятия мер, исключающих их взаимодействие. Связь по перекрестным помехам через общий источник питания является наиболее распространенной и должна быть исключена.

Блок–схема стандартного фильтра для акустического диапазона, выполненного в виде однокристалльной микросхемы компании Lantiq REF 24901H, приведена на рисунке 13.

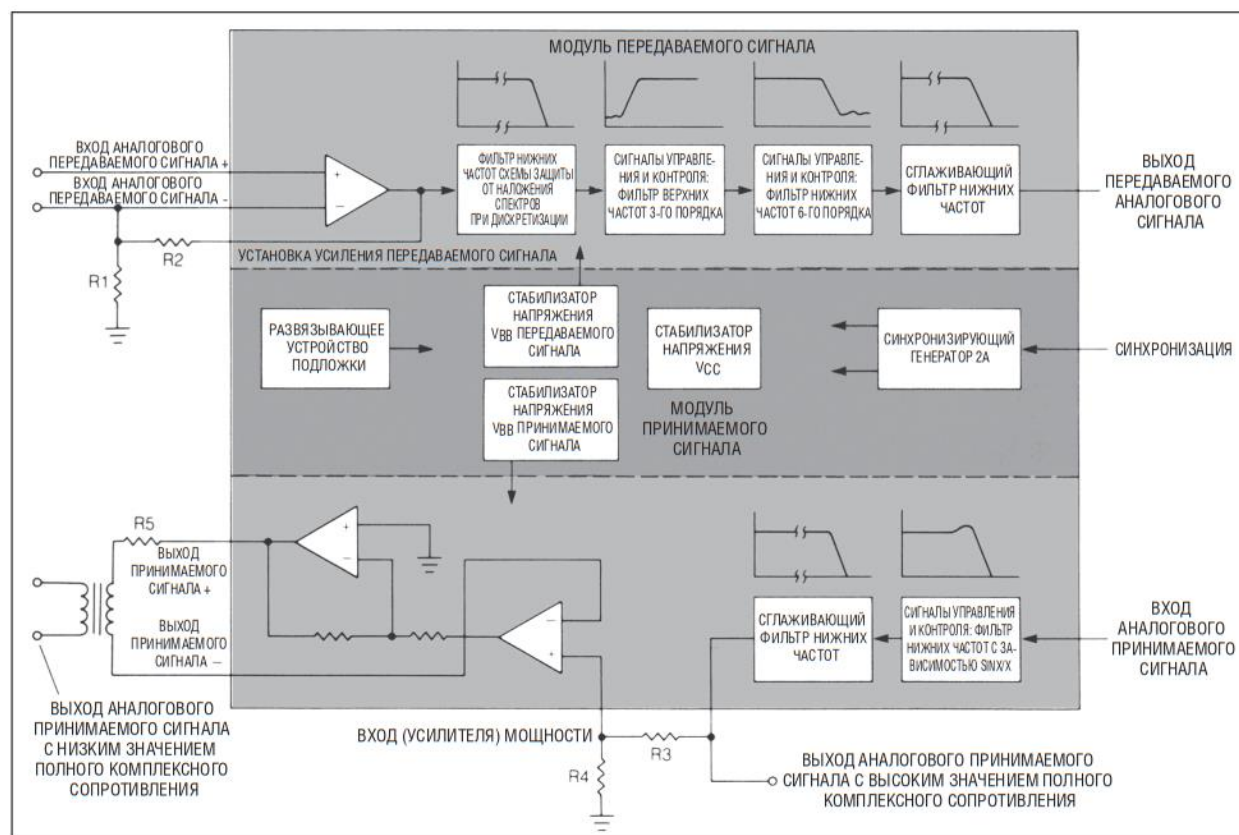


Рисунок 13 – фильтр линии связи с импульсно–кодовой модуляцией

Схема разделена на три блока. В блоке, относящемся к передаваемому сигналу, имеются акустический фильтр верхних частот третьего порядка и фильтр нижних частот шестого порядка, предназначенные обеспечить фильтрацию в полосе пропускания нежелательных коммутационных и низкочастотных шумов. В блоке принимаемого сигнала для кодека используется коррекция вида $\sin x/x$ и фильтруются высокочастотные сигналы, возникающие при коммутации. Интегральная микросхема REF 24901Н разрабатывалась для того, чтобы обеспечить фильтрацию в полосах пропускания передаваемого и принимаемого сигналов для соединительных линий с кодово-импульсной модуляцией или оконечных устройств абонентских линий. В ней применяется метод переключаемых конденсаторов и используются n-канальные МОП-структуры. Коэффициент усиления передаваемого сигнала может устанавливаться отношением сопротивлений резисторов R1 и R2. Если используются электронные гибридные схемы, имеющие высокое значение комплексного входного сопротивления, то выход принимаемого сигнала (HIGH IMPEDANCE REC ANLG OUT) может управлять гибридной схемой напрямую. Если требуется согласование со схемой, имеющей низкое значение полного комплексного сопротивления, встроенный в микрокристалл усилитель мощности может быть включен по схеме, приведенной на рисунке 13. Величины резисторов R3 и R4 подбираются для обеспечения необходимого усиления, а резистор R5 служит для согласования величины полного комплексного сопротивления. Третий блок схемы содержит тактовые генераторы, стабилизаторы напряжения для принимающего и передающего фильтров и развязывающее устройство подложки интегральной микросхемы, снижающее перекрестные помехи. Интегральная микросхема REF 24901Н содержит фильтры как для передаваемого, так и принимаемого сигналов. Интегральная схема управления абонентской линией связи, REF 3201, обеспечивает необходимый переход от четырехпроводной к двухпроводной линии связи.

Для экономии на ряде видов оборудования, достаточно большая часть абонентских линий может мультиплексироваться с использованием одного кодека. Такой способ является, однако, компромиссным. Большое количество телефонных аппаратов окажутся неработоспособными в случае выхода из строя одного кодека. Дополнительно к этому, аналоговое мультиплексирование с разделением по времени является достаточно сложным процессом и осложняется проблемами перекрестных помех. Здравый смысл подсказывает, что индивидуальный кодек для каждой абонентской линии является наилучшим решением проблемы.

Представленная на рисунке 11 схема относится к случаю использования для одной абонентской линии или одного канала связи. Также возможна ситуация, когда один кодек используется несколькими линиями с использованием мультиплексного оборудования. Такой подход позволяет экономить на оборудовании, так как требуется использовать меньшее количество кодеков. Однако при использовании одного кодека на несколько абонентских линий возникает ряд существенных проблем. Первая связана с надежностью или простоем. Если большое количество линий мультиплексируются с использованием одного кодека, то все эти линии окажутся неработоспособными в случае выхода из строя одного кодека. Вторая проблема связана с тем, что метод аналогового мультиплексирования с разделением во времени является очень сложным и менее гибким процессом по сравнению с цифровым мультиплексированием. Кроме этого, подобные кодеки, предназначенные обслуживать большое количество линий, гораздо сложнее при проектировании в виде однокристалльной интегральной микросхемы, прежде всего, из-за перекрестных помех, возникающих между отдельными каналами связи, и требований обеспечения необходимой производительности по скорости передачи.

Из-за перечисленных проблем, а также потенциально очень большого объема линий (существует более 600 миллионов абонентских телефонных

					ПГУ 2.090301.25.001 ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		46

линий связи), стоимость которых могла бы быть снижена, большинство производителей в мире нацелены на использование персонального кодека для каждой телефонной линии. Ожидаемое снижение стоимости следует из стандартной зависимости стоимости разработки, в соответствии с которой стоимость изделия электронной техники значительно снижается при больших объемах его производства.

Основные области использования кодеков в системах телефонной связи связаны с районными АТС, оборудованием и устройствами сопряжения цифровых каналов связи с голосовыми каналами, автоматическими офисными телефонными станциями и цифровыми телефонными аппаратами. Также возможно использование кодеков в других, не связанных с телефонией, областях, но объемы такого применения будут незначительными.

1.15 Вывод

В данной главе были рассмотрены возможные микросхемы для реализации порта интерфейса FXS. Рассмотрена работа протоколов, а также работа аудио кодеков и микросхемы для их реализации.

2 Реализация

2.1 Фрагмент структурной схемы разрабатываемого модуля FXS

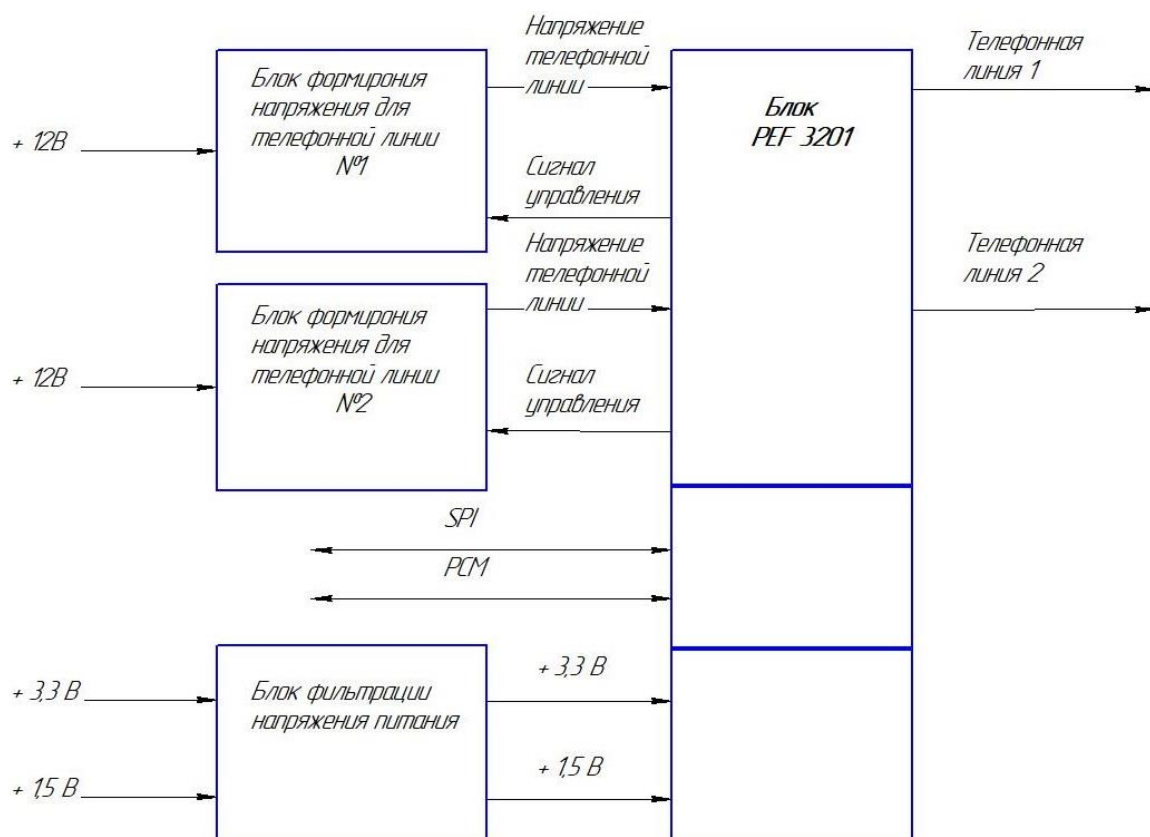


Рисунок 14 – фрагмент структурной схема модуля FXS

На рисунке 14 видно, что для каждой телефонной линии должен быть разработан свой блок формирования напряжения. Сам блок формирования напряжения питается от 12 В. Он передает в блок PEF 3201(SLIC) напряжение телефонной линии, а получает в ответ сигнал управления. Так же в блок SLIC поступает последовательный периферийный интерфейс SPI, который передает данные в режиме полного дуплекса, и ИКМ для оцифровки аналоговых сигналов. Последний блок фильтрации напряжения питания служит для уменьшения пульсации напряжения. Структурная схема приведена в приложении А.

2.2 Принципиальная схема разрабатываемого модуля FXS

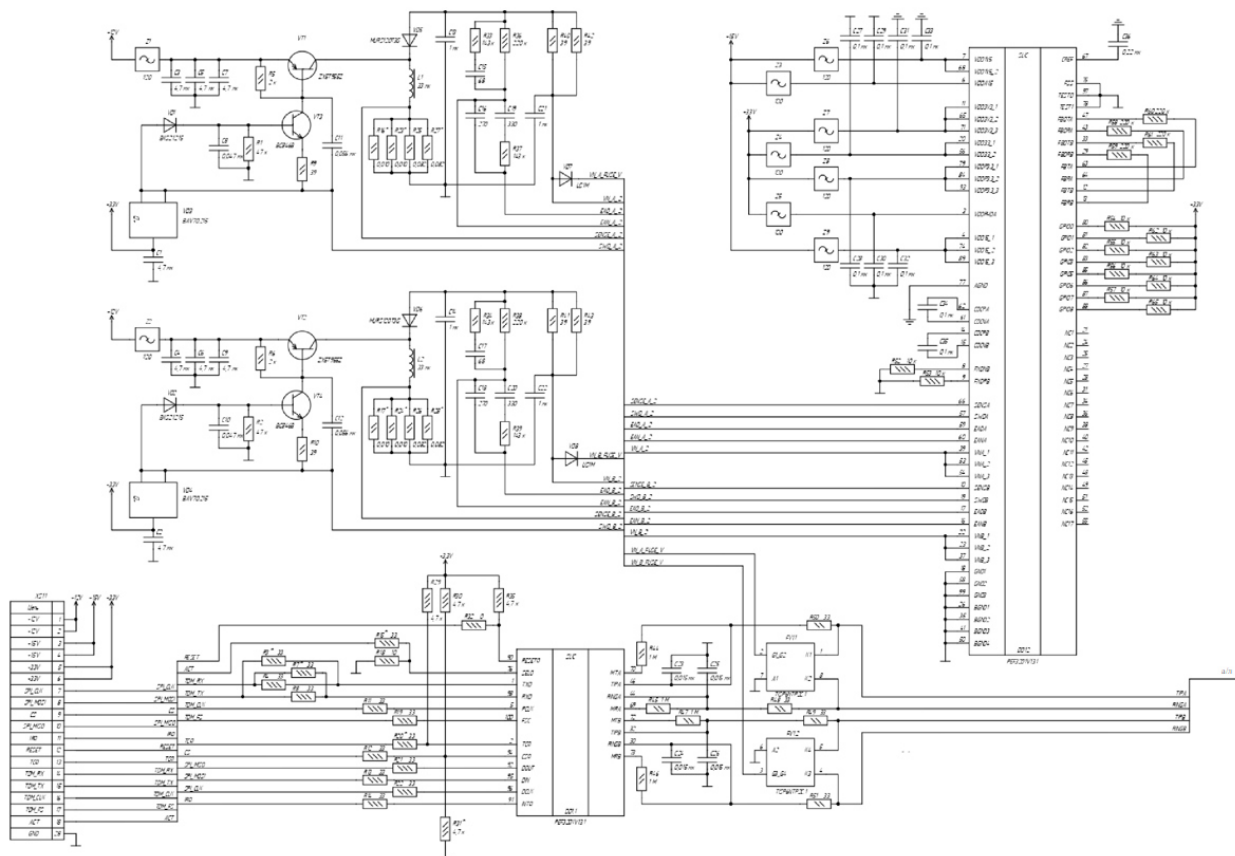


Рисунок 15- Принципиальная схема разрабатываемого модуля FXS

3.Разработка программного обеспечения

3.1 Программа управления модулем FXS.

Для управления модулем FXS используется стек драйверов:

- Tapi (TAPI High Level);
- DXT (TAPI Low Level);
- IFXOS ;
- TDM .

Драйвер устройства PEF3201 представляет собой программный модуль, позволяющий управлять PEF3201 с помощью телефонного интерфейса (TAPI), для продуктов УЦАТС. Использование стека позволяет обеспечить взаимодействие аппаратной части с программной, на всех уровнях.

Рассмотрим уровни взаимодействия драйверов модуля FXS, приведённый на рисунке16.

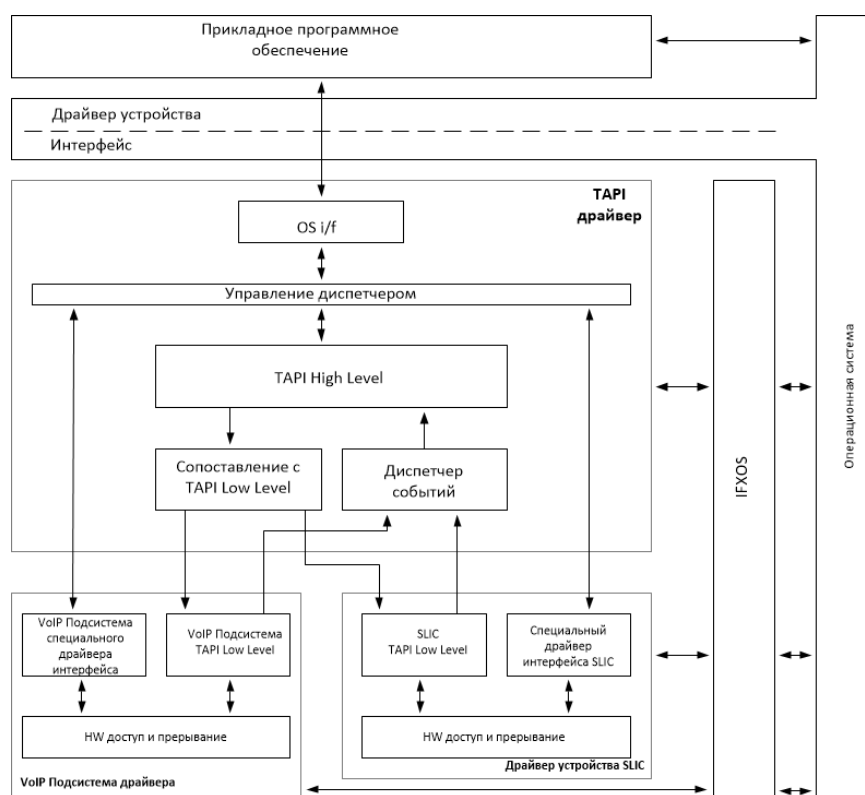


Рисунок 16 – взаимодействие драйверов модуля FXS.

Драйвер Tapi реализован в двух уровнях: TAPI High Level (HL), абстрагирование функций до уровня, не зависящего от устройства, и TAPI Low

Level (LL), реализующего специфичную для устройства часть (например, HW). / FW доступ). TAPI HL поставляется в виде отдельного драйвера, а TAPI LL реализован в отдельном двоичном файле для каждого поддерживаемого устройства. На рисунке 16 представлен обзор архитектуры TAPI, показывающий конфигурацию, в которой устройства PEF3201 управляются с помощью TAPI. В состав драйвера TAPI High Level входят следующие файлы исходного кода:

- drv_api.h
- drv_tapi.h
- drv_tapi_api.h
- drv_tapi_cid.c
- drv_tapi_cid.h
- drv_tapi_debug.c
- drv_tapi_debug.h
- drv_tapi_dev_io.c
- drv_tapi_dial.c
- drv_tapi_event.c
- drv_tapi_event.h
- drv_tapi_fxo.c
- drv_tapi_fxo.h
- drv_tapi_init.c
- drv_tapi_ioctl.c
- drv_tapi_ioctl.h
- drv_tapi_ioctl_handlers.h
- drv_tapi_kpi.c
- drv_tapi_kpi.h
- drv_tapi_linux.c
- drv_tapi_linux.h
- drv_tapi_ll_if_version.h

- drv_tapi_meter.c
- drv_tapi_misc.c
- drv_tapi_opcontrol.c
- drv_tapi_osmap.h
- drv_tapi_pcm.c
- drv_tapi_polling.c
- drv_tapi_polling.h
- drv_tapi_ppd.c
- drv_tapi_ppd.h
- drv_tapi_qos.c
- drv_tapi_qos.h
- drv_tapi_ring.c
- drv_tapi_stat.c
- drv_tapi_stream.c
- drv_tapi_stream.h
- drv_tapi_tone.c
- drv_tapi_voice.c

Фрагмент листинга TAPI High Level предоставлен в приложении А,
В состав драйвера TAPI Low Level входят следующие файлы исходного кода:
ar9.h

- danube.h
- easy3201.evs.h
- easy3201.h
- template.h
- drv_dxt.h
- drv_dxt_access.c
- drv_dxt_access.h
- drv_dxt_alm.c
- drv_dxt_alm.h

- drv_dxt_alm_gr909.c
- drv_dxt_alm_gr909.h
- drv_dxt_alm_lt.c
- drv_dxt_alm_lt.h
- drv_dxt_alm_priv.h
- drv_dxt_api.c
- drv_dxt_api.h
- drv_dxt_bbd.c
- drv_dxt_bbd.h
- drv_dxt_cid.c
- drv_dxt_cid.h
- drv_dxt_cid_priv.h
- drv_dxt_cid_sm.c
- drv_dxt_config.h
- drv_dxt_debug.c
- drv_dxt_debug.h
- drv_dxt_dtmf.c
- drv_dxt_dtmf.h
- drv_dxt_dtmf_priv.h
- drv_dxt_dtmf_sm.c
- drv_dxt_dwld.c
- drv_dxt_dwld.h
- drv_dxt_dwld_asdsp.c
- drv_dxt_eval.h
- drv_dxt_evalproc.h
- drv_dxt_event.c
- drv_dxt_fw_cmd_aop.h
- drv_dxt_fw_cmd_eop.h
- drv_dxt_fw_cmd_sdd.h

- drv_dxt_fw_events.h
- drv_dxt_fw_headers.h
- drv_dxt_gpio.c
- drv_dxt_gpio.h
- drv_dxt_gpio_priv.h
- drv_dxt_gpio_user.h
- drv_dxt_init.c
- drv_dxt_init.h
- drv_dxt_ioctl.c
- drv_dxt_irq.c
- drv_dxt_irq.h
- drv_dxt_linux.c
- drv_dxt_ll_if_version.h
- drv_dxt_pcm.c
- drv_dxt_pcm.h
- drv_dxt_pcm_priv.h
- drv_dxt_tg.c
- drv_dxt_tg.h
- drv_os.h

Фрагмент листинга TAPI LL предоставлен в приложении Б.

IFXOS это промежуточный драйвер между ОС и TAPI, предназначенный для раскрытия использованных вызов функций–оболочек драйверами TAPI HL и LL, под конкретную операционную систему. В состав драйвера IFXOS входят следующие файлы исходного кода:

- src
- common
- dev_io
- include

- linux
- xapi
- src\common
- ifxos_debug.c
- ifxos_sys_show.c
- ifxos_version.c
- ifx_crc.c
- ifx_fifo.c
- ifx_getopt.c
- ifx_getopt_ext.c
- ifx_ini_access.c
- src\dev_io
- ifxos_device_io.c
- ifxos_device_io_access.c
- src\include
- ifxos_common.h
- ifxos_copy_user_space.h
- ifxos_debug.h
- ifxos_device_access.h
- 7я586 ifxos_device_io.h
- ifxos_event.h
- ifxos_file_access.h
- ifxos_interrupt.h
- ifxos_linux_drv.h
- ifxos_lock.h
- ifxos_memory_alloc.h
- ifxos_memory_map.h
- ifxos_misc.h
- ifxos_mutex.h

- ifxos_pipe.h
- ifxos_print.h
- ifxos_print_io.h
- ifxos_rt_if_check.h
- ifxos_select.h
- ifxos_socket.h
- ifxos_std_defs.h
- ifxos_sys_show.h
- ifxos_sys_show_interface.h
- ifxos_termios.h
- ifxos_thread.h
- ifxos_time.h
- ifxos_version.h
- ifx_crc.h
- ifx_fifo.h
- ifx_getopt.h
- ifx_getopt_ext.h
- ifx_ini_access.h
- ifx_types.h
- src\include\lint
- ifxos_lint_std_defs.h
- src\include\linux
- ifxos_linux_common.h
- ifxos_linux_device_access.h
- ifxos_linux_event.h
- ifxos_linux_file_access.h
- ifxos_linux_interrupt.h
- ifxos_linux_lock.h
- ifxos_linux_memory_alloc.h

- ifxos_linux_misc.h
- ifxos_linux_mutex.h
- ifxos_linux_pipe.h
- ifxos_linux_print.h
- ifxos_linux_print_io.h
- ifxos_linux_select.h
- ifxos_linux_socket.h
- ifxos_linux_std_defs.h
- ifxos_linux_termios.h
- ifxos_linux_thread.h
- ifxos_linux_time.h
- src/linux
- ifxos_linux_copy_user_space_appl.c
- ifxos_linux_copy_user_space_drv.c
- ifxos_linux_device_access_appl.c
- ifxos_linux_event_appl.c
- ifxos_linux_event_drv.c
- ifxos_linux_file_access_appl.c
- ifxos_linux_lock_appl.c
- ifxos_linux_lock_drv.c
- ifxos_linux_memory_alloc_appl.c
- ifxos_linux_memory_alloc_drv.c
- ifxos_linux_memory_map_appl.c
- ifxos_linux_memory_map_drv.c
- ifxos_linux_misc_appl.c
- ifxos_linux_module_drv.c
- ifxos_linux_mutex_appl.c
- ifxos_linux_mutex_drv.c
- ifxos_linux_pipe_appl.c

- ifxos_linux_print_appl.c
- ifxos_linux_print_io_appl.c
- ifxos_linux_select_appl.c
- ifxos_linux_select_drv.c
- ifxos_linux_socket_appl.c
- ifxos_linux_socket_drv.c
- ifxos_linux_termios_appl.c
- ifxos_linux_thread_appl.c
- ifxos_linux_thread_drv.c
- ifxos_linux_time_appl.c
- ifxos_linux_time_drv.c
- src\xapi
- ifxos_xapi_event.c
- ifxos_xapi_lock.c
- ifxos_xapi_memory_alloc.c
- ifxos_xapi_mutex.c
- ifxos_xapi_print.c
- ifxos_xapi_select.c
- ifxos_xapi_thread.c
- ifxos_xapi_time.c

Фрагменты листинга предоставлены в приложении В.

TDM– подразумевает под собой объединение нескольких входных, низкоскоростных каналов в один составной высокоскоростной канал. Объединение канала FXS, происходит путем объединения 32 каналов DS0, один из которых используется для кадровой синхронизации, другой для передачи сигнализации. Кадр синхронизации состоит из 32 тайм слотов по 8 бит каждый. Частота потока равна 8КГц, скорость потока в таком случае будет равна 2мб/с, она получается путем умножения количества каналов на частоту

и размер (32x8x8=2048 Кбит/с.). Под каналом сигнализации понимается информация, необходимая сетевым элементам, для выполнения их функций (установка/ отбой телефонных соединений.). В состав драйвера TDM входят следующие файлы исходного кода:

- fsl_sai.h
- fsl_sai.mod.c
- fsl_sai_pltfrm.c

Фрагмент листинга предоставлен в приложении Г.

3.2 Вывод

В данном блоке подробнее изучили драйвер TAPI и разработали программу для управления модулем FXS.

4. Охрана труда

4.1 Основные воздействия на программиста за ПЭВМ

В настоящее время все больше и больше на рабочих местах появляются персональные компьютеры (ПК). Работа все чаще требует большего взаимодействия с ПК. Так же для тестирования и контроля за УАТС, оператору тоже требуется ПК.

С каждым годом ПЭВМ (персональная электронно–вычислительная машина) хоть и улучшается для меньшего вреда при работе с ним, но все же некоторые негативные воздействия остались. Для уменьшения этих воздействий следует придерживаться определенным правилам охраны труда.

В данном разделе ВКР разберем правила пользования ПЭВМ, которые способствуют уменьшению воздействия на пользователя.

4.2 Требования охраны труда перед началом работы

Каждый пользователь ПЭВМ перед работай с ним, должен выполнить следующие действия:

- осмотреть и привести в порядок рабочее место, убрать посторонние предметы, которые могут отвлекать внимание и затруднять работу;
- проверить визуальным осмотром достаточность освещенности рабочей поверхности, отсутствие свисающих и оголенных концов электрической проводки, состояние полов, окон. Если были выявлены оголенные провода, то работать и подходить к такой ПЭВМ опасно и запрещено. Нужно незамедлительно обесточить систему и сообщить электрику;
- проверить исправность оборудования и правильность его подключения к сети, проверить устойчивость производственных столов, стеллажей, прочность крепления оборудования. Если оборудование закреплено не надежно, то оно может привести к травмам в процессе работы. Следует сначала устранить эти недостатки;

- включить питание ПК, соблюдая последовательность: сетевой фильтр, монитор, периферийные устройства, процессор. Включать, следует не касаясь другого оборудования, а иначе может привести к «удару» статического напряжения. Так же не разрешается приступать к работе в случае обнаружения неисправности оборудования.

4.3 Требования охраны труда во время работы

Во время работы пользователь ПЭВМ обязан:

- следить за соблюдением требований охраны труда своими коллегами;
- не загромождать рабочее место и проходы к нему;
- соблюдать правила использования оборудования.

При работе с ПЭВМ программист обязан:

- Соблюдать установленные режимы рабочего времени: регламентированные перерывы в работе и выполнять в физкультпаузах рекомендованные упражнения для глаз, шеи, рук, туловища, ног
- Соблюдать расстояние глаз до экрана в пределах 60–70 см., но не ближе 50 см с учетом размеров алфавитно–цифровых знаков и символов. На рисунке 17 наглядно показано правильное и неправильное нахождение за ПЭВМ.

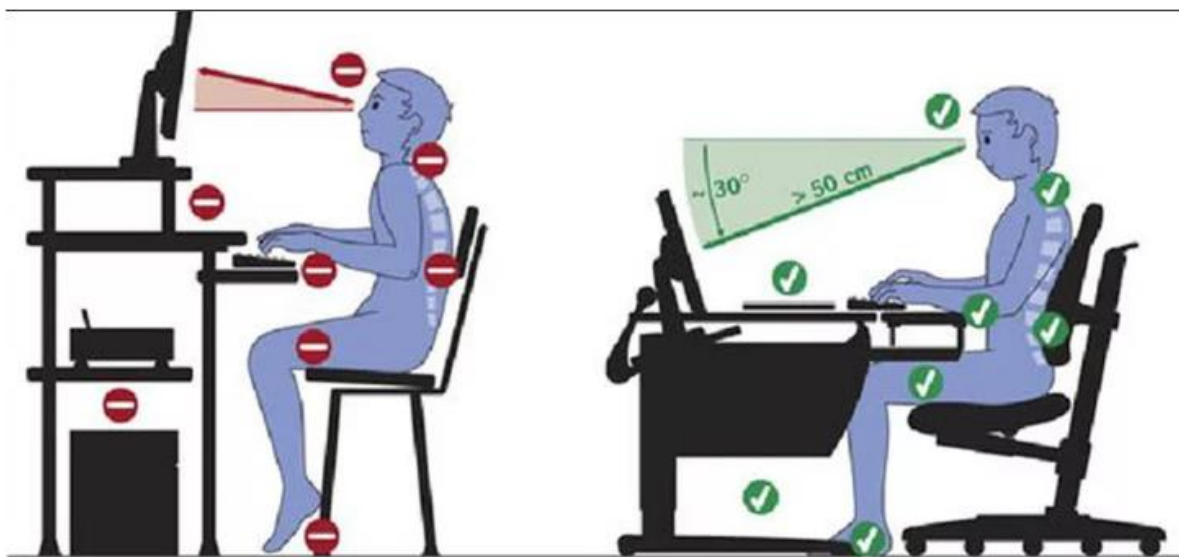


Рисунок 16 – правильное и неправильное нахождение за ПК.

Изм.	Лист	№ докум	Подпись	Дата

Программисту при работе на ПК не должен:

- касаться одновременно экрана монитора и клавиатуры;
- допускать попадания влаги на поверхность системного блока, монитора, рабочую поверхность клавиатуры, принтера и других устройств;
- производить самостоятельное вскрытие и ремонт оборудования.

Программист обязан отключить ПК из электросети:

- при обнаружении неисправности;
- при внезапном снятии напряжения электросети;
- во время чистки и уборки оборудования.

Продолжительность непрерывной работы с ПК не должна превышать 2 часов.

4.4 Требования охраны труда по окончании работы

По завершению работы необходимо выключить электрические приборы, отключить от электрической сети все оборудования. По окончании работ с ПЭВМ программист обязан соблюдать следующую последовательность выключения техники:

- произвести закрытие всех активных задач;
- выполнить парковку считывающей головки жесткого диска (если не предусмотрена автоматическая парковка головки);
- убедиться, что в дисководов нет дисков;
- выключить питание системного блока (процессора);
- выключить питание всех периферийных устройств;
- отключить блок питания.

Так же необходимо закрыть окна в помещении, выключить электрическое освещение. Перед уходом программист обязан осмотреть и привести в порядок рабочее место и вымыть с мылом руки и лицо. Закрыть дверь и ключ сдать на вахту. При обнаружении недостатков, возникших во время работы, сообщать об этом начальнику отдела.

5. Заключение

В ходе выполнения выпускной квалификационной работы были рассмотрены некоторые микросхемы разных производителей на которых проектируется порт интерфейса FXS для УАТС, разобраны принципы работы протоколов, взаимодействующих с портом FXS, разработана структурная схема, а также принципиальная схема модуля SLIC для УАТС на современной электронной базе.

Была написана программа драйвера TAPI, которая позволяет взаимодействовать разным устройствам с модулями SLIC в одной УАТС.

А также в ВКР были рассмотрены правила охраны труда, установленные для безопасной работы контролера УАТС за ПЭВМ.

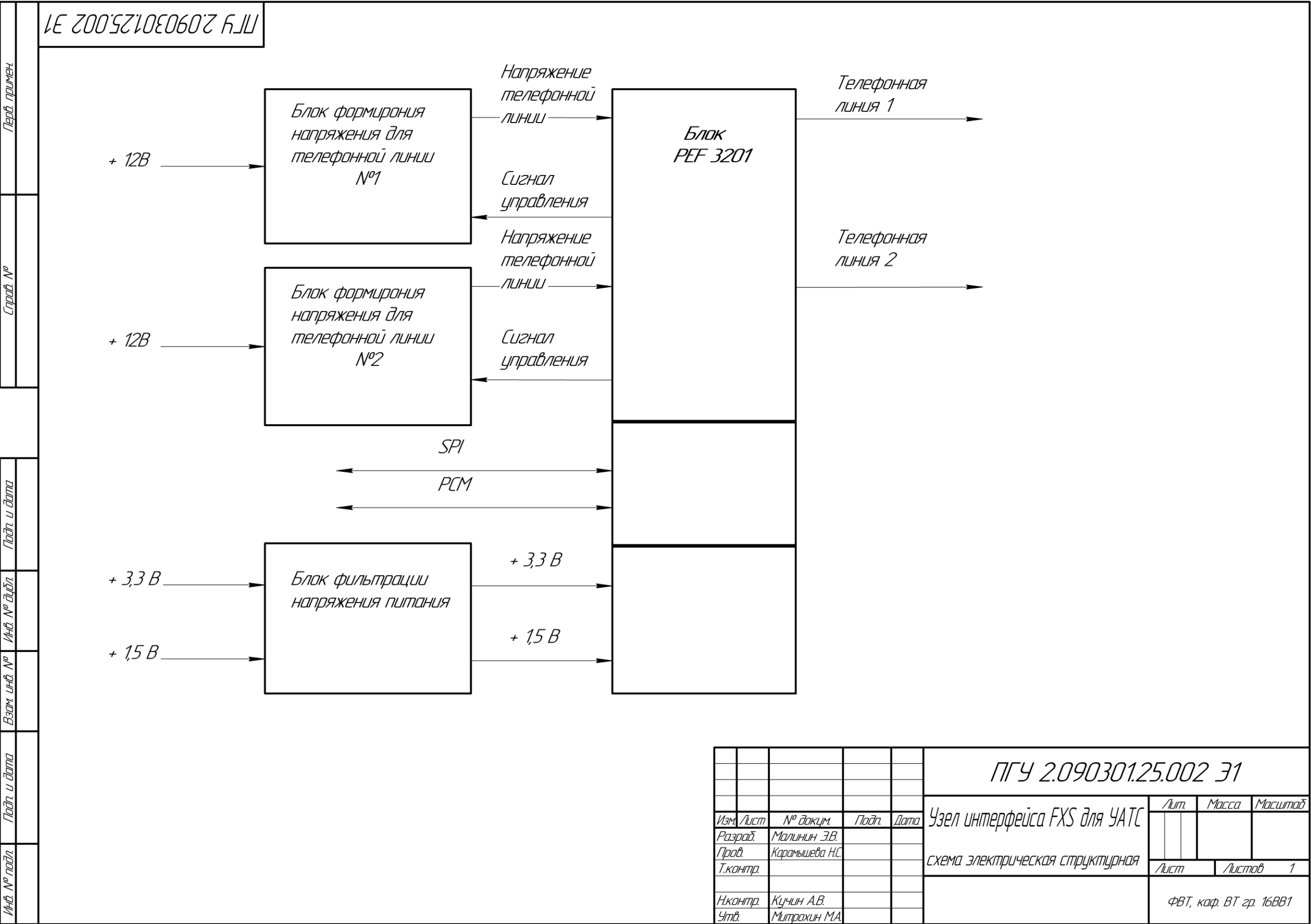
Список использованных источников

1. Официальный сайт аппаратного обеспечения Lantiq, URL: <https://www.lantiq.com/products>, дата обращения 20.02.2020 г.
2. Официальный сайт аппаратного обеспечения microsemi. URL: <https://microsemi.kde.org/>, дата обращения 20.02.2020 г.
3. Соколов Н.А. Эволюция местных телефонных сетей. ТОО Типография "Книга". г. Пермь, 2016. – 375 с.
4. Соколов Н.А. Сети абонентского доступа. Принципы построения. ИПК "Звезда". г. Пермь, 2018. – 254 с.
5. Официальный сайт аппаратного обеспечения Nuvoton, URL: <http://nuvoton.io/>, дата обращения 20.02.2020 г.
6. Соколов Н.А. М., Альварес Паблиншинг. Телекоммуникационные сети Монография в 4–х главах Часть 1 (Глава 1) Принципы построения телекоммуникационных систем. 2003. – 128 с.
7. Родичев Юрий Андреевич. Нормативная база и стандарты в области информационной безопасности. Учебное пособие. Издательство: “Питер”, 2017. – 256 с.
8. Официальный сайт аппаратного обеспечения PEF3201 UML URL: <http://pef.tigris.org/>, дата обращения 20.02.2020 г.
9. Системы коммутации. Гольдштейн Б.С. Спб.: BHV–Санкт–Петербург, 2003
<http://niits.ru/public/books>

Приложение А

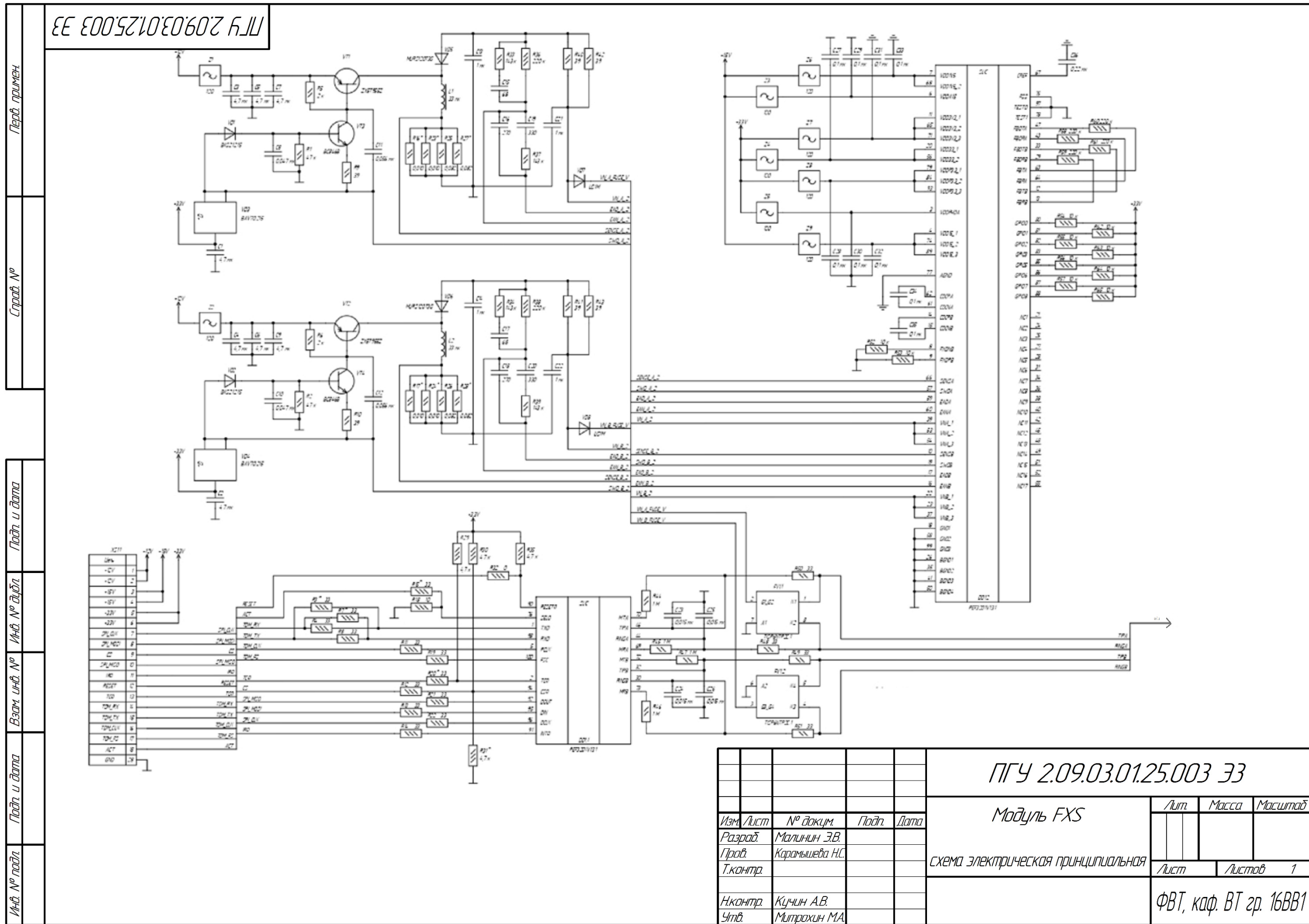
Структурная Схема

					ПГУ 2.090301.25.001 ПЗ	Лист
						65
Изм.	Лист	№ докум	Подпись	Дата		



Приложение Б
Принципиальная Схема

					ПГУ 2.090301.25.001 ПЗ	Лист
						65
Изм.	Лист	№ докум	Подпись	Дата		



Приложение В
Перечень элементов

					ПГУ 2.090301.25.001 ПЗ	Лист
						67
Изм.	Лист	№ докум	Подпись	Дата		

Поз. обозна- чение	Наименование	Кол.	Примечание
	Конденсаторы		
C1-C7	GRM21BR61H475KE51L	7	Импорт
C8	GCJ188R92A473KA01D	1	Импорт
C9	GRM21BR61H475KE51L	1	Импорт
C10	GCJ188R92A473KA01D	1	Импорт
C11,C12	C1206C563J5GACTU	2	Импорт
C13,C14	GRM21BC72A105KE01L	2	Импорт
C15	GCM1885C2A680JA16D	1	Импорт
C16	GRM21A5C2D271JW01D	1	Импорт
C17	GCM1885C2A680JA16D	1	Импорт
C18	GRM21A5C2D271JW01D	1	Импорт
C19,C20	GCM1885C2A331JA16D	2	Импорт
C21,C22	GRM21BC72A105KE01L	2	Импорт
C23-C26	GRM21BR72E153KW03L	4	Импорт
C27-C35	GCM188L81H104KA57D	9	Импорт
C36	GCM188R71H224KA64D	1	Импорт
C38*	GCJ43DR72E474KXJ1L	1	Импорт
C39,C40	GCJ43DR72E334KXJ1L	2	Импорт
C41,C42	GCJ188R92A473KA01D	2	Импорт
C43,C44	GCJ43DR72E334KXJ1L	2	Импорт

					ПГУ 2.090301.25.003 ПЭ					
Изм.	Лист	№ докум.	Подп.	Дата						
Разраб.		Малинин Э.В.			Модуль FXS Перечаяень элементов		Лит.	Лист	Листов	
Пров.		Карамышева Н.С.							1	4
Н.контр		Кучин А.В.					ФВТ, каф ВТ гр. 16ВВ1			
Утв.		Митрохин М.А.								

Поз. обозна- чение	Наименование	Кол.	Примечание
	Микросхемы		
DA6*	CPC1008N	1	Импорт
DA7*	B4S-E3/80	1	Импорт
DA8,DA9*	PS2703-1-F3-A	2	Импорт
DA10*	CPC1008N	1	Импорт
DD1	PEF3201V1.3	1	Импорт
	Диоды		
VD1,VD2	BAS21.215	2	Импорт
VD3,VD4	BAV70	2	Импорт
VD5,VD6	MURS120T3G	2	Импорт
VD7,VD8	US1M	2	Импорт
	Катушки индуктивности		
L1,L2	SRR1050A-330Y	2	Импорт
L3,L4	SRR1005-332K	2	Импорт
	Резисторы		
R1,R2	RC0805JR-0747KL	2	Импорт
R3*	RC0603JR-0733RL	1	Импорт
R4	RC0603JR-0733RL	1	Импорт
R5,R6	RC0805JR-072KL	2	Импорт
R7*	RC0603JR-0733RL	1	Импорт
R8	RC0603JR-0733RL	1	Импорт
R9,R10	RC0805JR-0739RL	2	Импорт

					ПГУ 2.090301.25.003 ПЭ	Лист
Изм	Лист	№ докум.	Подп.	Дата		2

Поз. обозна- чение	Наименование	Кол.	Примечание
	Резисторы		
R11-R14	RC0603JR-0733RL	4	Импорт
R15*	RC0603JR-0733RL	1	Импорт
R16,R17*	RL1206FR-070R01L	2	Импорт
R18	RC0603JR-0710RL	1	Импорт
R19	RC0603JR-0733RL	1	Импорт
R20*	RC0603JR-0733RL	1	Импорт
R21,R22	RC0603JR-0733RL	2	Импорт
R23,R24*	RL1206FR-070R01L	2	Импорт
R25,R26	RL1206FR-070R082L	2	Импорт
R27,R28*	RL1206FR-070R082L	2	Импорт
R29,R30	RC0603JR-074K7L	2	Импорт
R31*	RC0603JR-074K7L	1	Импорт
R32	RC0603JR-070RL	1	Импорт
R33,R34	RC0805FR-07143KL	2	Импорт
R35	RC0603JR-074K7L	1	Импорт
R36	RC0603JR-07220KL	1	Импорт
R37	RC0805FR-07143KL	1	Импорт
R38	RC0603JR-07220KL	1	Импорт
R39	RC0805FR-07143KL	1	Импорт
R40-R43	RC1206JR-0739RL	4	Импорт
R44-R47	RT1206FRE071ML	4	Импорт
R48-R51	RC1206JR-0733RL	4	Импорт
R52-R57	RC0603JR-0710KL	6	Импорт
R58-R61	RC0603JR-07220KL	4	Импорт
R62-R65	RC0603JR-0710KL	4	Импорт
R69,R70*	RC0603JR-07620RL	2	Импорт
R71,R73*	RC0603JR-073K3L	2	Импорт
R77	RC0805JR-070RL	1	Импорт

					ПГУ 2.090301.25.003 ПЭ		Лист
Изм.	Лист	№ докум.	Подп.	Дата			3

[illegible]

Приложение Г

Листинг TAPI High Level

drv_tapi

```
#ifndef DRV_TAPI_H
#define DRV_TAPI_H
    \file drv_tapi_cid.h
    Interface of the TAPI caller id implementation.
    This file contains the declaration of the functions for the CID
    module.
    */

#if (TAPI_CFG_FEATURES & TAPI_FEAT_CID)

extern IFX_int32_t IFX_TAPI_CID_SetPredefAlertTones (IFX_void_t);

extern IFX_int32_t IFX_TAPI_CID_Initialise_Unprot (TAPI_CHANNEL
*pChannel);

extern IFX_void_t IFX_TAPI_CID_Cleanup (TAPI_CHANNEL *pChannel);

extern IFX_int32_t TAPI_Phone_CID_SetConfig (TAPI_CHANNEL *pChannel,
    IFX_TAPI_CID_CFG_t const *pCidConf);

extern IFX_int32_t TAPI_Phone_CID_Info_Tx (TAPI_CHANNEL *pChannel,
    IFX_TAPI_CID_MSG_t const *pCidInfo);

extern IFX_int32_t TAPI_Phone_CID_Seq_Tx (TAPI_CHANNEL *pChannel,
    IFX_TAPI_CID_MSG_t const *pCidInfo);

extern IFX_int32_t TAPI_Phone_CID_Stop_Tx(TAPI_CHANNEL *pChannel);

extern IFX_int32_t TAPI_Phone_CidRx_Start (TAPI_CHANNEL *pChannel,
    IFX_TAPI_CID_RX_CFG_t const *pCidRxCfg);

extern IFX_int32_t TAPI_Phone_CidRx_Stop (TAPI_CHANNEL *pChannel,
    IFX_TAPI_CID_RX_CFG_t const *pCidRxCfg);

extern IFX_int32_t TAPI_Phone_Get_CidRxData (TAPI_CHANNEL *pChannel,
    IFX_TAPI_CID_RX_DATA_t *pCidRxData);

extern IFX_int32_t TAPI_Phone_CidRx_Status (TAPI_CHANNEL *pChannel,
    IFX_TAPI_CID_RX_STATUS_t *pCidRxStatus);

extern IFX_boolean_t TAPI_Cid_UseSequence (TAPI_CHANNEL
*pChannel);

/* trigger function called by ringing to indicate a ring pause */
extern IFX_void_t IFX_TAPI_CID_OnRingpause (TAPI_CHANNEL *pChannel);

/* trigger function to indicate a hook event to CID */
extern IFX_void_t IFX_TAPI_CID_OnHookEvent (TAPI_CHANNEL *pChannel,
    IFX_boolean_t bOffhook, IFX_boolean_t *bSendHookEvent);

/* trigger function to indicate a DTMF event to CID */
extern IFX_boolean_t IFX_TAPI_CID_OnDtmfEvent (TAPI_CHANNEL *pChannel,
```

					ПГУ 2.090301.25.001 ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		72

```

        IFX_uint8_t nDtmfAscii);

/* trigger function to indicate a CidRxEnd event to CID */
extern IFX_void_t IFX_TAPI_CID_OnCidRxEndEvent (TAPI_CHANNEL
*pChannel);

extern IFX_int32_t TAPI_Cid_RingStart (TAPI_CHANNEL *pChannel);

#endif /* (TAPI_CFG_FEATURES & TAPI_FEAT_CID) */

#endif /* _DRV_TAPI_CID_H */

```

drv_tapi_ring

```

/** \file drv_tapi_ring.c
    Contains TAPI Ringing Services.
    Implementation of the TAPI ring state machine which controls
    ringing on
    analog lines.
    \remarks
    All operations done by functions in this module are operating on
    analog
    phone lines and require a file descriptor with an ALM module. */

/* ===== */
/* Includes */
/* ===== */
#include "drv_tapi.h"
#include "drv_tapi_errno.h"
#include "drv_tapi_cid.h"
#include "drv_tapi_ppd.h"

/* ===== */
/* Local macros and definitions */
/* ===== */

/* ===== */
/* Type declarations */
/* ===== */
struct TAPI_RING_DATA
{
    /* timer id for ringing service */
#ifdef TAPI_HAVE_TIMERS
    /* timer id for voice path establishment */
    Timer_ID RingTimerID;
#endif /* TAPI_HAVE_TIMERS */
    /* is channel in ringing mode? */
    volatile IFX_boolean_t bRingingMode;
    /* is channel in ring_burst mode? */
    IFX_boolean_t bIsRingBurstState;
    /* cadence data */
    IFX_TAPI_RING_CADENCE_t RingCadence;
    /* counter to keep the current position in the cadence */
    IFX_uint16_t nBitPos;
    /* pointer to the cadence we currently use */
    IFX_uint8_t *pCurrentCadence;
}

```

					ПГУ 2.090301.25.001 ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		73

```

/* copy of the number of bits in the cadence we currently use */
IFX_uint16_t          BitsInCurrentCadence;
/* indicates that this is the final sequence of the last ring */
/* Set to false until proven otherwise */
*bLast = IFX_FALSE;

/* Get the current cadence bit pattern */
pBuf = pRingData->pCurrentCadence;
/* nBitsInCadence has a range from 1 to n */
nBitsInCadence = pRingData->BitsInCurrentCadence;

/* Get the first byte of the current sequence

the duration of the sequence.

\param   pChannel   Pointer to TAPI_CHANNEL structure.

\return
- TAPI_statusOk: if successful
- IFX_ERROR: in case of an error
*/
static IFX_int32_t ifx_tapi_ring_cadence_play(TAPI_CHANNEL *pChannel)
{
    TAPI_RING_DATA_t *pRingData = pChannel->pTapiRingData;
    IFX_boolean_t bBurst, bLast;
    IFX_uint32_t nTime = 0;
    IFX_int32_t ret = TAPI_statusOk;

    /* The final flag indicates if this call was done after the last
sequence
of the last cadence was played and so final actions need to be
performed.
If not the next sequence is determined the linemode set and the
timer
started for the duration of the sequence. */
    if (pRingData->bFinal == IFX_FALSE)
    {
        /* get the duration of the sequence and if this is a burst or a
pause */
        nTime = ifx_tapi_ring_get_next_time(pRingData, &bBurst, &bLast);

        /* shows the values of each step that is done
TRACE(TAPI_DRV, DBG_LEVEL_LOW,
("RINGING: time:%d, burst:%d, last:%d\n", nTime, bBurst,
bLast));
*/

        if (bBurst)
        {
            /* ring burst */
            pRingData->bIsRingBurstState = IFX_TRUE;
            ret = TAPI_Phone_Set_Linefeed (pChannel,
IFX_TAPI_LINE_FEED_RING_BURST);
        }
        else
        {
            /* ring pause */
            pRingData->bIsRingBurstState = IFX_FALSE;

```

```

        ret = TAPI_Phone_Set_Linefeed (pChannel,
IFX_TAPI_LINE_FEED_RING_PAUSE);
    }
    if (!TAPI_SUCCESS (ret))
    {
        /* On error set flag that this is the final sequence */
        pRingData->bFinal = IFX_TRUE;
    }

    /* if this is the last sequence of one cadence decrease ringing
count */
    if ((bLast == IFX_TRUE) && (pRingData->nRingsLeft > 0))

```

					ПГУ 2.090301.25.001 ПЗ	Лист
						75
Изм.	Лист	№ докум	Подпись	Дата		

Приложение Д

Листинг TAPI Low Level

drv_dxt

```

#ifndef _DRV_DXT_H_
#define _DRV_DXT_H_
\file drv_dxt.h
    This file contains all device specific definitions, e.g. offsets,
    registers,
    bits, etc.
*/

/* Examples how to use the register macros

Set register bits with one access:
*SSC_MCON = SSC_MCON_BM_VAL(7) | SSC_MCON_PH | SSC_MCON_HB;
Set one field value:
SSC_MCON_BM_SET(*SSC_MCON, 7);
Read out, modify and write back:
regVal = *SSC_MCON;
SSC_MCON_BM_SET(regVal, 7);
SSC_MCON_PH_SET(regVal, 1);
*SSC_MCON = regVal;
*/
/*
=====
==== */
/*                                     Includes
*/
/*
=====
==== */

/*
=====
==== */
/*                                     Macro definitions
*/
/*
=====
==== */
#define DXT_HOST_BASE (0x0000)

/* DuSLIC-xT has two PCM highways internally */
#define DXT_PCM_HIGHWAY      2

/* Size of the command in mailbox (Command Box In) in bytes */
#define DXT_CBI_SIZE         64

/* Size of the event mailbox (Event Box Out) in bytes */
#define DXT_EBO_SIZE         64

/*****
*****
* Register offsets

```

					ПГУ 2.090301.25.001 ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		76

```

*****
*****/
#define DXT_HOST_CFG      ((IFX_vuint32_t) (DXT_HOST_BASE + 0x06))
#define DXT_HOST_IEN1     ((IFX_vuint32_t) (DXT_HOST_BASE + 0x08))
#define DXT_HOST_STAT1    ((IFX_vuint32_t) (DXT_HOST_BASE + 0x0C))
#define DXT_HOST_INT1     ((IFX_vuint32_t) (DXT_HOST_BASE + 0x10))
#define DXT_HOST_IEN2     ((IFX_vuint32_t) (DXT_HOST_BASE + 0x0A))
#define DXT_HOST_STAT2    ((IFX_vuint32_t) (DXT_HOST_BASE + 0x0E))
#define DXT_HOST_INT2     ((IFX_vuint32_t) (DXT_HOST_BASE + 0x12))
#define DXT_HOST_DATA_CB  ((IFX_vuint32_t) (DXT_HOST_BASE + 0x14))
#define DXT_HOST_DATA_EB  ((IFX_vuint32_t) (DXT_HOST_BASE + 0x16))
#define DXT_HOST_LEN_CB   ((IFX_vuint32_t) (DXT_HOST_BASE + 0x18))
#define DXT_HOST_LEN_EB   ((IFX_vuint32_t) (DXT_HOST_BASE + 0x1A))
#define DXT_HOST_CMD      ((IFX_vuint32_t) (DXT_HOST_BASE + 0x1C))
#define DXT_HOST_REG0     ((IFX_vuint32_t) (DXT_HOST_BASE + 0x20))
#define DXT_HOST_REG1     ((IFX_vuint32_t) (DXT_HOST_BASE + 0x22))
#define DXT_HOST_REG2     ((IFX_vuint32_t) (DXT_HOST_BASE + 0x24))
#define DXT_HOST_REG4     ((IFX_vuint32_t) (DXT_HOST_BASE + 0x28))
#define DXT_HOST_REG5     ((IFX_vuint32_t) (DXT_HOST_BASE + 0x2A))
#ifdef EVALUATION
#define DXT_HOST_REG15     ((IFX_vuint32_t) (DXT_HOST_BASE + 0x3E))
#endif /* #ifdef EVALUATION */

/*****
*****
* HOST Configuration Register

*****
*****/
/* Reset value */
#define DXT_HOST_CFG_RESET      0x0000

/* Reserved (15:8) */
#define DXT_HOST_CFG_RES0      (0xff << 8)
#define DXT_HOST_CFG_RES0_VAL(val)  (((val) & 0xff) << 8)
#define DXT_HOST_CFG_RES0_GET(val)  (((val) & DXT_HOST_CFG_RES) >> 8) & 0xff)
#define DXT_HOST_CFG_RES0_SET(reg,val) ((reg) = ((reg & ~DXT_HOST_CFG_RES) | (((val) & 0xff) << 8)))
/* Reset (7) */
#define DXT_HOST_CFG_RST      (0x1 << 7)
#define DXT_HOST_CFG_RST_VAL(val)  (((val) & 0x1) << 7)
#define DXT_HOST_CFG_RST_SET(reg,val) ((reg) = ((reg & ~DXT_HOST_CFG_RST) | ((val) & 1) << 7))
/* Reserved (6:2) */
#define DXT_HOST_CFG_RES1      (0x1f << 2)
#define DXT_HOST_CFG_RES1_VAL(val)  (((val) & 0x1f) << 2)
#define DXT_HOST_CFG_RES1_GET(val)  (((val) & DXT_HOST_CFG_RES) >> 2) & 0x1f)
#define DXT_HOST_CFG_RES1_SET(reg,val) ((reg) = ((reg & ~DXT_HOST_CFG_RES) | (((val) & 0x1f) << 2)))
/* Self-Clearing Mode (1) */
#define DXT_HOST_CFG_SC_MD      (0x1 << 1)
#define DXT_HOST_CFG_SC_MD_VAL(val)  (((val) & 0x1) << 1)
#define DXT_HOST_CFG_SC_MD_GET(val)  (((val) & DXT_HOST_CFG_SC_MD) >> 1) & 0x1)

```

```

/*
=====
==== */
/*
                                Type definitions
*/
/*
=====
==== */
typedef struct
{
    /* address offset */
    IFX_uint16_t      offset;
    /* name of the register */
    IFX_char_t        name[15];
    /* result value in unsigned format */
    IFX_uint16_t      value;
} DXT_REG_INFO_t;

/* DuSLIC-xT specific device Structure.
   Structure element is part of the host global structure */
typedef struct
{
    /* device specific registers */
    /* PCM control */
    DXT_REG_INFO_t    regHostCfg;
    /* Extended configuration register */
    DXT_REG_INFO_t    regHostIen1;
    DXT_REG_INFO_t    regHostIen2;
} DXT_DEV_REG_t;

/*
=====
==== */
/*
                                Function prototypes
*/
/*
=====
==== */

#endif /* _DRV_DXT_H_ */

```

drv_tapi_voice

```

/**
    \file   drv_tapi_voice.c
    Contains TAPI Voice Services : Play, Recording, Conferencing.
*/

/* ===== */
/* Includes */
/* ===== */

#include "drv_tapi.h"
#include "drv_tapi_errno.h"

/* ===== */

```

					ПГУ 2.090301.25.001 ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		78

```

/* Local Macros & Definitions */
/* ===== */

/* ===== */
/* Local variable definition */
/* ===== */

/* ===== */
/* Local function definition */
/* ===== */

/* ===== */
/* Global function definition */
/* ===== */

#if (TAPI_CFG_FEATURES & TAPI_FEAT_VOICE)
/**
    This interface adds a data channel to an analog phone device

    \param pChannel      Pointer to TAPI_CHANNEL structure.
    \param pMap          Pointer to IFX_TAPI_MAP_DATA_t structure.

    \return
    IFX_SUCCESS or IFX_ERROR
*/
IFX_int32_t TAPI_Data_Channel_Add (TAPI_CHANNEL *pChannel,
                                   IFX_TAPI_MAP_DATA_t const *pMap)
{
    IFX_TAPI_DRV_CTX_t *pDrvCtx = pChannel->pTapiDevice->pDevDrvCtx;
    IFX_int32_t ret = TAPI_statusOk, retLL = TAPI_statusOk;

    switch (pMap->nPlayStart)
    {
        case IFX_TAPI_MAP_DATA_UNCHANGED:
            /* do nothing */
            break;
        case IFX_TAPI_MAP_DATA_START:
            if (IFX_TAPI_PtrChk (pDrvCtx->COD.DEC_Start))
                retLL = pDrvCtx->COD.DEC_Start (pChannel->pLLChannel);
            else
                RETURN_STATUS (TAPI_statusNotSupported, 0);
            break;
        case IFX_TAPI_MAP_DATA_STOP:
            if (IFX_TAPI_PtrChk (pDrvCtx->COD.DEC_Stop))
                retLL = pDrvCtx->COD.DEC_Stop (pChannel->pLLChannel);
            else
                RETURN_STATUS (TAPI_statusNotSupported, 0);
            break;
    }
    if (TAPI_SUCCESS(retLL))
    {
        switch (pMap->nRecStart)
        {
            case IFX_TAPI_MAP_DATA_UNCHANGED:
                /* do nothing */
                break;
            case IFX_TAPI_MAP_DATA_START:
                if (IFX_TAPI_PtrChk (pDrvCtx->COD.ENC_Start))

```

```

        retLL = pDrvCtx->COD.ENC_Start (pChannel->pLLChannel);
    else
        RETURN_STATUS (TAPI_statusNotSupported, 0);
    break;
case IFX_TAPI_MAP_DATA_STOP:
    if (IFX_TAPI_PtrChk (pDrvCtx->COD.ENC_Stop))
        retLL = pDrvCtx->COD.ENC_Stop (pChannel->pLLChannel);
    else
        RETURN_STATUS (TAPI_statusNotSupported, 0);
    break;
}
}

/* call low level function here */
if (TAPI_SUCCESS (retLL))
{
    /* Pass parameters to LL implementation. */
    ret = pDrvCtx->CON.Module_Find_Connected_Data_Channel (
        pChannel->pLLChannel, nModType,
pTapiCh);
}
else
{
    /* Report a single straight connection between module and data
channel. */
    if (*pTapiCh == IFX_NULL)
    {
        *pTapiCh = pChannel;
    }
    else
    {
        *pTapiCh = IFX_NULL;
    }
}

return ret;
}

/**
Set the AMR codec specific parameters.

\param pChannel    Pointer to TAPI_CHANNEL structure.
\param pTapiAMR    Pointer to IFX_TAPI_COD_AMR_t structure.

\return
TAPI_statusOk if successful, otherwise error code
*/
IFX_int32_t IFX_TAPI_COD_AMR_Set (TAPI_CHANNEL *pChannel,
                                IFX_TAPI_COD_AMR_t *pTapiAMR)
{
    IFX_TAPI_DRV_CTX_t *pDrvCtx = pChannel->pTapiDevice->pDevDrvCtx;
    IFX_int32_t retLL;

    if (!IFX_TAPI_PtrChk (pDrvCtx->COD.AMR_Set))
        RETURN_STATUS (TAPI_statusLLNotSupp, 0);

    retLL = pDrvCtx->COD.AMR_Set (pChannel->pLLChannel, pTapiAMR);
}

```

```

    if (!TAPI_SUCCESS(retLL))
        RETURN_STATUS (TAPI_statusLLFailed, retLL);

    return TAPI_statusOk;
}

/**
    Get the AMR codec specific parameters.

    \param pChannel      Pointer to TAPI_CHANNEL structure.
    \param pTapiAMR      Pointer to IFX_TAPI_COD_AMR_t structure.

    \return
    TAPI_statusOk if successful, otherwise error code
*/
IFX_int32_t IFX_TAPI_COD_AMR_Get(TAPI_CHANNEL *pChannel,
                                IFX_TAPI_COD_AMR_t *pTapiAMR)
{
    IFX_TAPI_DRV_CTX_t *pDrvCtx = pChannel->pTapiDevice->pDevDrvCtx;
    IFX_int32_t retLL;

    if (!IFX_TAPI_PtrChk (pDrvCtx->COD.AMR_Get))
        RETURN_STATUS (TAPI_statusLLNotSupp, 0);

    retLL = pDrvCtx->COD.AMR_Get (pChannel->pLLChannel, pTapiAMR);

    if (!TAPI_SUCCESS(retLL))
        RETURN_STATUS (TAPI_statusLLFailed, retLL);

    return TAPI_statusOk;
}

```

Приложение Е

Листинг драйвера IFXOS

ifxos_common

```
#ifndef _IFXOS_COMMON_H
#define _IFXOS_COMMON_H

/** \file
    This file contains common OS independent definitions for the IFX
    OS.
*/

/** \defgroup IFXOS_INTERFACE IFXOS Interface Layer

    This Group contains the definitions and functions of the Infineon /
    Lantiq OS Layer.

    Therefore the IFX OS defines and provides wrapper functions to hide
    the
    OS specifc implementations.

    Currently supported:
    - eCos
    - Linux
    - VxWorks
*/

/** \defgroup IFXOS_LAYER_LINUX IFXOS Implementation (Linux)

    This Group contains the mapping and implementation layer for the
    Linux OS.

    Therefore the Linux Layer is splitted in 2 parts:
    - Linux Kernel Layer
    - Linux User Layer
*/

/** \defgroup IFXOS_LAYER_VXWORKS IFXOS Implementation (VxWorks)

    This Group contains the mapping and implementation layer for the
    VxWorks OS.

    The VxWorks OS doesn't enforce a clear split between driver and
    application code, but for a clear structure and compatibility to
    Linux the
    IFX OS Layer keeps this border between driver and application code.

    \note
    Concerning to this split, there are functions which are related to
    driver code
*/

/** \defgroup IFXOS_COMMON IFXOS Common Defines

    This Group contains the common OS independant definition.
*/
```

```

#ifdef __cplusplus
    extern "C" {
#endif

```

ifxos_linux_common

```

#ifdef LINUX
/** \file
    This file contains common definitions used for the Linux OS
    adaptation
    */

```

```

#ifdef __cplusplus
    extern "C" {
#endif

```

```

#ifndef _IFXOS_COMMON_H
#   error "missing IFXOS endian defines, include 'ifx_common.h' at
    first"
#endif

```

```

#ifdef __KERNEL__
/*

```

```

=====
=====

```

IFX Linux adaptation - Linux Kernel Space

```

=====
=== */

```

```

/** \defgroup IFXOS_IF_LINUX_DRV Defines for Linux Kernel Adaptaion

```

This Group contains the Linux specific definitions and function.

```

\par Linux Kernel Endianess

```

Under Linux in Kernel space the Endianess is defined within the corresponding architecture header files.

The plattform endianess is mapped to the internal used IFXOS endianess definitons.

```

\attention

```

Under Linux only the `_LITTLE_ENDIAN` or the `__BIG_ENDIAN` is set, so avoid to use this settings directly !!!

```

\ingroup IFXOS_INTERFACE

```

```

*/
/*

```

```

=====
=====

```

IFX Linux adaptation - Includes (Linux Kernel)

```

=====
=== */

```

```

#include <linux/kernel.h>
#include <asm/byteorder.h>

```

```

/*

```

					ПГУ 2.090301.25.001 ПЗ	Лист
						83
Изм.	Лист	№ докум	Подпись	Дата		

Приложение Ж

Листинг драйвера TDM

fsl_sai

```
#ifndef FSL_SAI_H_
#define FSL_SAI_H_

// #define DEBUG
#include <linux/device.h>
#include <linux/types.h>
#include <linux/busfreq-imx.h>
#include <linux/clock.h>
#include <linux/delay.h>

#include <linux/module.h>
#include <linux/of_address.h>
#include <linux/platform_device.h>
#include <linux/regmap.h>
#include <linux/slab.h>
#include <linux/interrupt.h>
#include <linux/irqreturn.h>
#include <linux/cdev.h>
#include <linux/fs.h>
#include <linux/uaccess.h>
#include <linux/wait.h>
#include <linux/sched.h>

#include <linux/dmaengine.h>
#include <linux/dma-direction.h>
#include <linux/dma-mapping.h>

/* SAI Register Map Register */
#define FSL_SAI_TCSR 0x00 /* SAI Transmit Control */
#define FSL_SAI_TCR1 0x04 /* SAI Transmit Configuration 1 */
#define FSL_SAI_TCR2 0x08 /* SAI Transmit Configuration 2 */
#define FSL_SAI_TCR3 0x0c /* SAI Transmit Configuration 3 */
#define FSL_SAI_TCR4 0x10 /* SAI Transmit Configuration 4 */
#define FSL_SAI_TCR5 0x14 /* SAI Transmit Configuration 5 */
#define FSL_SAI_TDR 0x20 /* SAI Transmit Data */
#define FSL_SAI_TFR 0x40 /* SAI Transmit FIFO */
#define FSL_SAI_TMR 0x60 /* SAI Transmit Mask */
#define FSL_SAI_RCSR 0x80 /* SAI Receive Control */
#define FSL_SAI_RCR1 0x84 /* SAI Receive Configuration 1 */
#define FSL_SAI_RCR2 0x88 /* SAI Receive Configuration 2 */
#define FSL_SAI_RCR3 0x8c /* SAI Receive Configuration 3 */
#define FSL_SAI_RCR4 0x90 /* SAI Receive Configuration 4 */
#define FSL_SAI_RCR5 0x94 /* SAI Receive Configuration 5 */
#define FSL_SAI_RDR 0xa0 /* SAI Receive Data */
#define FSL_SAI_RFR 0xc0 /* SAI Receive FIFO */
#define FSL_SAI_RMR 0xe0 /* SAI Receive Mask */

#define FSL_SAI_xCSR(tx) (tx ? FSL_SAI_TCSR : FSL_SAI_RCSR)
#define FSL_SAI_xCR1(tx) (tx ? FSL_SAI_TCR1 : FSL_SAI_RCR1)
#define FSL_SAI_xCR2(tx) (tx ? FSL_SAI_TCR2 : FSL_SAI_RCR2)
#define FSL_SAI_xCR3(tx) (tx ? FSL_SAI_TCR3 : FSL_SAI_RCR3)
#define FSL_SAI_xCR4(tx) (tx ? FSL_SAI_TCR4 : FSL_SAI_RCR4)
```

```
#define FSL_SAI_xCR5(tx)      (tx ? FSL_SAI_TCR5 : FSL_SAI_RCR5)
#define FSL_SAI_xDR(tx)      (tx ? FSL_SAI_TDR : FSL_SAI_RDR)
#define FSL_SAI_xFR(tx)      (tx ? FSL_SAI_TFR : FSL_SAI_RFR)
#define FSL_SAI_xMR(tx)      (tx ? FSL_SAI_TMR : FSL_SAI_RMR)
```

Приложение 3

Презентация

					ПГУ 2.090301.25.001 ПЗ	Лист
						86
Изм.	Лист	№ докум	Подпись	Дата		

Тема работы: «Узел интерфейса FXS для учрежденческих автоматических телефонных станций»

Автор работы: студент группы 16BB1 Малинин Э.В.
Научный руководитель : к.т.н., доцент каф.ВТ ПГУ Карамышева Н.С.

Постановка задачи

Цель работы: разработка узла интерфейса FXS для
учрежденческих автоматических телефонных
станций.

Основные задачи:

- анализ взаимодействия существующих микросхем узла интерфейса FXS;
- разработка программного обеспечения для взаимодействия с портом FXS.

Основные функции FXS

- питание телефонного аппарата;
- определения положения телефонной трубки;
- наличие детекторов цифр, набранных на телефонном аппарате;
- формирование напряжения на телефонном аппарате при приходе вызова с IP–сети.

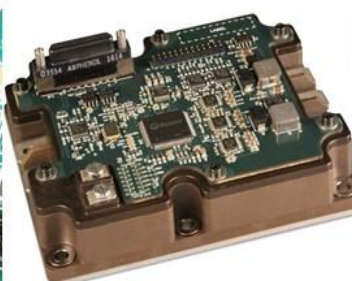
Подзадачи

- Провести анализ взаимодействия узла FXS с абонентской линией;
- Рассмотреть существующие аналоги;
- Разработать принципиальную схему узла FXS;
- Написать программу взаимодействия с модулем FXS.

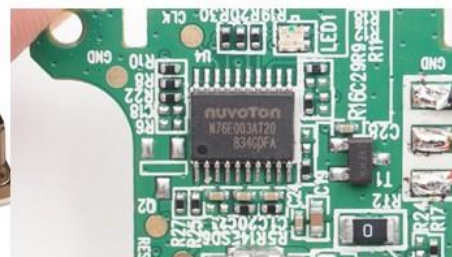
Существующие аналоги FXS



Ref3201 фирмы Lantiq

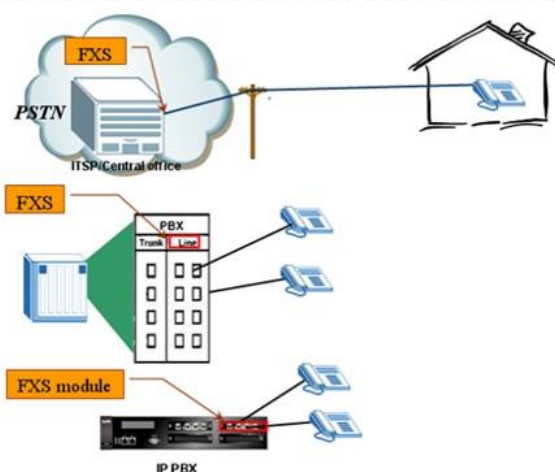


ZLR964321L фирмы Microsemi



N681622 фирмы Nuvoton

Анализ взаимодействия узла FXS с абонентской линией

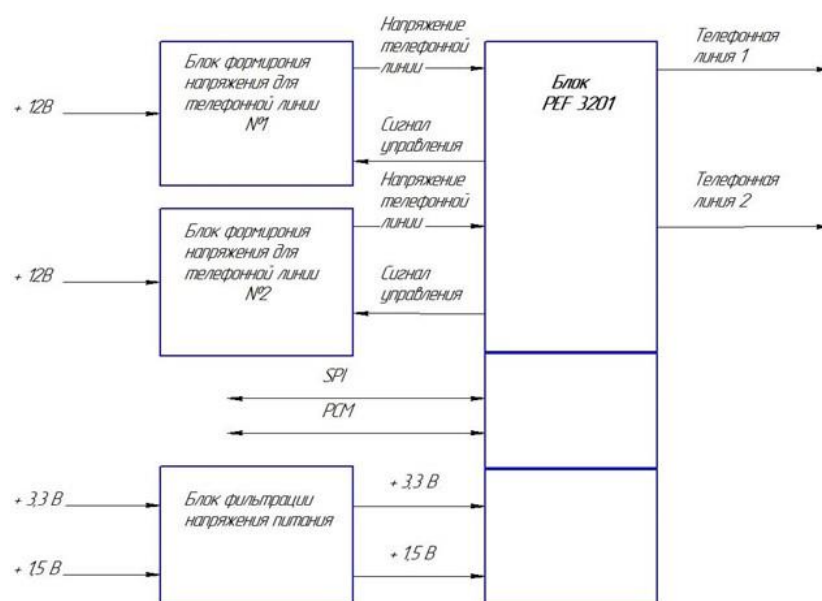


Протоколы взаимодействия:

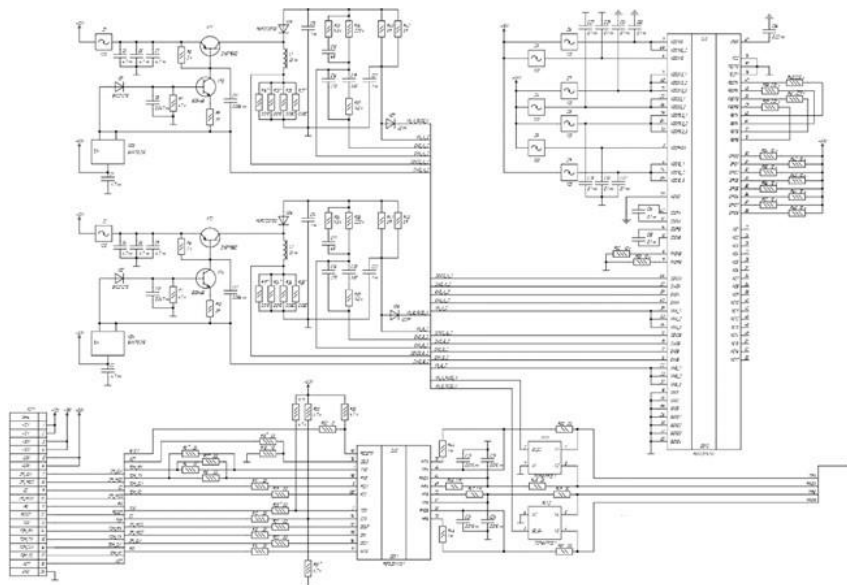
- SIP
- RTP
- Интерфейс TAPI

Изм.	Лист	№ докум	Подпись	Дата

Структурная схема



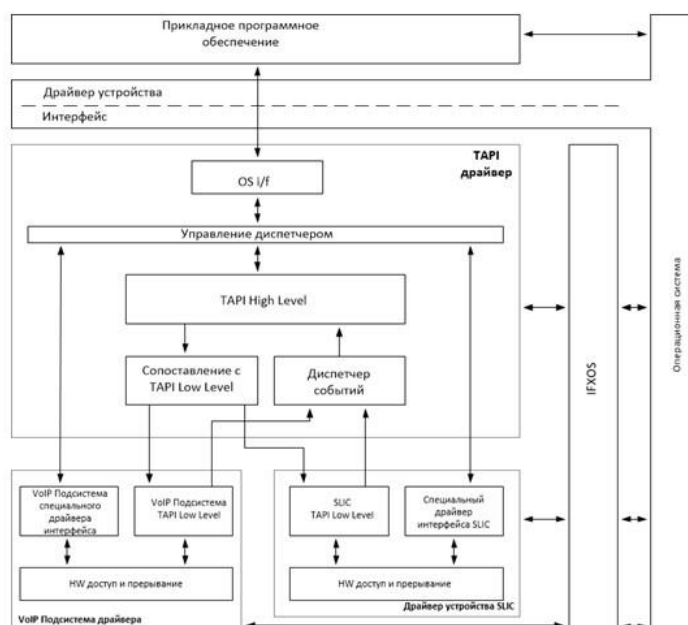
Принципиальная схема



Программа взаимодействия с модулем FXS

Основной листинг программы:

- TAPI High Level
- TAPI Low Level
- IFXOS
- TDM



Заключение

В процессе ВКР было сделано:

- провели анализ взаимодействия узла FXS с абонентской линией;
- рассмотрели существующие аналоги микросхем;
- разработали структурную и принципиальную схему модуля узла интерфейса FXS;
- разработана программа взаимодействия с модулем FXS.

Спасибо за внимание

					ПГУ 2.090301.25.001 ПЗ	Лист
						92
Изм.	Лист	№ докум	Подпись	Дата		