МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Интеллектуальные технические системы

А.Д. Королев, Н.А. Кореневский, Д.Н. Кузнецов,
Нгуен Тхе Кыонг, Ю.П. Муха,
В.И. Сырямкин, Д.В. Титов

ИНТЕЛЛЕКТУАЛЬНЫЕ ПРОГРАММНО-АППАРАТНЫЕ КОМПЛЕКСЫ ПЕРЕДАЧИ ИНФОРМАЦИИ В ТЕЛЕМЕДИЦИНСКИХ СЕТЯХ

2-е издание, исправленное и дополненное

Томск
Издательский Дом Томского государственного университета
2019

УДК 330.51-7
ББК 65.011.151
К68

Королев А.Д., Кореневский Н.А., Кузнецов Д.Н., Нгуен Тхе Кыонг, Муха Ю.П., Сырямкин В.И., Титов Д.В.
К68 Интеллектуальные программно-аппаратные комплексы передачи информации телемедицинских сетях / под ред. Ю.П. Мухи, В.И. Сырямкина. – 2-е изд., испр. и доп. – Томск : Издательский Дом Томского государственного университета, 2019. – 360 с. (Серия: Интеллектуальные технические системы)

ISBN 978-5-94621-806-1

Монография посвящена проблеме разработки передачи данных медицинской информации. Рассмотрены вопросы теории универсального преобразования систем кодирования, проектирование узлов цифровой обработки с использованием сигнальных процессоров, синтеза и метрологического обеспечения гибкого индивидуального преобразователя, а также схемная реализация и применение универсальных программно-аппаратных комплексов в телемедицине.

Для широкого круга специалистов в области телемедицины, приборостроения и робототехники.

УДК 330.51-7
ББК 65.011.151

Работа выполнена при финансовой поддержке РФФИ (грант № 16-29-04388/16 от 19.04.2016 г.), а также по программе повышения конкурентоспособности Национального исследовательского Томского государственного университета

© Королев А.Д., Кореневский Н.А., Кузнецов Д.Н., Нгуен Тхе Кыонг, Муха Ю.П., Сырямкин В.И., Титов Д.В., 2019
ISBN 978-5-94621-806-1
© Томский государственный университет, 2019

СПИСОК СОКРАЩЕНИЙ

ПТК – программно-технический комплекс
ПАК – программно-аппаратный комплекс
РЗ – рабочая зона
ДИ – датчик изображения
БП – блок подсветки
БОИ – блок обработки изображения
БА – блок адаптации
БОП – блок оценки параметров
ГИИ – гибкий интеллектуальный интерфейс
DICOM – Digital Imaging and COmmunications in Medicine (цифровые изображения и обмен ими в медицине)
АИС ТМК – автоматизированная информационная система телемедицинских консультаций

ОПРЕДЕЛЕНИЯ

Интеллектуальные системы (ИС) – это техническая или программная система, способная решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти такой системы. Структура интеллектуальной системы включает три основных блока – базу знаний, механизм вывода решений и интеллектуальный интерфейс

ИС – это компьютерная система, состоящая из основных взаимодействующих компонентов: языковой подсистемы (механизм обеспечения связи между пользователем и другими компонентами), информацией подсистемы (хранилище данных и средств их обработки), подсистемы управления знаниями (хранилище знаний о проблемной области, таких как процедуры, эвристики и правила, и средства обработки знаний), подсистемы управления моделями и подсистемы обработки и решения задач (связующее звено между другими подсистемами).

Интерфейсы – общая граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил взаимодействия (управления, контроля и т.д.) между элементами системы.

Интерфейс (ударение на вторую «е») это, в широком смысле, набор инструментов для взаимодействия человека и компьютерной системы. Этими инструментами чаще всего являются кнопки, галочки, текстовые поля, подсказки, переключатели, выпадающие списки и прочие ухищрения. В дословном переводе с английского слово «interface» означает «[взаимодействие] между лицами». В объектно ориентированных языках программирования интерфейсом называется шаблон класса, все методы которого являются абстрактными, и поэтому должны быть реализованы в каждом классе, наследующем этот интерфейс.

ПРЕДИСЛОВИЕ

В современном мировом обществе актуальной является развитие телемедицины, обеспечивающая эффективную диагностику и лечение людей. Телемедицина является основой современной цифровой медицины, реализующей дистанционную диагностику и терапию пациентов. Здесь используются различные виды связи (интернет, телефония и специальная), включая спутниковую связь, которая осуществляет связь в труднодоступных районах и чрезвычайных ситуациях. При этом актуальной является повышение быстродействия помехоустойчивости и помехозащищённости телекоммуникационных систем, а также создание медицинских систем с элементами искусственного интеллекта.

В монографии рассмотрены актуальные вопросы проектирования, синтеза, анализа интеллектуальных программно-аппаратных комплексов передачи информации в телемедицинских сетях. Проведены обзор распределенных телемедицинских систем, основы теории универсального преобразования систем кодирования, результаты моделирования, реализации и метрологического обеспечения гибких интеллектуальных подсистем, примеры практического применения интеллектуальных программно-аппаратных комплексов (обмен видеоинформацией в телемедицине, дистанционная идентификация личности человека).

Разработанное алгоритмическое, аппаратурное и программное обеспечение может быть успешно использовано в системах технического зрения и системах управления робототехническими комплексами (коллективами роботов) различного назначения: медицинскими роботами, роботами спасателями, роботами-разведчиками, роботами охранниками.

Отзывы, критические замечания и предложения по монографии просьба направлять по адресу: 634050, г. Томск, пр. Ленина 36, Национальный исследовательский Томский государственный университет или e-mail: egs@sibmail.com.

1. ОБЗОР СОВРЕМЕННЫХ РАСПРЕДЕЛЕННЫХ ИЗМЕРИТЕЛЬНО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

1.1. Описание предметной области

С бурным ростом производительности современной вычислительной техники, а также с развитием технологий приема и передачи информации, широкое распространение получают сложные распределенные измерительно-вычислительные системы. Одним из важнейших аспектов их функционирования является передача данных, их сохранность и верная интерпретация. Такие измерительно-вычислительные системы внедряются повсеместно: в промышленность, энергетику, медицину, образование. Сложность данных систем, объем обрабатываемых ими данных, последующая транспортировка сопровождаются рядом технических проблем, решение которых носит приоритетный характер.

Наиболее показательным примером работы измерительно-вычислительной системы является сфера медицины. Количество получаемой информации от медицинских приборов постоянно увеличивается, получает распространение удаленное консультирование, хранение и обработка информации. Все это требует применения специального оборудования по транспортировке, приему и передаче специализированной информации.

Современную медицину невозможно представить без соответствующего технического обеспечения, которое позволяет прогнозировать, ставить диагноз и проводить процесс лечения максимально эффективно с высокой точностью. Это приводит к применению в ЛПУ большого количества различных аппаратов, приборов, систем, комплексов и сетей различных уровней.

Общая структурная схема технических средств, применяемых в медицине для решения задач медико-биологических исследований, с указанием основных связей взаимодействия ТС различных типов и назначений, показана на рис. 1.1.

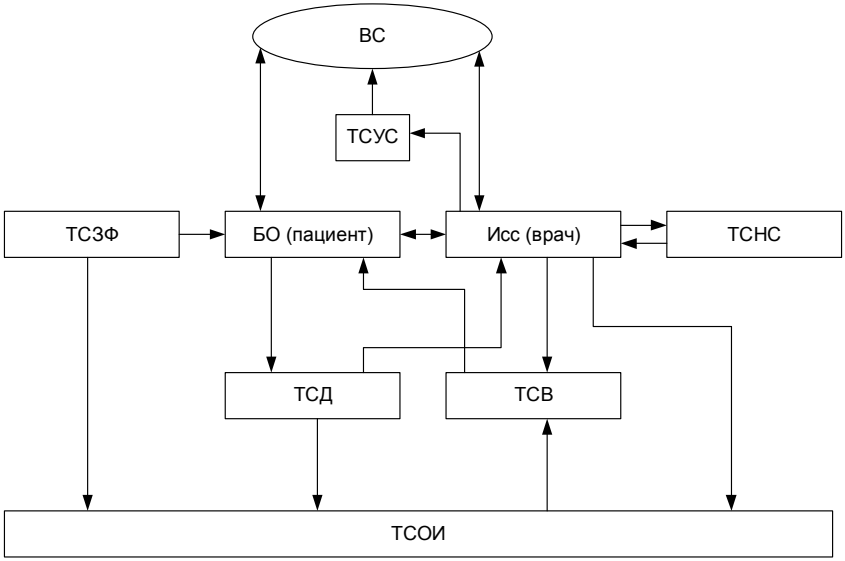


Рис. 1.1. Классификация медицинской техники

1.2. Система передачи данных медицинской информации

Как видно из приведенного примера, важнейшую роль играют методы и средства передачи, получаемой диагностической и управляющей информации. Справиться с обработкой этой информации, ее своевременной передачей можно только с внедрением информационных технологий (ИТ). Целью внедрения информационных технологий является создание информационных систем для анализа и принятия на их основе управленческих решений. Конкретным воплощением ИТ выступают автоматизированные сети и системы, и лишь в этом случае принято говорить о компьютерных технологиях. Большой объем передаваемых специализированных данных, интерактивный режим обработки приводят к необходимости применения МИС. МИС – это телемедицинская сеть. Она повышает эффективность использования медицинской информации,

оперативность ее получения, достоверность информации. Кроме того, появляется возможность оптимизации стоимости пребывания пациента в стационаре.

Телемедицинские технологии дают возможность врачу контролировать состояние пациентов в амбулаторных, домашних условиях, что особенно актуально для пациентов с хроническими заболеваниями, а также в период проведения восстановительных мероприятий после стационарного лечения, особенно в случаях оперативных вмешательств.

Видеосвязь помогает врачам-терапевтам поддерживать контакт с медицинскими центрами, осуществлять телемониторинг нетрудоспособных пациентов, находящихся дома.

Весьма актуальным поэтому является организация региональных и межрегиональных телемедицинских сетей.

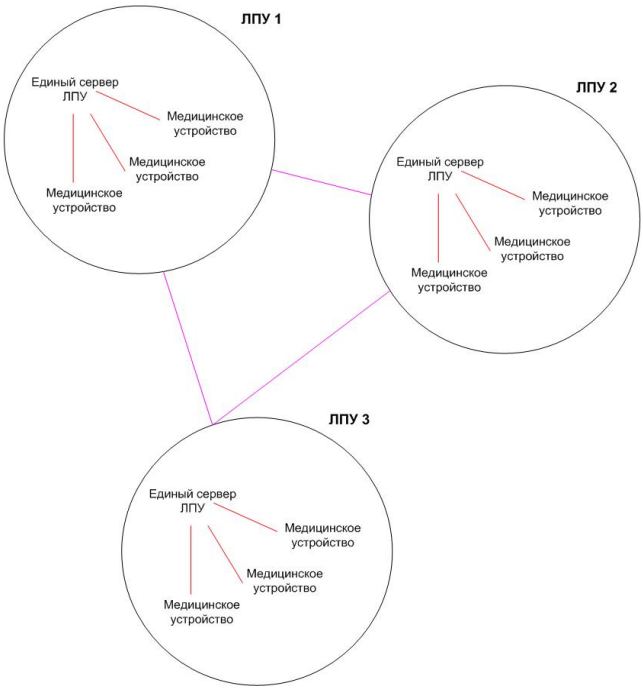


Рис. 1.2. Схема информационного взаимодействия ЛПУ

Передача медицинской информации осуществляется на нескольких уровнях: уровне взаимодействия медицинской техники внутри одного ЛПУ и на уровне взаимодействия различных ЛПУ, что показано на рис. 1.2.

Телемедицинские сети могут резко улучшить систему здравоохранения регионов. В этом случае возможно использование консультации ведущих специалистов областного центра и ведущих российских и зарубежных медицинских центров.

1.3. Классификация телемедицинских систем

Различные специалисты и научные школы рассматривают телемедицину с нескольких разных позиций от того, что телемедицина не является отдельной медицинской дисциплиной, до того что телемедицина является отдельной отраслью медицины. В данном случае определим телемедицину, как направление медицины, основанное на использовании компьютерных и телекоммуникационных технологий для обмена медицинской информацией между специалистами с целью повышения качества диагностики и лечения конкретных пациентов.

К основным областям применения телекоммуникационных технологий в медицине относят:

1) телемедицинские консультации;
2) теленаблюдение за пациентами;
3) телемедицина ургентных состояний, чрезвычайных ситуаций и катастроф;
4) телеобучение;
5) телехирургия и дистанционное обследование;
6) военная телемедицина;
7) космическая телемедицина.

Наиболее распространенными и часто применяемыми на практике являются телемедицинские консультации. Телемедицинские консультации могут быть как отложенными, так и осуществляться в режиме реального времени. Консультации в отложенном времени не требуют скоростных телекоммуникационных каналов и

сложного оборудования и, по этой причине, сравнительно дешевы. Простейшими способами реализации отложенных телеконсультаций могут быть консультации по e-mail, консультации на специализированных медицинских форумах, с помощью специализированных телемедицинских систем и т.п.

Телеконсультации в режиме реального времени требуют более сложного технического оснащения и проводятся с использованием широкополосных каналов связи и видеоаппаратуры. В отличие от отложенных, консультации в режиме реального времени могут применяться при оказании неотложной помощи. Различают плановые, экстренные видеоконсультации и видеоконсилиумы. Во всех этих случаях обеспечивается непосредственное общение между консультантом и лечащим врачом. Сеанс видеоконференции может проходить как между двумя абонентами (режим «точка-точка»), так и между несколькими (многоточечный режим). При этом обеспечивается возможность передачи практически всей необходимой информации для выработки квалифицированного заключения – выписок из истории болезни, результатов клинических анализов, различных визуальных материалов, таких как рентгенограммы, компьютерные томограммы, снимки УЗИ и т.п. Для организации передачи широкого спектра необходимой информации во время телеконсультаций в режиме реального времени используются различные протоколы передачи данных, ориентированные на работу с мультимедиа, например, такие как SIP (Session Initiation Protocol), H.320-H.324 и ряд других.

Теленаблюдение за пациентами осуществляется с помощью телемедицинских систем динамического наблюдения для наблюдения за пациентами, страдающими хроническими заболеваниями, а также на промышленных объектах повышенной опасности, таких как атомные электростанции, химические производства, для контроля состояния здоровья работников. Теленаблюдение может проводиться как в условиях стационара, так и вне его, например, в домашних условиях (home telehealth). Телемедицинские системы динамического наблюдения позволяют регистрировать физиологические параметры организма человека и передавать их на базовую станцию для дальнейшего анализа. Подобные системы целе-

сообразно использовать у пациентов, нуждающихся в непрерывном наблюдении и частых обследованиях. В телемедицине ургентных состояний, чрезвычайных ситуаций и катастроф используются мобильные телемедицинские комплексы. Такие комплексы могут быть как переносным, так и размещаться на базе передвижных средств, например реанимобиля. Современный мобильный телемедицинский комплекс строится на базе мощного компьютера, который соединен с разнообразным медицинским оборудованием и включает в себя средства для проведения видеоконференций, дальней и ближней беспроводной связи, передачи данных по сетям интернет. Такими комплексами целесообразно оснащать бригады медицины катастроф и санитарной авиации, медицинские формирования МЧС и подразделений МО. Другим интенсивно развивающимся направление телемедицины является телеобучение. В результате использования телекоммуникационных технологий у практикующегося врача появилась реальная возможность непрерывного профессионального образования без отрыва от места работы. Типовой зал для телеобучения представляет собой помещение на 20–50 человек с двумя экранами большого размера для визуализации докладчика и вспомогательного материала, а также с микрофонной и акустической системой. Такой зал предназначен для приема трансляций из операционных, удаленного чтения курса лекций, проведения видеосеминаров и конференций. Во время занятий преподаватель может иметь интерактивный контакт с аудиторией. Лекции, как и видеоконсультации могут проходить в многоточечном режиме, таким образом, лекция может быть прочитана сразу для слушателей из нескольких регионов.

Отличительной особенностью дистанционного обучения и, в особенности, телехирургия несмотря на то, что относятся в настоящее время к экспериментальному направлению телемедицины, считаются достаточно перспективными. Отличительной особенностью этого направления является активное воздействие на организм пациента специалиста, находящегося на расстоянии. Развитие в настоящее время ведется в двух направлениях: дистанционное управление медицинской диагностической аппаратурой и дистанционное проведение лечебных воздействий, хирургических

операций на основе использования дистанционно управляемой робототехники.

Военная телемедицина подразумевает использование телемедицинских технологий при подготовке и проведении военных операций и использует в основном спутниковые системы связи.

Примером современной телемедицинской системы военного назначения является мобильная система LifeBot® DREAMS™ Advanced Telemedicine. Система разработана исследовательским центром американской армии и исследовательским центром телемедицины и технологий специально для использования в войсках. Система размещается на базе автомобиля повышенной проходимости и использует для передачи данных сотовые стандарты 3G и 4G, а также Wi-Fi, WiMAX, LTE, спутниковый канал, а также военный канал цифровой связи. Она автоматически согласует ширину канала и скорость передачи с текущим потоком данных. Помимо возможности передачи голоса и видеоизображения в данной системе предусмотрена возможность регистрации и передачи различных физиологических параметров, таких как ЭКГ в 12 отведениях, давления крови, газов крови и ряда других. Все данные пациента передаются в зашифрованном виде с помощью алгоритма Advanced Encryption Standard (AES), – симметричного алгоритма блочного шифрования, принятого в качестве стандарта шифрования правительством США по результатам конкурса.

В рамках программ космической телемедицины разрабатываются специальные методы и аппаратура для дистанционной регистрации основных физиологических и биохимических параметров организма человека в условиях космического полета, в том числе и длительного, специалисты на земле следят за принятием своевременных мер по коррекции возникающих нарушений. В настоящее время космическая телемедицина не только повышает эффективность медицинского сопровождения полетов, но и позволяет на современном уровне координировать проведение международных исследовательских проектов по космической биологии и медицине.

Аппаратура для организации телемедицинских систем производится различными российскими и зарубежными компаниями, сре-

ди которых «Витанет», «СТЭЛ-Компьютерные системы», «DiViSy», «TREDEX», «AMD Global Telemedicine», «American Telecare», «Aerotel medical system» и многие другие. Сложность производимой аппаратуры, как и медицинские области применения, самые различные – от простых приборов для удаленного слежения за ЭКГ, рассчитанных на каналы связи с низкой пропускной способностью, до сложных многофункциональных комплексов, с развитой периферией, например, с рентгенологическими и морфологическими лабораториями. Так фирма «СТЭЛ–Компьютерные системы» производит мобильный телемедицинский комплекс STEL TKmobile, предназначенный для оказания оперативной дистанционной консультативно-диагностической медицинской помощи, который выполнен в виде двух пылевлагозащищенных чемоданов. В состав первого входят аппаратура и программное обеспечение для проведения видеоконференций, в состав второго-компьютерный электрокардиограф на 12 отведений, пульсоксиметр, автоматический измеритель артериального давления, глюкометр и цифровой измеритель температуры. Встроенный кодек видеоконференций обеспечивает проведение сеансов связи с высококачественным видеоизображением и звуковым сопровождением. Поддерживается работа в IP сетях на скоростях до 2 Мбит/с с использованием протокола H.323. Сбор, хранение и передача медицинских данных осуществляются с помощью компьютера, подключенного к базовой точке беспроводного доступа WiFi 802.11g. Разработчики считают эффективным использование этого комплекса в случаях оперативной оценки состояния пострадавших в авариях и катастрофах, оказания медицинской помощи жителям труднодоступных населенных пунктов, оказание медицинской помощи пассажирам на транспорте, в работе служб скорой медицинской помощи. Другим продуктом этой компании являются передвижные телемедицинские комплексы STEL TKP5, TKP6, TKP7 и TKP8, которые предназначены для проведения телеконсультаций и телеконсилиумов, дистанционной диагностики, мониторинга сложных медицинских манипуляций с использованием видеосвязи в режиме реального времени. Комплексы реализованы на базе мощного производительного компьютера, аппаратного кодека ви-

деоконференцсвязи компании Polycom и необходимого периферийного оборудования.

Примером телемедицинских систем для персонального применения являются системы, производимые фирмой Aerotel Medical Systems. Основным направлением деятельности фирмы является производство системы для мониторинга электрокардиосигнала (ЭКС) – Aerotel Heartline T. Система состоит из набора персональных регистраторов и центра приема, где производится обработка ЭКС. Спектр персональных регистраторов позволяет проводить исследование ЭКС как в одном, так и в нескольких, до 12-ти, отведениях. Передача ЭКС в центр приема осуществляется по линиям телефонной связи.

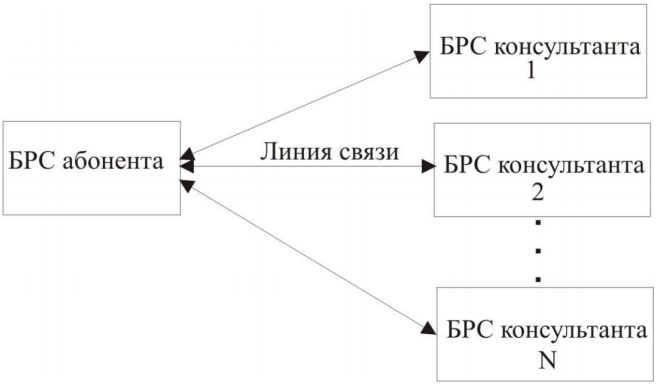
Другим направлением Aerotel Medical Systems, связанным с телемедицинскими системами, является создание телемедицинских агрегаторов, которые позволяют записывать основные физиологические параметры и передавать их на базовую станцию для дальнейшей обработки и хранения. Так, например, персональный домашний центр сбора данных Tele-CliniQ™ обеспечивает одновременную передачу данных от нескольких (до 4) медицинских контрольно-измерительных приборов в центр приема информации.

Внутренняя память, часы и автоматический режим передачи данных повышают надежность работы и сводят к минимуму ошибки оператора. В качестве регистрируемых параметров могут быть данные, полученные от измерителей артериального давления, глюкозы, пульсоксиметра, спирометра, электрокардиографа и т.д.

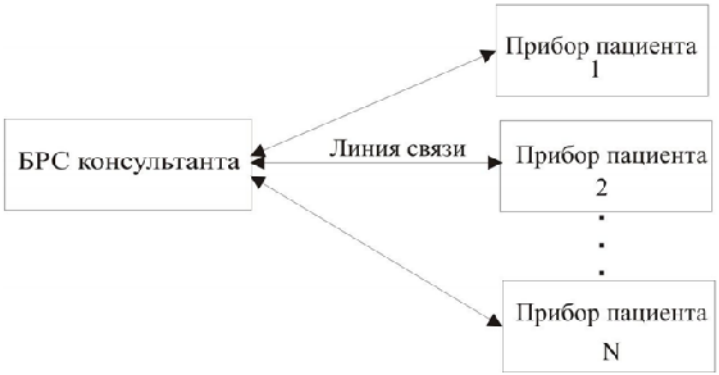
1.4. Варианты обобщенных структур телемедицинских систем

В основе построения телемедицинских систем лежат комплексы аппаратно-программных средств, представляющие собой многопрофильные и многозадачные рабочие места специалиста с возможностями обработки основных видов медицинской информации, а также проведения телеконференций (БРС). По базовой структуре телемедицинские системы можно упрощенно разделить

на две группы: системы для удаленного консультирования, диагностики и обучения, а также системы удаленного мониторинга жизненных функций (биотелеметрические системы). Структура систем первого типа представлена на рис. 1.3, *а*, второго – на рис. 1.3, *б*.



а)



б)

Рис. 1.3. Упрощенные структурные схемы:
а – удаленного консультирования; б – биотелеметрии

Несмотря на большое многообразие телемедицинских систем, для них характерен ряд типовых операций, связанных с получением, обработкой, представлением и передачей информации. Обобщенная модель информационного тракта телеметрической системы приведена на рис. 1.4.

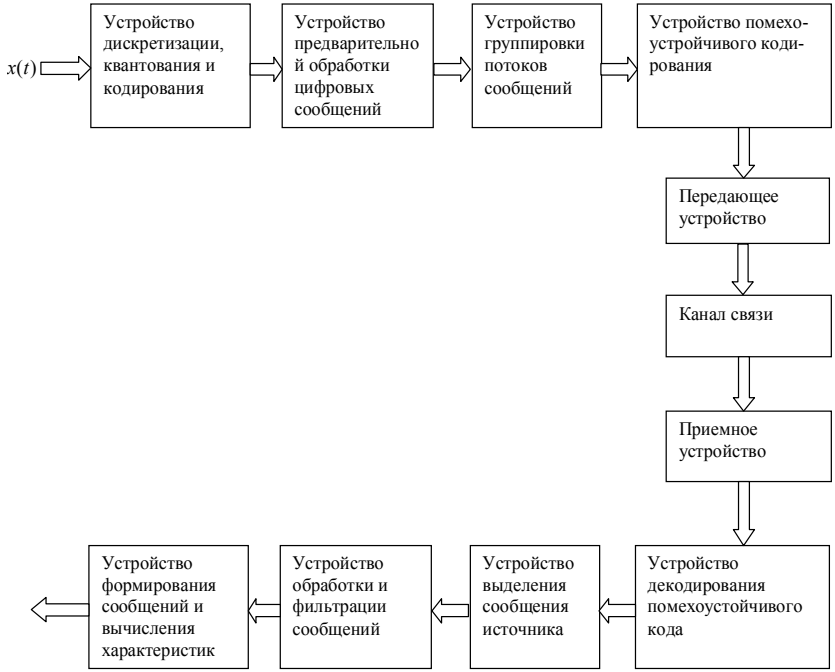


Рис. 1.4. Модель информационного тракта телемедицинской системы

В этой модели, на вход поступает аналоговый векторный сигнал (сообщение) x(t), с произвольного источника, например, микрофона, видеокамеры, медицинского диагностического прибора, который далее дискретизируется, квантуется и кодируется. Далее сигнал в цифровой форме подается на устройство предварительной обработки сообщений. Типичными операциями предварительной обработки являются фильтрация, сжатие сигнала, а также вы-

числение характеристик сигнала. Последняя операция может быть использована, если получателя не интересует исходный сигнал, а только отдельные его параметры. Кроме того, вычисление характеристик сигнала дает возможность сгруппировать потоки сообщений в соответствии с их динамическими характеристиками. Устройство группировки потоков сообщения формирует общий канальный сигнал, который поступает на устройство помехоустойчивого кодирования и, далее, в передающее устройство, где модулирует несущую. Распространяясь по каналу, передаваемый сигнал искажается шумами и помехами, имеющими как аддитивную, так мультипликативную составляющую, и поступает на устройство декодирования, после которого происходит выделение сообщений источников. Учитывая, что сигнал при передаче может искажаться, в устройстве отбраковки и фильтрации производится исключение аномальных значений сообщений, а также дополнительная фильтрация. На последнем этапе обработки происходит восстановление исходных сигналов и вычисление характеристик сигналов, необходимых пользователю. При построении информационного тракта телемедицинской системы алгоритмы и устройства их реализующие оптимизируются под решение основной задачи- получению от объекта и доставки потребителю требуемого количества информации с заданной точностью.

Для организации связи в телемедицинских системах используются различные линии связи, в которых на основе стандартных коммуникационных протоколов организуются каналы связи. Организация связи осуществляется по всем доступным типам телекоммуникационных каналов достаточной пропускной способности для решения конкретной телемедицинской задачи. Подавляющее количество современных телемедицинских систем для организации канала связи на большие расстояния использует протоколы транспортного и прикладного уровня сети интернет: TCP/IP,UDP, DHCP, ARP, FTP, Telnet, HTTP, HTTPS, SOAP и XML.

Телемедицинская система также должна обладать возможностью непосредственного подключения медицинской диагностической аппаратуры по портам LAN, IEEE-1394, IEEE-802.11, USB, COM, Bluetooth, S-video, RCA и т.д.

В предыдущих разделах и специальной литературе подробно рассматриваются возможности и техника связи по протоколу Bluetooth. К перспективным решениям с использованием радиопередачи относят использование приемопередатчиков типа ADuCRF101 работающих в субгигагерцовых ISM-диапазонах (от 433.05 до 434.79 МГц и от 902 до 928 МГц), обеспечивающих надежную передачу данных в полосе частот, альтернативной 2.4 ГГц. Приемопередатчик выполнен на основе процессора ARM Cortex™-M3, необходимые средства отладки этого процессора разработаны фирмами-партнерами компании Analog Devices. Потребляемый ток ADuCRF101 составляет 300 мкА/МГц. В спящем режиме он снижается до 1 мкА, что позволяет использовать этот процессор в медицинских приборах с батарейным питанием, в частности, в инфузионных насосах и носимых мониторах. Экономичный приемопередатчик в режиме приемника потребляет не более 12.8 мА, при этом имеет лучшую в своем классе чувствительность – 107,8 дБ. В режиме передатчика ADuCRF101 потребляет не более 9 мА, причем потребление зависит от задаваемой пользователем мощности передаваемого сигнала.

Для организации пакетной радиосвязи общего пользования используют канал GPRS (General Packet Radio Service – «пакетная радиосвязь общего пользования»), являющийся надстройкой над технологией мобильной связи GSM, осуществляющей пакетную передачу данных. GPRS позволяет пользователю сети сотовой связи производить обмен данными с другими устройствами в сети GSM и с внешними сетями, в том числе интернет. GPRS использует общий физический ресурс радиоинтерфейса совместно с существующими ресурсами системы GSM с коммутацией каналов. Службу GPRS можно рассматривать как наложенную на сеть GSM. Это позволяет использовать одну и ту же физическую среду в сотах как для передачи речи с коммутацией каналов, так и для передачи данных с коммутацией пакетов. Ресурсы GPRS могут выделяться под передачу данных динамически в периоды, когда отсутствует сессия передачи информации с коммутацией каналов. На структурном уровне систему GPRS можно разделить на две части: подсистему базовых станций (BSS) и опорную сеть GPRS

(GPRS Core Network). Ядро системы GPRS (GPRS Core Network) состоит из двух основных блоков – SGSN (Serving GPRS Support Node – узел поддержки GPRS) и GGPRS (Gateway GPRS Support Node – шлюзовой узел GPRS). SGSN контролирует доставку пакетов данных пользователям, взаимодействует с реестром абонентов HLR, проверяя, разрешены ли запрашиваемые пользователями услуги, ведет мониторинг находящихся онлайн пользователей, организует регистрацию абонентов, вновь «проявившихся» в зоне действия сети и т.п. GGSN является шлюзом между сотовой сетью (вернее, ее частью для передачи данных GPRS) и внешними информационными магистралями (Internet корпоративными интранет-сетями, другими GPRS-системами и т.д.). Основной задачей GGSN, таким образом, является маршрутизация данных, идущих от и к абоненту через SGSN. Вторичными функциями GGSN является адресация данных, динамическая выдача IP-адресов, а также отслеживание информации о внешних сетях и собственных абонентах. При использовании GPRS информация собирается в пакеты и передаётся через неиспользуемые в данный момент голосовые каналы. Такая технология предполагает более эффективное использование ресурсов сети GSM.

При построении телемедицинской системы с использованием GPRS канала необходимо учитывать относительно невысокую скорость передачи по каналу и ее изменчивость в зависимости от уровня помех. Это не позволяет строить системы, требующие большой пропускной способности канала, например, при передаче видеосигнала с малым коэффициентом сжатия в реальном масштабе времени. Однако для систем персонального мониторинга, где не требуется высокая скорость обмена данными, GPRS имеет определенные преимущества, связанные, прежде всего, с доступностью подключения к этой службе. Для организации связи с интернетом по каналу GPRS можно использовать модемы и модули GSM-GPRS различных производителей, которые имеют встроенный TCP/IP стек (например, SIM900D).

Разработчики телемедицинской аппаратуры в части выбора радиоканалов рекомендуют руководствоваться стандартом IEE 802.15.6 регламентирующим физический и частично канальный

уровни сетевой модели для сетей передачи данных, локализованных на теле человека или внутри него.

В качестве конкретных примеров в работе [81] приводятся варианты построения удаленного диагностического центра (рис. 1.5) и телемедицинского консультативного центра с видеовыходом (рис. 1.6) с микроконтроллерными ядрами.

В варианте маломощного диагностического центра (рис. 1.5) используется микроконтроллер (MCU) типа LM359B9x со встроенными Ethernet и USB портами.

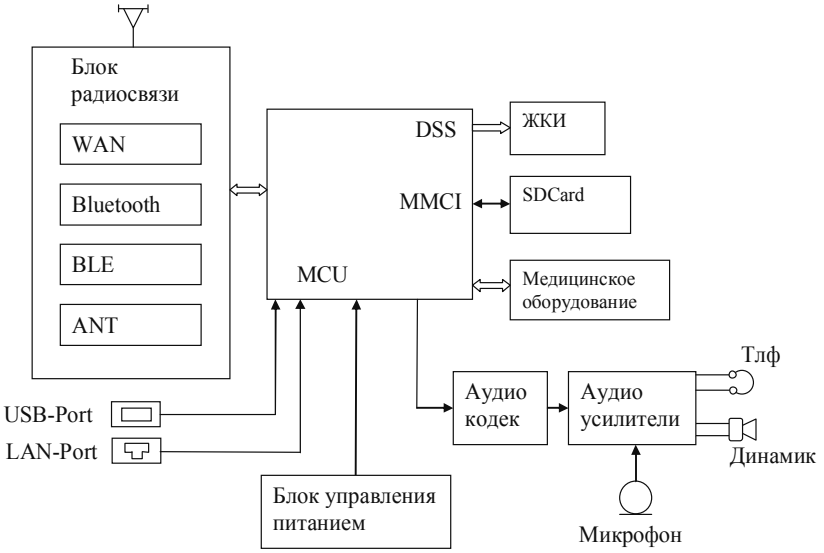


Рис. 1.5. Структура маломощного диагностического центра

Для решения вопросов сертификации, проектируемого устройства требуется наличие хотя бы одного PAN интерфейса таких как USB или Bluetooth, одного интерфейса типа WAN (например Ethernet) и локального интерфейса типа LAN (например, ZigBee).

На рис. 1.5 показано подключение различных радиоинтерфейсов, использующих технологии ANT, ZigBee, WIAN, Bluetooth и BLE.

ANT – технология беспроводной датчиковой сети, работающей в не лицензируемом диапазоне 2,4 ГГц, предназначенная для по-

строения беспроводных сетей малого радиуса действия автономных устройств с низким энергопотреблением. Эта технология отличается низким энергопотреблением трансиверов, простотой развертывания, эффективностью и масштабируемостью.

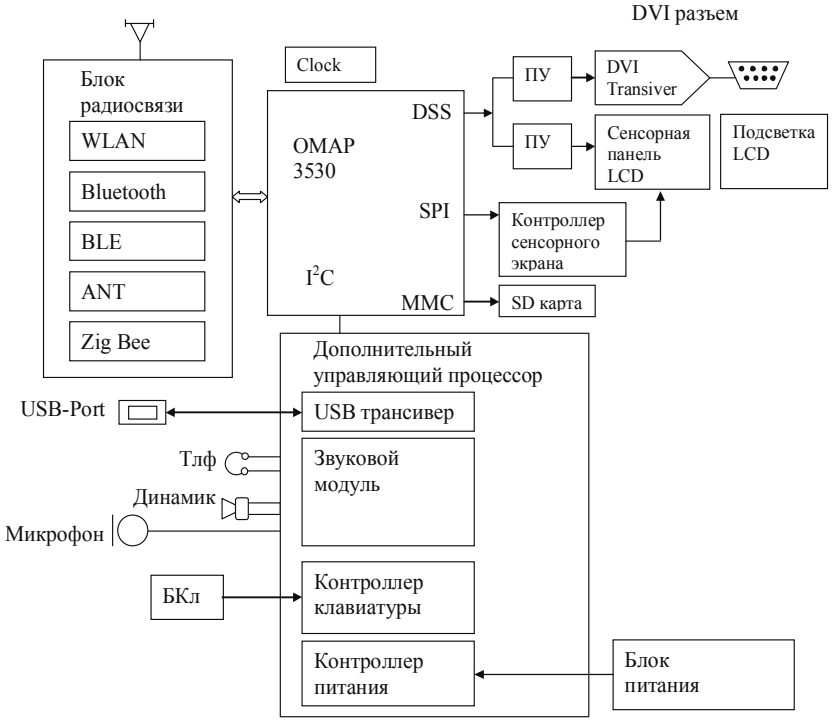


Рис. 1.6. Структура медицинского консультативного центра

Основная особенность технологии ZigBee заключается в том, что она при малом энергопотреблении поддерживает не только простые технологии (точка-точка, дерево и звезда), но и самоорганизующуюся ячеистую технологию с ретрансляцией и маршрутизацией сообщений.

WLAN – беспроводная локальная сеть, обеспечивающая доступ на всей территории ее покрытия. Как правило, радиус покрытия около 100 м. Частотным протоколом WLAN является стандарт Wi-Fi.

BLE – Bluetooth low energy (Bluetooth 4.0) работает в том же диапазоне частот, что Wi-Fi и ZigBee разработана специально для работы с устройствами на батарейном питании. Модель поведения модулей, использующих технологию BLE при установлении соединений различных типов устройств, поддерживающих технологию Bluetooth определяется их профилем (функциями и возможностями).

На рис. 1.6 приведен вариант микроконтроллерного исполнения консультативного центра с прикладным процессором OMAP3530, реализующим аудио и видео интерфейсы периферийных устройств.

Процессор OMAP3530 имеет HD видеоускоритель для включения потокового видео с высоким разрешением, что позволяет через протокол DVI поддерживать разрешение дисплеев от VGA до UXGA. Управление высококачественными дисплеями осуществляется через DSS, преобразователи уровня (ПУ) и DVI-трансивер.

Для работы с прикладными процессорами OMAP-35xx разработан дополнительный управляющий процессор содержащий: высокоскоростной USB-трансивер; звуковой модуль с цифровым фильтром и усилителями класса D; контроллер клавиатуры через который подключается блок клавиатуры (БКл); преобразователь, низкоуровневый регулятор и зарядное устройство (контроллер питания).

Если по техническому заданию нет необходимости работать с потоковыми HD-видео к прикладному процессору может быть подключен преобразователь типа AMP3517, обеспечивающий работу LAN(интернет) и USB портов.

Анализ современных телемедицинских систем показывает, что современная промышленность позволяет создавать мощные системы удаленного взаимодействия врачей и пациентов различных типов и назначений.

При этом состояние техники, формирующей сложноформатированные пакеты информации, характеризуется большим разнообразием подготовки пакетов в соответствии с техническим разнообразием систем кодирования, используемых датчиками изображения (ДИ). Это разнообразие является некоторым ограничением при выборе инструментальной базы для создания входного канала на базе телевизионных датчиков.

Такая же причина ограничивает реализацию алгоритмов обработки изображений с целью улучшения информационных характеристик информационного пакета для дальнейшей передачи его с помощью магистральных каналов.

Для снятия ограничений необходимо универсальное кодовое преобразование, которое подготовит передаваемое изображение к любой системе кодирования магистральной шины.

С этой целью целесообразно создание программно-технических комплексов (ПТК) , содержащих высококачественное видеопреобразование с помощью ДИ, использующих коррекцию дисторсии изображений, высококачественное управление подготовкой последовательности пакетов изображений и высококачественные универсальные кодовые преобразователи для подготовки последовательностей пакетов видеоизображений к передаче на магистральные шины. В этом случае структурная схема ПТК имеет следующий вид (рис. 1.7).

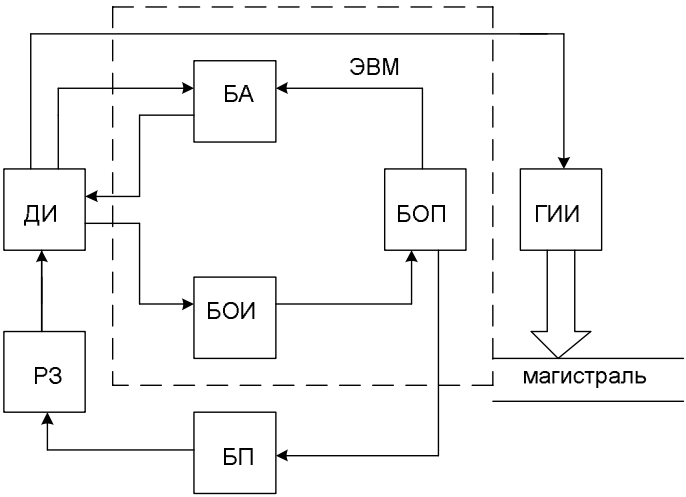


Рис. 1.7. Структурная схема универсального ПТК:
РЗ – рабочая зона; ДИ – датчик изображения; БП – блок подсветки;
БОИ – блок обработки изображения; БА – блок адаптации;
БОП – блок оценки параметров; ГИИ – гибкий интеллектуальный интерфейс

1.5. Проблемы построения медицинских информационных систем

Общие трудности построения медицинских информационных систем

Нерешенные проблемы информатизации медицины приводят к невозможности постановки правильного диагноза, принятию ошибочных решений в выборе воздействия на биообъект (БО).

В настоящее время основным мировым стандартом при передаче диагностической информации внутри ЛПУ является DICOM. Универсальный стандарт DICOM позволяет решить проблемы совместимости форматов данных различных производителей, вводя определенные требования для разработчиков медицинского оборудования. Стандарт DICOM описывает «паспортные» данные пациента, условия проведения исследования, пространственное положение пациента в момент получения изображения и т.п., для того, чтобы впоследствии провести медицинскую интерпретацию полученных данных. Стандарт позволяет организовать цифровую связь между различным диагностическим и терапевтическим оборудованием, использующимся в системах различных производителей. Рабочие станции, КТ и МРТ, микроскопы, УЗ-сканеры, общие архивы, хост-компьютеры и серверы от разных производителей, расположенные в одном городе или нескольких городах, могут «общаться» друг с другом на основе DICOM с использованием открытых сетей по стандартным протоколам, например, TCP/IP. Непосредственная работа с медицинским оборудованием, прием-передача принятой медицинскими приборами информации осуществляется в DICOM – формате.

Однако, несмотря на широкое распространение стандарта DICOM и электронных систем обмена информацией в ЛПУ, существуют значительные трудности использования телемедицинских технологий. Основными из них являются:

1. Поддержка оборудования, производитель которого перестал существовать или был поглощён другим производителем. Проблема состоит в том, что медицинская техника, выпускавшаяся произ-

водителем, после его реорганизации перестала получать техническую поддержку, из-за чего появились проблемы состыковки и интеграции данной техники в современные телемедицинские сети.

2. Наличие медицинских устройств, изготовленных без интерфейса DICOM. Проблема состоит в том, что современные цифровые медицинские устройства обработки изображений изначально не были оснащены интерфейсом DICOM. Это происходит либо тогда, когда производители устройств не хотят углубляться в производство специализированной медицинской техники; либо тогда, когда изготовитель работает в других областях получения изображений или производитель считает медицинскую визуализацию слишком сложной.

3. Необходимость задействовать в работе общие немедицинские устройства (цифровые или аналоговые). Часто в наличии у ЛПУ есть устройства общего назначения – фотокамеры, вебкамеры, сканеры, при помощи которых возможно получение медицинских изображений, не имеющих DICOM-выхода. Эти аппаратные средства не вписываются в процесс получения и обмена достоверной медицинской информацией. Например, каждое изображение должно содержать имя пациента и его идентификатор, что важно для точной передачи. Если используется устройство получения изображения без DICOM-выхода, то дополнение изображения вышеописанными данными возможно при участии человека, что часто приводит к ошибкам ввода данных и потерям времени [6]. В то время как устройства с DICOM-выходом получают такие данные автоматически.

4. Наличие в ЛПУ аналоговых устройств. Использование аналоговых устройств в ЛПУ требует преобразования аналоговых данных в цифровые для DICOM-совместимости. Задачу оцифровки и DICOM-преобразования обычно выполняют при помощи DICOM-преобразователя – устройства, способного превращать аналоговые изображения и видео в цифровые версии DICOM-совместимых изображений. DICOM-преобразователи предназначаются, в первую очередь, для устаревших аналоговых систем обработки изображений, но могут использоваться для работы с любым аналоговым устройством визуализации в целом. DICOM-

преобразователи позволяют расширить возможности передачи цифровых, но DICOM-несовместимых изображений и преобразовать сторонние медицинские форматы для работы с DICOM. Таким образом, решая задачу передачи аналоговой информации, необходимо привлекать целый набор DICOM-преобразующей аппаратуры, но вместе с тем снова возникает проблема ПО, неправильно поддерживающего стандарт DICOM. Устройства, преобразующие форматы, могут вносить погрешности в виде артефактов при преобразовании. Артефактом изображения в компьютерной диагностике является любое несоответствие между реальным изображением и полученным посредством прибора. Артефакты могут проявляться в виде полос, колец, затемнений, зашумлений. Например, появление артефакта при передаче стоматологической информации (рис. 1.8). При рассмотрении зубов (особенно вдоль оси зуба) с имеющимся запломбированным каналом совсем рядом с ним можно обнаружить артефакт, похожий на еще один, но незапломбированный канал зуба – это артефакт «ложный канал зуба». Критериями отличий артефакта является близкое расположение запломбированного канала, а также оценка зуба в аксиальной проекции.

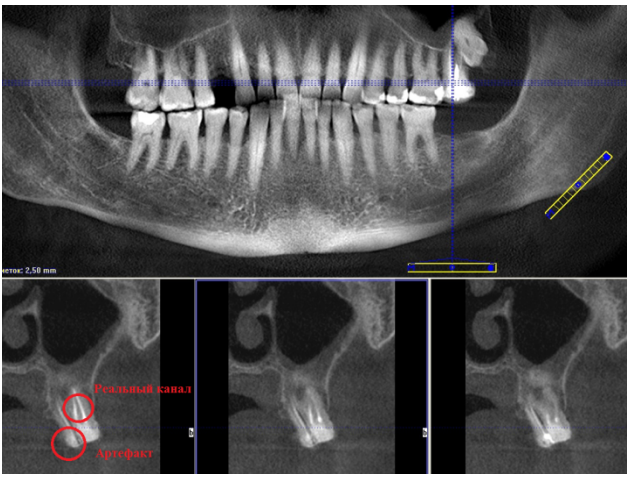


Рис. 1.8. Артефакт в стоматологии

Артефакты, возникающие из-за преобразования формата, могут появиться только по причине неисправности аппаратуры и несовершенства методов сжатия и восстановления изображений.

5. Взаимодействие между ЛПУ затруднено наличием множества разных систем обмена медицинской информацией от различных производителей. Происходит распыление ресурсов на разработку программного обеспечения в каждом отдельном медицинском учреждении, отсутствует возможность структурировать и организовать информацию в одной базе данных, существуют отдельные недоработки в программном обеспечении каждой клиники. Все это приводит к невозможности взаимодействия с ЛПУ, отсутствию единой телемедицинской сети на уровне района, города, области, страны.

6. Передача медицинских изображений по низкоскоростным (узкополосным) линиям передачи данных, что может приводить к потерям информации, увеличению времени передачи информации между различными ЛПУ, невозможности передачи большого объема информации. Низкая пропускная способность телефонных каналов – один из основных сдерживающих факторов на пути полномасштабного внедрения телемедицинских услуг. В отсутствии широкополосного доступа, единственным способом осуществить передачу в режиме реального времени является эффективное сжатие изображений. Однако, необходимо учитывать, что при сжатии возможна потеря информации, появление артефактов (п.4), что может привести к постановке неверного диагноза и выбору несоответствующего лечения.

7. Отсутствие единого стандарта взаимодействия между ЛПУ. Если западные производители телемедицинских систем решают проблемы объединения различных ЛПУ в единую цифровую сеть, то в России пока еще приходится решать подобные задачи на уровне одного медицинского учреждения.

8. Неправильная поддержка DICOM-стандарта программным обеспечением возможна при разработке ПО сторонними производителями, если недосконально изучен этот стандарт, содержащий в себе большое количество информации. Это происходит из-за большого количества управляющей информации, которая включе-

на в стандарт DICOM. Кроме этого, стандарт постоянно претерпевает обновления, которые несут изменения и вынуждают осуществлять настройку ПО для правильного функционирования медицинских систем.

Все вышеперечисленные проблемы справедливы для всех сфер применения сложных измерительно-вычислительных систем. Данные проблемы могут приводить не только к отсутствию обмена информацией между элементами систем, но, что более опасно и сложно в обнаружении, к неверной интерпретации информации и дальнейшим неверно принятым управленческим решениям. То есть возможно говорить о появлении различных видов метрологических погрешностей, влияющих на результат измерений.

Существующие решения интерфейсов в области телемедицинских систем

Аналоговые интерфейсы биомедицинских систем

В настоящее время с целью решения задач структурно-функциональной оптимизации систем обрабатывающих аналоговые сигналы, включая биотехнические системы многие производители, выпускают интегрированные микросхемы для сбора и предварительной обработки аналоговых сигналов. В зарубежной литературе микросхемы такого типа обозначают термином Analog Front End (AFE).

Производственным лидером в этой области считают американскую фирму Texas Instruments (TI). Значительное число микросхем типа AFE выпускается специально для медицинских приложений. Например: микросхема AD9671 представляет собой восьмиканальный аналоговый входной интерфейс с цифровым демодулятором и портом стандарта YESD 204 B и предназначена для систем ультразвуковой диагностики; микросхема AD8232 выполнена как аналоговый интерфейс для решения задач контроля ЭКГ и в частности ЧСС для одного отведения; ADAS 1000 – аналоговый интерфейс с низким энергопотреблением для пятиэлектродных электрокардиографов; микросхемы AD9278 и AD9279 – восьмика-

нальные приемники сигналов ультразвуковых частот; AD5791 – прецизионный 20-битный ЦАП для систем магниторезонансной томографии; ADAS1128 – 24-битный преобразователь тока в цифровой код для систем компьютерной томографии; ADuM4160 – однокристальный изолятор USB для медицинского оборудования с критическими системными требованиями и т.д.

Использование микросхем типа AFE значительно упрощает процесс проектирования. При этом освободившееся конструкторское время может быть перераспределено в пользу значительного улучшения математического обеспечения и пользовательских интерфейсов.

Типовая обобщенная структура микросхем типа AFE приведена на рис. 1.9.

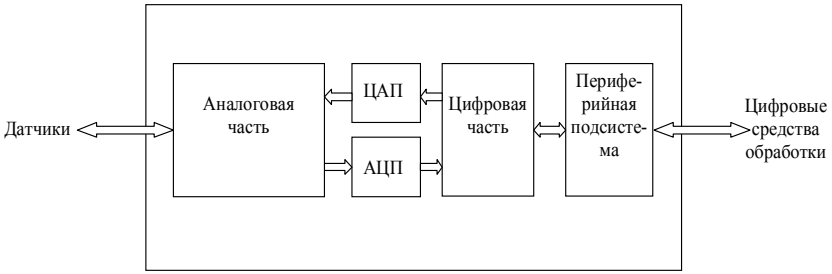


Рис. 1.9. Обобщенная структура устройств типа AFE

Конкретные структура и функции AFE определяются решающей задачей. Например, AFE ориентированные на регистрацию ЭКГ в аналоговой части содержат коммутаторы, усилители с программируемым коэффициентом усиления, фильтры, драйверы обратной связи на биообъект и экран и т.д. Как правило, при проектировании аналоговой части используют те же схемы, что и в «классическом» аналоговом исполнении. После достаточно качественного аналогово-цифрового преобразования дальнейшую предварительную обработку производит цифровая часть, в состав которой могут входить как классические микросхемы, так и DSP процессоры, которые решают задачи подавления синфазной поме-

хи, вычисления отведений, фильтрации и т.д. Периферийная подсистема обеспечивает реализацию стандартных протоколов (чаще всего SPI). Через периферийную систему AFE подключается к другим вычислительным средствам или средствам передачи данных (например, Bluetooth). Через неё же осуществляется настройка параметров AFE. Для воздействия на биообъект или формирования тестовых сигналов (например, при измерении надежности контактов электродов; при измерении импеданса и др.) в схему AFE включают ЦАП. Естественно, что в зависимости от решаемых задач некоторые блоки приведенные на рис. 1.9 могут отсутствовать, например в микросхеме для измерения РН отсутствуют ЦАП, АЦП и цифровая часть.

На рис. 1.10 приведен вариант схемы кардиомонитора с использованием микросхемы ADAS 1000 реализующей функции аналогового интерфейса.

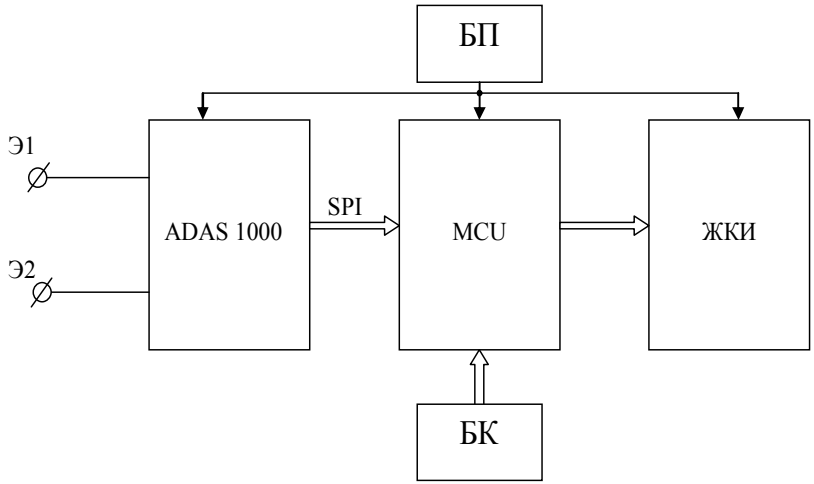


Рис. 1.10. Структура кардиомонитора с использованием аналогового интерфейса

В схеме приведённой на рис. 1.10 аналоговый интерфейс ADAS 1000 подключается непосредственно к двум грудным электродам Э1 и Э2.

Этой микросхемой сигнал усиливается с подавлением синфазной помехи, фильтруется и преобразуется в последовательный цифровой код стандарта SPI. Микроконтроллер MCU выполняет требуемый набор вычислений (расчет ЧСС, выделение и классификацию аритмий, определяет наличие тахикардий и брадикардий и т.д.) и обеспечивает формирование кодов для жидкокристаллического индикатора (ЖКИ). Управление работой кардиомонитора осуществляется блоком клавиатуры (БК). Питается прибор от блока питания (БП) (аккумулятор) напряжением 3В.

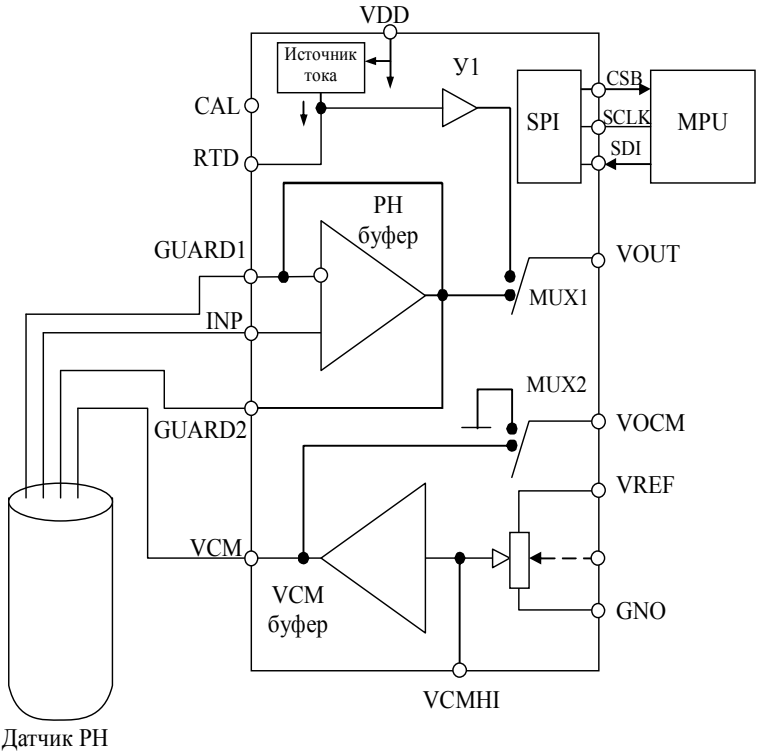


Рис. 1.11. Микроконтроллерная схема измерения кислотности с аналоговым интерфейсом LMP91200

На рис. 1.11 показан вариант структуры измерителя кислотности на специализированной микросхеме типа LMP91200. Эта микросхема сделана специально для измерителей РН на двухэлектродных датчиках кислотности. Дополнительно по входу RTD эта микросхема поддерживает температурные датчики типа Pt 1000, Pt 100. При измерении температуры на выход VOUT подается сигнал с У1 через MUX1. На выводе VOCM устанавливается напряжение смещения в зависимости от диапазона измерения РН. Измерение значения кислотности через РН-буфер передается на выход при низком положении переключателя MUX1. Микросхема программируется через стандартный интерфейс с протоколом SPI. Имеются встроенные функции тестирования, с помощью которых проверяется правильность подключения и работы датчиков.

Как видно из рис. 1.11 выходной сигнал VOUT характеризующий значение РН или температуры является аналоговым. Поэтому для его обработки цифровыми средствами следует использовать отдельную микросхему АЦП.

Специально для разработки многофункциональных систем работающих с различными типами датчиков в медицине и экологии фирма Analog Devices разработала прецизионные и энергоэффективные чипы (meter – on – a – chip) типа ADuCM350. Основное назначение этого чипа объединение различных устройств, следящих за состоянием здоровья, таких как медицинское диагностическое оборудование у постели больного, многофункциональные мониторы для оценки состояния здоровья и окружающей среды, домашние медицинские устройства, а также носимые индикаторы основных показателей состояния организма, браслеты для фитнеса и спорта.

Структурная схема микрочипа ADuCM350 приведена на рис. 1.12. Из приведенной структуры следует, что эта микросхема может быть отнесена к разряду прикладных процессоров имеющих в качестве ядер достаточно мощный микропроцессор APM Cortex M3 и акселератор выполняющий функции DSP к которым добавлены элементы аналогово-цифровой части типа AFE.

Подключение множества пассивных датчиков (электроды, емкостные датчики, тензодатчики, электрохимические датчики и т.д.)

к инструментальным усилителям (TIA) осуществляется через матрицу аналоговых коммутаторов (Switch Matrix). Микросхема ADuCM350 реализует функцию автокалибровки датчиков. После TIA аналоговый сигнал фильтруется фильтром низких частот (F) и через мультиплексор (MUX) поступает на 16-разрядный аналогово-цифровой преобразователь (ADC), работающий с частотой дискретизации 160 кГц. Для измерения импеданса пассивных датчиков и работы с емкостными датчиками в микросхеме имеется программно-управляемый генератор (Waveform Generator) расположенный в блоке аппаратного акселерометра формирования сигналов и фильтрации (Accelerators). Цифровой код с этого генератора преобразуется в аналоговый сигнал двенадцатиразрядным ЦАП (DAC), фильтруется фильтром низкой частоты (F) и через усилитель AMP и программируемую матрицу аналоговых коммутаторов подается на биообъект или соответствующий датчик. Температурный режим микросхемы контролируется температурным датчиком (T sensor).

Данные температурного датчика могут использоваться для оценки и коррекции погрешностей измерений. Для проведения биоимпедансных измерений используется быстрое преобразование Фурье с аппаратной поддержкой блоком DFT. С целью улучшения соотношений сигнал/шум реализуются программируемые цифровые фильтры (блок Filters). Работа аналогово-цифровой части поддерживается контроллером AFE (Analog Front End Controller). Взаимодействие аналогово цифровой части AFE с другими блоками микросхемы ADuCM350 и «внешним цифровым миром» обеспечивает процессор ARM Cortex –M3 (частота работы процессора 16 мГц).

Микропроцессорная часть микросхемы содержит: контроллер сенсорной клавиатуры (CAP TOUCH); интерфейс USB; контроллер ЖК дисплея (Display) с параллельным и последовательными интерфейсами; контроллер сегментного ЖК-дисплея; блок памяти (Memory). Блок памяти состоит из Flash (384 кбайт), EEP ROM (16 кбайт, электрически перепрограммируемое ПЗУ), SRAM (32 кбайт, статическое ОЗУ).

С внешними цифровыми устройствами и с активными датчиками (например AFE – акселерометрами) микросхема ADuCM350

«связывается» посредством периферийного интерфейса (Peripheral Interface). Аппаратно поддерживаются последовательные интерфейсы UART, SPI, I^2C, I^2S, а так же побитно программно-управляемый интерфейс GPIO.

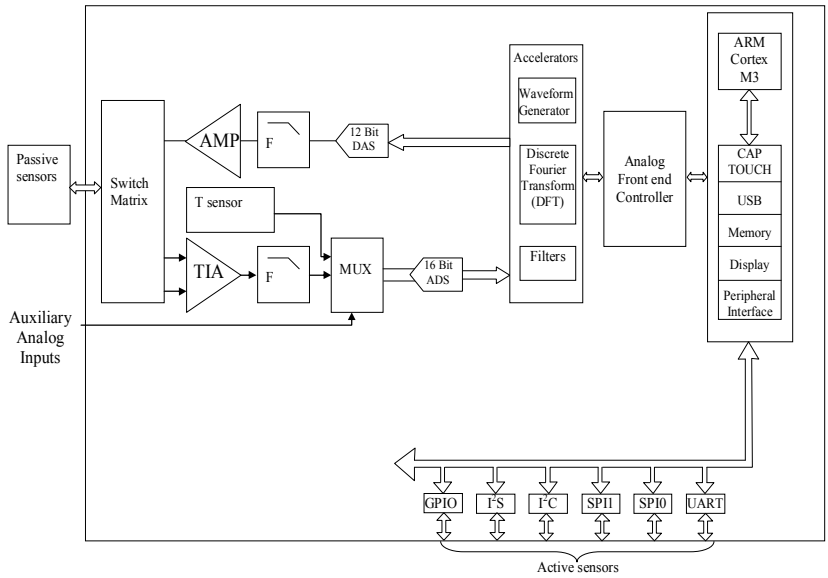


Рис. 1.12. Структура прецизионного чипа ADuCM350

С помощью интерфейса GPIO пользователь может создавать как «свои» протоколы обмена, так и стандартные интерфейсы типа дополнительных SPI, I^2C и т.д.

Микросхема ADuCM350 обеспечивает широкие возможности управления питанием и может питаться от одноэлементной дисковой батарейки.

Для подключения к ПЭВМ необходимо обеспечить соответствующую гальваническую развязку.

Американские фирмы Texas Instruments и Analog Devices продолжают активно разрабатывать и полнить рынок интегральных микросхем (ИМС), включая чипы AFE ориентированные на меди-

цинскую промышленность. Свои новые разработки они описывают в соответствующих информационных бюллетенях. Например, в бюллетене [35] они описывают набор аналоговых интерфейсов для ультразвуковых систем нового поколения:

– AD9278 и AD9279 – восьмиканальные приемники ультразвуковых сигналов;

– AD8021 – быстродействующий усилитель с низким уровнем собственных шумов;

– ADA4841-2 – усилитель с малым потреблением и малыми искажениями входного сигнала;

– AD7982 – АЦП семейства PulSAR с разрешением 18 бит и частотой выборки до 1 МГц;

– ADR433 – малошумящий опорный источник с выходным напряжением 3 В;

– AD9516 и AD9520 – генераторы синхросигналов;

– ADCLK946 и ADCLK846 – разветвляющиеся по выходу буферы синхросигналов.

В этом же бюллетене приводится краткое описание еще нескольких разработок.

ИМС AD5791 – 20-разрядный ЦАП с выходом по напряжению, разрешением 1 ppm и интегральной нелинейностью ±1 EMP. ЦАП обеспечивает высокоточное управление магнитным градиентом сильного магнитного поля, что, в свою очередь, позволяет получить высокие разрешение и контрастность изображения МРТ-аппаратуры. ЦАП имеет сверхнизкий уровень шумов, снижающий число нежелательных артефактов, что дает возможность уменьшить число операций сканирования органов пациента. Таким образом, снижение продолжительности сканирования одного пациента позволяет увеличить пропускную способность дорогой аппаратуры. Высокая точность ЦАП AD5791 дает возможность уменьшить число его калибровок в процессе эксплуатации МРТ-аппаратуры, что тоже приводит к увеличению пропускной способности этой аппаратуры. Алгоритмы калибровки МРТ-аппаратуры при использовании прецизионного ЦАП AD5791 могут быть существенно упрощены, что позволяет уменьшить как сроки, так и сложность проектирования системы в целом.

Для обеспечения гальванической развязки компания Analog Devices представляет набор ИМС выполненных по технологии i-Coupler, краткое описание которых представлено табл. 1.1.

Таблица 1.1

Цифровые i-Coupler изоляторы, сертифицированные на соответствие требованиям стандарта IEC 60601-1

Тип ИМС	Тип интерфейса, число каналов	Скорость обмена данными, макс., Мбит/с	Мощность, потребляемая при скорости передачи данных 1 Мбит/с, мВт	Вых. мощность изолированного источника питания, мВт	Тип корпуса	Цена, $
ADuM240x	4	90	20	–	16-SOIC	3.20
ADuM4160	USB	12	35	–	16-SOIC	4.89
ADuM225x	I2C	1	50	–	16-SOIC	2.77
ADuM640x	4	25	–	500	16-SOIC	6.33

Компания Analog Devices выпускает прецизионные усилители и преобразователи, предназначенные для использования в недорогих мониторах сердечного ритма. Эти ИМС могут быть дополнены изолированными USB-интерфейсами, MEMS-акселерометрами, ВЧ-приемниками, сенсорными интерфейсами для управления различными датчиками, температурными сенсорами, аналоговыми интерфейсами и пр. Перечисленные компоненты позволяют создавать различное медицинское оборудование, включая мониторы сердечного ритма.

ИМС AD7798 – 16-разрядный сигма-дельта АЦП с низким потреблением и малым уровнем шумов, который имеет три аналоговых входных канала. ИМС AD7794 – 24-разрядная версия этого АЦП. К особенностям преобразователей следует отнести наличие ключа для заземления внешнего источника сигнала, детектора уровня опорного напряжения, генератора тактовых импульсов, источников токов для питания, например, мостовых датчиков. В АЦП имеются цифровые входы для программирования коэффи-

циента усиления VGA-усилителя. Напряжение питания АЦП 2.7–5.25 В, типовой ток потребления 300 мкА, корпус типа 16-TSSOP.

ИМС AD8235 – измерительный усилитель, выполненный в корпусе типа WLCSP размерами 1.6×2.0 мм (его аналог AD8236, выполнен в корпусе типа MSOP). Максимальный ток потребления 40 мкА, в спящем режиме ток потребления не превышает 6 нА, входной ток смещения 1 пА. Это идеальный усилитель для использования в миниатюрных портативных медицинских приборах, таких как мониторы сердечного ритма и другие медицинские мониторы, к которым предъявляются высокие требования по габаритам и потребляемой мощности.

ИМС AD8619 – счетверенный микромощный ОУ, предназначенный для построения различной медицинской аппаратуры. Его используют вместе с измерительным усилителем AD8235 и АЦП AD7798. Напряжение питания AD8619 от 1.8 до 5 В, спектральная плотность шума 22 нВ/√Гц, максимальный входной ток смещения 1 пА, ток потребления 38 мкА, тип корпуса 14-TSSOP или 14-SOIC.

ИМС ADIS16334 представляет собой инерционный MEMS сенсор обеспечивающий точное слежение за движениями хирурга, преобразуя его (с помощью джойстика) в движение хирургического инструмента. Возможно использование этой ИМС в автоматическом режиме, реализуя движение по заданной программе.

Для систем управления инфузионными насосами разработаны микросхемы серии ADM116X.

Для использования в дозиметрической аппаратуре, МРТ-сканерах, фотодиодных сенсорных устройствах, спектроскопической аппаратуре предлагается микросхема ADAS1128, представляющая собой 24-разрядный преобразователь тока в код, позволяющий одновременно опрашивать 128 каналов фотодиодной матрицы и т.д.

Для медицинских телеметрических систем фирмы Analog Devices предлагает микромощные приёмопередатчики, выполненные по технологии SoC, обеспечивающей более точную приемопередачу, чем Bluetooth и ZigBee. Приемопередатчик ADuCRF101 работает в субгегагерцовом ISM диапазоне (от 433.05 до

434,79 МГц и от 902 до 928 МГц). Ток потребления ADuCRF101 составляет 300 мкА/МГц, а в спящем режиме снижается до 1 мкА, что позволяет его использовать в медицинских приборах с батарейным питанием, в частности в инфузионных насосах, носимых мониторах и т.д.

Таким образом, рынок микросхем типа AFE, ориентированных на медицинскую промышленность постоянно пополняется, что необходимо учитывать при проектировании биотехнических систем.

Популярные интерфейсы сопряжения цифровых устройств между собой и внешним оборудованием

Важной задачей проектирования биотехнических систем, содержащих в своем составе цифровые узлы, блоки и целые вычислительные устройства является их соединение между собой и с другими элементами разрабатываемых систем. Решение этой задачи обеспечивается грамотным использованием соответствующих интерфейсов. В данном разделе рассмотрим наиболее широко используемые интерфейсы, ориентируясь на их аппаратную поддержку, современными микроконтроллерами семейства STM32.

Одним из наиболее простых интерфейсов является **двухпроводной интерфейс I2C**, в котором по одному проводу (шине) передается сигнал синхронизации SCL, а по другому – данные (шины SDA). Устройства, подключаемые к шинам должны иметь управляемый выход с высоко импедансным состоянием, чтобы не мешать работающим в данный момент устройствам. Технически это обеспечивается использованием схем с открытым коллектором, реализующих «Монтажное И». Для формирования высокого уровня сигналов коллекторы подтягиваются к питанию соответствующими резисторами (рис. 1.13).

Когда устройство не является ни приемником, ни передатчиком ее выходной транзистор переводится в закрытое состояние, и он не влияет на передачу данных.

Активное устройство, передавая данные, переводит выходной транзистор в открытые и закрытые состояния. Когда транзистор

открыт, на его коллекторе и шине SDA потенциал близок к потенциалу земли (логический 0), когда транзистор закрыт, на его коллекторе «почти» потенциал питания (логическая 1). Аналогично управляется шина SCL. Количество устройств, подключаемых к шине, ограничивается полем адреса и емкостными характеристиками в 400пФ (скоростные данные шины).

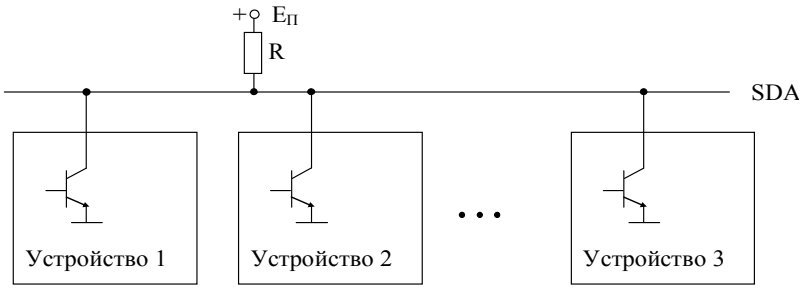


Рис. 1.13. Физическая организация линии интерфейса I2C

Обмен данными организован таким образом, что одно устройство обязательно является ведущим, оно инициирует передачу данных и синхроимпульсы. Подчиненное (ведомое) устройство принимает данные и может организовать передачу данных только по команде ведущего устройства. Каждое устройство, подключенное к шине, имеет свой уникальный адрес. Когда ведущий передает данные, то вначале он передает код адреса, а каждое из ведомых устройств, подключенных к шине проверяет этот адрес, решая задачу опознания «свой», «чужой». В состав байта адреса входит один бит направления передачи данных, по которому определяется выполнение чтения из «ведомого» или запись в него.

При отсутствии передачи данных на шинах SCL и SDA поддерживается потенциал логической единицы (близкий к напряжению питания за счет подтягивающего резистора при всех закрытых выходных транзисторах).

Биты начала (START) и конца (STOP) передачи пакета данных формируются ведущим для определения начала и окончания передачи данных соответственно. Бит START формируется переходом

сигнала SDA из высокого уровня в низкий при высоком уровне сигнала SCL. Бит STOP формируется переходом SDA из низкого уровня в высокий при высоком уровне SCL (рис 1.14).

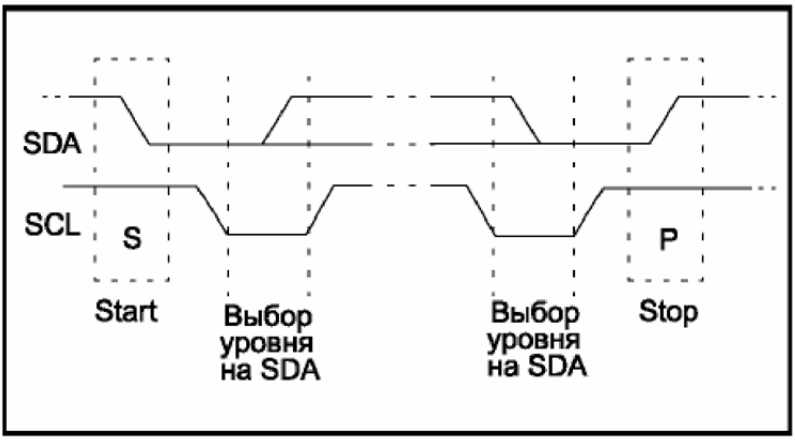


Рис. 1.14. Механизм формирования начальных и конечных границ, передаваемого пакета данных

При передаче данных сигнал SDA переключается только при низком потенциале SCL, что позволяет надежно различать коды нулей и единиц.

В протоколе интерфейса I2C предусмотрено фва формата адрессации: простой 7-разрядный формат с битом чтения/записи R/W (рис. 1.15); 10-разрядный формат, передается двумя байтами. В первом байте передается: пять битов, определяющих, что это 10-разрядный адрес; два старших бита адреса: бит записи/чтения. Во втором байте передается 8 младших бит адреса.

При передаче данных после каждого переданного байта приемник должен подтвердить получения байта сигналом АСК (рис. 1.16). Если ведомый не подтверждает получение байта адреса или данных, ведущий должен прервать передачу сформировав сигнал STOP (ведомый должен отпустить SDA для формирования STOP ведущим).

Рис 1.15. Форматы простой адресации.

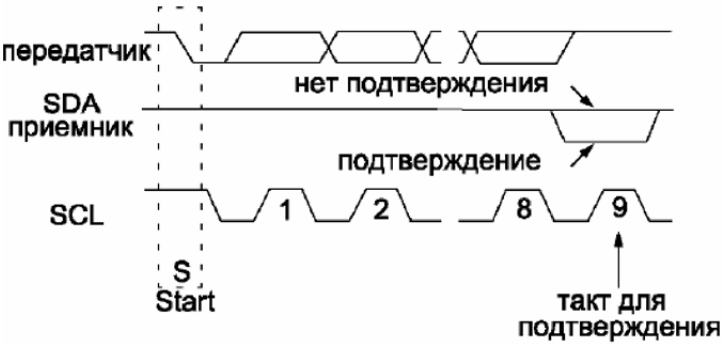


Рис. 1.16. Механизм подтверждения приема данных

Когда ведущий шины принимает данные, то на каждый принимаемый байт формируется бит подтверждения, если принятый байт не последний. Для сообщения ведомому о том, что ведущий прекращает принимать данные по приему последнего байта – ACK не формируется. Ведомый отпускает SDA, чтобы ведущий смог передать бит STOP. Ведущий может формировать бит STOP на месте бита подтверждения.

Если ведомому необходимо задержать передачу данных, то он может удерживать SCL в низком логическом уровне. Передача данных продолжится, когда ведомый отпустит SCL. Это позволяет ведомому подготовить новые данные для передачи.

На рис. 1.17 показан механизм передачи данных ведущим устройствам.

Рис. 1.17. Передача данных ведущим устройствам

Протокол передачи данных I2C позволяет иметь более одного ведущего на шине. Для разрешения конфликтов на шине при инициализации передачи используются функции арбитража и синхронизации.

К широко используемым интерфейсам относится **последовательный интерфейс SPI**. Шина SPI организована по принципу ведущий-ведомый (подчиненный). Одним из основных технических элементов интерфейса SPI является сдвиговый регистр, с выхода которого передается последовательный код данных. Обязательным также является сигнал синхронизации. В приемнике данных также используется регистр сдвига (рис. 1.18).

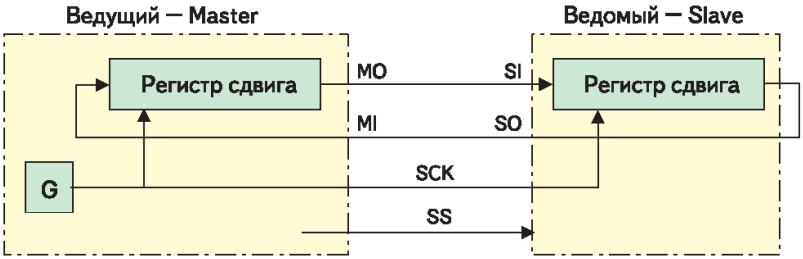


Рис. 1.18. Структура взаимодействия ведущего и ведомого интерфейса, SPI

Протокол SPI использует следующий набор сигналов:

MOSI (Master Out Slave In) – выход данных ведущего (она же вход данных ведомого);

MISO (Master In Slave Out) – вход данных ведущего (она же выход данных ведомого);

• SCK (Serial ClocK) – тактирование (синхронизация);

• SS (Slave Select) – выбор ведомого.

Существуют три типа подключения к шине SPI, в каждом из которых участвуют четыре сигнала. В первом варианте подключения (рис. 1.19) участвуют две микросхемы.

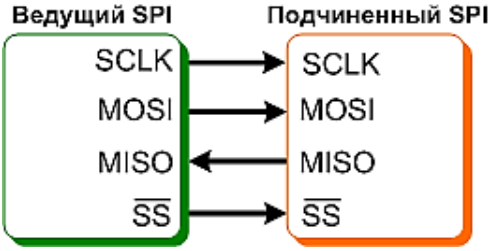


Рис. 1.19. Простейшее подключение к шине SPI

В этом варианте ведущий шины передает данные по линии MOSI синхронно со сгенерированным им же сигналом SCLK, а подчиненный захватывает переданные биты данных по определенным фронтам принятого сигнала синхронизации. Одновременно с этим подчиненный отправляет свою посылку данных. Представленную схему можно упростить исключением линии MISO, если используемая подчиненная ИС не предусматривает ответную передачу данных или в ней нет потребности. Одностороннюю передачу данных можно встретить у таких микросхем как ЦАП, цифровые потенциометры, программируемые усилители и драйверы. Таким образом, рассматриваемый вариант подключения подчиненной ИС требует 3 или 4 линии связи. Чтобы подчиненная ИС принимала и передавала данные, помимо наличия сигнала синхронизации, необходимо также, чтобы линия SS была переведена в низкое состояние.

При необходимости подключения к шине SPI нескольких микросхем используется либо независимое (параллельное) подключе-

ние (рис. 1.20), либо каскадное (последовательное) (рис. 1.21). Независимое подключение более распространенное, так как достигается при использовании любых SPI-совместимых микросхем. Здесь, все сигналы, кроме выбора микросхем, соединены параллельно, а ведущий шины, переводом того или иного сигнала SS в низкое состояние, задает, с какой подчиненной ИС он будет обмениваться данными.

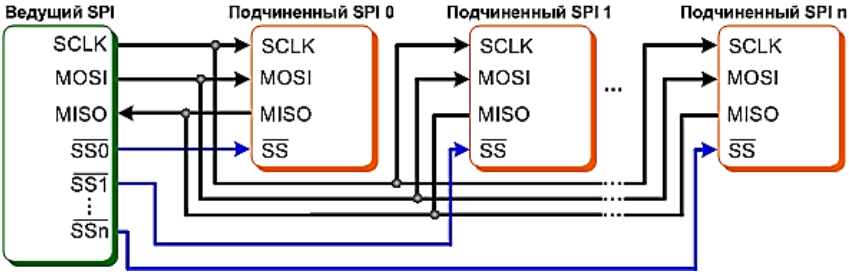


Рис. 1.20. Независимое подключение к шине SPI

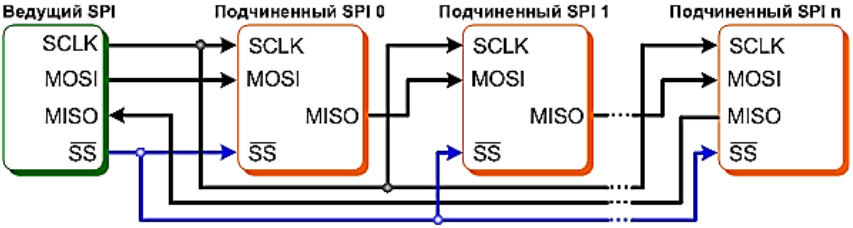


Рис. 1.21. Каскадное подключение к шине SPI

Частота следования битовых интервалов в линиях данных SPI определяется синхросигналом SCK, который генерирует один из абонентов – ведущий (Master). Прочие абоненты – ведомые (Slave), которых может быть как один, так и несколько, используют синхросигнал для определения моментов изменения битов на линии данных.

Передача битов осуществляется пакетами. Длина пакета чаще всего составляет 1 байт (8 битов), хотя встречаются реализации

SPI с иной длиной пакета. Как в ведущем устройстве, так и в ведомом имеется (обычно программно недоступный) счетчик импульсов синхронизации (битов). Счетчик в ведомом устройстве позволяет последнему определить момент окончания передачи пакета. Счетчик битов сбрасывается в ноль при выключении (деактивации) подсистемы SPI, такая возможность всегда имеется в ведущем устройстве. В ведомом устройстве счетчик битов обычно сбрасывается деактивацией интерфейсного сигнала SS. Передача пакета, то есть запуск генератора синхроимпульсов в ведущем, обычно осуществляется автоматически при записи передаваемого байта в регистр сдвига ведущего.

Интерфейсом SPI предусмотрено несколько вариантов обмена (различные временные диаграммы работы). На рис. 1.22 приведен один из таких вариантов.

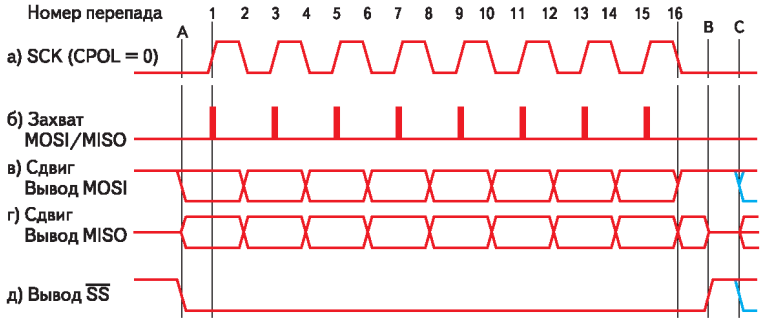


Рис. 1.22. Вариант формирования временных диаграмм интерфейса SPI (режима)

До начала передачи пакета, левее момента времени A сигнал SS на выходе ведущего в высоком состоянии – рис. 1.22, *д*), ведомый неактивен, его выход MISO в отключенном состоянии, а входы ведомого MOSI и SCK не воспринимают сигналов, формируемых ведущим. Тактовый сигнал на выходе ведущего в этот период имеет низкий уровень.

Для активации ведомого ведущий переводит в активное (низкое) состояние сигнал SS выбора ведомого (момент времени A). Ведомый включается, его вход тактирования способен воспринимать тактовый

сигнал, а вывод MISO переходит в режим выхода. Тактовые импульсы пока еще не начались, выходные сигналы на линях данных MOSI и MISO теперь отражают значения бита, который будет первым передан другому абоненту (интервал времени от A до 1).

Через некоторое время ведущий начинает формирование пакета сдвиговых импульсов (рис. 1.22, *а*), сигнал SCK. Перепады тактового сигнала вызывают сдвиг содержимого сдвиговых регистров ведущего и ведомого, а также захват новых значений битов после сдвига. Моменты сдвига рис. 1.22, *б* и *в* и захвата (рис. 1.22, *б*) должны быть разнесены во времени, поэтому эти действия выполняются в интерфейсе SPI по разным перепадам тактовых импульсов. Диаграмма рис. 1.22, *а* изображена в предположении, что захват производится по нарастанию тактового сигнала (по нечетным перепадам на рисунке рис. 1.22), а сдвиг в регистрах – по спаду (четные перепады). Таким образом, порядок событий при действии каждого тактового импульса можно описать как «захват, затем сдвиг» (Latch, then Shift).

После окончания обмена пакетами (правее перепада 16-тактовой последовательности) выходной сигнал на каждой из линий данных MOSI и MISO отражает значение бита, который был передан первым. Передача пакета завершается снятием сигнала SS на выходе ведущего (момент B), что деактивирует ведомого абонента. Через некоторое время возможно начало передачи следующего пакета (момент C).

Сигнал SS может формироваться на выходе ведущего разными способами. В некоторых МК он формируется аппаратно: после того как будет программно выполнена запись передаваемого байта в регистр данных передатчика, выходной сигнал SS автоматически переключается в низкий уровень, а по окончании пакета – возвращается в исходное неактивное значение (высокий). Если в ведущем устройстве отсутствует аппаратно реализованное формирование сигнала SS, его формируют программно, используя для этого выход параллельного порта общего назначения. Однако при работе с некоторыми периферийными устройствами приходится формировать сигнал SS программно даже при наличии аппаратного формирователя.

Временные диаграммы различных режимов работы протокола SPI приведены на рис. 1.23.

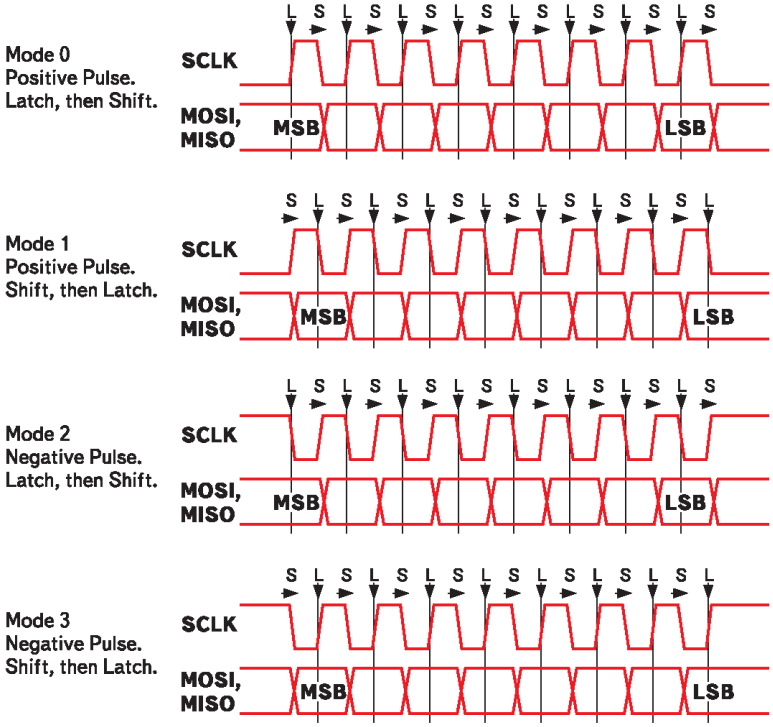


Рис 1.23. Четыре режима работы интерфейса SPI

Буквы S и L над графиками обозначают соответственно Shift (сдвиг) и Latch (захват). Рассмотренная ранее диаграмма (рис. 1.23) соответствует Mode 0. Считается, что наиболее употребительны Mode 0 и Mode 3.

Для выбора в ведущем устройстве (в МК) одного из режимов, различающихся полярностью тактовых импульсов и их «фазой» (порядком действий), в одном из периферийных регистров подсистемы SPI обычно есть два управляющих бита: CPOL (полярность) и CPHA (фаза) – приведенные обозначения битов и обозначения

режимов являются широко употребительными. Периферийные устройства (ведомые) чаще всего поддерживают единственный режим, и перед началом обмена ведущий (МК) должен быть настроен на этот режим.

Универсальный синхронно-асинхронный приемопередатчик USART (Universal Synchronous Asynchronous Receiver Transmitter) – это модуль последовательного ввода-вывода, который может использоваться для работы с периферийными устройствами, такими как терминалы или персональные компьютеры, модемы, микросхемы ЦАП, АЦП, последовательные EEPROM и т.д.

USART может работать в трех режимах: асинхронный, полный дуплекс; ведущий синхронный, полудуплекс;ведомый синхронный, полудуплекс.

Модуль приемо-передатчика обеспечивает полнодуплексный обмен по последовательному каналу, при этом скорость передачи данных может варьироваться в довольно широких пределах. Длина посылки может составлять от 5 до 9 битов. В модуле присутствует схема контроля и формирования бита четности.

Модуль USART может обнаруживать следующие внештатные ситуации: переполнение; ошибка кадрирования; неверный старт-бит.

Для уменьшения вероятности сбоев в модуле также реализована функция фильтрации помех. Для взаимодействия с программой в микроконтроллере, как правило, предусмотрены прерывания, запрос на генерацию которых формируется при наступлении следующих событий: «передача завершена»; «регистр данных передатчика пуст»; «прием завершен».

Интерфейс USART задействует 3 линии ввода-вывода: TxD – передача данных; RxD – прием данных; XCK – тактовый сигнал (используется только в синхронном режиме) (рис. 1.24).

Рис. 1.24. Соединение устройств по интерфейсу USART

Передача данных осуществляется кадрами. Кадр – совокупность одного слова данных и сопутствующей информации. Кадр начинается со старт-бита, за которым следует младший бит слова данных. Слово данных может состоять из 5, 6, 7, 8 или 9 битов. После старшего бита слова данных следует один или два стопбита. Если включена схема формирования бита четности (P), он включается между старшим битом слова данных и первым стопбитом (рис. 1.25).

Рис. 1.25. Структура кадра

Схема контроля четности предполагает два возможных режима работы:

– контроль четности (even parity) – исключающее ИЛИ всех битов слова данных: $P_{EVEN} = d_n \oplus d_{n-1} \oplus d_1 \oplus d_0$.

– контроль нечетности (odd parity) – исключающее ИЛИ всех битов слова данных с логической единицей: $P_{ODD} = d_n \oplus d_{n-1} \oplus d_1 \oplus d_0 \oplus 1$.

В асинхронном режиме, а также в синхронном режиме при работе в качестве ведущего скорость приема и передачи данных задается контроллером скорости передачи, работающим как делитель системного тактового сигнала с программируемым коэффициентом деления.

Скорость передачи данных, как правило, выбирается из стандартного ряда: 300; 600; 1200; 2400; 4800; 9600; 19200; 38400; 57600; 115200; 230400; 460800; 921600 бод.

Реальное значение скорости передачи может отличаться от стандартного ряда, поскольку определяется тактовой частотой микроконтроллера и ограничено разрядностью регистра конфигурации.

Для обеспечения приема в асинхронном режиме работы используются схемы восстановления тактового сигнала и данных. Схема восстановления тактового сигнала предназначена для син-

хронизации внутреннего тактового сигнала, формируемого контроллером скорости передачи, и кадров, поступающих на вывод RxD микроконтроллера. Схема восстановления данных осуществляет считывание и фильтрацию каждого бита принимаемого кадра. Схема восстановления тактового сигнала осуществляет опрос входа приемника с целью определения старт-бита кадра.

Обнаружение спадающего фронта на выводе RxD интерпретируется как возможное появление переднего фронта старт-бита. После этого проверяется значение заданных значений выборок входного сигнала. Если значение хотя бы двух выборок из указанных равно «1», старт-бит считается ложным (помеха), а приемник переходит к ожиданию следующего изменения входного сигнала с «1» в «0». В противном случае считается, что обнаружен старт-бит новой последовательности, с которым синхронизируется внутренний тактовый сигнал приемника. После этого начинает работать схема восстановления данных.

Решение о значении принятого бита принимается также по результатам заданных выборок входного сигнала. Состоянием бита считается логическое значение, которое было получено, хотя бы в двух из трех выборок. Процесс распознавания повторяется для всех битов принимаемого кадра, включая первый стоп-бит.

Старт-бит нового кадра может передаваться сразу же после последней выборки, используемой для определения значения бита.

При работе в синхронном режиме в качестве ведомого скорость приема и передачи определяется частотой сигнала, поступающего на вход XCK.

При работе модуля USART в синхронном режиме состояние вывода RxD считывается по одному из фронтов сигнала XCK, а выдача сигнала вывод TxD – по другому фронту.

Протокол работы USART совмещается с одним из вариантов подключения интерфейса RS232 (при наличии модуля согласующего уровни сигналов). В реальных цифровых системах используется широкий набор интерфейсов различного типа. В данной работе нет возможности их подробного описания. В последующих разделах остановимся на тех из них, которые реально достаточно широко используются в медицинском приборостроении.

Универсальные интерфейсы DICOM

Решением проблем совместимости медицинской техники с форматами передачи данных внутри одного ЛПУ являются аппаратно-программные комплексы, преобразующие исходный формат информации к виду, воспринимаемому ЛПУ. Такие устройства уже существуют. Перечислим их:

1. DICOM box (CA-, CA+, DVI).
2. Mediphan MedRecorder
3. TIMS DICOM System

В табл. 1.2. приведены основные технические характеристики устройств. Рассмотрим каждое устройство подробнее.

Таблица 1.2

Сравнительные технические характеристики устройств

Устройство	Поддержка DICOM	Разрешение изображения	Разрешение видео	Сжатие данных	Кадров в секунду	Поддержка DVI/VGA
DICOM box CA-	+	2048x1024	–	Нет	–	Нет
DICOM box CA+	+	2048x1024	2048x1024	Нет	25	Нет
DICOM box DVI	+	2048x1024	2048x1024	Нет	25	Да
TIMS 200 DICOM	+	1280x1024	–	Да	30	Да
TIMS 1000 DICOM	+	1280x1024	–	Нет	–	Да
TIMS 2000 SP DICOM	+	1280x1024	1280x1024	Нет	30	Нет
Mediphan MedRecorder	+	Нет	1600x1200	Да	40	Да

DICOM box является универсальным интерфейсом DICOM, позволяющий пользователям подключаться устройства в МИС без наличия DICOM-интерфейса.

DICOM box CA-

DICOM box CA- – захватывает изображение из стоп-кадра видеосигнала и преобразует изображения в формат DICOM (рис. 1.26). В состав МИС могут входить:

– DICOM-принтеры (DICOM Laser, Non-DICOM Laser через переходник Interpreter);

– DICOM-рабочие станции (Workstation) – рабочее место специалиста;

– DICOM-серверы (DICOM Archive HIS/RIS);

– медицинское оборудование, подключаемое через переходник DICOM CA: CT/MRI (компьютерный томограф), Angio (КТ ангиограф), Fluoro (флюорограф), Mobile C-arm (хирургическая рентгеновская система), Ultrasound (УЗИ), DICOM network (МИС DICOM).

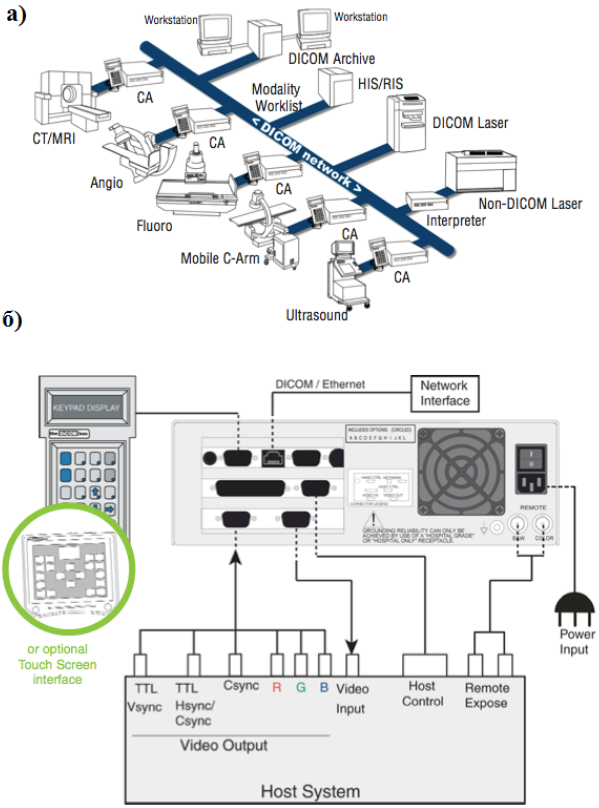


Рис. 1.26. *а* – МИС DICOM; *б* – Устройство DICOM box CA-

Переходник (рис. 1.26, *а*) располагается между медицинскими приборами. Устройства, имеющие интерфейс DICOM, подключаются напрямую в МИС. На рис. 1.26, *б* изображен внешний блок DICOM CA-. При помощи имеющихся разъемов (TTL Vsync, CSync, R,G,B, Video Input), а также сетевых разъемов (Ethernet) возможен захват изображений медицинской техники, не имеющей интерфейса DICOM [12]. DICOM CA- обеспечивает работу стандарта и используется для создания гибкой и масштабируемой сети DICOM. Захватывает изображение из видео с параметрами до 166 МГц и разрешением 2048x1024 пикселей.

DICOM box CA+

DICOM CA+ – осуществляет захват динамического видео в реальном времени и преобразование из собственного формата в DICOM монохромных и цветных источников видеосигнала (рис. 1.27, *б*).

DICOM box CA+ является универсальным интерфейсом DICOM, который позволяет пользователям просматривать динамическое видео. В отличие от DICOM box CA– осуществляет работу лишь с видеосигналом. Предоставляет возможность преобразования монохромного и цветного видео в реальном времени, захвата от десяти секунд до нескольких минут видеоданных. DICOM box CA + упрощает процесс подключения и обеспечивает получение видео, печать. CA + может быть полностью интегрирован в МИС системы. В состав МИС могут входить:

– DICOM-принтеры (DICOM Laser, Non-DICOM Laser через переходник Interpreter);

– DICOM-рабочие станции (Workstation) – рабочее место специалиста;

– DICOM-серверы (DICOM Archive HIS/RIS);

– медицинское оборудование, подключаемое через переходник DICOM CA+: Fluoro (флюорограф), Mobile C-arm (хирургическая рентгеновская система), Ultrasound (УЗИ), DICOM network (МИС DICOM).

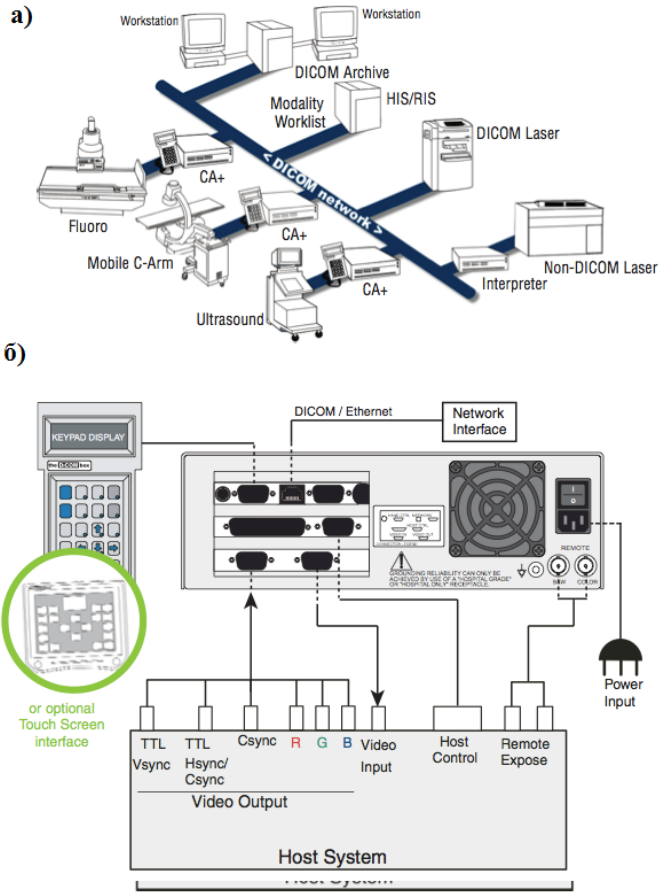


Рис. 1.27. *а* – МИС DICOM; *б* – устройство DICOM box CA+

Переходник (рис. 1.27, *а*) располагается между медицинскими приборами. Устройства, имеющее интерфейс DICOM, подключаются напрямую в МИС. На рис. 1.27, *б* изображен внешний блок DICOM CA+. При помощи имеющихся разъемов (TTL Vsync, CSync, R,G,B, Video Input), а также сетевых разъемов (Ethernet) возможен захват изображений медицинской техники, не имеющей интерфейса DICOM.

DICOM box DVI

DICOM box DVI – позволяет захватывать изображение от сторонних цифровых форматов медицинской техники, включая различные модели сканеров, для дальнейшего преобразования в DICOM. В отличии от DICOM box CA- и DICOM box CA+ осуществляет работу с цифровым не DICOM-сигналом (рис. 1.28).

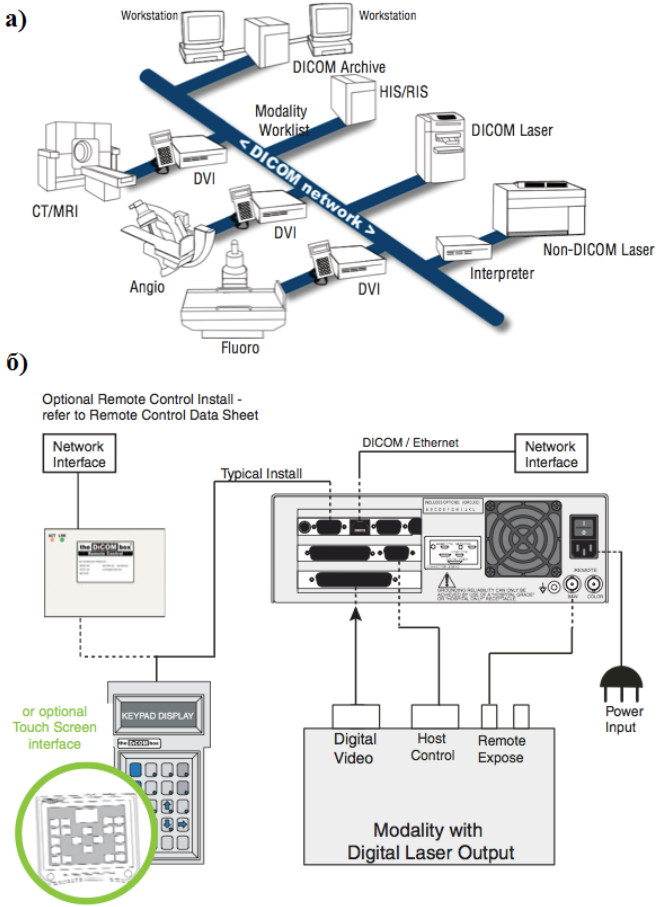


Рис. 1.28. *а* – МИС DICOM; *б* – устройство DICOM box DVI

DICOM DVI обеспечивает стандарт DICOM для различных сетевых приложений. Захватывает цифровое изображение от родного интерфейса для преобразования в DICOM и дальнейшей передачи по сети Ethernet [12]. DVI система предлагает полную поддержку «паспорта» пациента, то есть его имени, идентификатора и регистрационного номера.

В состав сети входят:

– DICOM-принтеры (DICOM Laser, Non-DICOM Laser через переходник Interpreter);

– DICOM-рабочие станции (Workstation) – рабочее место специалиста;

– DICOM-серверы (DICOM Archive HIS/RIS);

– медицинское оборудование, подключаемое через переходник DICOM DVI: CT/MRI (компьютерный томограф), Angio (КТ ангиограф), Fluoro (флюорограф).

Mediphan MedRecorder

Mediphan MedRecorder (рис. 1.29) – устройство для захвата, архивирования, систематизирования и просмотра медицинских изображений (в том числе, в стандарте DICOM) с любой медицинской аппаратуры, имеющей VGA или DVI дисплей.

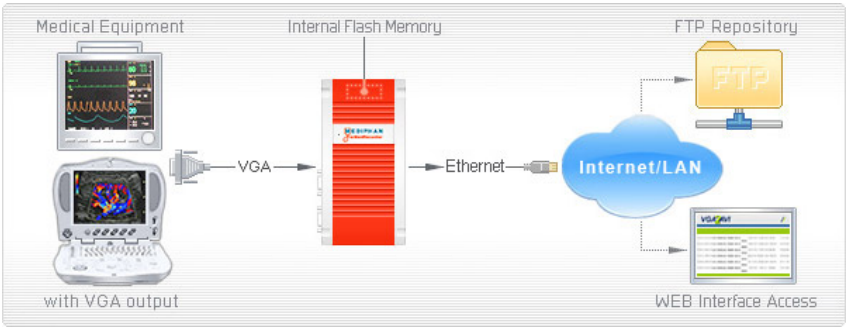


Рис. 1.29. Общий вид Mediphan MedRecorder

В состав сети входят:

– медицинское оборудование, с которого происходит захват, передача по средствам VGA-канала и преобразование изображения (Medical Equipment);

– Mediphan MedRecorder – интерфейс для преобразования сигнала к DICOM-формату;

– FTP – хранилище данных (FTP Repository);

– Web-интерфейс управления системой (Web Interface Access).

MedRecorder работает в системе PACS (Picture Archiving and Communication System) и может использоваться для повышения безопасности хранения и передачи данных в медицинских учреждениях. Оно позволяет захватывать изображение и преобразовывать в формат DICOM. Запись производится практически с любого устройства, имеющего дисплей [13].

MedRecorder подключается к DVI или VGA разъемам медицинской аппаратуры, захватывает изображение с экрана в исходном качестве, записывает и архивирует его для последующего изучения. После сохранения записей (видео или изображений в стандарте DICOM) их можно сразу же загружать на сервер (например, PACS). Доступ к записям может быть произведен через удобный пользовательский интерфейс MedRecorder Portal, для чего не требуется установка дополнительных программ. MedRecorder – это полноценное решение для захвата, каталогизации и архивирования медицинских изображений и видео.

TIMS DICOM

TIMS DICOM (рис. 1.30) – система цифровой визуализации медицинской системы. TIMS преобразует любой не-DICOM формат в DICOM. В результате цифровые исследования могут быть отправлены на серверы, хранящие информацию, записаны на CD / DVD / USB, а также распечатаны на пленку или бумагу. TIMS может преобразовать любой медицинской формат в DICOM, включая данные УЗИ, рентгеноскопии, эндоскопии, движение рентген, ангиография, ядерная медицина, МР, КТ, и многое другое. TIMS отражает статические изображения, а также потоковое видео при частоте 30 кадров в секунду. TIMS преобразует изображение и видео в DICOM JPG, BMP, AVI.

В состав сети входят:

– медицинское оборудование, с которого происходит захват, передача и преобразование изображения.

– TIMS – интерфейс для преобразования сигнала к DICOM-формату.

– Network телемедицинская сеть;

– Web-интерфейс управления системой (Web Interface Access).

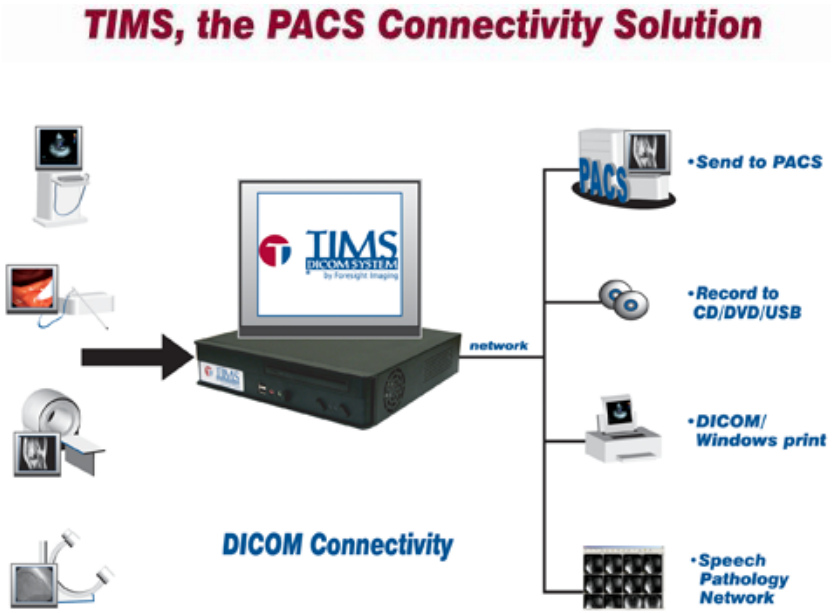


Рис. 1.30. Общий вид TIMS DICOM

Различают три версии TIMS DICOM:

1) TIMS 500 DICOM:

– Используется для всех аналоговых и DICOM-медицинских устройств;

– Захват только статического изображения;

– Преобразование JPG, BMP, и AVI файлов в DICOM;

– Отправка и прием DICOM;

– Малые габариты компьютера.

2) TIMS 1000 DICOM:

– Используется для всех аналоговых и DICOM-медицинских устройств;

– Захват только статического изображения;

– Преобразование JPG, BMP, и AVI файлов в DICOM;

– Отправка и прием DICOM файлов;

– Малые габариты компьютера с CD / DVD–RW.

3) TIMS 2000 SP DICOM:

– Для всех аналоговых и DICOM медицинских условий

– Захват статического и потокового видеосигнала;

– Синхронный захват аудио сигнала;

– Преобразование JPG, BMP, и AVI файлов в DICOM;

– Отправка и прием DICOM файлов;

– Малые габариты компьютера с CD / DVD–RW и монитором.

Программные средства преобразования

Кроме перечисленных выше программно-аппаратных комплексов по преобразованию форматов к виду DICOM существуют программные средства, устанавливаемые на рабочую станцию врача и осуществляющие преобразования форматов. Примером таких программ-конверторов могут служить Fadonics Dicom и Dicomapps.

Данные приложения преобразуют в DICOM изображения *.JPEG, *. BMP, *. GIF, *. PNG, *. TIFF, *. WMF форматов. Их отличает простота и надежность. Однако для использования данных систем необходим ПК с возможностью подключения медицинской техники и передачи сигнала для дальнейшей обработки.

В табл. 1.3 проведем сравнение рассмотренных устройств относительно обозначенных проблем.

Дополнительно в табл. 1.3 введен критерий преобразования вышеперечисленными средствами входной информации в не-DICOM формат. Из табл. 1.3 видно, что существующие средства не позволяют решить все имеющиеся проблемы.

Таблица 1.3

Возможность решения проблем устройствами

	Поддержка оборудования, производитель которого перестал существовать или был поглощён другим производителем	Наличие медицинских устройств, изготовленных без интерфейса DICOM	Необходимость задействовать в работе общие немедицинские устройства (цифровые или аналоговые)	Взаимодействие между ЛПУ затруднено наличием множества разных систем обмена медицинской информацией от различных производителей	Передача медицинских изображений по низкоскоростным (узкополосным) линиям передачи данных	Отсутствие единого стандарта взаимодействия между ЛПУ	Неправильная поддержка DICOM-стандарта программным обеспечением возможна при разработке ПО сторонними производителями	Возможность поддержки не-DICOM стандартов
DICOM box CA-	Да	Да	Да	Нет	Нет	Нет	Нет	Нет
DICOM box CA+	Да	Да	Да	Нет	Нет	Нет	Нет	Нет
DICOM box DVI	Да	Да	Да	Нет	Нет	Нет	Нет	Нет
TIMS 200 DICOM	Да	Да	Да	Нет	Нет	Нет	Нет	Нет
TIMS 1000 DICOM	Да	Да	Да	Нет	Нет	Нет	Нет	Нет
TIMS 2000 SP DICOM	Да	Да	Да	Нет	Нет	Нет	Нет	Нет
Mediphan MedRecorder	Да	Да	Да	Нет	Нет	Нет	Нет	Нет
Программные средства	Нет	Да	Да	Нет	Нет	Нет	Нет	Нет

Постановка задачи исследования

Проведенный анализ измерительно-вычислительных систем, применяемых в различных областях телемедицины, несмотря на различие сфер их применения, позволяет выделить общие проблемы при передаче информационных пакетов:

1. Необходимость поддержки оборудования, производитель которого перестал существовать или был поглощён другим производителем.
2. Наличие устройств, изготовленных без современных интерфейсов.
3. Наличие в системах аналоговых устройств. Использование аналоговых устройств требует преобразования аналоговых данных в цифровые для совместимости с современными интерфейсами.
4. Взаимодействие между системами затруднено наличием множества разных систем обмена информацией от различных производителей.
5. Реализация передачи информации по низкоскоростным (узкополосным) линиям передачи данных.
6. Отсутствие единого стандарта взаимодействия внутри информационно-измерительных систем и между информационно-измерительными системами.
7. Неправильная поддержка современных стандартов передачи данных программным обеспечением.

На примере медицинской техники, как наиболее показательной области применения измерительно-вычислительных систем, анализ существующих решений показывает, что основные проблемы решаются при помощи DICOM-стандарта. Однако, существующие решения не обладают универсальностью, позволяя преобразовывать информацию только к виду DICOM-стандарта. Под универсальностью будем понимать способность программно-аппаратного комплекса преобразовывать входную информацию к выбранному из множества выходных форматов. Разработка данного комплекса должна опираться на математический аппарат, позволяющий проектировать универсальные средства преобразования, производить их оптимальный выбор и формализовать всю совокупность требуемых действий.

Проектирование данного комплекса должно учитывать многоуровневое представление взаимодействия устройств. Это позволит использовать принятые стандарты передачи данных в процессе функционирования системы. Процесс принятия решений при преобразовании систем представления информации требуется обосновывать и выбирать наиболее подходящий по критериям сложности и ясности алгоритма. При этом важно учесть масштабируемость задачи, то есть легкость увеличения объема решаемой задачи и построения решающего правила. Для хранения вариантов возможных преобразований необходимо иметь базу данных (БД), хранящую все варианты входного и выходного формата, последовательность преобразований для каждого отдельного случая входного/выходного формата.

Программно-аппаратный комплекс, решающий вышеперечисленные проблемы, обязан учитывать, что при работе и взаимодействии сложных вычислительно-измерительных систем возможно не только отсутствие полного «взаимопонимания» между системами, но, что более опасно, появление различных видов метрологических погрешностей, влияющих на результат измерений и приводящих к искажению первоначальной информации, что в свою очередь может привести к принятию неверных решений. Следовательно, при решении вышеописанных проблем необходимо особенно тщательно подойти к вопросу метрологических оценок взаимодействия измерительно-вычислительных систем. Таким образом, одним из важнейших свойств разрабатываемого программно-аппаратного комплекса должна являться метрологическая надежность. Под метрологической надежностью будем понимать надежность средств измерений в части сохранения метрологической исправности, т.е. такого состояния, в котором все метрологические характеристики средств измерений соответствуют установленным нормам.

При разработке комплекса необходимо будет учесть методические погрешности по всем типам:

1) основная / дополнительная;
2) случайная / систематическая;
3) статическая / динамическая (датирования);
4) абсолютная / относительная.

Кроме этого, программно-аппаратный комплекс должен обеспечивать высокую эффективность процесса передачи информации. Под эффективностью передачи информации будем понимать обеспечение наивысшего заданного быстродействия при сохранении требуемой достоверности. Где заданное быстродействие, в нашем случае, – скорость и время выполнения преобразований программно-аппаратным комплексом.

Назовем программно-аппаратный комплекс, позволяющий производить подключение техники в информационные сети, делать автоматическую настройку, трансформацию передаваемых данных, ориентируясь на существующие стандарты, приводить передаваемые информационные пакеты к виду, воспринимаемому как источником, так и приемником сигнала, обладать высокими быстродействием, эффективностью и надежностью, Гибким Интеллектуальны Интерфейсом (ГИИ). Под гибкостью подразумеваем возможность преобразования информации ко всем известным БД форматам. Под интеллектуальностью подразумеваем возможность программно-аппаратного комплекса к самостоятельному принятию решения в процессе преобразования, приема и передачи информации.

1.6. Типовые схемы включения микропроцессоров и микроконтроллеров в аппаратуре контроля за состоянием пациентов

При проектировании медицинского оборудования, использующего микропроцессоры и их наборы, следует иметь в виду, что некоторые из них не содержат внутренней памяти для хранения программ и промежуточных данных или объем этой памяти недостаточен для решения реальных задач. Поэтому для решения задач управления и обработки данных в составе медицинских приборов, аппаратов и систем собственно микропроцессоры (МП) обрамляются дополнительным оборудованием, которое обычно объединяется на основе шинной структуры организации связей.

Шинная структура организации связей – один из основных архитектурных принципов, характеризующих способ организации связей между устройствами внутри самого МП и с внешней средой. Выделяют три типа шин: адресная, данных и управления. На практике с целью сокращения количества вводов – выводов большой интегральной схемы (БИС) часто используют объединенную шину адресов и данных.

Полученные в результате такого объединения микропроцессорные системы (МПС), как и МП, могут быть универсальными и специализированными. На рис. 1.31 показана наиболее характерная структура МПС с одним центральным МП.

Совокупность шин передачи информации в МПС называют магистралью. В каждый момент времени по магистралям может передаваться только одно сообщение, посылаемое одним источником и предназначенное для одного или нескольких приемников. По шине адреса указывается код адреса функциональных блоков, для которых предназначается информация, передаваемая по шине данных.

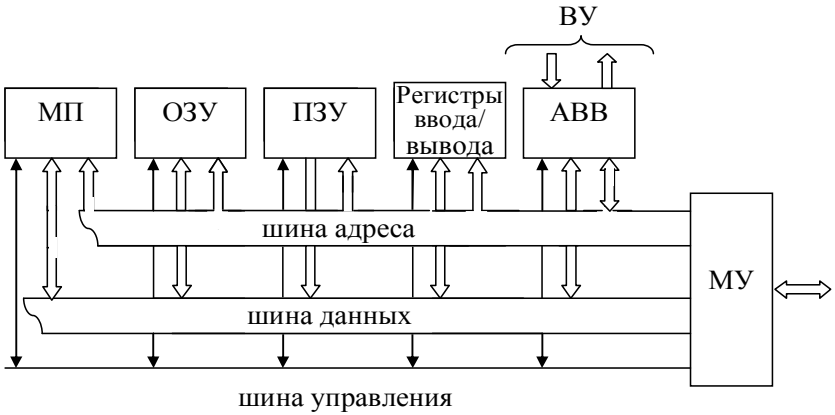


Рис. 1.31. Схема типовой МПС

Постоянное запоминающее устройство (ПЗУ) служит для хранения неизменяемой части программы и может работать только в режиме выдачи информации. Оперативное запоминающее устрой-

ство (ОЗУ) используется для хранения программ и промежуточных данных и может работать в режимах записи, чтения и хранения информации. Эти термины определены по отношению к МП. То есть термин «чтение» (сигнал IOR) означает, что МП читает информацию, например, из ОЗУ, а термин «запись» (сигнал IOW) означает, что информация из МП помещается на хранение в ОЗУ, передается во внешние устройства и т.д.

Для согласования МП с внешними устройствами (ВУ) используют специальные адаптеры ввода-вывода (АВВ), которые в простейшем случае могут быть представлены регистрами ввода-вывода. Обмен данными с ВУ может осуществляться тремя способами: программно, в режиме прерывания и прямого доступа к памяти (ПДП). При программном обмене внешние устройства рассматриваются МП, как функциональные узлы, снабженные своими уникальными адресами, позволяющими однозначного определять адресата взаимодействующего с МП по программе микропроцессора. Режим прерываний (сигнал IRQ) характеризуется тем, что внешние устройства обращаются к МП по мере их готовности к обмену, что исключает необходимость постоянного программного опроса внешних устройств, высвобождая тем самым ресурсы МП для другой работы. В режиме ПДП внешним устройствам разрешается обмен с памятью МПС без участия МП, что позволяет увеличивать соответствующую скорость обмена информацией с ВУ.

В тех случаях, когда разработчики планируют для потребителя подсоединение к магистрали дополнительных функциональных блоков (открытый вариант МПС), магистраль обычно снабжается магистральными усилителями (МУ), увеличивающими ее нагрузочную способность.

В более сложных вариантах, когда при передаче сигналы необходимо не только усиливать по мощности, но и преобразовывать, к магистралям подключают соответствующие адаптеры.

На рис. 1.31 показан набор основных функциональных блоков, входящих в МПС. Каждый из этих блоков может состоять из одной или нескольких БИС. Выбор МП и МПК для создания той или иной МПС не всегда является тривиальной задачей из-за значительного количества различных серий МПК, разнообразных и не

всегда совместимых характеристик, сложности требований к вычислительным и управляющим устройствам и др.

К основным характеристикам, которые используются при сопоставлении и выборе МП, относят:

– вид МП (универсальный, специализированный, однокристальный или многокристальный);

– технологию изготовления (p-МОП, n-МОП, КМОП, 1^2Л, ТТЛ, ЭСЛ и др.); информация о технологии позволяет ориентировочно судить о потребляемой мощности и среднем быстродействии;

– разрядность (4, 8, 16, 32);

– длину информационного слова, одновременно обрабатываемую процессором;

– емкость адресуемой памяти, характеризующую возможности МП по взаимодействию с запоминающими устройствами;

– принцип управления с «жесткой логикой» или микропрограммный;

– быстродействие, которое может характеризоваться продолжительностью выполнения одной операции, числом операций типа обмен регистр-регистр, выполняемых в секунду, тактовой частотой;

– напряжение электропитания и потребляемую мощность;

– конструктивные данные.

К основным критериям, которые позволяют оценить требования к структуре и элементной базе МПС относят:

– число МП, их разрядность, количество уровней прерывания, число регистров связи и объемы ОЗУ и ПЗУ;

– необходимость реализации команд умножения, деления, работы с плавающей запятой, вычисления специальных функций;

– ориентировочный объем программ позволяющий определить требуемые объемы ОЗУ и ПЗУ с учетом использования средств отладки, роль которых возрастает с ростом объема программы;

– наличие соответствующих средств отладки и его контролепригодность, определяющая удобство контроля и поиска неисправностей;

– при использовании нескольких МП оценивается единство программного обеспечения;

– возможность работы МПС в реальном масштабе времени.

Последний критерий означает, что средняя скорость обработки данных должна выше скорости поступления данных. Если это затруднительно при выбранных типах МП и структуры МПС используют буферизацию данных на входе МПС, или строят многопроцессорные системы различной архитектуры.

Грамотное использование микропроцессорных систем в приборах медицинского назначения позволяет значительно расширить их функциональные возможности часто при более простых структурах. В качестве примера рассмотрим два варианта построения многоканальных микропроцессорных электрокардиографов с использованием различных типов промышленных микросхем. Отличительной чертой современного кардиографа является наличие высоких требований к точности регистрации ЭКГ, которая в основном обеспечивается специальными схемотехническими решениями входных цепей обеспечивающих низкую чувствительность к шумам от сетевых наводок, высокочастотных электромагнитных полей, электродов, движения пациента и т.д.

Рассмотрим вначале вариант, в котором основную задачу подавления различного рода помех решают в основном аналоговые цепи (рис. 1.32). В этой схеме сигнал снимаемый с электродных отведений $ЭО_1$,…, $ЭО_n$ нормализуется высококачественными инструментальными усилителями ($ИУ_1$…$ИУ_n$) с подавлением синфазных помех. Фильтры низких частот (ФНЧ) обеспечивают подавление высокочастотных радиопомех. Фильтры высоких частот (ФВЧ) рассчитаны на подавление частот в диапазоне 0,…,0,05 Гц с целью уменьшения дрейфа изолинии, полосовые фильтры (ПФ) обеспечивают подавление сетевой помехи. Операционные усилители (ОУ) формируют ЭКГ сигналы требуемой амплитуды, которые аналоговым коммутатором (АК) управляемым со стороны МПС поочередно подключаются к аналого-цифровому преобразователю (АЦП). Кардиограф работает под управлением МП. В ПЗУ хранятся стандартные программы обработки и визуализации электрокардиосигнала (ЭКС). В ОЗУ хранятся фрагменты ЭКС и результаты вычислений (если это предусмотрено проектом). Для подключения электрокардиографа к другим вычислительным устройствам, например, к ПЭВМ включен микроконтроллер со-

пряжения (МКС) обеспечивающий выбранные протоколы обмена (RS 232C, USB и т.д.). С блока клавиатуры (БКл.) задаются режимы работы, блок отображения информации (БОИ) на экране монитора формирует изображение ЭКС и результаты вычислений. При расширении функциональных задач решаемых БОИ для его управления может быть выделен специальный микроконтроллер. Для формирования графической записи ЭКС на бумажном носителе в состав МКС может быть включен регистратор (Р) со своим микроконтроллером (МК).

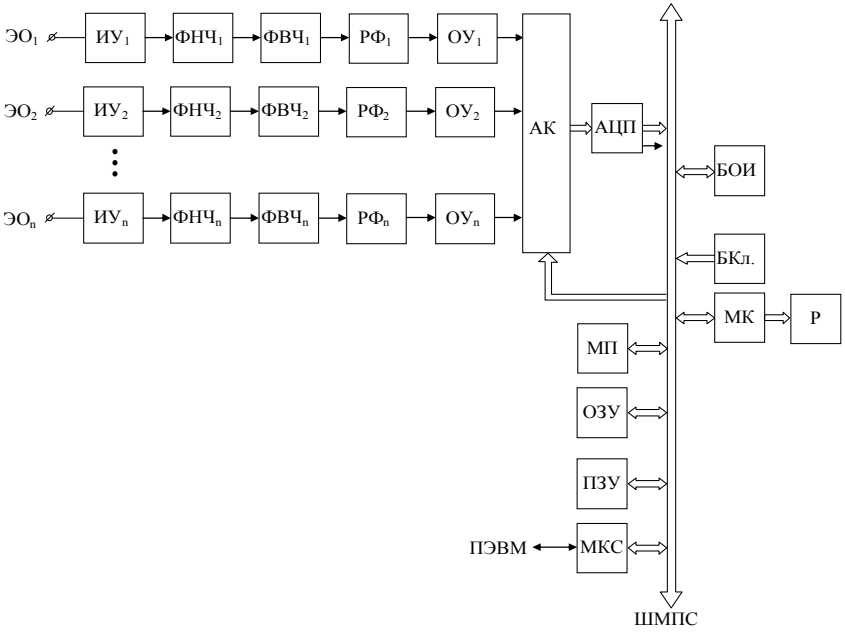


Рис. 1.32. Структурная схема многоканального микропроцессорного кардиомонитора с аналоговой фильтрацией

На рис. 1.32 три шины изображённые на рис. 1.31 условно изображены одной шиной МПС (Ш МПС).

Приведенная на рис. 1.32 схема относится к ранним разработкам электрокардиографов. К наиболее существенным недостаткам

этой схемы относят достаточно большие аппаратурные затраты, трудоемкость в подстройке активных фильтров, значительные неуправляемые фазовые сдвиги от фильтров, дополнительные шумы дискретизации от работы АЦП.

В более современных конструкциях вместо множества активных фильтров и мультиплексора используют сигма-дельта АЦП ($\sum-\Delta$ АЦП). Улучшение медико-технических характеристик медицинских приборов, включая электрокардиографы, использующих $\sum-\Delta$ АЦП обеспечивается особыми свойствами этих преобразователей изучаемыми в специальных курсах и описываемых в соответствующей литературе. В этом разделе остановимся кратко только на описании этих свойств. Сигма-дельта АЦП имеют большой динамический амплитудный диапазон позволяющий передать на обработку в цифровые блоки весь сигнал без первичного отделения различных его составляющих. Большинство микросхем АЦП этого типа по своим входам ориентированы сразу на подключение к датчикам и не требуют специальной нормализации сигнала, что позволяет заменить сложные и относительно дорогие инструментальные усилители на обычные операционные усилители (ОУ). Кроме того в этих типах АЦП имеются программируемые цифровые фильтры, что позволят уменьшить требования к ФНЧ и исключить ФВЧ. Выходные каскады $\sum-\Delta$ АЦП устроены таким образом, что они легко коммутируют множество своих и внешних цифровых выходов на один управляемый высокоимпедансный выход. Всё это позволяет упростить схемотехнику электрокардиографической и другой медицинской аппаратуры при сохранении высокого разрешения и производительности обработки многоканальных сигналов. Пример структуры многоканального электрокардиографа с использованием $\sum-\Delta$ АЦП приведен на рис. 1.33.

В этой схеме с учетом высокой частоты дискретизации в $\sum-\Delta$ АЦП вместо активных фильтров могут быть использованы пассивные R-C-цепи. Остальные обозначения те же, что и на рис. 1.32.

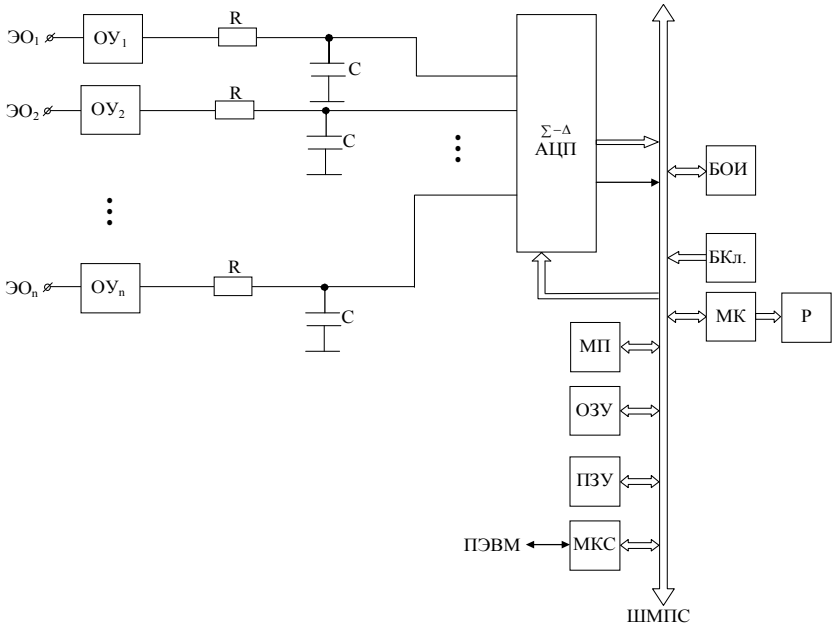


Рис. 1.33. Структурная схема многоканального электрокардиографа на основе $\sum -\Delta$ АЦП

Одним из основных недостатков использования микропроцессоров в медицинской аппаратуре является относительная громоздкость МПС, требующая наличия ОЗУ, ПЗУ, буферных регистров и т.д. Лучшими технико-экономическими показателями обладают микроконтроллеры, в корпусе БИС которых имеются собственные перепрограммируемые ПЗУ, ОЗУ, таймеры, аналого-цифровые преобразователи, порты со встроенными схемами управления и другие элементы, полезные для построения гибких и высокоэффективных устройств обработки данных и управления узлами и блоками медицинской аппаратуры. Конкретный состав аппаратно-программного обеспечения микроконтроллеров определяется их серией и модификацией.

Отличительной особенностью микроконтроллеров является их ориентация на решение задач управления, хотя они так же предназначены для решения логических и арифметических задач. Для обмена с «внешним миром» в микроконтроллерах используют специально организованные регистры-порты, и интерфейсные блоки, реализующие различные протоколы взаимодействия с внешними устройствами.

На рис. 1.34 приведена типовая структура простейшего микроконтроллера.

Рис. 1.34. Типовая структура микроконтроллера

В наиболее простых и дешевых микроконтроллерах обмен данными, адресами, командами организуется через программно управляемые порты, разряды которых, в зависимости от выбираемых режимов работы, могут выполнять разные функции (адресация внутренней и внешней памяти, ввод/вывод данных в последовательных и параллельных режимах, прием и выдача команд управления, организация прерываний и т.д.).

Кроме портов работа микроконтроллеров поддерживается дополнительными сигналами и выводами (подключение кварцевого генератора, строб адреса, задание режимов работы и т.д.)

На рис. 1.35 приведен вариант взаимодействия микроконтроллера с медицинским прибором ($МП_Р$) режимы работы которого задаются блоком клавиатуры (БК).

В этой схеме порт Р0 запрограммирован на прием данных от медицинского прибора. Порт Р1передает медицинскому прибору результаты обработки данных, например для их отображения на экране монитора. Кварцевый резонатор Q задаёт скорость обработки данных.RST реализует общий сброс МК.

Сигнал ALE (выходной сигнал разрешения фиксации адреса) может быть использован для синхронизации обмена данных между МК и МПр. Порт Р2 запрограммирован на формирование команд управления медицинским прибором, и через порт Р3 осуществляется контроль состояния узлов и блоков МПр. Кроме того, часть разрядов этого порта могут быть использованы для обработки прерываний (сигналы INT_i).

По этой схеме можно составить представление о том, как МК общается со своим внешним окружением. Более мощные МК реализуют дополнительные наборы возможностей: встроенные монитор/отладчик программ; обработка прерываний от разных источников, аналоговый ввод/вывод (встроенные АЦП и ЦАП); управление работой жидкокристаллическими индикаторами и аудиокодеками; взаимодействие с различными типами памяти; реализация стандартных протоколов обмена типа SPI, I2C, I2S, UART и т.д.

Микроконтроллеры удобно использовать вместо микропроцессоров в схеме рис. 1.32 и 1.33, что значительно повышает функциональные возможности, позволяя при этом исключать аналогово-цифровые преобразователи, сократить объемы внешних ОЗУ и ПЗУ. Упростить средства сопряжения с ПЭВМ и т.д.

Значительное число интегральных микросхем (ИМС), включая микропроцессоры, микроконтроллеры, сигнальные процессоры и т.д. выпускаются с расчетом на использование батарейного питания, что очень важно, например, при создании носимых мониторных систем.

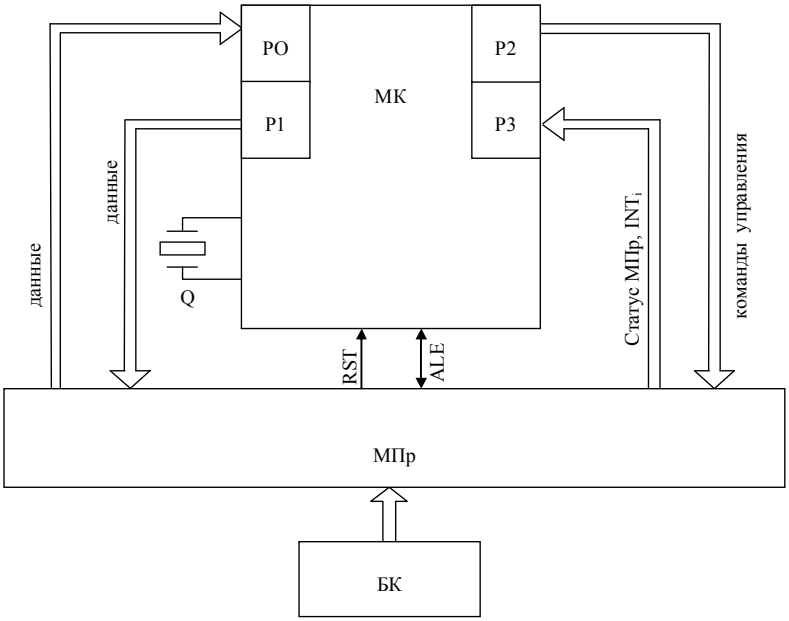


Рис. 1.35. Вариант подключения МК к медицинскому прибору

Для таких микросхем характерно низкое энергопотребление с переходом, по возможности, на различные режимы энергоэкономии с автоматическим отключением незадействованных в те или иные моменты времени узлов и блоков. Для управления режимами энергопотребления часто используется ИМС частов реального времени которые всегда находятся в активном режиме.

С ориентацией на такие часы фирма Analog Devices разработала оригинальный механизм оптимизации мощности энергопотребления с использованием линейки ИМС стабилизаторов с низкими и сверхнизкими показателями. На рис. 1.36 приведена схема организации батарейного питания микропроцессора с типовым периферийным оборудованием от фирмы Analog Devices.

В схеме приведенной на рис. 1.36 ИМС ADP160 – микромощный LDO-стабилизатор, который обеспечивает питание часов реального времени – центрального устройства систем управления электропитанием.

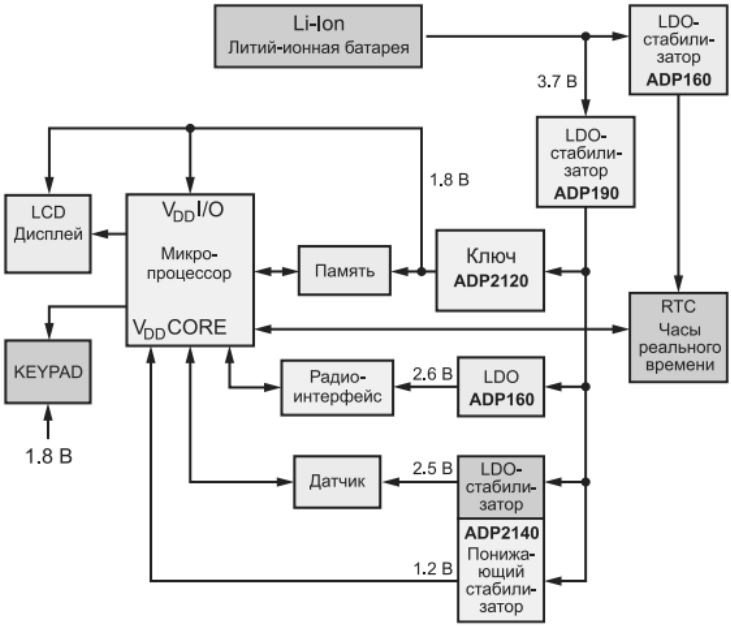


Рис. 1.36. Типовая схема организации питания в приборах с батарейным питанием

Без нагрузки ADP160 потребляет не более 560 нА и только 3 мкА – при токе нагрузки 100 мкА. ИМС ADP190 представляет собой ключ, позволяющий отключить находящуюся в неактивном режиме нагрузку от источника питания. При подключенной нагрузке ток, протекающий через ADP190 в шину «земли», не превышает 2 мкА. ИМС ADP2140 содержит понижающий стабилизатор с током нагрузки до 600 мА и LDO-стабилизатор – до 300 мА. Она выполнена в корпусе 10-LFCSP размерами 3×3 мм. На печатной плате ИМС ADP2140 вместе с внешними компонентами занимает площадь размерами 7,5×12 мм, что позволяет использовать ее в портативных медицинских приборах. ИМС ADM6305/ADM6306 – сдвоенные микромощные супервизоры напряжения, предназначенные для мониторинга двух напряжений питания и формирующие сиг-

нал сброса для сигнального процессора или микропроцессора, используемых в проектируемой системе. Малый ток потребления (до 5 мкА) и миниатюрные размеры позволяют успешно использовать эти ИМС в портативной медицинской аппаратуре с батарейным питанием. Для более полного представления о механизмах сопряжения микроконтроллеров, микропроцессоров и сигнальных процессоров с внешними устройствами в последующих разделах будет произведено описание наиболее популярных протоколов обмена (интерфейсов).

2. ПРОЕКТИРОВАНИЕ УЗЛОВ ЦИФРОВОЙ ОБРАБОТКИ С ИСПОЛЬЗОВАНИЕМ СИГНАЛЬНЫХ ПРОЦЕССОРОВ

В биотехнических системах различного типа информация об исследуемых объектах чаще всего поступает в виде аналоговых сигналов различной природы, которую необходимо обработать, интерпретировать и в некоторых случаях превратить обратно в управляющий сигнал. В силу сложности извлечения требуемой информации из исследуемого сигнала высокой его зашумленности, разнообразия и сложности алгоритмов интерпретации обработку данных целесообразно производить современными средствами цифровой обработки сигналов (ЦОС). Блок схема типичной дискретной системы ЦОС приведена на рис. 2.1.

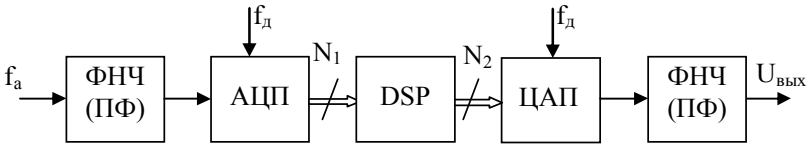


Рис. 2.1. Блок-схема системы обработки сигнала цифровым способом

В этой системе исследуемый аналоговый сигнал подвергается фильтрации фильтром низкой частоты (ФНЧ) или полосовым фильтром (ПФ) с целью улучшения соотношения сигнал/шум и преобразуется в цифровой сигнал разрядностью N_1 аналогово-цифровым преобразователем АЦП с частотой дискретизации $f_Д$.

Если к системе ЦОС предъявляются требования к обработке аналогового сигнала в реальном времени, то сигнальный процессор DSP должен закончить свои вычисления в пределах интервала дискретизации $1/f_Д$ и передать на свой выход результат обработки разрядностью N_2 до поступления новой порции информации со стороны АЦП. В случае если скорости работы DSP недостаточно,

может быть организовано хранение кодов принятых от АЦП в оперативной памяти DSP, но при этом алгоритм обработки не должен переполнить внутреннюю память приведя к потере информации. Блоки ЦАП и выходные фильтры требуются для задач, когда необходимо от цифрового кодирования вновь перейти к аналоговой форме представления информации, например при выводе сигналов электрокардиограммы на бумажные носители. В других многочисленных приложениях перехода от цифры к аналоговой форме сигнала не требуется, тогда ЦАП и выходные фильтры из схемы рис. 2.1 исключаются.

2.1. Аналогово-цифровое преобразование

Подавляющее число сигналов в биотехнических системах медицинского назначения, решающих задачи оценки состояния человека, имеют аналоговое представление и для использования средств ЦОС требуют их преобразования в цифровую форму. Эти преобразования осуществляют аналогово-цифровые преобразователи (АЦП).

Как правило, регистрируемые сигналы имеют весьма малые амплитуды с высоким уравнением шумов, к которым добавляются собственные шумы и погрешности АЦП. При неудачном выборе технических средств «суммарные» шумы биообъектов, датчиков и АЦП могут возрасти настолько, что из них будет очень трудно, а иногда практически невозможно, выделить информацию с требуемым качеством. Для того, чтобы грамотно оценить погрешности возникающее при работе АЦП рассмотрим более подробно механизмы их возникновения.

В реальных процедурах аналогово-цифрового преобразования существует два основных процесса: дискретизация по времени и квантование по амплитуде, которые определяют разрешающую способность производимых АЦП операций. Дискретизация непрерывных аналоговых сигналов осуществляется через интервал $t_д=1/f_д$, который необходимо тщательно выбирать для точного представления аналогового сигнала поступающего на вход АЦП с частотой f_a.

Рис. 2.2 иллюстрирует механизм возникновения погрешностей преобразования при неудачном выборе частоты $f_д$.

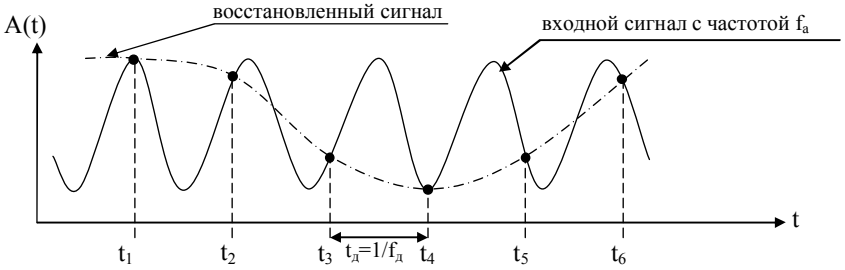


Рис. 2.2. Эффект получения «ложных» сигналов после преобразования

Как видно из рисунка при выбранном интервале (частоте) дискретизации из входного сигнала с частотой f_a будет формироваться выборка чисел, значения которых соответствуют амплитудам точек (A(t1), A(t2), A(t3)…). Если произвести обратное преобразование сигнала с помощью ЦАП, то будет получен сигнал, изображенный на рис. 2.2 пунктиром, который значительно отличается от того сигнала который был подан на АЦП.

Чтобы избежать таких искажений необходимо частоту дискретизации сигнала выбирать в соответствии с известной теоремой Котельникова (удовлетворять критерию Найквиста). В соответствии с этой теоремой (критерием) частота дискретизации сигнала, должна быть, по крайней мере, вдвое больше частоты исследуемого сигнала ($f_д>2f_a$). В противном случае информация об исследуемом сигнале будет полностью потеряна.

С учетом теоремы Котельникова самая высокая частота, содержащаяся в исследуемой частоте f_a, интересующая разработчиков аппаратуры связана с частотой дискретизации соотношением: $f_a<0{,}5f_д$. Частоту 0,5 $f_д$ называют частотой Найквиста, а частотный диапазон от 0 до 0,5 $f_д$. называют первой зоной Найквиста. Интервал от $0{,}5f_д$ до $f_д$ называют второй зоной Найквиста от $f_д$ до $1{,}5f_д$ – третьей зоной Найквиста и т.д.

В реальных сигналах присутствуют множество частотных составляющих, включая составляющие шума. Следует отметить, что

если не принять специальных мер, то все эти составляющие при подаче на вход АЦП будут подвергаться дискретизации и вносить свой вклад в погрешность преобразования. Причем различные составляющие шума несут в своем составе низкочастотные компоненты, попадающие в частотный диапазон исследуемого сигнала. Чтобы уменьшить влияние низкочастотных составляющих высокочастотного «шума» перед АЦП, как правило, ставят фильтры низкой частоты (ФНЧ) ослабляющие сигналы помехи. Естественно для этих фильтров частоту сопряжения выбирают с учетом полосы пропускания полезного сигнала f_a (рис. 2.3). Выше этой частоты сигналы ослабляются, а сигналы выше частоты $f_в$ на вход АЦП практически не проходят. В реальных условиях на частотах ниже f_a имеются различные паразитные составляющие обусловленные как особенностью работы биообъекта (например, на сигнал ЭЭГ накладываются сигналы ЭКГ в информативном частотном диапазоне), так и шумы от вешних систем, включая собственные шумы АЦП.

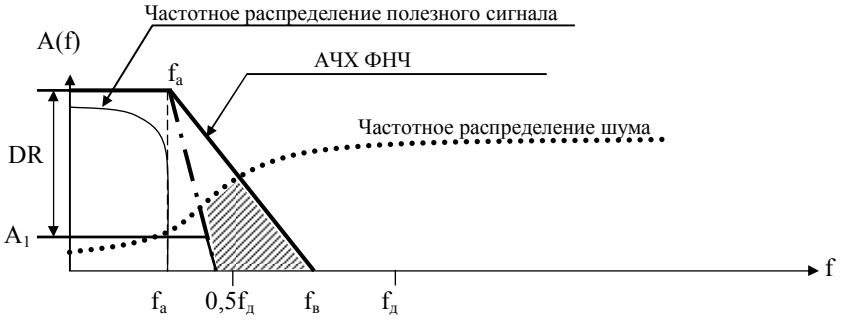


Рис. 2.3. Взаимодействие ФНЧ с полезным сигналом и шумовой составляющей АЦП

На рис. 2.3 показан вариант наложения частотных распределений полезного сигнала и шумов ниже частоты f_a. Эта ситуация означает, что ниже амплитуды A_1 полезный сигнал неотделим от сигнала шумов, а выделение полезного сигнала осуществляется в диапазоне DR называемом динамическим диапазоном системы. При проектирова-

нии систем ЦОС динамический диапазон системы выбирается исходя из требований точности воспроизведения сигнала.

Из рис. 2.3 видно, что крутизна спада амплитудно-частотной характеристики (АЧХ) ФНЧ определяет долю шумовой составляющей попадающей на вход АЦП. Чем круче сход АЧХ (штрихпунктирный спад), тем меньшая доля шума попадает на вход АЦП. При этом следует иметь ввиду что увеличение крутизны АЧХ ФНЧ приводит к увеличению затрат на изготовление аналоговых фильтров и увеличению фазовых искажений передаваемого сигнала поскольку различные его частотные составляющие будут подвергаться различному сдвигу фазы. Эти недостатки аналоговых фильтров легко устраняются алгоритмами цифровой фильтрации, поэтому одной из важных функций сигнальных процессов является аппаратная поддержка методов и алгоритмов фильтрации.

Одним из способов снижения требований к крутизне аналоговых фильтров является увеличение частоты дискретизации АЦП в К раз, что реализуется в так называемых сигма-дельта АЦП.

Схемотехника различных типов АЦП описана в различных источниках. В этой же литературе приводятся их характеристики, включая погрешности преобразований, которые позволяют подобрать нужный тип АЦП под конкретную решаемую задачу.

Учитывая все ввозрастающий интерес разработчиков биотехнических систем к использованию сигма-дельта АЦП ($\sum -\Delta$ АЦП), остановимся более подробно на их конструкции и принципах работы.

В современной литературе можно найти множество различных вариантов описаний функционирования $\sum -\Delta$ АЦП, но значительное их число переполнено достаточно сложными интегральными математическими выражениями достаточно сложными для понимания. С другой стороны на инженерных аспектах их построения можно выделить две основные части: аналоговую и цифровую, принцип работы которых можно объяснить доступным техническим языком (рис. 2.4).

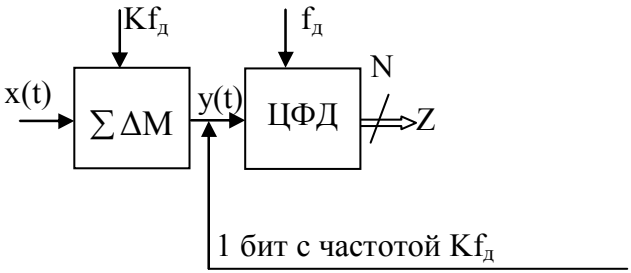


Рис. 2.4. Укрупненная схема $\sum -\Delta$ АЦП

Аналоговая часть имеет достаточно простую структуру (аналоговые сумматоры, интеграторы, компаратор, источник опорного напряжения, и коммутатор) и представляет собой $\sum -\Delta$ модулятор ($\sum -\Delta$ M), который превращает аналоговый сигнал x(t) в последовательный набор нулей и единиц y(t) следующих с частотой в К раз большей, чем обычно выбираемая для классических АЦП частота дискретизации. Таким образом, биты информации меняются с частотой $Kf_д$. Причем $\sum -\Delta$ модулятор работает так, что в последовательности y(f) нулей и единиц содержится среднее значение входного сигнала x(t). Цифровая часть $\sum -\Delta$ АЦП имеет достаточно сложную структуру. По существу она представляет собой сигнальный процессор реализующий функцию цифрового фильтра, на выходе которого с частотой $f_д$ вырабатываются искомые цифровые коды z формирующиеся из потока нулей и единиц y(t). Часто в литературе этот тип специализированного сигнального процессора называют цифровым фильтром с децимацией (ЦФД).

Использование высокой частоты дискретизации в аналоговой части схемы ($Kf_д$) и переход к «обычной» более низкой частоте выдачи выходных цифровых кодов $f_д$, определяемой по Котельникову (Найквисту), позволяет реализовывать аналогово-цифровое преобразование в широком амплитудном диапазоне напряжения

x(t) при снижении такой существенной погрешности работы АЦП, как шум квантования.

Рассмотрим вначале механизм формирования последовательного двоичного потока y(t) с частотой смены информации f_T на примере $\sum -\Delta$ АЦП первого порядка приведенного на рис. 2.5.

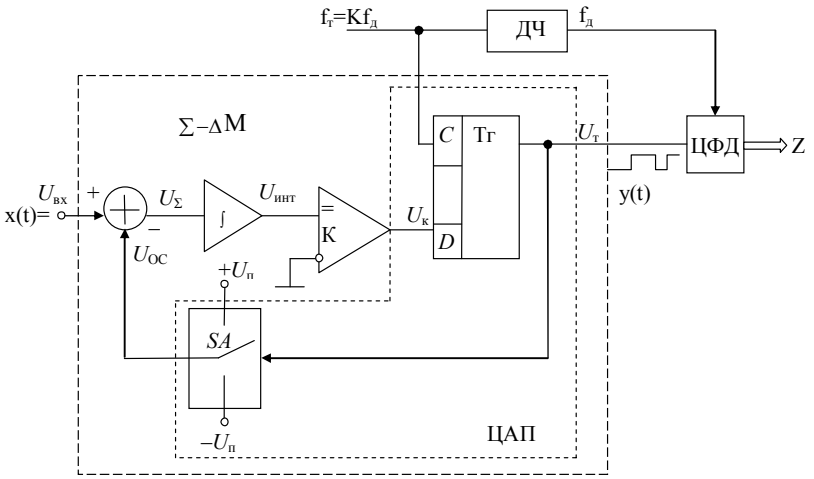


Рис. 2.5. Упрощенная схема ∑–Δ АЦП первого порядка

Основная часть этой схемы работает по принципу преобразования входного напряжения ($U_{вх}$) в частоту, которая преобразуется затем в последовательность усредненных цифровых кодов. На вход интегратора (∫) подается напряжение U_{Σ}, которое представляет собой разность между входным напряжением и напряжением обратной связи ($U_{ОС}$), формируемым цифроаналоговым преобразователем (ЦАП), состоящим из D-триггера (Тг) и аналогового коммутатора SA, подключающего вход обратной связи сумматора (∑) к одному из выходов двухполярного источника постоянного напряжения ($\pm U_{п}$) в зависимости от состояния D-триггера. Напомним, что D-триггер задерживает сигнал со входа D на один такт,

переключаясь по переднему фронту сигнала $f_т$ входа C, если вход D изменил свое состояние. Компаратор (К) сравнивает напряжение на выходе интегратора с нулем, поскольку его инверсный вход подключен к «земле». Частота опроса состояния компаратора через D-триггер выбирается в K раз большей, чем частота $f_д$ формирования выходного кода Z цифровым фильтром с децимацией (ЦФД).

Принцип преобразования аналогового сигнала в цифровой код с помощью ∑–Δ АЦП поясняют временные диаграммы, приведенные на рис. 2.6.

Пусть входное напряжение постоянно и составляет 0,6 В (см. рис. 2.6, *б*), а работа схемы тактируется с частотой $f_т$ (см. рис. 2.6, *а*). Выберем постоянную времени интегратора, численно равную периоду тактовых импульсов, а напряжения $+U_п = 1$ В и $–U_п = –1$ В. В нулевой период (см. рис. 2.6, *а*, такт 0) выходное напряжение интегратора $U_и$ (см. рис. 2.6, *д*) сбрасывается в нуль. Нулевое напряжение $U_{ОС}$ формируется на выходе ключа *SA* (см. рис. 2.6, *в*).

Таким образом, в нулевом цикле на входе интегратора $U_∑$ находится напряжение $U_{вх} = 0{,}6$ В, поскольку $U_{ОС} = 0$. После снятия сигнала нулевого периода интегратор интегрирует положительное напряжение $U_∑$. На его выходе формируется пилообразное положительное напряжение, переводящее компаратор в состояние логической единицы ($U_к = 1$) (см. рис. 2.6, *е*).

Первым тактовым импульсом T_1, являющимся счетным для триггера, последний передает единичное состояние компаратора на свой выход (см. рис. 2.6, *ж*).

В результате управляемый от триггера ключ *SA* переводится в верхнее положение и напряжение $U_п$=+1 В вычитается из $U_{вх}$. Таким образом напряжение на входе интегратора ($U_∑$=$U_{вх}$–$U_п$= –0,4 В) скачком переходит в отрицательную область (см. рис. 2.6, *г*), что приводит к линейному уменьшению напряжения на выходе интегратора (см. рис. 2.6, *д*).

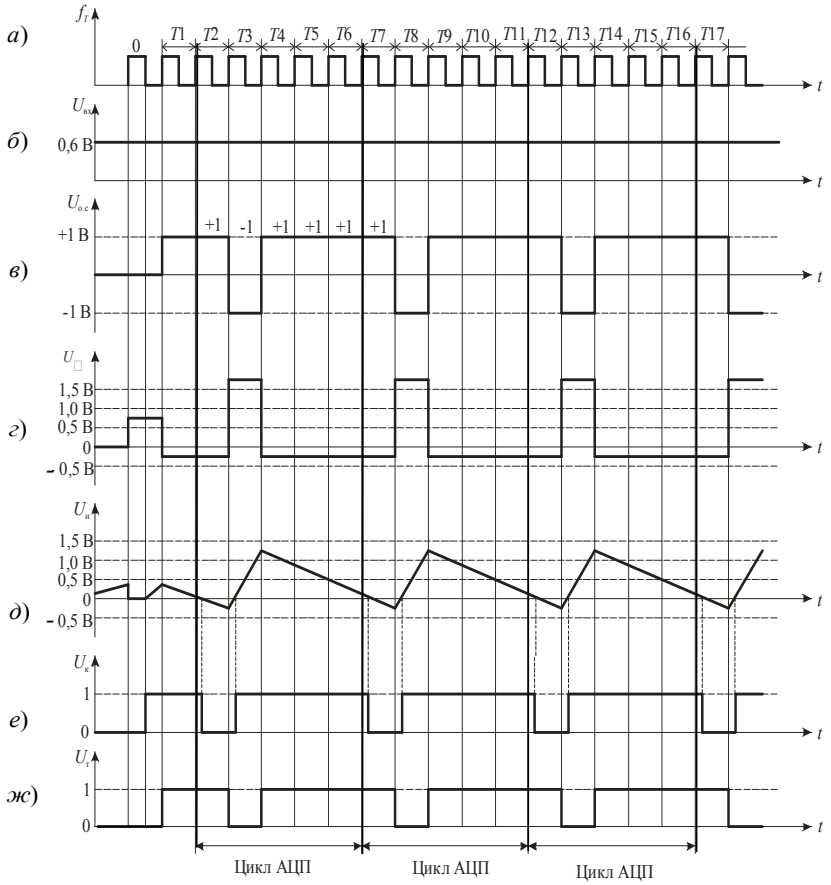


Рис. 2.6. Временные диаграммы, поясняющие принцип работы ∑–Δ-АЦП

Когда напряжение на выходе интегратора перейдет в отрицательную область, компаратор переключается в состояние «0» (см. рис. 2.6, *е*), которое тактовым импульсом T_3 передается на выход триггера (см. рис. 2.6, *ж*). Переход триггера в состояние «0» переключит *SA* в положение «$-U_п$» (см. рис. 2.6, *в*), поэтому на входе интегратора формируется скачок положительного напряжения ($U_∑$=0,6+1=1,6 В), которое интегратором преобразуется в линейно

нарастающее напряжение (см. рис. 2.6, *д*). Когда напряжение на выходе интегратора перейдет в положительную область, компаратор перейдет в состояние «1» (см. рис. 2.6, *е*), которое будет передано на выход Тг тактовым импульсом T_4 (см. рис. 2.6, *ж*). Ключ *SA* переводится в верхнее положение, U_{Σ} становится равным –0,4 В (см. рис. 2.6, *в–г*), интегратор формирует линейно спадающий участок (см. рис. 2.6, *д*). Когда напряжение $U_{и}$ перейдет в отрицательную область, последовательно, как и ранее, переключается компаратор, триггер и ключ *SA*, повторяя цикл преобразований до тех пор, пока АЦП не будет выключен или не изменится $U_{вх}$.

На рис. 2.6 повторяющиеся циклы выделены четырьмя вертикальными линиями. Нетрудно заметить, что цикл работы занимает пять тактовых периодов, а усредненный подсчет сигналов за цикл с выхода ЦАП (сигнал $U_{ОС}$), (см. рис. 2.6, *в*) дает усредненное число:

$$U_{\text{ср ЦАП}} = (+1-1+1+1+1)/5 = 0,6 .$$

Анализируя приведенные временные диаграммы, можно увидеть, что при увеличении $U_{вх}$ в положительную сторону амплитуда сигнала $U_{и}$ будет расти, что будет «затягивать» время нахождения компаратора и триггера в положительной области; если $U_{вх} = 1,0$ В, триггер всегда будет оставаться в единичном состоянии, и наоборот, уменьшение $U_{вх}$ будет приводить к увеличению доли нахождения триггера в нулевом состоянии, и при $U_{вх} = –1,0$ В он будет оставаться в состоянии «0». Если $U_{вх} = 0$, то триггер «генерирует» меандровые импульсы, а на выходе ЦАП будет нулевое среднее значение.

Таким образом, по выходу $U_{т}$ приведенная схема работает как преобразователь «напряжение – длительность». По идеологии построения АЦП сигнал на выходе триггера, представляющий собой униполярный код, необходимо преобразовать в последовательный или параллельный двоичный позиционный код. Эта задача может быть решена при использовании двоичных цифровых счетчиков с фиксированным циклом преобразования. После окончания цикла счета информация из него передается в регистр-защелку, откуда она считывается, а счетчик обнуляется. Тактирование этого преобразования удобно производить на частоте $f_{т} = Kf_{д}$.

В специальной литературе показывается, что с точки зрения помехоподавляющих свойств в ∑–Δ АЦП вместо счетчиков целе-

сообразно использовать цифровые фильтры с конечной импульсной характеристикой (КИХ-фильтры) реализуемых сигнальными процессорами, входящими в состав ∑–Δ АЦП. Со схемотехникой цифровых КИХ-фильтров можно ознакомиться в специальной литературе. В настоящем учебном пособии КИХ-фильтры будут рассматриваться в разделах посвященных структурной организации сигнальных процессов.

Для более полной оценки преимуществ ∑–Δ АЦП важно вспомнить такие понятия как шум квантования, избыточная дискретизация, формирование формы кривой распределения шума квантования, цифровой фильтрации и децимации.

Реальные преобразования сигналов с помощью АЦП производится с некоторыми искажениями обусловленными механизмами аналогово-цифрового преобразования. Выделяют погрешности смещения нуля преобразователя, погрешности усиления и линейности, дрожание аппаратуры, ограничения полосы, шумы различной природы, включая шумы квантования и т.д..

Одной из наиболее существенных шумовых характеристик любого типа АЦП считается наличие шума квантования вызванного самой природой аналогово-цифрового преобразования. Напомним вкратце механизм возникновения шумов квантования, который иллюстрируются передаточной функцией идеального АЦП с однополярным входным сигналом x(t) (рис. 2.7).

Идеальный N-разрядный АЦП имеет погрешности связанные только с процессами дискретизации и квантования. При нулевом и близком к нулю значении x(t) при переходе через некоторый порог x_{n_1} аналогового напряжения на выходе АЦП формируется код $z = 000,$ по мере роста x(t) на выходе АЦП формируется код $z = 010$ и т.д. Таким образом, между порогами входного напряжения выходной код АЦП остается неизменным, что порождает так называемую неопределённость квантования связанную с ошибкой квантовая. Если принять за идеальную передаточную характеристику прямую проходящую через центры «ступенек» преобразования, где ошибка преобразования равна нулю, то отступление от идеальной линии вверх и вниз следует рассматривать как ошибку (шум) квантования. График мгновенных значений такого шума приведен на рис. 2.8.

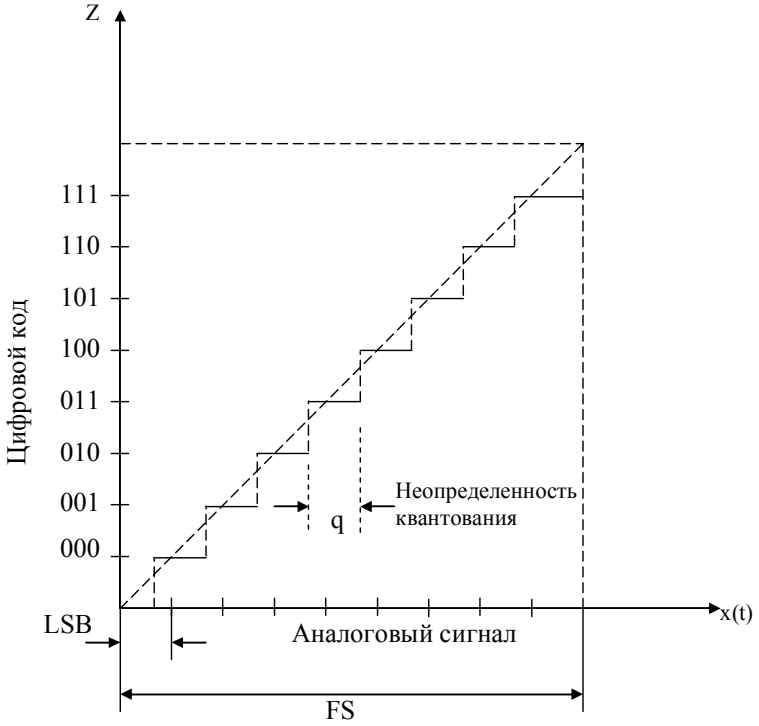


Рис. 2.7. Передаточная функция идеального трехразрядного АЦП

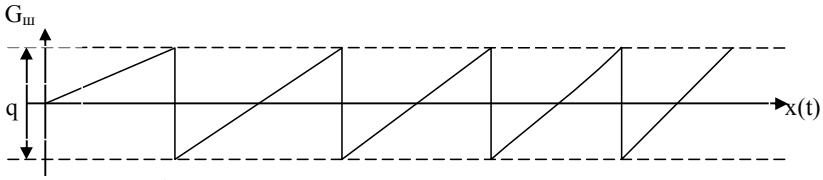


Рис. 2.8. График распределения шума квантования по значениям амплитуд входного сигнала

Максимальная погрешность, которую имеет идеальный АЦП относительно входного сигнала равна ± ½ LBS, где LBS одна из характеристик АЦП определяемая весом младшего разряда преобразователя выражаемая милливольтами. Если, например LBS=10mV, то при росте x(t) от 0 до 10mV на выходе АЦП формиру-

ется код 000, а при превышении x(t) величины 10 mV и при условии, что $x(t)<20mV$ на выходе АЦП будет сформирован код $Z = 001$ и т.д.

На рис. 2.8 величина $q=1LSB$.

Еще одной характеристикой АЦП является значение полной шкалы FS определяемое максимальными значениями входного сигнала x(t) с которым «работает» аналогово-цифровой преобразователь (рис. 2.7).

В работе [43] показано, что среднеквадратичное значение энергии шума квантования (среднеквадратичное значение погрешности) приближается к величине $G_{Ш} = q / \sqrt{12}$, а его частотный состав в значительной степени зависит от спектрального состава x(t).

Установлено, что если полоса частот исследуемого сигнала BW меньше величины 0,5 $f_д$ ($f_a<0,5\ f_д$), то отношение сигнал/шум (SNR) порождаемое работой АЦП определяется выражением:

$$SNR=6,02 \cdot N+1,76 дБ+10\log\left[\frac{f_Д}{2 \cdot BW}\right],$$

где N – разрядность АЦП.

Рассмотрим теперь, как влияет применение избыточной дискретизации используемой в ∑–Δ–АЦП на распределение шума квантования в сравнении с обычным АЦП. Для классического АЦП с частотой дискретизации $f_д$ шум квантования равномерно распределен в пределах полосы Найквиста от 0 до 0,5 $f_д$ со среднеквадратичным значением энергии $G_{Ш} = q / \sqrt{12}$ (q-значения младшего значащего бита, рис. 2.9, *а*).

При этом частотный спектр W(f) полезного сигнала с верхней граничной частотой f_a с учетом теоремы Котельникова (Найквиста) лежит в диапазоне частот $[0,\ldots, 0,5f_Д]$.

На рис. 2.9 ENOB определяет количество разрядов АЦП представляющих полезный сигнал без шумов квантования (эффективное число разрядов).

Если выбрать частоту дискретизации в К раз большую, чем частота, определяемая по Котельникову (Найквисту), то шум квантования будет формироваться с более высокой частотой.

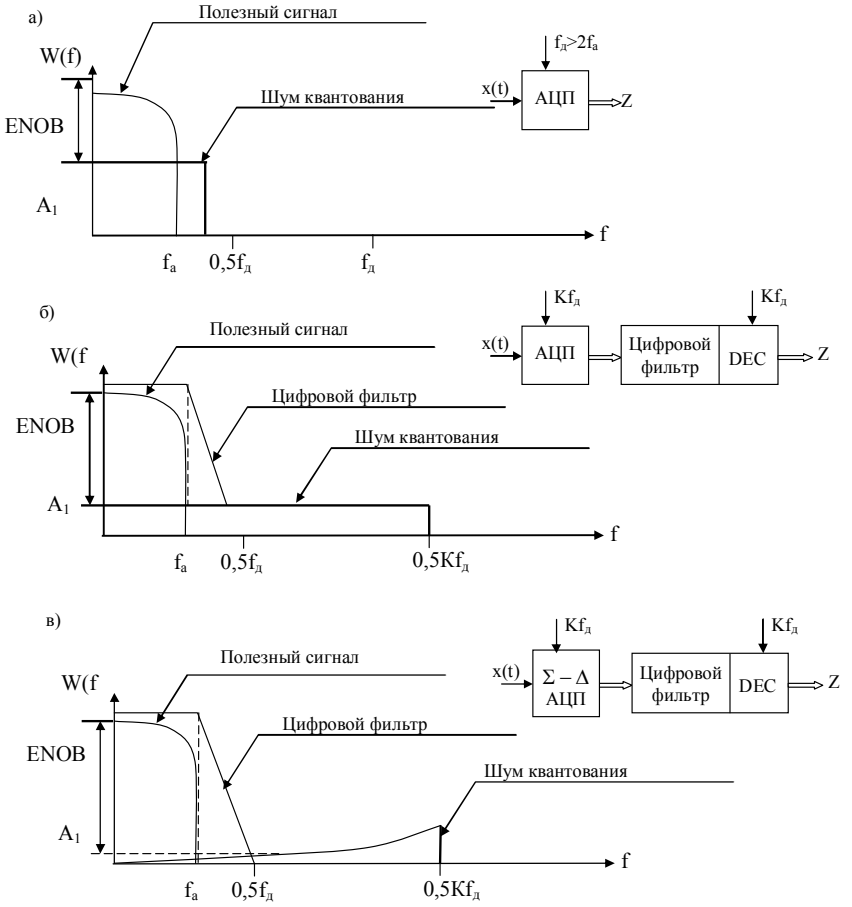


Рис. 2.9. Иллюстрация избыточной дискретизации, цифровой фильтрации и прореживания в АЦП различного типа

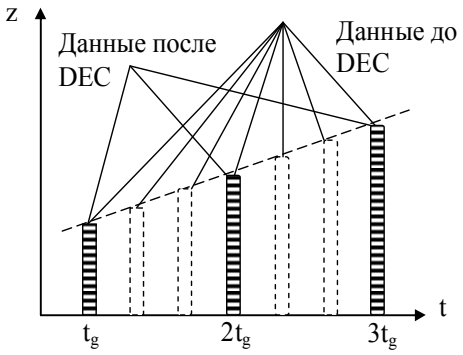
То есть при неизменной энергии шума квантования он займет более широкую полосу от 0 до $0{,}5\,Kf_Д$ и при этом его средняя амплитуда уменьшится (рис. 2.9, *б*).

Если на выходе АЦП, работающего с частотой дискретизации $Kf_Д$, поставить цифровой фильтр низкой частоты, сохраняющий полезный сигнал, то шум квантования будет значительно снижен, а величина ENOB расширена. Таким образом, выполняется аналогово-цифровое преобразование с более высоким разрешением при использовании АЦП без высоких требований к его собственной разрешающей способности. Поскольку цифровой фильтр сужает полосу пропускания до 0,5 $f_Д$ выдача выходных результатов может осуществляться с частотой меньшей чем 0,5 $Kf_Д$ (вплоть до частоты 0,5 $Kf_Д$ без нарушений требований теоремы Котельникова (Найквиста) по восстановлению исследуемого сигнала x(t) с требуемой точностью. Уменьшение частоты дискретизации осуществляется блоком, называемым дециматором (DEC), который из сформированных наборов цифровых кодов выбирает каждый м-й результат, отбрасывая все остальные (рис. 2.10).

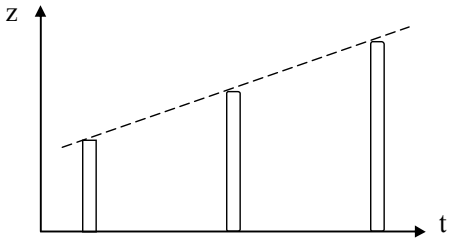
На рис. 2.10,*а* величины исходных цифровых кодов до децимации условно обозначены вертикальными ступенями, а моменты их формирования дециматором – отчетами времени t_g, $2t_g$, $3t_g$,…. В результате децимации коды чисел формируемых АЦП со структурой, приведенной на рис. 2.10, *б*, подаются на выход Z в темпе, показанном на рис. 2.10, *б*.

При использовании ∑–Δ модуляторов в ∑–Δ–АЦП производится не только перевод шумов квантования в более высокочастотную область, но и трансформирование формы кривой распределения шума квантования таким образом, что большая его часть выходит за пределы полосы пропускания цифрового фильтра (рис. 2.9, *в*), что еще больше расширяет зону эффективного числа разрядов ENOB.

Шумоподавляющие свойства ∑–Δ–АЦП удобно исследовать с помощью линейных непрерывных моделей этого класса АЦП так, как это делается в теории управления. С этой точки зрения схема ∑–Δ модулятора первого порядка имеет структуру, представленную на рис. 2.11, *а*.



а)



б)

Рис. 2.10. Временные диаграммы, поясняющие работу цифровых дециматоров

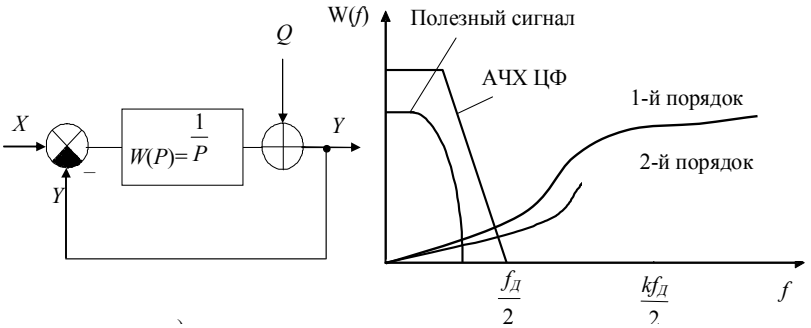


Рис. 2.11. Подавление шума квантования в ∑–Δ–АЦП

Используя преобразование Лапласа, с учетом передаточной функции интегратора для выходного сигнала Y(P) относительно входного сигнала $X(P)$ и шума квантования $Q(P)$ можно записать:

$$Y(P) = \frac{1}{1+P} X(P) + \frac{P}{1+P} Q(P)$$

Учитывая представимость оператора P частотой f, из последнего выражения следует, что с уменьшением частоты влияние шумовой составляющей на выходной сигнал снижается, поскольку $f \to 0$, а входной составляющей увеличивается, поскольку составляющая $1(1+P) \to 1$ при $f \to 0$. В области высоких частот, наоборот, $1/(1+P) \to 0$, а $P/(1+P)$ возрастает (см. рис. 2.10, *б*).

Используя в ∑-Δ-АЦП несколько интеграторов и сумматоров (переходя к схемам более высокого порядка), добиваются большего подавления шумовой составляющей. На рис. 2.12 приведена структура ∑-Δ-АЦП второго порядка, амплитудно-частотная характеристика которого показана на рис. 2.11, *б*.

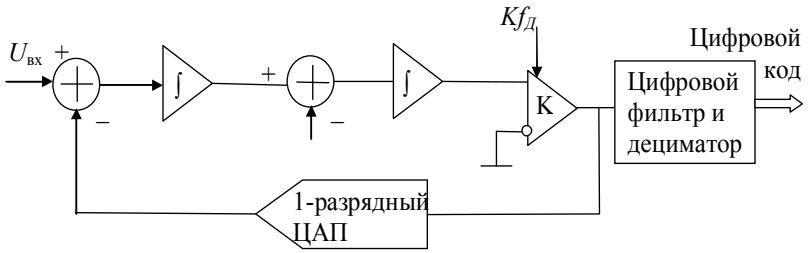


Рис. 2.12. Подавление шума квантования в ∑-Δ-АЦП

Еще более хорошие характеристики по широте динамического диапазона и полосы частот дают ∑-Δ-АЦП более высокого порядка. Цифровые фильтры ∑-Δ-АЦП реализуются по схеме с конечной импульсной характеристикой (КИХ-фильтры) с аппаратной поддержкой операций фильтрации, требующей реализации умножения с накоплением за один машинный цикл. Схемотехника построения таких фильтров специализированными микроконтроллерами описана.

Современные технологии управления ∑-Δ-АЦП позволяют строить схемы с параллельными АЦП и ЦАП, что позволяет полу-

чать более широкий динамический диапазон преобразования (рис. 2.13).

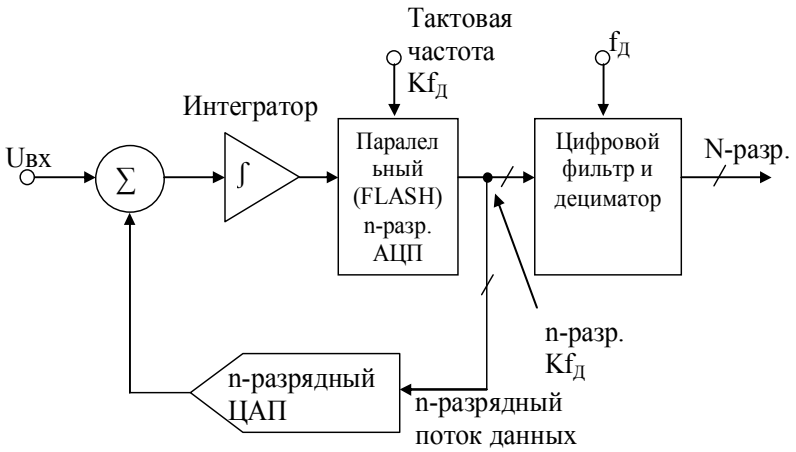


Рис 2.13. Многоразрядный сигма-дельта АЦП

Большинство сигма-дельта-АЦП имеют собственный внутренний цифровой фильтр. Частота среза этого фильтра (и скорость выходного потока данных АЦП) привязана к частоте задающего генератора. В качестве конкретного примера рассмотрим архитектуру рекомендованного для медицинских приложений АЦП AD7725, который представляет собой 16-разрядный сигма-дельта АЦП с программируемым внутренним цифровым фильтром с максимальной частотой дискретизации преобразователя 19,2 МГц (рис. 2.14).

Следующий за преобразователем перестраиваемый фильтр с конечной импульсной характеристикой выполняет прореживание выходных данных преобразователя с коэффициентом децимации 8, снижая скорость выходного потока данных до 2,4 МГц. На выходе перестраиваемого фильтра расположен программируемый цифровой фильтр.

Программное управление фильтром позволяет гибко оперировать длиной фильтра и коэффициентом децимации. Фильтр может иметь до 108 коэффициентов, до 5 режимов прореживания и ко-

эффициенты децимации от 2 до 256. При обработке коэффициентов поддерживается точность 24 разряда, а при арифметических операциях – 30 разрядов. AD7725 содержит процессор постобработки PulseDSP™, компании Systolix, который позволяет запрограммировать характеристики фильтра через параллельный или последовательный интерфейс микропроцессора.

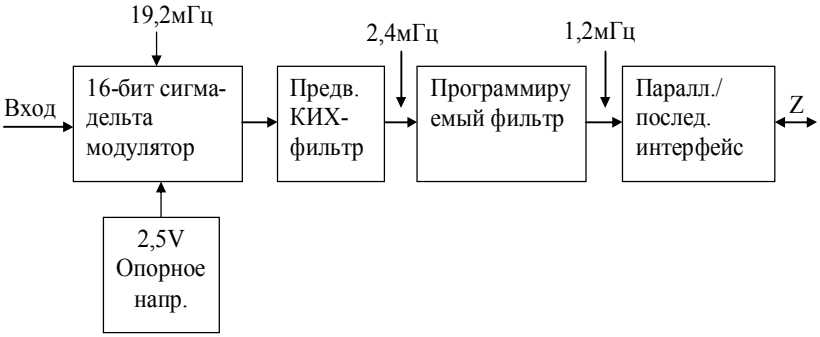


Рис. 2.14. Шестнадцатиразрядный сигма-дельта АЦП с программируемым цифровым фильтром AD7725

Процессор постобработки имеет полностью программируемое ядро, которое обеспечивает производительность обработки до 130 миллионов операций умножения с накоплением в секунду (MAC). Процесс программирования процессора сводится к редактированию пользователем конфигурационного файла, который содержит все необходимые данные для программирования функций фильтра. Этот файл создан с помощью компилятора FilterWizard, который поставляется Analog Devices. Компилятор AD7725 воспринимает значения коэффициентов фильтра как входные данные и автоматически генерирует необходимый программный код устройства.

Файл коэффициентов отклика фильтра может быть сгенерирован с помощью пакетов проектирования цифровых фильтров типа Systolix FilterExpress™ (http://www.systolix.co.uk) или QEDESIGN™ компании Momentum Data Systems (http://www.mds.com). Отклик фильтра может быть построен на основе данных, известных пользователю

до генерации коэффициентов фильтра. Скорость потока входных данных процессора – 2,4 МГЦ. Если прореживание применяется в многоступенчатом фильтре, первый фильтр будет обрабатывать поток данных со скоростью 2,4 мГц, и пользователь может затем производить децимацию между каскадами. Фильтр может иметь характеристики НЧ-фильтра, ВЧ-фильтра, режекторного или полосового фильтра и может быть выполнен как КИХ- или БИХ-фильтр (фильтр с бесконечной импульсной характеристикой).

AD7725 работает от однополярного источника питания +5В. Он имеет внутрикристальный источник опорного напряжения 2,5 В и выпускается в 44-выводном PQFP корпусе. При работе на максимальной тактовой частоте потребляемая мощность не превышает 350 мВт. Возможна работа в режиме сниженной в два раза максимальной частоты задающего генератора – 10 МГц. Максимальная потребляемая мощность в этом режиме составляет 200 мВт.

2.2. Цифровая фильтрация

Характерной особенностью работы биотехнических систем является наличие большой доли помех в регистрируемых сигналах, значительное количество которых «убирается» методами фильтрации, проводимой аналоговыми и цифровыми способами.

Причем, цифровая фильтрация обладает рядом существенных преимуществ перед аналоговой.

Эти преимущества связаны с тем, что устраняются ошибки фильтрации, связанные с изменением параметров пассивных и активных компонентов фильтров, фазовая характеристика линейна, легко реализуется фильтрация с «крутыми» переходными участками. Увеличение сложности фильтра часто реализуется без увеличения затрат и т.д.

Цифровой фильтр в отличие от аналогового оперирует с дискретными во времени данными, а каждый отсчет, соответствующий отклику фильтра, формируется по окончании периода дискретизации.

С учетом дискретной природы обработки данных на формируемые в результате фильтрации отсчеты часто ссылаются по их номерам, например, отсчет1, отсчет2, …, отсчет n, …., отсчет N.

Следует иметь в виду, что для работы в реальном времени DSP процессор данных должен быть рассчитан на выполнение всех шагов по программе фильтрации в пределах промежутка времени, соответствующего одному такту дискретизации ($t_Д = 1/f_Д$), что является одним из существенных ограничений при обработке высокоскоростных сигналов.

Преодоление ограничений DSP-процессоров по быстродействию обеспечивается различными программно-аппаратными способами, реализуемыми универсальными специализированными процессорами. Например, высокопроизводительный, универсальный DSP-процессор с фиксированной точкой типа ADSP-2189M, обладающий быстродействием 75 миллионов операций в секунду (MIPS), способен выполнить операцию умножения с накоплением при реализации одного каскада фильтра за 13,3 нс. DSP-процессор ADSP-2189M затрачивает N+5 инструкций при реализации фильтра с количеством каскадов N. Для 100-каскадного фильтра полное время вычисления составляет приблизительно 1,4 мкс. Это соответствует максимально возможной частоте дискретизации 714 кГц, ограничивая, таким образом, ширину полосы частот обрабатываемого сигнала несколькими сотнями килогерц.

Можно заменить универсальной DSP-процессор специализированным аппаратным цифровым фильтром, способным работать на частотах дискретизации, соответствующих видеосигналу. В другом варианте технической реализации ограничения по быстродействию могут быть преодолены сохранением выборки данных, поступающих с большой скоростью от АЦП, в буферной памяти. Затем буферная память читается со скоростью, совместимой с быстродействием цифрового фильтра, основанного на DSP. Используя данный метод, может осуществляться обработка сигнала в псевдореальном масштабе времени в таких системах, как радар, где обычно обрабатываются пакеты данных, накапливаемые после каждого излучаемого импульса.

Еще один подход заключается в использовании специализированных микросхем цифровых фильтров, подобных фильтрам PulseDSP™ компании Systolix. Сигма-дельта-АЦП AD7725 имеет на своем кристалле фильтр PulseDSP, который может выполнять за секунду 125 миллионов операций умножения с накоплением. В дискретных системах, даже с высокой степенью избыточной дискретизации, требуется наличие аналоговых ФНЧ перед АЦП и после ЦАП для устранения эффекта наложения спектра. Более того, с ростом частоты сигналы выходят за рамки возможностей доступных АЦП, и цифровая фильтрация становится невозможной, На крайне высоких частотах и активная аналоговая фильтрация тоже невозможна из-за ограничений, связанных с полосой пропускания и искажениями ОУ, и в этих случаях требования фильтрации удовлетворяются пассивными элементами.

Что касается биотехнических приложений, то мощности современных DSP контроллеров хватает для решения практически всех прикладных задач.

Для более детального понимания разности аналоговых и цифровых способов фильтрации рассмотрим в сравнении их частотные характеристики для одной и той же частоты среза в 1 кГц (рис. 2.15).

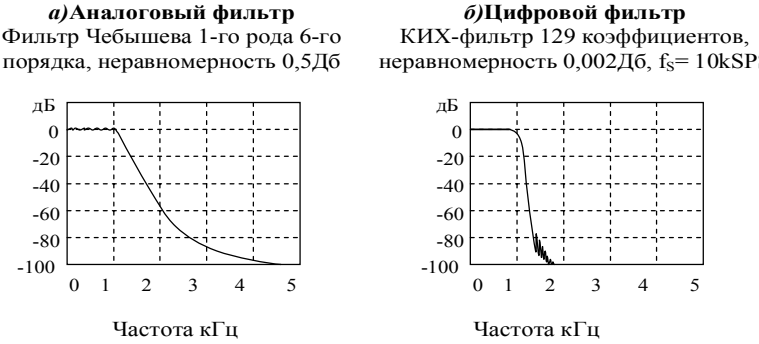


Рис. 2.15. Сравнительные частотные характеристики:
а – аналогового фильтра; *б* – цифрового фильтра

Аналоговый фильтр реализован в виде фильтра Чебышева первого рода 6-го порядка (характеризуется неравномерностью коэф-

фициента передачи в полосе пропускания и равномерностью коэффициента передачи вне полосы пропускания). На практике этот фильтр может быть собран на трех фильтрах второго порядка, каждый из которых состоит из операционного усилителя и нескольких резисторов и конденсаторов. Проектирование фильтра 6-го порядка является непростой задачей, а удовлетворение техническим требованиям по неравномерности характеристики в 0,5 дБ требует точного подбора компонентов.

С другой стороны, представленный цифровой фильтр с конечной импульсной характеристикой (КИХ) имеет неравномерность характеристики всего 0,002 дБ в полосе пропускания, линейную фазовую характеристику и значительно более крутой спад частотной характеристики. Такие показатели аналоговыми фильтрами практически не достижимы.

В практике построения цифровых фильтров известны две их основных разновидности: фильтры с конечной импульсной характеристикой (КИХ) и фильтры с бесконечной импульсной характеристикой (БИХ).

В практике построения биотехнических систем наибольшее применение получили КИХ фильтры. Поэтому на принципах их реализации остановимся более подробно. Наиболее простым КИХ-фильтром является фильтр скользящего среднего, реализующий формулу вида:

$$y(n) = \frac{1}{M} \sum_{k=0}^{M-1} x(n-k),$$

где M – число точек, по которому производится усреднение результатов отсчетов, k – текущий отсчет входного сигнала фильтра; n – текущий отсчет выходного сигнала фильтра.

На рис. 2.16 показан процесс фильтрации сигнала x(t), оцифрованного АЦП по 4 точкам (4-точечный фильтр).

Четырехточечный фильтр реализует формула вида:

$$y(n) = \frac{1}{4} \sum_{k=0}^{3} x(n-k).$$

Естественно, что эта формула не «начнет работать» пока не наберется четыре первых отсчета x(0), x(1), x(2), x(3), чтобы посчитать их первое среднее значение y(t) по формуле:

$$y(3) = 0,25[x(0) + x(1) + x(2) + x(3)].$$

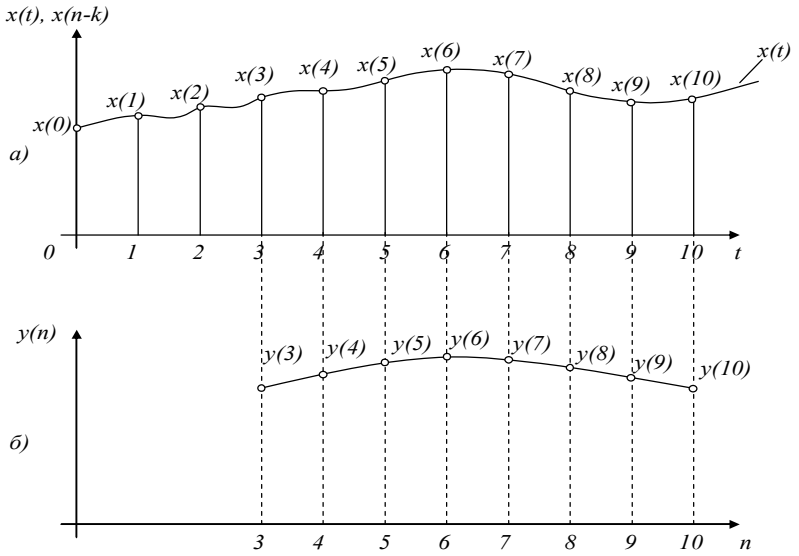


Рис. 2.16. Диаграммы, поясняющие принципы работы четырехточечного фильтра

Второе среднее значение рассчитывается по формуле:

$$y(4) = 0,25[x(1) + x(2) + x(3) + x(4)].$$

Далее расчеты ведутся последовательностью формул:

$$y(5) = 0,25[x(2) + x(3) + x(4) + x(5)];$$

$$y(6) = 0,25[x(3) + x(4) + x(5) + x(6)];$$

$$y(7) = 0,25[x(4) + x(5) + x(6) + x(7)];$$

. . . . . .

Анализируя полученные для вычисления y(n) строки легко заметить, что каждая последующая строка получается из предыдущей путем вычитания левого слагаемого в квадратных скобках и добавления нового текущего отсчета. Таким образом, каждый новый отсчет требует выполнения трех операций: вычитания, сложения и умножения.

С точки зрения технической реализации входные отсчеты фильтров целесообразно хранить в М регистрах (ячейках памяти) и для каждого нового вычисления заменять младший (самый ранний) входной отсчет на вновь поступающий отсчет. Все М отсчетов просуммировать и умножить на 0,25.

Доказано, что из всего многообразия линейных цифровых фильтров фильтр скользящего среднего дает самый низкий уровень шума при заданной крутизне фронта импульса, причем, уровень шума значительно понижается по мере увеличения числа точек, а крутизна частотной характеристики фильтра при этом увеличивается (рис. 2.17). Как видно из этого рисунка в полосе задержания не происходит полного подавления сигнала, что не всегда приемлемо.

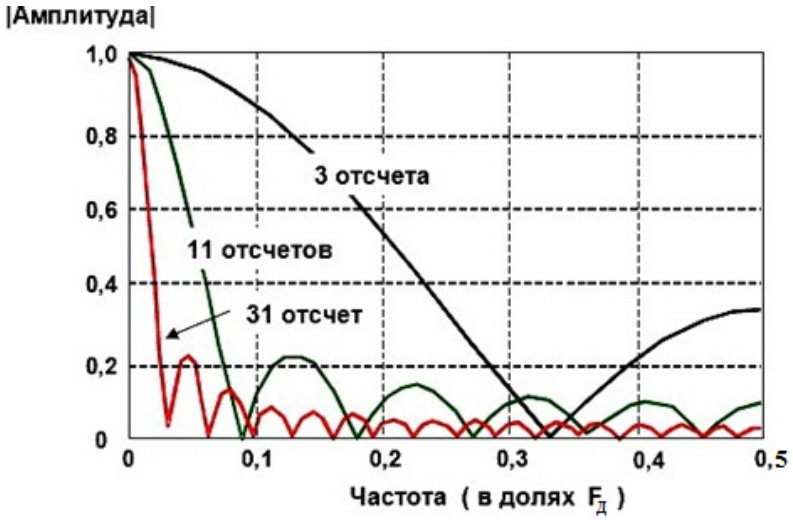


Рис. 2.17. Частотная характеристика фильтра скользящего среднего

Существенного улучшения характеристик КИХ фильтров можно достичь при использовании в них различных весов вместо постоянных множителей. В этом варианте работа обобщенного КИХ фильтра определяется выражением:

$$y(n) = \sum_{k=0}^{M-1} h(k) \cdot x(n-k),$$

где h(k) – массив коэффициентов фильтра; x(n-k) – входной массив данных фильтра.

С учетом выражения (5.4) крутизна спада частотной характеристики увеличивается добавлением большего количества звеньев в фильтр, а характеристики полосы затухания улучшаются выбором надлежащих коэффициентов фильтра. В отличие от фильтра скользящего среднего, для реализации каждой ступени обобщенного КИХ фильтра требуется цикл умножения с накоплением.

При изображении КИХ фильтров различного порядка пользуются их условным изображением (рис. 2.18).

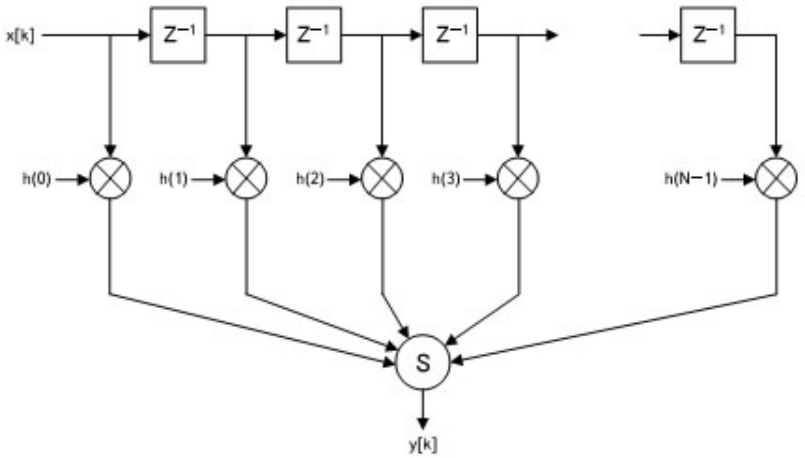


Рис. 2.18. Условное графическое изображение обобщенного КИХ фильтра М-точек

На рис. 2.18 входные отсчеты x(n) пропускаются через ряд регистров памяти (Z^{-1}), выполняющих также роль задержки при Z преобразовании сигналов. Операции умножения на весовые коэффициенты h(k) показаны символом ⊗. Выходной отсчет фильтра y(n) формируется сумматором S.

Часто диаграммы КИХ фильтров изображают более упрощенно (рис. 2.19). На этих диаграммах операции суммирования представляются стрелками, указывающими в точки, а операции умножения обозначают, помещая коэффициенты h(k) рядом со стрелками на линиях. Элемент задержки z показывают, помещая его обозначение выше или рядом с соответствующей линией.

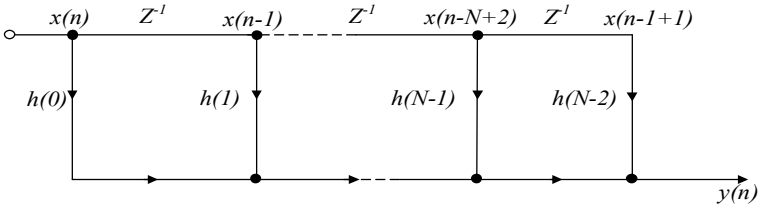


Рис. 2.19. Упрощенная схема фильтра.

При технической реализации обобщенных КИХ фильтров в памяти соответствующих сигнальных процессоров располагаются значения весовых коэффициентов h(k), которые должны перемножаться с потоками отсчетов входного сигнала x(n-k). Причем, операция перемножения h(k) на x(n-k) осуществляется циклически для чего целесообразно выделить в DSP процессоре циклический буфер фиксированного объема в оперативной памяти, работающий по схеме, приведенной на рис. 2.20. В этой схеме показана схема работы четырехточечного КИХ фильтра, в котором суммарный отсчет данных заменяется новым после каждой операции. При этом выборка из 4 (M) последних отсчетов всегда сохраняется в оперативной памяти.

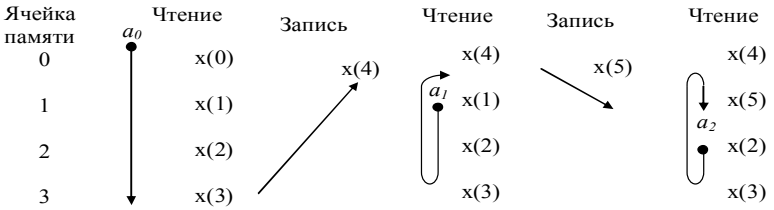


Рис. 2.20. Вычисление выходного сигнала КИХ-фильтра 4-го порядка с использованием циклического буфера

На рис. 2.20: a_0 – адрес начала счета на первом цикле вычислений y(3); a_1 – адрес начала счета на втором цикле вычислений y(4); a_2 – адрес начала счета на третьем цикле вычислений y(5) и т.д.

Чтобы упростить адресацию, чтение из памяти старых значений начинается с адреса, который следует непосредственно за адресом только что записанного нового элемента выборки. Например, если значение x(4) только что записано в ячейку памяти 0, то значения данных читаются из ячеек 1, 2, 3 и 0. Этот пример может быть расширен применительно к любому числу звеньев фильтра. Используя адресацию ячеек памяти таким способом, адресный генератор должен лишь вычислять последовательные адреса, независимо от того, является ли данная операция чтением памяти или записью. Такой буфер в памяти данных называется циклическим, потому что, когда достигается его последняя ячейка, указатель автоматически позиционируется на начало буфера.

Выборка коэффициентов из памяти осуществляется одновременно с выборкой данных, реализуя последовательное вычисление выходных отсчетов фильтра в соответствии с формулами:

$$y(3) = h(0)x(3) + h(1)x(2) + h(2)x(1) + h(3)x(0)\ ;$$

$$y(4) = h(0)x(4) + h(1)x(3) + h(2)x(2) + h(3)x(1)$$

$$y(5) = h(0)x(5) + h(1)x(4) + h(2)x(3) + h(3)x(2)$$

. . . . . .

В соответствии с описанной схемой адресации самый старый отсчет данных выбирается первым. Поэтому сначала должна осуществляться выборка из памяти последнего коэффициента. При использовании адресного генератора, поддерживающего инкрементную адресацию, коэффициенты могут быть сохранены в памяти в обратном порядке: h(N – I) помещается в первую ячейку, а h(0) – в последнюю. И наоборот, коэффициенты могут быть сохранены в порядке возрастания их номеров, если использовать адресный генератор, поддерживающий декрементную адресацию. В примере, показанном на рис. 2.20, коэффициенты сохранены в обратном порядке.

Проектирование КИХ фильтров широко поддерживается современными средствами систем автоматизированного проектирования (САПР). В качестве исходных посылок при проектировании

КИХ-фильтров используют положения о том, что частотная характеристика фильтра определяется импульсной характеристикой и что коэффициенты фильтра определяются его квантованной импульсной характеристикой.

Математический аппарат, применяемый при проектировании фильтров (аналоговых или цифровых), в основном базируется на преобразованиях Фурье. В непрерывных по времени системах в качестве обобщенного преобразования Фурье может рассматриваться преобразование Лапласа. Подобным способом можно обобщить преобразование Фурье для дискретных по времени систем, и результат такого обобщения известен как z-преобразование.

Существует достаточно много программ, использование которых позволяет рассчитать параметры КИХ-фильтров для желаемых пользователем характеристик.

В качестве примера рассмотрим возможности программы Паркса-Маккилана, позволяющей программировать низкочастотные, высокочастотные и режекторные КИХ-фильтры с использованием алгоритма Ремеза. Проектирование фильтра начинается с определения его параметров, например, по желаемой картинке, представленной на рис. 2.21.

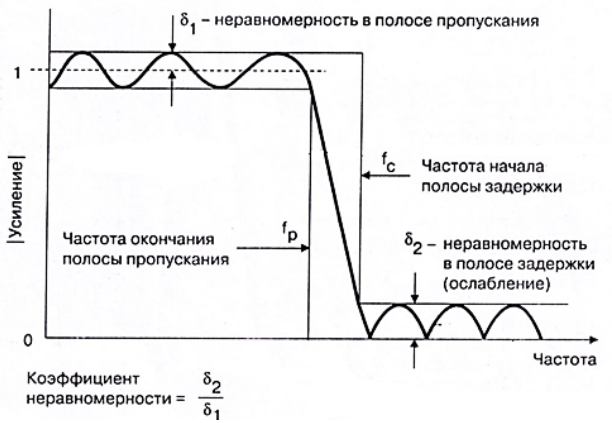


Рис. 2.21. Исходная желаемая частотная характеристика ФНЧ

Пусть требуется спроектировать ФНЧ, для которого определена частота дискретизации в 44,1 кГц, желаемые характеристики выбраны следующим образом: граничная частота пропусскания – 18 кГц; полоса пропускания начинается с 21 кГц; неравномерность полосы пропускания 0,01 дБ; неравномерность полосы задержки 96 дБ.

Кроме этого по условию работы программы необходимо определить длину слова (разрядность) коэффициентов, которая ориентируется на используемый DSP процессор. Например, может быть выбран 16-разрядный процессор с фиксированной точкой.

В ходе расчетов программа рассчитывает количество звеньев (для рис. 2.21. – это 69 звеньев), определяет квантованные коэффициенты фильтров, реакции на импульсное воздействие, выдает данные анализа в S и Z плоскостях и импульсную характеристику. Кроме того, генерируются ассемблерные программы для загрузки фильтров в некоторые популярные DSP-процессоры, включая процессоры Analog Devices.

2.3. Аппаратная реализация DSP процессоров

С учетом специфики задач, решаемых DSP-процессорами в работе [50] для них определена наиболее важная операция и сформирован ряд основных требований (табл. 2.1). Рассмотрим более подробно основные требования предъявляемые к DSP.

Как отмечено ранее основное требование к быстродействию определяется скоростью выполнения операции умножения с накоплением (MPC), однако нельзя забывать и о необходимости скоростного выполнения и других арифметических и логических операций, которые аппаратно-поддерживаются арифметико-логическими устройствами (АЛУ).

Выполнение требований по высокой точности представления результатов обусловлено рядом причин. Например, когда перемножаются два 16-битных слова, результат представляется 32-битным словом. Ядро процессоров компании Analog Devices семейства ADSP-21xx с фиксированной точкой имеет встроенный

40-битный аккумулятор, который обеспечивает большой запас суммирования без переполнения. Хотя использование DSP с плавающей точкой автоматически устраняет большинство проблем, связанных с точностью и переполнением, процессоры с фиксированной точкой остаются очень популярными для многих приложений, и поэтому при их использовании нужно обращать достаточное внимание на возможное переполнение, потерю результатов (выход результата операции за пределы разрядной сетки) и масштабирование операндов.

Таблица 2.1

Наиболее важная операция и основные требования к DSP

№ п/п	Операции, требования	Формулы, содержание
1	Цифровая фильтрация	• Базовая формула: $y(n) = h(0) \cdot x(n) + h(1) \cdot x(n-1) + \ldots + h(N-1) \cdot x(n-N)$ • Многократное умножение значений входных отсчетов на коэффициенты фильтра (или на поворотные множители при БПФ) • Накопление результатов умножения в регистре-аккумуляторе • Повторение этих действий N раз
2	Требования, предъявляемые к DSP	• Быстрое выполнения умножения с накоплением • Высокая точность представления результата (в аккумуляторе) • Одновременная выборка двух операндов • Наличие циклических буферов • Реализация циклов с автоматической проверкой условий
3	За один цикл ядро DSP типа ADSP-21xx осуществляет	• Выборку значения отсчета из памяти данных • Выборку значения коэффициента из памяти программ • Умножение с накоплением

Для быстрой и непрерывной обработки данных блоком MAC целесообразно одновременно выбирать два операнда. Такая выборка двух операндов в DSP осуществляется по двум независимым шинам: шине данных памяти программ и шине данных памяти

данных. Кроме того, имеются отдельные шина адреса памяти программ и шина адреса памяти данных. Таким образом, МАС может получать входные данные по каждой шине данных одновременно. Такая архитектура обычно называется гарвардской.

В предыдущем разделе при описании вычислительной процедуры реализации КИХ фильтра было показано, что при одновременной выборке данных и коэффициентов организуется циклический вычислительный процесс с задержкой на выбор новых пар операндов.

Использование циклических буферов является специфическим для цифровой обработки сигналов и для достижения максимальной эффективности циклические буферы должны поддерживаться аппаратно. Аппаратная реализация циклических буферов позволяет установить параметры буфера (такие как адрес начала буфера, длина и т.д.) в программе вне тела цикла, непосредственно вычисляющего алгоритм. Это позволяет избежать включения дополнительных команд в тело цикла. Отсутствие аппаратной реализации циклических буферов может существенным образом ухудшить возможности DSP-процессора по реализации алгоритмов цифровой обработки сигналов.

Необходимость поддержки циклов с автоматической проверкой условий завершения вызвана циклическим характером алгоритмов ЦОС. Функция умножения с накоплением и выборка данных повторяются N раз при каждом вычислении типового алгоритма. В традиционных микропроцессорах организация цикла предполагает наличие в заголовке цикла команд для проверки условия окончания цикла. Архитектура DSP-процессоров компании Analog Devices обеспечивает аппаратную поддержку программных циклов без необходимости программной проверки условия продолжения или завершения в теле цикла.

Для типичной DSP-архитектуры различие в производительности при аппаратной поддержке цикла с автоматической проверкой условия завершения и при программной проверке условия завершения цикла может превышать 20% времени выполнения цикла.

Перечисленные выше требования поддерживаются в DSP компании Analog Devices, сигнальные процессоры которой выполняют

все операции, связанные с реализацией звена КИХ-фильтра за один машинный цикл. Это достигается тем, что данные и код программы хранятся в различных блоках памяти и доступ к ним осуществляется через разные шины «работающие» одновременно.

В качестве примера на рис. 2.22 приведена архитектура ядра процессора с фиксированной точкой компании Analog Devices типа ADSP-21xx. Архитектура этого процессора оптимизирована для цифровой обработки сигнала и для других высокоскоростных приложений.

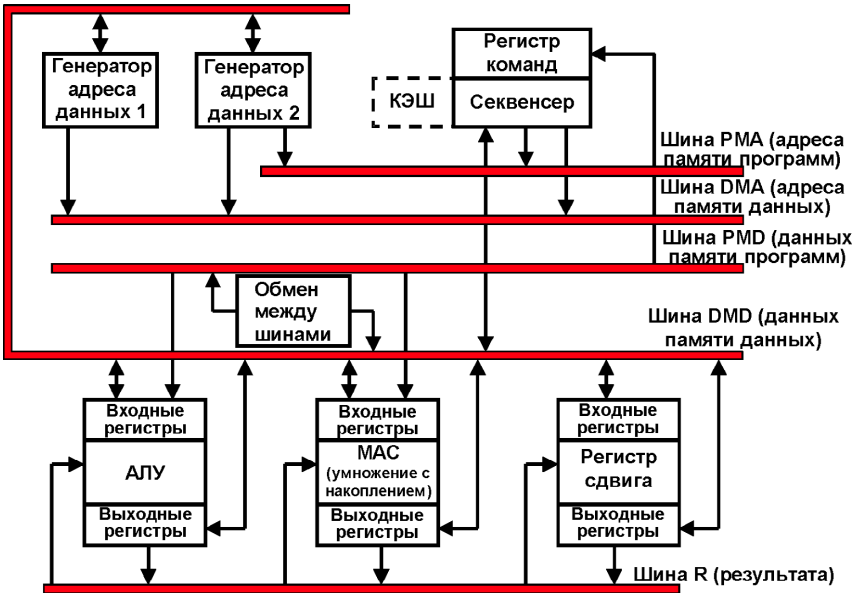


Рис. 2.22. Архитектура ядра процессора ADSP-21xx

Данное семейство DSP включает ядро семейства ADSP-2100 (три вычислительных блока, адресные генераторы, устройство управления выполнением команд), два синхронных последовательных порта, программируемый таймер, развитую систему обработки прерываний, встроенную статическую память программ и данных. Существуют также версии с встроенным ПЗУ.

Гибкая архитектура процессоров семейства ADSP-21xx и соответствующий набор команд позволяют достичь высочайшей степени параллелизма при выполнении операций. За один цикл DSP семейства ADSP-21xx может сгенерировать адрес следующей команды программы, выбрать следующую команду, выполнить 1 или 2 перемещения данных, обновить 1 или 2 указателя адреса, выполнить вычислительную операцию, передать или принять данные от двух последовательных портов и обновить регистр таймера.

Процессоры семейства ADSP-21xx имеют пять внутренних шин для повышения эффективности передачи данных. Шины адреса памяти программы (PMA) и адреса памяти данных (DMA) используются одновременно для адресации в пределах адресных пространств памяти программ и памяти данных. Шины данных памяти программ (PMD) и шина данных памяти данных (DMD) используются для передачи данных из соответствующих областей памяти. При выводе шин на корпус за пределы кристалла они объединяются в одну внешнюю шину адреса и в одну внешнюю шину данных. Области памяти выбираются соответствующими сигналами управления. Шина результата (R) используется для пересылки промежуточных результатов напрямую между различными вычислительными блоками. 14-разрядная шина PMA позволяет получить прямой доступ к 16К слов памяти программ. Шина DMD является 16-разрядной. Она обеспечивает доступ к содержимому любого регистра процессора для передачи его в любой другой регистр или любую ячейку памяти данных за один цикл. Адрес памяти данных может поступать от двух источников: абсолютное значение, содержится в коде команды (прямая адресация) или на выходе адресного генератора (косвенная адресация). Для выборки данных из памяти программ используется только косвенная адресация. Шина данных памяти программы (PMD) может также использоваться для передачи данных между вычислительными блоками напрямую или через блок обмена между шинами PMD-DMD. Блок обмена между шинами PMD-DMD позволяет передавать данные от одной шины к другой. Он содержит технические средства, которые позволяют, когда это необходимо, преодолевать разницу в 8 бит между двумя шинами.

В памяти программ могут храниться как команды, так и данные, позволяя DSP семейства ADSP-21xx одновременно осуществлять выборку двух операндов в одном цикле, один из памяти программы, а другой из памяти данных. Команды поступают либо прямо из памяти программ путем осуществления двойного доступа в одном машинном цикле (процессоры серии ADSP-218x), либо из кэш-памяти программ (в процессоре серии ADSP-219x и SHARC).

Процессор содержит три независимых вычислительных блока: арифметико-логическое устройство (АЛУ); умножитель с накоплением (MAC) и устройство сдвига. Вычислительные блоки способны обрабатывать 16-разрядные данные и могут поддерживать вычисления с повышенной точностью. АЛУ обладает флагом переноса CI, который позволяет поддерживать 32-разрядные арифметические действия. АЛУ обеспечивает стандартный набор арифметических и логических функций: сложение, вычитание, смену арифметического знака, инкремент, декремент, получение абсолютного значения, логическое И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ и инверсию. Также поддерживаются примитивы деления.

MAC способен выполнить за один машинный цикл операцию умножения, умножения/сложения или умножения/вычитания. Он также содержит 40-разрядный аккумулятор, который обеспечивает дополнительные 8 разрядов для накопления результата без потери информации. Данное решение допускает возникновение 256-ти переполнений, прежде чем произойдет потеря данных. Специальные команды обеспечивают поддержку блочной плавающей точки, при которой одна экспонента приписывается некоторому набору данных.

Для ускорения обработки прерываний в MAC также может использоваться набор вспомогательных регистров. Если после окончания процедуры обработки сигнала установлен флаг MV, это означает, что регистр результата содержит слово, длина которого больше 32 разрядов. В этом случае значение, содержащееся в регистре, может быть заменено максимально или минимально возможной величиной, представимой в пределах 32-разрядной сетки в формате 1.32, в зависимости от природы переполнения.

Устройство сдвига производит операции логического и арифметического сдвига, нормализации и денормализации, а также вычисления блочной экспоненты. Устройство может быть использовано для эффективного управления численными форматами, включая представление чисел в формате с плавающей точкой с повышенной точностью.

Вычислительные блоки располагаются параллельно, а не последовательно, так что результат работы любого блока может использоваться как исходное данное для другого блока в следующем машинном цикле. Для обеспечения такой возможности используется шина промежуточных результатов (R).

Два специализированных адресных генератора и устройство управления последовательностью выполнения команд (секвенсер) делают использование вычислительных блоков еще более эффективным. Адресные генераторы (DAG) вычисляют адреса при перемещении данных из памяти в регистры и обратно.

Каждый генератор DAG обладает четырьмя регистрами-указателями. Всякий раз, когда указатель используется для адресации данных (косвенная адресация), он модифицируется значением, содержащимся в специализированном регистре-модификаторе. Для реализации автоматической циклической буферизации каждому регистру-указателю приписывается регистр, хранящий длину циклического буфера. При использовании двух независимых адресных генераторов DAG процессор может генерировать одновременно два адреса для обеспечения одновременной выборки двух операндов.

DAG1 может осуществлять адресацию только в памяти данных. DAG2 может осуществлять адресацию и в памяти данных, и в памяти программ. Когда в конфигурационном регистре (MSTAT) установлен соответствующий управляющий бит, адресный генератор DAG1 осуществляет бит-реверсивную адресацию. Бит-реверсивная адресация существенно упрощает реализацию алгоритма БПФ по основанию 2.

Программный секвенсер осуществляет вычисление адреса инструкции, выборка которой должна осуществляться по мере исполнения программы. Ключевым компонентом устройства являет-

ся регистр команд, который хранит информацию о выполняемой в текущее время команде. Регистр команд образует одноуровневый конвейер в потоке обрабатываемых команд. Команды выбираются и загружаются в регистр команд в одном цикле шины процессора и выполняются в следующем цикле, в то время как осуществляется выборка следующей команды. Чтобы уменьшить количество дополнительных циклов, устройство поддерживает выполнение переходов по условию, вызовов подпрограмм и возвращений к выполнению главной программы за один машинный цикл. Используя внутренний счетчик цикла и стек цикла процессор может выполнять программу цикла с автоматической проверкой условия завершения, дополнительных затрат процессорного времени на организацию цикла.

Таким образом, чтобы организовать цикл, команды явного перехода не требуются. Устройство также способно с минимальной задержкой реагировать на прерывания, поступающие от контроллера прерываний. Появление прерывания вызывает переход к определенной ячейке памяти, где хранится подпрограмма обработки прерывания. Короткая подпрограмма обработки прерывания может располагаться непосредственно в таблице векторов прерываний, где для этих целей зарезервировано четыре ячейки памяти.

Для выполнения более сложной подпрограммы обработки прерывания приходится осуществлять переход с помощью команды JUMP в область памяти программ, где может быть размещена более длинная программа.

Для организации гибкого взаимодействия с внешними устройствами ядро процессоров семейства ADSP-21xx дополняется встроенными периферийными устройствами, которые имеют различную конфигурацию и разные возможности в зависимости от конкретного исполнения представителя семейства. В качестве примера на рис. 2.23 приведена архитектура процессора семейства ADSP-218x.

Семейство 21xx имеет множество различных вариантов процессоров с разным объемом встроенной памяти; в более новом семействе 218x имеются модели, включающие до 48K слов памяти программ и 56K слов памяти данных. Все DSP процессоры семейства

используют модифицированную гарвардскую архитектуру, которая предполагает раздельные адресные пространства памяти программы и памяти данных и позволяет хранить данные в памяти программы. Интерфейс внешней памяти поддерживает как быструю, так и медленную память с программируемыми состояниями ожидания.

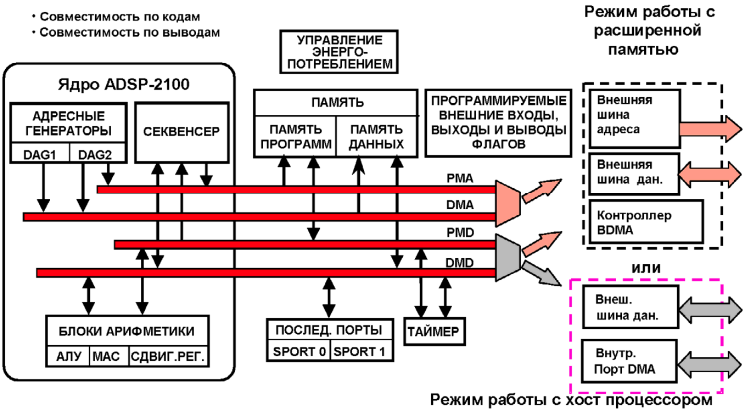


Рис. 2.23. Архитектура процессоров семейства ADSP-218x

Процессоры семейства ADSP-218x также поддерживают отдельное адресное пространство портов ввода-вывода. Все представители семейства 21xx (кроме сигнальных процессоров ADSP-2105) имеют два последовательных порта с двойной буферизацией (SPORT) для приема и передачи последовательных данных. Каждый SPORT является двунаправленным и имеет свой собственный программируемый генератор битовой и фреймовой синхронизации. Длина слова SPORT может изменяться от 3 до 16 разрядов. Данные могут передаваться с использованием фреймовой синхронизации или без нее. Каждый SPORT способен генерировать прерывания и поддерживает логарифмическое сжатие данных.

IDMA-порт процессоров семейства ADSP-218xx поддерживет возможность начальной загрузки процессора от хост-компьютера и возможность доступа со стороны хост-компьютера во внутрен-

нюю память DSP для чтения и записи «на лету», когда DSP занимается выполнением своей программы. Порт IDMA позволяет главному процессору осуществлять доступ ко всей внутренней памяти DSP без использования почтовых регистров. Порт IDMA поддерживает передачу 16- и 24-разрядных слов, при этом передача 24-разрядных слов происходит за два машинных цикла.

Процессоры семейства ADSP-218xx обладают также интерфейсом для взаимодействия с памятью, которая имеет байтовую организацию. Данный интерфейс может использоваться для начальной загрузки процессора и для передачи данных из внутренней памяти и во внутреннюю память «на лету». Максимальный размер адресуемой внешней восьмибитовой памяти составляет 4MB. Данное адресное пространство играет роль загрузочной области (boot memory), характерной для представителей семейства 21xx. Байтовая память имеет организацию 256 страниц по 16Kx8 бит. Передача данных в байтовую память и из нее может осуществляться с упаковкой или распаковкой 24-разрядного, 16-разрядного и 8-разрядного (с выравниванием по младшему или старшему байту) форматов. При доступе к внутренней памяти DSP контроллер DMA занимает один машинный цикл, во время которого ядро не может осуществлять доступ в память.

Процессоры ADSP-21xx, ADSP-218x и ADSP-21msp5x имеют специальный режим работы с низкой потребляемой мощностью, который позволяет достигнуть потребления меньше 1 мВт. Переход в данный режим может осуществляться аппаратно или программно. Это особенно важно для устройств, работающих от автономных источников питания. В некоторых режимах работы с низкой потребляемой мощностью отключается внутренний тактовый сигнал, но содержимое памяти и регистров при этом сохраняется.

Краткое описание архитектуры сигнальных процессоров позволяет оценить их возможности. Для использования их в биотехнических системах естественно нужно более детально ознакомиться как с их конструкцией, так и с системой команд, которые можно найти в соответствующей литературе и технических описаниях.

Важной составляющей проектирования систем ЦОС является наличие набора средств аппаратной и программной отладки DSP процессоров.

Для более полного понимания процесса проектирования ЦОС на основе DSP процессоров рассмотрим пример разработки такой системы.

Первый шаг в процессе проектирования – это описание архитектуры системы, которое включает такую информацию, как тип процессора, периферийные устройства (внешняя память, кодеки, хост-процессор, каналы связи), конфигурацию и т.д. Эта информация помещается в файл, называемый LDF (файл описания связей).

Следующий шаг в процессе проектирования – генерирование необходимой программы DSP. Программа может создаваться на языке высокого уровня (обычно C или C++), на языке ассемблера для DSP или с использованием обоих языков. Программа DSP, разработанная на C, должна быть откомпилирована для получения кода на языке ассемблера. Нужно учитывать, что преимуществом использования языка C является простота программирования, в то время как результат компилирования такой программы не так эффективен, как при программировании непосредственно на ассемблере. По этой причине многие программисты DSP программируют на C, но используют ассемблер для проектирования таких моментов в программе, которые требуют наибольшего быстродействия. Язык ассемблера компании Analog Devices для DSP использует алгебраический синтаксис и достаточно прост при непосредственном использовании. В конце этапа компиляции компоновщик генерирует исполняемый файл.

Затем программное обеспечение должно быть отлажено с использованием программного симулятора в сочетании с некоторой аппаратной отладочной системой, такой как, например, плата EZ-ICE или другая плата, встраиваемая в слот персонального компьютера и поставляемая какой-либо фирмой. После того, как программа отлажена с использованием отладочной платы, она должна быть протестирована в разрабатываемой системе (на проектируемой плате с DSP).

Внутрисхемный эмулятор, такой как EZ-ICE, обычно взаимодействует с проектируемой платой через интерфейс PCI или JTAG-интерфейс через соответствующий кабель.

Конечный этап в процессе создания программы требует генерации загрузочного модуля с использованием программы PROM Splitter.

Новые программные средства разработки компании Analog Devices для DSP – пакеты VisualDSP® и VisualDSP ++ – являются весьма простыми в использовании инструментами, работающими под управлением Windows 95|98|NT. Пакет VisualDSP является интегрированной программной средой, включающей все необходимые для отладки средства, поддерживающий ассемблер с алгебраическим синтаксисом и оптимизированный С-компилятор. Мультипроцессорное окружение также может быть смоделировано и отлажено. VisualDSP++ обеспечивает поддержку языка С++.

2.4. Взаимодействие DSP процессоров с внешними устройствами

В базовой схеме цифровой обработки сигналов (рис. 2.24) DSP процессоры взаимодействуют с АЦП и ЦАП, которые являются внешними устройствами по отношению к сигнальным процессорам. Современные DSPпроцессоры могут иметь интегрированные АЦП и ЦАП (например, ADSP-21ESP202). В таком варианте проблема организации интерфейса между определенными компонентами ЦОС не существует. Имеются технические решения, в которых ЦАП и АЦП ориентированы на взаимодействие и синхронизацию, но все таки требуют понимания процессов обмена между отдельными компонентами ЦОС.

Рассмотрим вначале пример высокоскоростного параллельного обмена между DSP семейства ADSP21xx и АЦП с параллельным цифровым выходом (рис. 2.24).

Схема на рис. 2.24 несколько упрощена, на ней показаны только сигналы, используемые для чтения данных из периферийных устройств, в качестве которого выступает АЦП с параллельным выходом, имеющий выходную шину данных с переводом в высокоомное состояние при отсутствии обмена.

Временная диаграмма цикла чтения для процессоров семейства ADSP-21XX показана на рис. 2.25.

В этом примере подразумевается, что АЦП производит выборку с постоянной частотой, которая задается внешним тактовым генератором (генератор импульсов запуска), ВТГ асинхронно по отношению к внутренней тактовой синхронизации DSP-

процессора. Использование отдельного задающего генератора для АЦП является предпочтительным, поскольку сигнал внутреннего генератора DSP-процессора может иметь высокий уровень помех и фазовый шум (jitter), который в процессе аналого-цифрового преобразования приведет к увеличению уровня шумов АЦП.

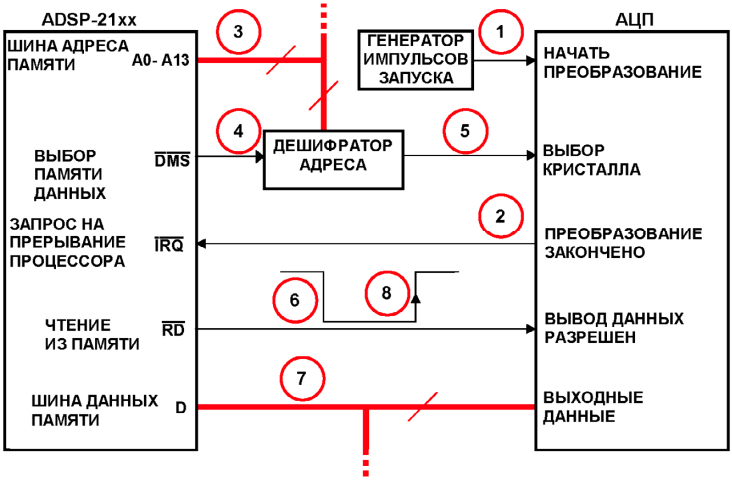


Рис. 2.24. Подключение АЦП к процессорам семейства ADSP-21xx через параллельный интерфейс

Тактовый импульс задающего генератора на входе «начать преобразование» (convert start) АЦП инициирует процесс преобразования входных данных (шаг № 1). По переднему фронту этого импульса внутренняя схема выборки-хранения АЦП переключается из режима выборки в режим хранения и таким образом начинается процесс преобразования.

После выполнения преобразования на выходе АЦП выставляется строб преобразование закончено (шаг № 2). Когда этот сигнал поступает на вход запроса прерывания DSP-процессора ($\overline{\text{IRQ}}$), начинается процесс чтения данных из АЦП, в начале которого процессор выставляет на шине адрес периферийного устройства, инициировавшего запрос на прерывание (шаг № 3). В то же самое

время процессор переводит в активное состояние сигнал доступа к памяти ($\overline{\text{DMS}}$) (шаг № 4). Две внутренние шины адреса в процессоре ADSP-21XX (шина адреса памяти программ и шина адреса памяти данных) совместно используют внешнюю шину адреса, а две внутренние шины данных (шина данных памяти программ и шина данных памяти данных) совместно используют одну внешнюю шину данных. Сигналы выбора памяти начальной загрузки ($\overline{\text{BMS}}$), выбора памяти данных ($\overline{\text{DMS}}$), выбора памяти программ ($\overline{\text{PMS}}$) и выбора памяти устройств ввода-вывода ($\overline{\text{IOMS}}$) указывают, для какой памяти в данный момент используются внешние шины. Эти сигналы обычно используются для разрешения внешней дешифрации адреса, как показано на рис. 2.24. Выходной сигнал дешифратора адреса подается на вход chip select выбора периферийного устройства (выбор кристалла) (шаг № 5).

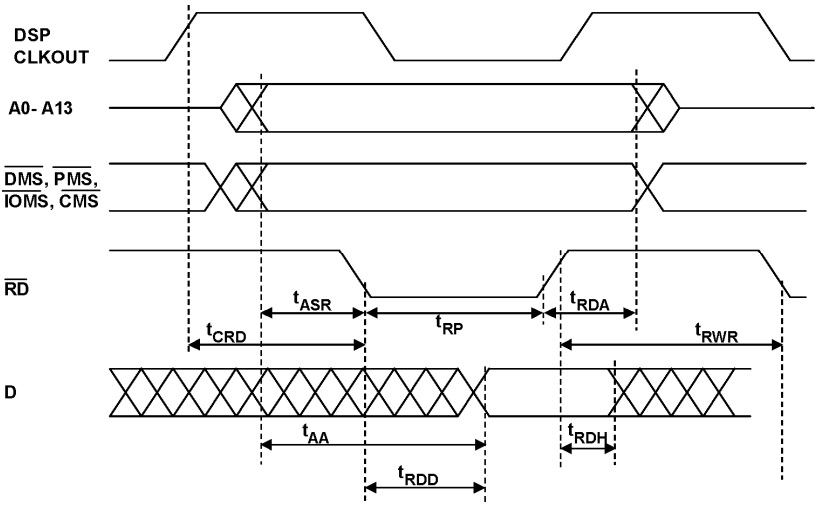


Рис. 2.25. Временная диаграмма чтения из памяти в процессорах семейства ADSP-21xx

Сигнал чтения памяти (memory read, $\overline{\text{RD}}$) выставляется через промежуток времени t_{ASR} после активации сигнала $\overline{\text{DMS}}$ (шаг № 6). Чтобы полностью использовать преимущество высокой скорости DSP-процессора, сумма времени задержки дешифрации ад-

реса и времени включения периферийного устройства после подачи сигнала выбора (chip select) не должна превышать время t_{ASR}. Сигнал чтения памяти (memory read, $\overline{RD}$) остается активным (низкий логический уровень) в течение времени t_{RP}. Этот сигнал используется для перевода в активное состояние параллельного выхода данных периферийного устройства (шаг № 7). Сигнал $\overline{RD}$ обычно подключается к соответствующему выводу периферийного устройства, называемому сигналом разрешения выхода или чтения (output enable или read). Восходящий (задний) фронт сигнала $\overline{RD}$ используется для ввода данных с шины в DSP-процессор (шаг № 8). После появления восходящего (заднего) фронта сигнала $\overline{RD}$ данные на шине должны удерживаться периферийным устройством в течение времени t_{RDH}, называемого временем удержания данных. Для большинства процессоров семейства ADSP-21XX это время равно нулю.

Следует отметить, что если используются скоростные DSP процессоры, то время доступа к АЦП или другому периферийному устройству получается больше, чем время работы сигнального процессора. В этом случае необходимо использовать созданные циклы ожидания или снижать тактовую частоту работы процессора. Например, процессор ADSP-2189M способен эффективно взаимодействовать с медленными периферийными устройствами при помощи имеющихся средств программирования длительности состояния ожидания. Имеется три специальных регистра для управления процессом ожидания: для памяти начальной загрузки, для памяти программ и для памяти данных и пространства ввода-вывода. Программист может задать от 0 до 15 тактов ожидания для каждого параллельного интерфейса памяти. Каждый такт ожидания увеличивает время доступа к внешней памяти на величину, равную по длительности одному такту генератора тактовых импульсов процессора (13.3 нс для процессора ADSP-2189M, работающего на тактовой частоте 75 МГц). В рассматриваемом примере сигналы адрес памяти данных, DMS и RD удерживаются неизменными в течение дополнительного времени, определяемого продолжительностью тактов ожидания. В качестве второго примера

рассмотрим организацию интерфейса семейства ADSP-218x с микросхемой, содержащей в своем составе $\sum\!-\!\Delta$ ЦАП и АЦП. Такие комбинированные устройства принято называть кодеками и аналоговыми оконченными устройствами (AFE). В рассматриваемом примере в качестве AFE используется микросхема AD73322.

Функциональная схема микросхемы AD73322 показана на рис. 2.26. Данный прибор представляет собой двойной AFE с двумя 16-разрядными АЦП и двумя 16-разрядными ЦАП с возможностью работы с частотой дискретизации 64 кГц. ИС AD73322 разработана для универсального применения, включая обработку речи и телефонию с использованием сигма-дельта АЦП и сигма-дельта ЦАП. Каждый канал обеспечивает отношение сигнал/шум на уровне 77 дБ в пределах голосовой полосы частот. Каналы АЦП и ЦАП имеют программируемые коэффициенты усиления по входу и выходу с диапазонами до 38 дБ и 21 дБ соответственно. Встроенный источник опорного напряжения допускает использование однополярного питающего напряжения величиной +2.7–5.5 В. Его потребляемая мощность при напряжении питания +3 В составляет 73 мВт.

Частота дискретизации кодека может быть запрограммирована на одно из четырех фиксированных значений: 64 кГц, 32 кГц, 16 кГц и 8 кГц при частоте опорного задающего генератора 16.384 МГц. Последовательный порт позволяет легко организовать интерфейс одного или нескольких кодеков, включенных каскадно, со стандартными DSP-процессорами.

Скорость передачи данных по последовательному порту может программироваться, что позволяет осуществлять интерфейс как с быстрыми, так и с медленными DSP-процессорами. Организация интерфейса кодека с процессором семейством ADSP-218X показана на рис. 2.27. Вывод SE (включение последовательного порта) может управляться от программируемого выхода общего назначения, например FL1, или, в случаях, когда не требуется перевод последовательного порта в энергосберегающий режим, на этот вывод может постоянно подаваться высокий уровень через подходящий нагрузочный резистор.

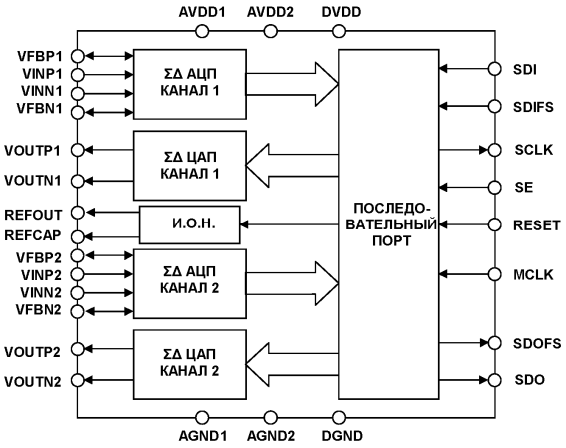


Рис. 2.26. Структурная схема ИС AD73322 – 16-разрядного кодека с частотой дискретизации 64 кГц с однополярным питанием и последовательным интерфейсом

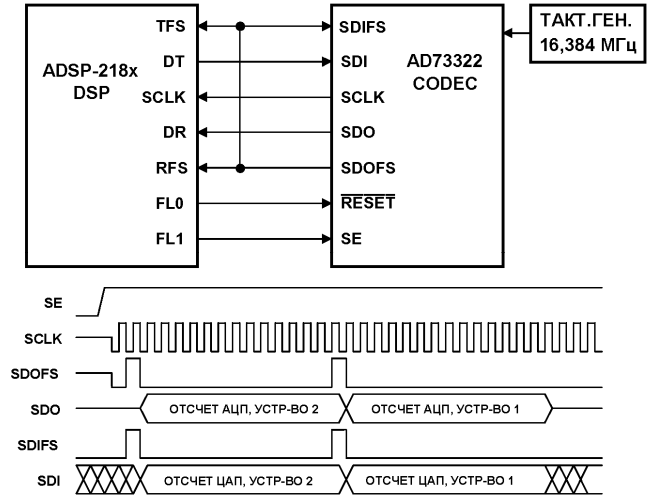


Рис. 2.27. Схема интерфейса кодека AD73322 и процессора серии ADSP-218x (режим пересылки данных)

Вывод сброса (RESET) может быть соединен с сигналом общего аппаратного сброса системы или может управляться любым программируемым выходом общего назначения.

В режиме программирования данные передаются от DSP-процессора в управляющие регистры микросхемы AD73322 для активации необходимого режима работы. После программирования конфигурации устройства, т.е. после правильной установки различных регистров управления, кодек может выйти из режима программирования и войти в режим передачи данных. Данные от двух АЦП передаются в DSP-процессор двумя блоками, состоящими из 16-разрядных слов. Точно так же данные для двух ЦАП передаются от DSP-процессора к микросхеме AD73322 аналогичными способом. Упрощенные временные диаграммы работы последовательного интерфейса показаны на рис. 2.27.

Приведенные примеры носят в основном демонстративный характер, более подробно вопросы проектирования ЦОС можно найти в специализированной литературе.

В современных медицинских приборах сигнальные процессоры могут использоваться и как ядро микропроцессорных систем, которые окружены полномасштабной памятью и различными интерфейсами с внешним оборудованием.

На рис. 2.28 приведён вариант системного интерфейса ADSP-2189M, который позволяет использовать соответствующий сигнальный процессор в составе микропроцессорной системы.

Приведенная схема включает два устройства, работающие через последовательные интерфейсы, 8-разрядную EPROM, внешнюю оверлейную память программ и данных. Возможность программной генерации циклов ожидания позволяет легко подключать быстрый процессор к более медленным периферийным устройствам. Процессор ADSP-2189M также поддерживает четыре внешних прерывания, семь универсальных сигналов ввода-вывода и два последовательных порта. Один из последовательных портов может быть сконфигурирован как источник двух дополнительных сигналов прерывания, один универсальный вход и один универсальный сигнал вывода, что даст в сумме шесть внешних сигналов прерывания, девять каналов ввода-вывода общего назначения при

сохранении одного полнофункционального последовательного порта.

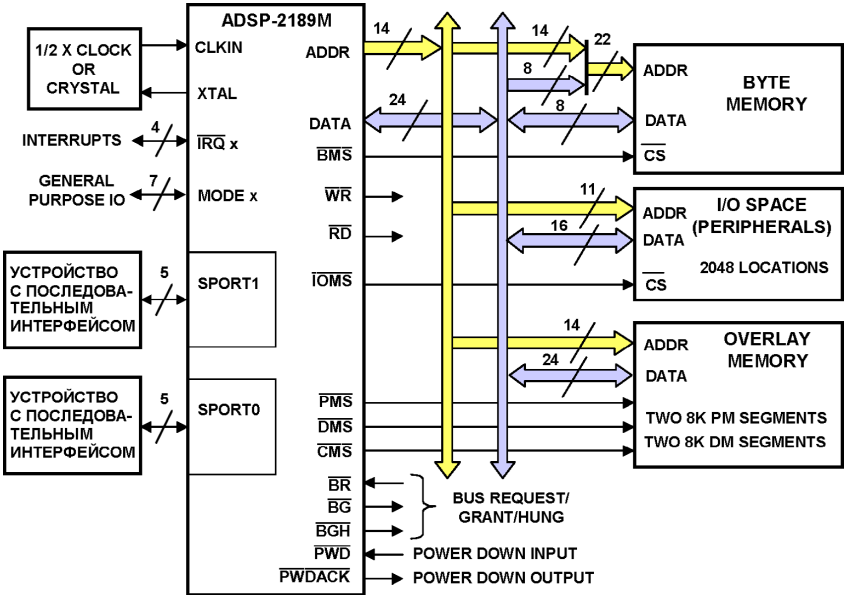


Рис. 2.28. Пример системы на базе ADSP-2189M

Процессор ADSP-2189M может также работать в режиме доступа к хост-памяти (host memory mode), который позволяет организовать доступ по всей ширине внешней шины данных, но ограничивает адресацию одним адресным битом. Дополнительные периферийные устройства могут быть подключены в режиме host memory mode при использовании внешних аппаратных средств для генерации и фиксации дополнительных адресных сигналов.

2.5. Особенности проектирования аппаратуры ЦОС

При проектирование систем цифровой обработки сигналов может возникать ряд специфических особенностей, среди которых к наиболее существенным с точки зрения эффективности и надеж-

ности аппаратуры ЦОС относят: различное напряжение питания используемых микросхем, разная природа сигналов в системе (аналоговая и цифровая), существенно различные скоростные и мощностные характеристики работы различных элементов, «пустой» монтаж и малые расстояния между выводами микросхем при высокой частоте преобразования информации и т.д. Недоучет этих особенностей при проектировании средств ЦОС может приводить к снижению эффективности их работы, отказам, сбоям и даже выходу аппаратуры из строя.

В связи с этим рассмотрим ряд рекомендаций по проектированию аппаратуры ЦОС, позволяющих повысить эффективность и надежность их работы.

Измененные тенденции, связанные с расчетом скорости обработки данных и повышение уровня компактности микросхем при минимальной стоимости приводят к разработке элементной базы с пониженным напряжением питания до 2,5 В и менее. При этом следует иметь в виду, что хотя низковольтное напряжение питания уменьшает размах используемых сигналов, уменьшая тем самым шум от переключения, но при этом уменьшается и допустимый для нормальной работы уровень шума (запас помехоустойчивости).

Если же в одном и том же приборе (системе) используются микросхемы с различным напряжением питания, то возникает ряд дополнительных проблем, связанных с тем что эти микросхемы могут «не понимать» друг друга, а с другой стороны более высоковольтные микросхемы создают большие выходные токи и требуют более мощных сигналов для своих выводов. Такое несоответствие может приводить не только к информационным сбоям, но и к повреждениям микросхем.

Для борьбы с этими явлениями используют либо микросхемы, которые по своей внутренней структуре обеспечивают совместимость работы с микросхемами другого напряжения питания, либо применяют специальные шинные переключатели (преобразователи).

На рис. 2.29 приведена схема формирования питающего напряжения микросхемы типа ADSP-21xx серии M вход которой, работая с логическими схемами с напряжением питания 2,5В «выдерживает» входные переключающие сигналы +3,3В, внутренняя ло-

гика питания от напряжения 2,5В (v_{DD} внутреннее (ядра)), а на выходе встроен двухтранзисторный драйвер, питающийся от напряжения v_{DD} (внешнее), которое в зависимости от окружения может быть выбрано как на уровне +3,3В, так и на уровне +2,5В.

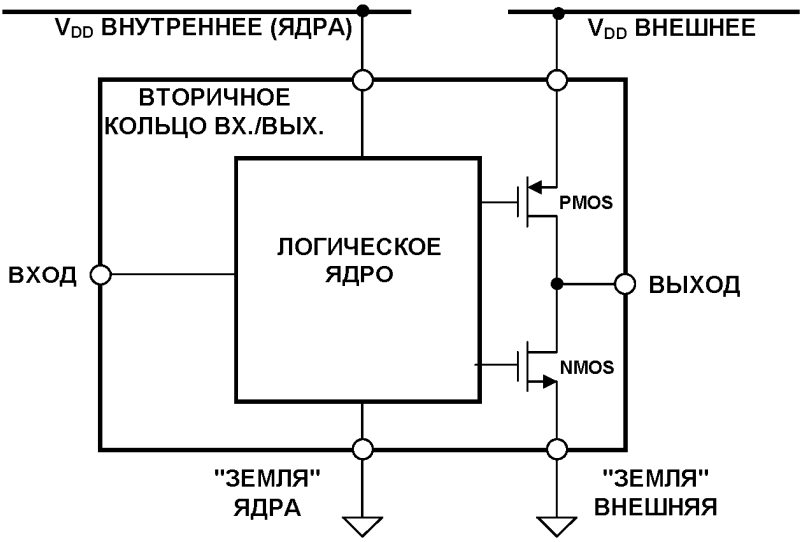


Рис. 2.29. КМОП ИС со вторичным контуром входов/выходов

Такая структура микросхемы обеспечивает устойчивую и совместимую работу двух типов микросхем, питающихся напряжением питания +3,3В и +2,5В.

Примером внешнего преобразования (приемо-передатчика) обеспечивающего устойчивое и совместим сопряжение между микросхемами с разноуровневым питанием (+3,3В и +2,5В) является микросхема 74VCX164245 (рис. 2.30).

Серия Fairchild 74VCX164245 – это низковольтные 16-битные преобразователи/приемопередатчики с двойным питанием и с тремя состояниями на выходе. Схема выходного драйвера питается от шины питания V_{DDB}, обеспечивая устойчивость и совместимость выхода с напряжением V_{DDB}.

Входная схема питается от шины питания V_{DDA}, и входная логическая схема регулирует уровни порогов на входе логики в соответствии с конкретным значением V_{DDA}.

Приведены различные варианты подключения цифровых схем с различными уровнями питания (рис 2.31–2.33).

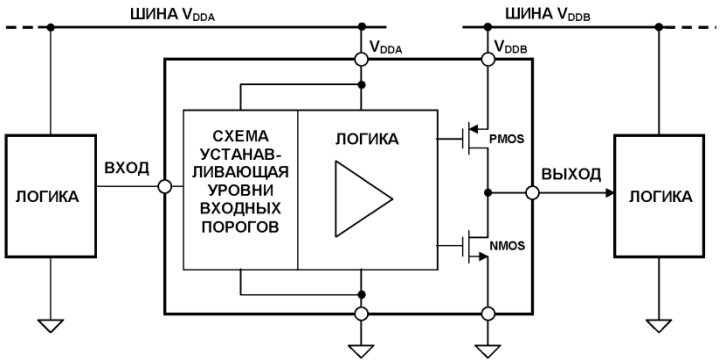


Рис. 2.30. Преобразователь-приемопередатчик логических сигналов Fairchild 74VCX164245

На рис. 2.31 показано два возможных варианта сопряжения 3.3-вольтовой и 2.5-вольтовой логики. На верхнем рисунке (*А*) показано прямое включение. Эта схема будет работать, если 2.5-вольтовая ИС обладает устойчивостью к 3.3 В на входе. Если 2.5-вольтовая ИС не является устойчивой к 3.3 В, то может использоваться преобразователь, приведенный на рис 2.30, как показано на рис. 2.31, *Б*.

На рис. 2.32, *А* показано прямое соединение между ИС с напряжениями питания 2.5 В и 3.3 В. Чтобы данная схема работала, выход 2.5-вольтовой микросхемы должен обеспечивать хотя бы 2 В. При отсутствии нагрузки на выходе 2.5-вольтовой микросхемы, вход 3.3-вольтовой ИС соединяется напрямую с шиной +2.5 В через внутреннее сопротивление транзистора микросхемы. Таким образом обеспечивается запас помехоустойчивости 0.5 В при номинальном напряжении питания 2.5 В.

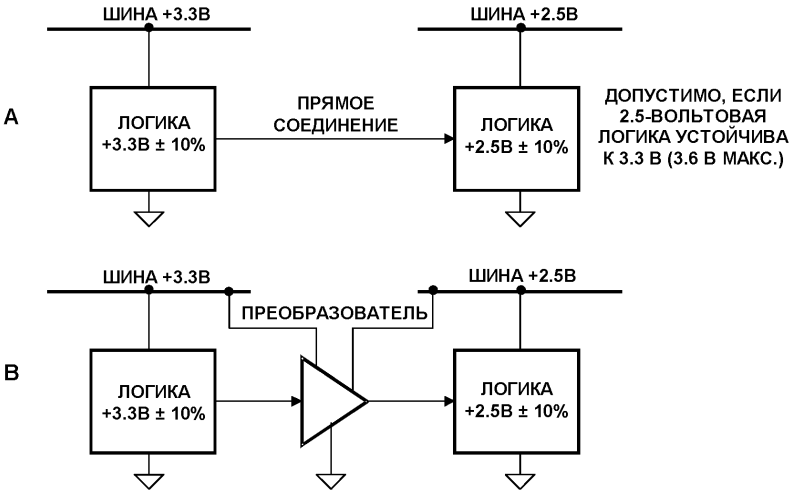


Рис. 2.31. Интерфейс между системами с напряжениями питания 3.3 В и 2.5 В

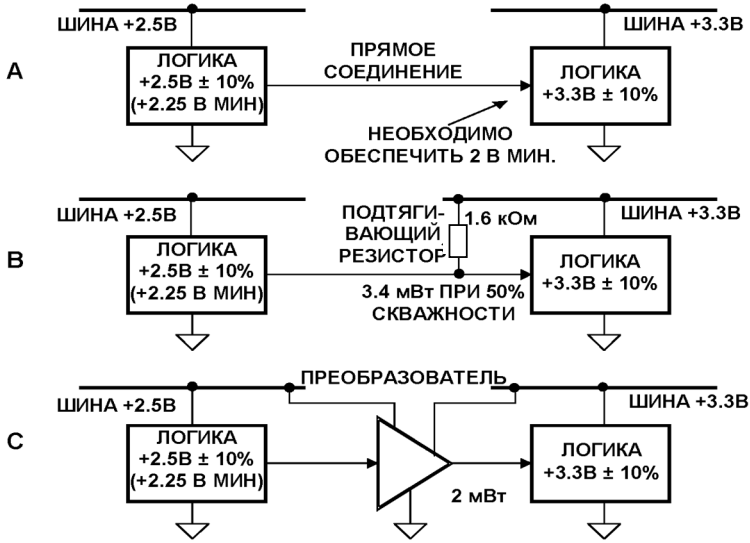


Рис. 2.32. Интерфейс между системами с напряжениями питания 2.5В и 3.3В

Однако ввиду допустимого 10%-ного разброса напряжение на шине 2.5 В может упасть до минимума в 2.25 В, и запас помехоустойчивости уменьшается до 0.25 В. Эта схема может тем не менее работать при сравнительно «тихом» окружении, но работать на пределе, если в напряжении источника питания присутствует шум.

Добавление «подтягивающего» резистора R сопротивлением 1.6 кОм, как показано на рис. 2.32, *B* гарантирует, что напряжение на выходе 2.5-вольтовой ИС не упадет ниже 2.5 В при наличии тока на входе 3.3-вольтового устройства, но запас помехоустойчивости все таки уменьшится при напряжении питания 2.25 В. При скважности 50% данный резистор добавляет примерно 3.4 мВт рассеиваемой мощности на каждый выход.

Более надежный интерфейс между 2.5-вольтовой и 3.3-вольтовой системами показан на рис. 2.32, *C*; здесь используется преобразователь. С его помощью решаются все проблемы, связанные с запасом помехоустойчивости, имеющиеся в схемах (A) и (B), и он потребляет около 2 мВт на каждый вход.

На рис. 2.33 приведен вариант сопряжения микросхем с напряжениями питания 3.5 В и 5В. С помощью шинного переключателя на МОП-транзисторе с каналами n-типа.

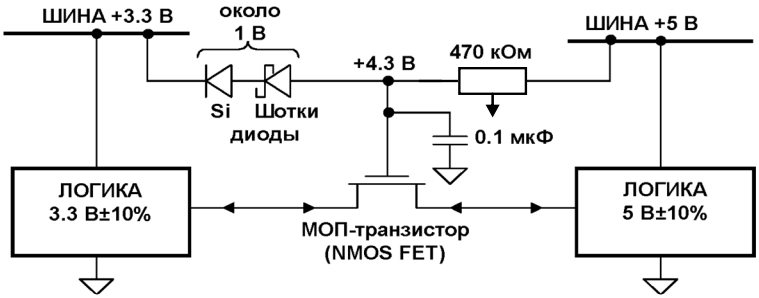


Рис. 2.33. Двунаправленный интерфейс сопряжения между 3.5В и 5В логикой

Если на затвор транзистора подано напряжение 4.3В, то максимальная величина напряжения проходящего сигнала составит 3.3В (примерно на 1 В меньше, чем напряжение на затворе МОП-транзистора). Если напряжение на входе и на выходе не превыша-

ет 3.3В, МОП-транзистор представляет собой небольшое сопротивление (RON = 5 Ом). Когда входной сигнал достигает величины 3.3В, сопротивление МОП-транзистора возрастает, ограничивая таким образом уровень сигнала на выходе.

Один из путей получения напряжения питания 4.3В на системной плате, где имеются 5В и 3.3В, является включение диода. В схеме на рис. 2.33 напряжение 4.3В генерируется кремниевым диодом и диодом Шотки, соединенными последовательно и подключенными к шине питания 3.3В. Этот метод позволяет получить более стабильное напряжение смещения на затворе МОП-транзистора с учетом допустимости 10% разброса напряжений питания 5В и 3.3В.

В современных высокоскоростных системах ЦОС относительно низкоскоростные АЦП и ЦАП работают в окружении достаточно мощных цифровых процессоров, создающих высокий уровень высокочастотных помех, которые попадают как во входные и выходные цепи ЦАП и АЦП, так и в шины питания.

Кроме того современные АЦП при относительно низких скоростях квантования используют для своей работы достаточно высокочастотные генераторы. Так, например, АЦП выборки (АЦП со схемой выборки-запоминания), используемые в системах обработки сигналов, работают с достаточно высокоскоростными генераторами тактовых импульсов с малым временем нарастания и спада (обычно несколько наносекунд) и должны рассматриваться как высокоскоростные устройства, даже если их производительность (частота отсчетов) представляется невысокой. Например, 12-разрядный АЦП последовательного приближения (SAR) типа AD7892 работает при внутренней тактовой частоте 8 МГц, тогда как его частота отсчетов составляет только 600 кГц.

Для сигма-дельта (Σ-Δ) АЦП также требуется высокочастотный тактовый генератор, так как такие АЦП имеют высокий коэффициент передискретизации. 16-разрядный АЦП AD7722 имеет частоту обновления на выходе (эффективную частоту отсчетов), равную 195 кГц, но в действительности производит выборку с частотой 12.5 МГц (в 64 раза выше). Даже так называемые низкочастотные сигма-дельта (Σ-Δ) АЦП промышленного назначения с

высоким разрешением (имеющие частоту обновления на выходе от 10 Гц до 7.5 кГц) работают при тактовой частоте 5 МГц или выше и обеспечивают 24-разрядное разрешение (например, микросхемы фирмы Analog Devices типа AD7730 и AD7731).

Еще более осложняет вопрос то, что ИС со смешанными сигналами содержит как аналоговую, так и цифровую часть

Работая в смешанных системах порождает проблему обеспечения большого динамического диапазона в условиях высокого уровня высокоскоростных помех. В современных системах ЦОС эта проблема разрешается технически грамотной трассировкой сигналов, развязкой и грамотно выполняемым заземлением. Эта проблема не является тривиальной. Она достаточно подробно описана в специальной литературе.

В данной работе, не вдаваясь в теоретические «тонкости» рассмотрим ряд общепринятых практических рекомендаций.

Обеспечение низкоимпедансных заземляющих поверхностей большой площади очень важно для всех современных аналоговых схем. Заземляющая поверхность действует не только как низкоимпедансный обратный тракт для развязки высокочастотных токов (вызванных работой скоростных цифровых схем), но также минимизирует электромагнитные радиочастотные (EMI/RFI) помехи. Благодаря экранирующему действию заземляющей поверхности чувствительность устройства к внешним помехам также уменьшается.

Заземляющие поверхности также позволяют передавать высокоскоростные цифровые и аналоговые сигналы с использованием технологий линий передач (полосковую или микрополосковую), там, где требуется получить определенное характеристическое сопротивление линии.

В качестве примера, поясняющего необходимость аккуратного отношения к решению задач «заземления» на рис. 2.34 показано два варианта подключения аналоговых и цифровых схем к референтной земле (GND REF).

На рис. 2.34 элементы R_A, L_A, R_D и L_D характеризуют импеданс земляных проводов аналоговой и цифровой части системы ЦОС по которым протекают соответствующие возвратные токи L_A и L_D. Для варианта соединения земляных шин аналоговых и цифровых

схем, приведенного на рис. 2.31, а возвратные токи земли различных частей схемы смешиваются (моделируются) приводя к значительным помехам. В схеме, приведенной на рис 2.31, *Б* такого смешения «токов земли» не происходит. Этот рисунок иллюстрирует фундаментальную концепцию заземления «звездой», называемую иногда системой с одной точкой.

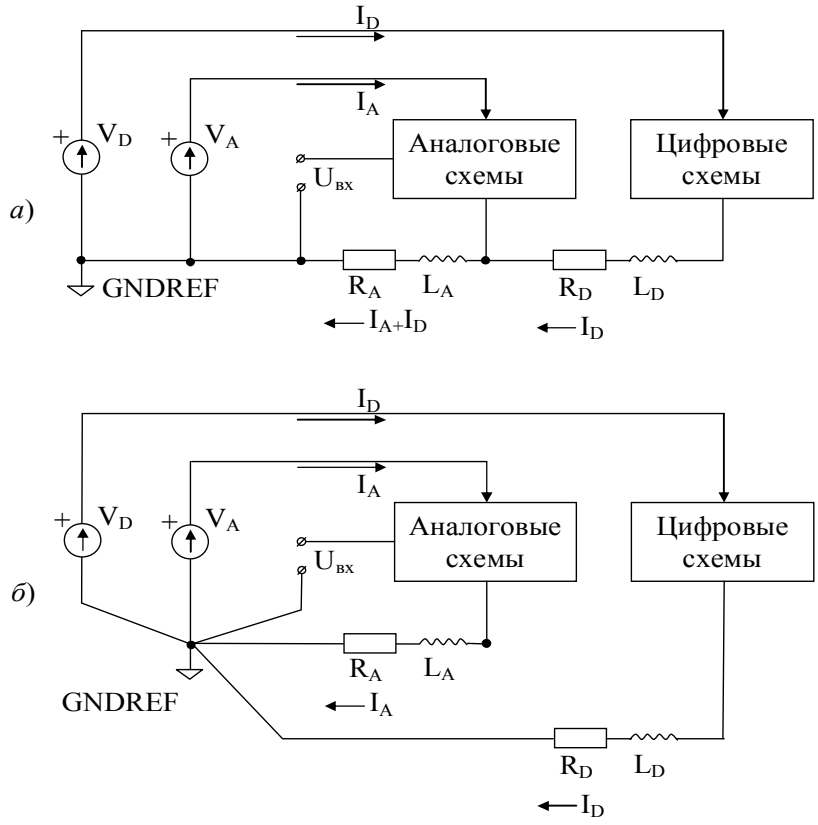


Рис. 2.34. Цифровые токи «земли» аналоговой и цифровой схем при их различном подключении к GND REF

Реализовать настоящее одноточечное заземление в системе, которая содержит большое количество высокочастотных трактов, сложно, так как физическая длина каждого провода, по которому течет обратный ток, будет вносить паразитное сопротивление и индуктивность, которые могут сделать затруднительным обеспечение низкоимпедансного заземления для токов высокой частоты. На практике тракт возвратного тока должен включать в себя заземляющие поверхности большой площади для того, чтобы обеспечить низкое сопротивления для высокочастотных токов. Таким образом, без низкоимпедансной заземляющей поверхности практически невозможно избежать появления общего для аналоговой и цифровой схем тракта заземления, особенно на высоких частотах.

Все выводы заземления микросхем должны соединяться с помощью пайки прямо с низкоимпедансной заземляющей поверхностью с целью минимизировать последовательную индуктивность и сопротивление. Использование традиционных панелек (разъемов) для микросхем в высокоскоростных устройствах не рекомендуется. Добавочная индуктивность и емкость даже «низкопрофильных» панелек может привести к нарушению работы схемы из-за появления дополнительных трактов.

Выводы питания должны быть развязаны прямо на заземляющую поверхность с помощью низкоиндуктивных керамических конденсаторов для поверхностного монтажа (SMD). Если используется конденсатор для обычного монтажа, то его выводы должны иметь длину не более 1 мм. Керамические конденсаторы должны быть расположены как можно ближе к выводам питания микросхемы. Для дополнительной развязки могут также потребоваться ферритовые бусины.

Каждая печатная плата в системе должна иметь хотя бы один слой, полностью отведенный под заземляющую поверхность. В идеале двусторонняя плата должна иметь одну сторону, полностью отведенную под заземление и вторую – для различных соединений. На практике это невозможно, так как частично заземляющая поверхность, разумеется, должна быть удалена для отведения места под дорожки питания и сигналов, межслойные переходы и сквозные монтажные отверстия. Тем не менее как можно

больше площади заземляющей поверхности должно быть сохранено, хотя бы 75% необходимо оставить. После окончания предварительной разводки платы поверхность заземления должна быть аккуратно проверена для того, чтобы убедиться, что не осталось изолированных «островков» заземления, так как выводы заземления микросхем, расположенные на таких островках, не будут иметь связи с заземляющей поверхностью. Также заземляющая поверхность должна быть проверена на предмет слишком тоненьких соединений между соседними большими площадями, которые могут значительно уменьшить эффективность заземляющей поверхности.

Системы, в которых интегральные микросхемы для поверхностного монтажа расположены тесно, будут иметь большое число соединений, поэтому здесь нужно использовать многослойные платы. Это позволит хотя бы один слой полностью отвести под заземление. В простой 4-слойной плате два внутренних слоя обычно используются для заземляющей поверхности и поверхности питания, а два внешних слоя – для выполнения соединений между установленными компонентами. Расположение питающей и заземляющей поверхностей в соседних слоях обеспечивает дополнительную межповерхностную емкость, которая способствует высокочастотной развязке тока питания. В большинстве систем четырех слоев недостаточно, и требуются дополнительные слои для трассировки линий сигналов, а также питания.

Лучший способ минимизировать импеданс заземления в многоплатной системе – использовать «материнскую плату» в качестве объединительной для организации соединения между платами, и, кроме того, обеспечить продолжение заземляющей поверхности на общую плату. В соединителе печатной платы хотя бы 30–40% выводов должно быть отведено под заземление, и эти выводы должны быть связаны с заземляющей поверхностью на материнской объединяющей плате. Для окончательного завершения устройства заземления системы существуют два способа:

1. Заземляющая поверхность на объединяющей плате может быть соединена с «землей» монтажной панели (шасси) во многих точках, таким образом равномерно распределяя различные пути возвратного тока.

2. Заземляющая поверхность может быть подключена по схеме «звезды» к единственной в системе точке заземления (обычно вблизи источника питания).

Первый подход чаще всего используется в чисто цифровых системах, но может быть использован и в системах со смешанными сигналами, если токи заземления цифровых схем достаточно малы и распределены на больших площадях. Низкий импеданс заземления обеспечивается на всем пути по плате, по объединяющей плате и далее по монтажной панели (шасси). Однако очень важно поддерживать хорошее электрическое соединение в местах, где «земля» связана с металлической монтажной панелью. Для этого необходимы металлические винты-«саморезы» или шайбы с насечками. Особое внимание соединению должно быть уделено там, где в качестве материала монтажной панели используется алюминий, так как его поверхность проявляет себя как изолятор.

Второй подход (заземление «звездой») часто используется в высокоскоростных системах с смешанными сигналами, имеющих отдельные аналоговую и цифровую системы заземления, и ниже обсуждается более подробно.

В системах со смешанными сигналами с большим числом цифровых схем весьма желательно физически отделить чувствительные аналоговые компоненты от шумных цифровых компонентов. Также пойдет на пользу использование раздельных заземляющих поверхностей для аналоговых и цифровых схем. Эти поверхности не должны перекрываться для того, чтобы уменьшить емкостную связь между ними. Раздельные аналоговая и цифровая заземляющие поверхности продолжаются на объединительной плате с использованием или заземляющей поверхности материнской платы, или «экранирующего заземления», которое выполняется при помощи проводов заземления, чередующимися в разъёме с сигнальными проводами. На структурной схеме, показанной на рис. 2.35, видно, что две заземляющих поверхности на всем своем протяжении идут отдельно до точки заземления «звездой», которая обычно располагается около источника питания.

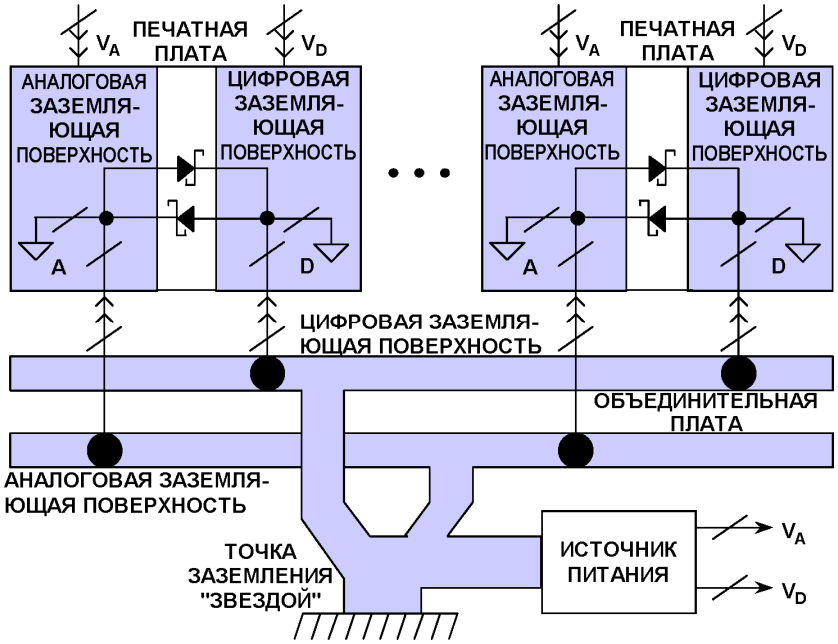


Рис. 2.35. Разделение аналоговой и цифровой заземляющих поверхностей

Соединение заземляющих поверхностей и источника питания в точке заземления «звездой» должно быть выполнено с помощью многочисленных шин или толстого медного жгута для минимизации сопротивления и индуктивности. Пара встречно-параллельных диодов Шотки имеется на каждой печатной плате для предотвращения случайного появления постоянного напряжения между двумя заземляющими системами в момент, когда платы вставляются или вынимаются. Это напряжение не должно превышать 300 мВ, чтобы избежать выхода из строя ИС, которая подключена как к аналоговой, так и к цифровой заземляющим поверхностям. Предпочтительно использовать диоды Шотки, так как они имеют малую емкость и малое падение напряжения в режиме прямого тока. Низкая емкость позволяет избежать связи по переменному току между аналоговой и цифровой заземляющими поверхностями. Диоды Шотки начинают проводить при прямом напряжении

около 300 мВ, и если ожидаются большие токи, может понадобиться несколько параллельно соединенных диодов. В некоторых случаях вместо диодов Шотки могут быть использованы дроссели с ферритовыми бусинами, однако они вызывают появление паразитных контуров с замыканием через «землю» по постоянному току, которые могут вызвать проблемы в прецизионных системах.

Обязательно нужно обеспечить сопротивление заземляющих поверхностей как можно меньшим на всем пути к точке заземления «звездой». Переменное или постоянное напряжение более чем 300 мВ между двумя заземляющими поверхностями может не только вывести из строя ИС, но и вызвать ошибочное включение логического элемента и, возможно, переход в фиксированное состояние.

Чувствительные аналоговые компоненты, такие как усилители и источники опорного напряжения, всегда подключаются и развязываются на аналоговой заземляющей поверхности. АЦП и ЦАП (и другие ИС со смешанными сигналами) с небольшими цифровыми токами обычно должны рассматриваться как аналоговые компоненты и также заземлены и развязаны на аналоговой заземляющей поверхности.

При работе АЦП на цифровые блоки рекомендуется на выходе преобразователя ставить буферные регистры с выходными каскадами, обладающими небольшой емкостью (например, на КМОП транзисторах).

Это позволяет изолировать цифровые цепи преобразователя от шумов, присутствующих на шине данных. Данный регистр также служит для минимизации нагрузки на цифровых выходах преобразователя и действует как экран между этими цифровыми выходами и шиной данных. Даже несмотря на то, что многие преобразователи имеют входы/выходы с тремя состояниями, применение подобного изолирующего регистра остается оправданным. В некоторых случаях для обеспечения большей развязки может быть желательным добавление еще одного буферного регистра на аналоговой заземляющей поверхности после выхода преобразователя.

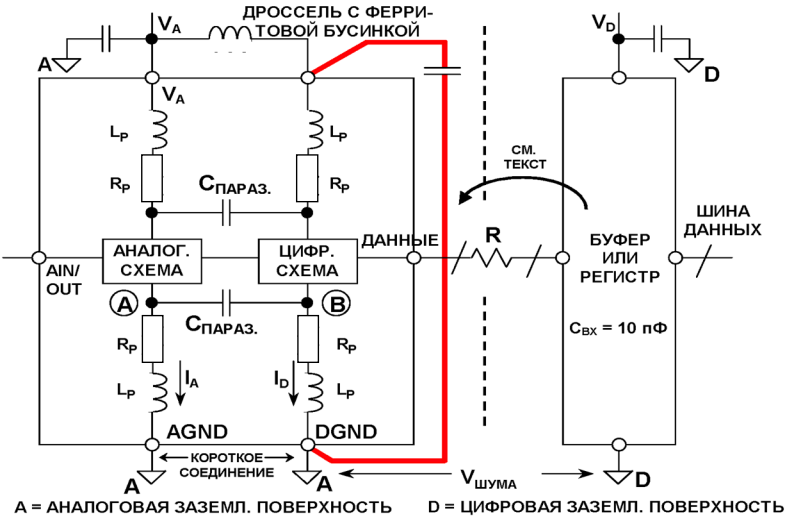


Рис. 2.36. Правильное заземление микросхемы со смешанными сигналами с небольшими цифровыми токами

Последовательно включенный резистор между выходом АЦП и входом буферного регистра помогает минимизировать цифровые импульсные токи, которые могут повлиять на качество работы преобразователя. Этот резистор изолирует драйвер цифрового выхода преобразователя от входной емкости буферного регистра. Кроме того, RC-цепочка, образуемая резистором R и входной емкостью буферного регистра, действует как фильтр низкой частоты и таким образом сглаживает резкие фронты.

Буферный регистр и другие цифровые схемы должны быть заземлены и развязаны на цифровой заземляющей поверхности печатной платы. Следует иметь в виду, что любой шумовой сигнал между аналоговой и цифровой заземляющими поверхностями уменьшает запас помехоустойчивости цифрового интерфейса преобразователя. Так как запас помехоустойчивости цифровой схемы составляет порядка сотен или тысяч милливольт, это едва ли будет иметь значение. Аналоговая заземляющая поверхность обычно не

бывает слишком «шумной», но если шум на цифровой заземляющей поверхности (относительно аналоговой заземляющей поверхности) превышает несколько сотен милливольт, то необходимо предпринять шаги для уменьшения импеданса цифровой заземляющей поверхности, таким образом обеспечивая приемлемый уровень запаса помехоустойчивости цифровой схемы. Ни при каких условиях напряжение между двумя заземляющими поверхностями не должно превышать 300 мВ, иначе ИС может выйти из строя.

Также весьма желательно наличие отдельных источников питания для аналоговой и цифровой схем. Для питания преобразователя необходим «аналоговый» источник питания. Если преобразователь имеет вывод, обозначенный как вывод питания цифровой части схемы (VD), он должен быть подключен или к отдельному «аналоговому» источнику питания, или подключен через фильтр, как показано на схеме. Все выводы питания преобразователя должны быть развязаны на аналоговой заземляющей поверхности, а все выводы питания цифровых схем должны быть развязаны на цифровой заземляющей поверхности. Если источник «цифрового» питания относительно тихий, он может оказаться вполне пригодным для питания аналоговых схем.

В некоторых случаях не представляется возможным подключить вывод VD к источнику питания аналоговой части. Некоторые из новейших высокоскоростных ИС могут быть рассчитаны на работу аналоговой части при напряжении питания 5 В, в то время как цифровая часть питается от источника +3 В для того, чтобы быть совместимым с 3-вольтовой логикой. В этом случае вывод питания +3 В микросхемы должен быть развязан непосредственно на аналоговую заземляющую поверхность. Также будет благоразумно включить дроссель на ферритовой бусине последовательно с линией питания, которая подключена к выводу питания +3 В цифровой части ИС.

В высокопроизводительных системах дискретизации для генерации тактовых импульсов преобразования АЦП (или ЦАП) необходимо использовать кварцевый генератор с низким фазовым шумом, так как фазовый шум (jitter) тактового генератора модулирует аналоговый входной/выходной сигнал и увеличивает уровень шу-

ма и искажений. Генератор тактовых импульсов должен быть изолирован от шумных цифровых цепей и заземлен и развязан на аналоговой заземляющей поверхности, точно так же как операционные усилители и АЦП.

Однако это не всегда возможно по различным причинам. Во многих случаях тактовые импульсы преобразователя необходимо получить из более высокочастотных тактовых импульсов всей системы, которые генерируются на цифровой заземляющей поверхности. Затем эти импульсы должны идти от места их генерации на цифровой заземляющей поверхности к АЦП, находящемуся на аналоговой заземляющей поверхности. Шум между двумя заземляющими поверхностями добавляется непосредственно к тактовому сигналу и приводит к увеличению фазового шума.

Этот фазовый шум может ухудшить соотношение сигнал/шум преобразователя, а также вызвать появление нежелательных гармоник. Данное явление иногда можно устранить, если передавать тактовые импульсы как дифференциальный сигнал с помощью либо небольшого высокочастотного трансформатора, как показано на рис. 2.37, либо с помощью быстродействующих интегральных микросхем дифференциального драйвера и приемника. Если используется активный дифференциальный драйвер и приемник, то они должны быть выполнены по технологии ECL, чтобы минимизировать фазовый шум. В системе с однополярным питанием +5 В микросхема ECL-логики может быть включена между шиной земли и питания +5 В (PECL), а сигнал с дифференциальных выходов преобразован для подачи на вход тактовых импульсов АЦП. В любом случае, изначальные тактовые импульсы должны быть генерированы с помощью кварцевого генератора с низким уровнем фазового шума.

Одним из средств размыкания паразитных контуров заземления является использование изолирующих методов. Аналоговые изолирующие усилители часто находят применение там, где требуется высокая степень изоляции, например, в медицинском оборудовании. Методы цифровой изоляции обеспечивают надежный способ передачи цифрового сигнала через интерфейс, не вызывающий помех на заземлении.

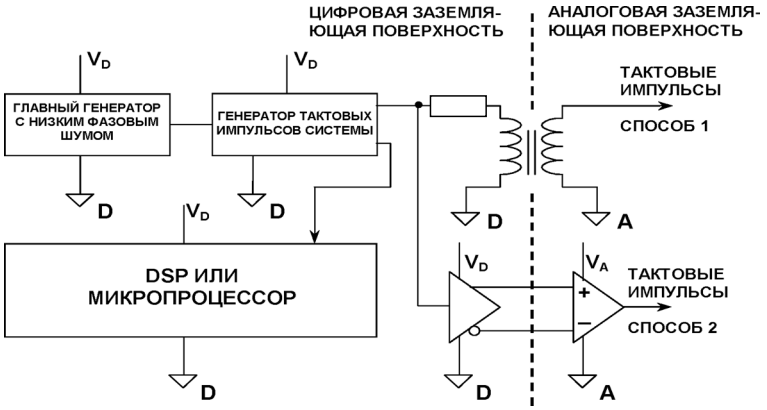


Рис. 2.37. Передача тактовых импульсов
с цифровой заземляющей поверхности на аналоговую

В цифровых системах используют развязывающие микросхемы различных фирм. Их описание можно найти в работах [43, 67].

Прецизионные аналоговые схемы обычно питаются от источника с хорошим линейным стабилизатором с низким уровнем шума. Однако за последние 10 лет в электронных схемах стали гораздо чаще использоваться импульсные источники (ИИП), и как следствие, они стали использоваться также для питания аналоговых схем. Причины их популярности – высокий КПД, малое повышение температуры, небольшой размер и вес.

Несмотря на эти преимущества, ИИП имеют отрицательные стороны, и самое главное – большой уровень помех на выходе. Эти помехи обычно занимают широкую полосу частот, проявляются в проводимых и излучаемых шумах, а также в нежелательных электрических и магнитных полях. Напряжение выходного шума ИИП представляет собой короткие импульсы или пики напряжения. Хотя значение частоты переключения лежит в пределах 20 кГц-1 МГц, выбросы могут содержать частотные компоненты, достигающие 100 МГц и выше. В спецификации на ИИП обычно указано значение среднеквадратичного напряжения шума, но потребитель должен знать амплитуду выбросов при определенной нагрузке, соответствующей проектируемой системе.

В работе [67] можно найти способы выбора фильтров, которые позволяют со сравнительно небольшими потерями довести качество напряжения на выходе ИИП до уровня пригодного для аналоговых устройств. Рассмотренные схемные решения фильтров обычно подходят для всех типов источников питания, использующих импульсные элементы в тракте питания. Это относится и к различным преобразователям постоянного напряжения (DC-DC), и к распространенным источникам питания на напряжение 5 В (компьютерным).

3. ТЕОРИЯ УНИВЕРСАЛЬНОГО ПРЕОБРАЗОВАНИЯ СИСТЕМ КОДИРОВАНИЯ

Проблемы взаимодействия современной медицинской техники и проблемы передачи данных, сформулированы ранее. Решением обозначенных проблем является ГИИ, для проектирования которого необходимо учесть основополагающую модель взаимодействия систем OSI. Использование при разработке ГИИ данной модели позволяет третьим сторонам создавать различные аппаратные или программные средства расширения и модификации, а также создавать программно-аппаратные комплексы из продуктов разных производителей, взаимодействующих между собой. Основываясь на ГОСТ Р ИСО/МЭК 8831-99 «Информационная технология. Взаимосвязь открытых систем. Концепции и услуги передачи и обработки заданий», рассмотрим каждый из уровней системы OSI, выявим реализуемые ими функции. Произведем формализацию спецификации OSI, что упростит управление функционированием ГИИ и преобразованием систем кодирования.

3.1. Многоуровневое представление средств сетевого взаимодействия

Организация взаимодействия между устройствами сети является сложной задачей. Для ее решения используется известный универсальный прием – декомпозиция. Декомпозиция состоит в четком определении моделей и функций каждого модуля, а также порядка их взаимодействия (то есть межмодульных интерфейсов). При таком подходе каждый модуль можно рассматривать как «черный ящик», абстрагируясь от его внутренних механизмов и концентрируя внимание на способе взаимодействия этих модулей. В результате такого логического упрощения задачи появляется возможность независимого разработки, тестирования и модификации модулей.

Наиболее эффективной концепцией, развивающей идею декомпозиции, является многоуровневый подход. После представления исходной задачи в виде множества модулей их группируют и упорядочивают по уровням, образующим иерархию. В соответствии с принципом иерархии для каждого промежуточного уровня можно указать непосредственно примыкающие к нему соседние: вышележащий и нижележащий уровни.

Такой подход дает возможность проводить разработку, тестирование и модификацию отдельного уровня независимо от других уровней. Иерархическая декомпозиция позволяет, двигаясь от более низкого уровня к более высокому, переходить ко все более и более абстрактному.

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют, по меньшей мере, две стороны, что требует организации согласованной работы двух иерархий аппаратных и программных средств на разных устройствах, участвующих в обмене информацией. Оба участника сетевого обмена должны принять множество соглашений: по уровням и форме электрических сигналов, по способу определения размера сообщений, по методам контроля достоверности. Соглашения должны быть приняты на всех уровнях, начиная от самого низкого – уровня передачи битов (физического), и заканчивая самым высоким (уровнем пользователя), реализующим обслуживание пользователей сети.

В начале 80-х годов ряд международных организаций по стандартизации – ISO, ITU-T и некоторые другие – разработали модель, которая сыграла значительную роль в развитии сетей. Эта модель называется моделью взаимодействия открытых систем OSI.

В модели OSI (рис. 3.1) средства взаимодействия делятся на семь уровней: прикладной, представительный, сеансовый, транспортный, сетевой, уровень звена данных и физический. Каждый уровень имеет дело с одним определенным аспектом взаимодействия сетевых устройств.

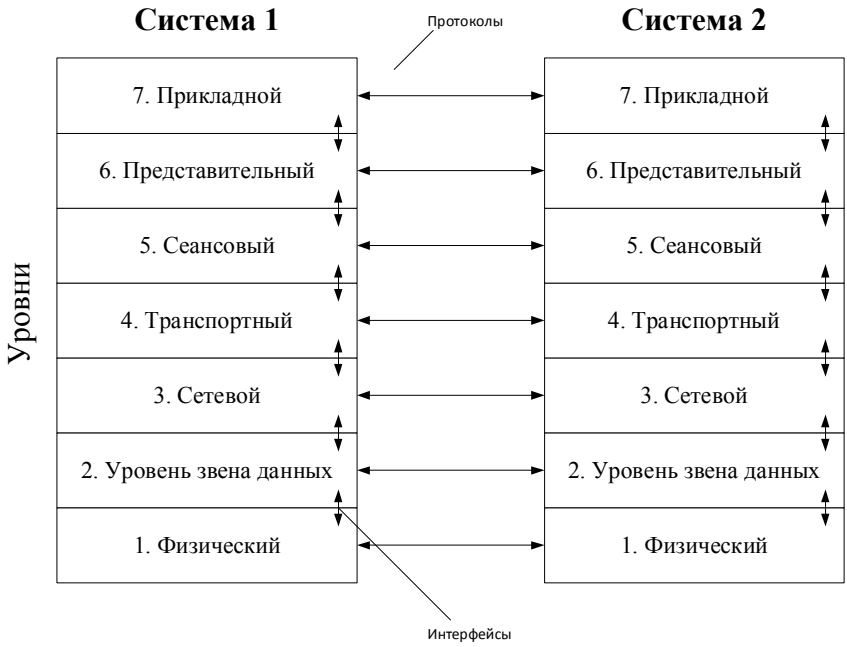


Рис. 3.1. Система OSI

Дадим подробное описание системы OSI для дальнейшей оптимизации ГИИ.

Физический уровень

Физический уровень обеспечивает механические, электрические, функциональные и процедурные средства для активизации, поддержки и деактивации физических соединений, предназначенных для побитовой передачи между логическими объектами уровня звена данных. Физическое соединение может проходить через промежуточные открытые системы, каждая из которых осуществляет ретрансляцию битового потока средствами физического уровня. Логические объекты физического уровня связаны через физическую среду.

Уровень звена данных

Уровень звена данных обеспечивает функциональные и процедурные средства для установления, поддержания и разрыва соединений уровня звена данных между сетевым и логическими объектами и для передачи сервисных блоков данных этого уровня. Соединение уровня звена данных строится на основе одного или нескольких физических соединений.

На уровне звена данных обнаруживаются и, по возможности, исправляются ошибки, которые могут возникнуть на физическом уровне. Кроме того, обеспечивается для сетевого уровня возможность управлять подключением каналов данных на физическом уровне.

Сетевой уровень

В рамках сетевого уровня предоставляются средства установления, поддержания и разрыва сетевого соединения между системами, содержащими связанные прикладные логические объекты, а также функциональные и процедурные средства для обмена по сетевому соединению сетевыми сервисными блоками данных между транспортными логическими объектами.

Элементы интерфейса сетевого уровня обеспечивают транспортным логическим объектам независимость от функций маршрутизации и ретрансляции, связанных с процессами установления и функционирования данного сетевого соединения.

Все функции ретрансляции и расширенные протоколы последовательного переноса данных, которые предназначены для поддержания сетевых услуг между оконечными открытыми системами OSI, функционируют ниже транспортного уровня, т.е. на сетевом уровне или ниже.

Функции сетевого уровня обеспечивают использование различных конфигураций для поддержки сетевых соединений: от соединений, поддерживаемых двухпунктовыми сетевыми конфигурациями, до сетевых соединений, поддерживаемых сочетаниями подсетей с различными характеристиками.

Транспортный уровень

Транспортные услуги обеспечивают передачу интерфейсом данных без каких-либо изменений между сеансовыми логическими объектами и освобождают их от выполнения операций, обеспечивающих надежную и экономически эффективную передачу данных.

Транспортный уровень оптимизирует использование доступных сетевых услуг, чтобы обеспечить пропускную способность, требуемую каждым сеансовым логическим объектом, при минимальных затратах. Эта оптимизация достигается с учетом ограничений, обусловленных совместными требованиями со стороны всех одновременно работающих сеансовых логических объектов, а также общим качеством и объемом сетевых услуг, предоставляемых транспортному уровню.

Все протоколы, определенные на транспортном уровне, имеют межоконечный характер, окончаниями считаются связанные транспортные логические объекты.

Поскольку сетевые услуги обеспечивают сетевые соединения между транспортными логическими объектами по принципу «каждый с каждым», включая использование последовательно соединенных подсетей, транспортный уровень освобождается от функций маршрутизации и ретрансляции.

На транспортном уровне имеются функции, обеспечивающие требуемое качество услуг на основе услуг, предоставляемых сетевым уровнем.

Сеансовый уровень

Сеансовый уровень обеспечивает средства, необходимые взаимодействующим логическим объектам уровня представления, для организации и синхронизации диалога и административного управления между ними. С этой целью сеансовый уровень предоставляет услуги по установлению сеансового соединения между двумя логическими объектами уровня представления, а также услуги по поддержанию упорядоченного обмена данными при взаимодействии.

В процессе существования сеансового соединения сеансовые услуги используются логическими объектами уровня представления для регулирования диалога между ними и обеспечения упорядоченного обмена сообщениями по сеансовому соединению. Сеансовое соединение существует до тех пор, пока оно не будет разорвано логическими объектами уровня представления или сеансовыми логическими объектами. Пока сеансовое соединение существует, сеансовые услуги поддерживают состояние диалога даже при потерях данных на транспортном уровне.

Уровень представления

Уровень представления устанавливает способы представления информации, которой обмениваются прикладные логические объекты или на которую они ссылаются в процессе этого обмена.

Уровень представления охватывает два взаимодополняющих аспекта таких способов представления информации:

а) представление данных, подлежащих передаче между прикладными логическими объектами;

б) представление структуры данных, которую прикладные логические объекты намереваются использовать в своем диалоге наряду с представлениями совокупности действий, выполняемых над этой структурой данных.

Указанные выше аспекты представления информации основываются на общем понятии синтаксиса передачи. Уровень представления имеет дело только с синтаксисом или со способами представления данных, а не с их семантикой или смысловым содержанием.

Уровень представления обеспечивает способы представления информации, которые являются общими (или одинаковыми) для взаимодействующих прикладных логических объектов. Таким образом, прикладные логические объекты освобождаются от функции представления информации, поскольку используется общий способ представления и для них обеспечивается синтаксическая независимость. Такая независимость может быть реализована двумя путями:

1. На уровне представления обеспечиваются элементы поддержки синтаксиса, являющиеся общими для использующих их прикладных логических объектов.

2. Прикладные логические объекты могут использовать произвольный синтаксис. Уровень представления обеспечивает преобразование этих синтаксисов, для обмена между прикладными логическими объектами используется общий синтаксис. Такое преобразование выполняется внутри открытой системы. На другие открытые системы это не влияет и, следовательно, не оказывает влияния на стандартизацию протоколов уровня представления.

Прикладной уровень

Прикладной уровень является наивысшим уровнем в эталонной модели OSI и обеспечивает прикладным процессам средства доступа к функциональной среде OSI. Поэтому прикладной уровень не имеет интерфейса с более высоким уровнем. Он является единственным средством доступа прикладных процессов к функциональной среде OSI.

Прикладной уровень служит окном между взаимодействующими прикладными процессами, использующими функциональную среду OSI для обмена смысловой информацией.

На прикладном уровне выполняются все функции связи между открытыми системами, которые не выполняются нижележащими уровнями. В их число включаются функции, выполняемые программными средствами, и функции, выполняемые людьми.

Когда определенный экземпляр прикладного процесса некоторой открытой системы намерен установить связь с некоторым экземпляром прикладного процесса в другой открытой системе, он должен обратиться к экземпляру прикладного логического объекта на прикладном уровне своей открытой системы. Тогда этот логический объект становится ответственным за установление ассоциации с экземпляром соответствующего прикладного логического объекта в открытой системе получателя. Этот процесс осуществляется посредством обращения к экземплярам логических объектов нижележащих уровней. После установления ассоциации меж-

ду двумя прикладными логическими объектами может устанавливаться связь между соответствующими прикладными процессами.

3.2. Формализация спецификации ГИИ

Для формализации спецификации ГИИ необходимо выделить выполняемые функции на каждом из уровней системы OSI, построить графы взаимодействия и оптимизировать их при помощи метода блочно-функционального распределения (БФР).

В соответствии с общим принципом синтеза структуры большой системы сформирован алгоритм БФР. Он основан на преобразованиях структуры с использованием алгоритма Бержа для нахождения наименьшего внешне устойчивого множества. Если дан граф $G(X, \Gamma_x)$, то множество $T \subset X$ внешне устойчиво, если для каждой вершины $x \notin T$ справедливо $\Gamma_x \cap T \neq \emptyset$, т.е. когда $\Gamma^{-1}T \supset X \setminus T$. Для каждого графа существует семейство внешне устойчивых множеств.

Алгоритм БФР состоит в следующем:

1. Устанавливается наиболее полный список функций, реализуемый системой в целом в любой форме.

2. Выбирается начальный способ преобразования информации, и на его основе записывается начальный оператор системы.

3. Выполняется расширение начального отображения за счет детализации объектов морфизма на основе возможных вариантов детализации. Множество вариантов детализации строится или на основе уже известных решений, или исходя из эвристического поиска. Вместе с расширением оператора проводится детализация списка функций и его перераспределения. Поэтому и множество известных решений, и эвристический поиск проводится, исходя из списка реализуемых функций, так как он определяет особенности объектов начального морфизма.

4. Детализированный морфизм заменяется категорной диаграммой, и уточняются детали объектов категорий и их отображений за счет нового распределения реализуемого списка функций.

5. Выполняется переход от категорной диаграммы к графовой форме записи начального детализированного оператора с учетом уточнения распределения функций на категорной диаграмме.

6. Выполняется поиск множества наименьшей внешней устойчивости $T_{\min}$.

7. На множестве $T_{\min}$ устанавливаются графовые отображения, сохраняющие общую (внешнюю) функцию, которая является основной для системы в целом. При этом каждая из вершин имеет оптимальное число входов, а в целом структура отличается в таких условиях минимальным числом вершин (блоков, субблоков, устройств и т.д.). Распределение функций при этом отличается близкой сложностью по всей совокупности вершин.

8. Происходит детализация отдельных вершин в соответствии с набором реализуемых функций, список которых относится к данной вершине, с помощью операции расширения. При этом снова происходит перераспределение списка функций каждой вершины между вершинами структуры расширения.

9. На совокупности расширенных вершин осуществляется поиск $T_{\min}$ для каждой расширенной структуры.

10. На множестве $T_{\min}$, как и ранее (шаг 7), устанавливаются графовые отображения, которые в целом сохраняют реализацию системы функций, присущую расширенной вершине. Снова возникает граф второго порядка по отношению к расширенной вершине, на котором заново перераспределена система вершинных функций. Как и ранее, распределение отличается оптимальностью по критерию сложности реализуемых функций, числу входов вершин и количеству вершин на структуре, подвергнутой преобразованиям.

Основной операцией алгоритма БФР является операция, в результате которой строится граф системы на множестве наименьшей внешней устойчивости. Необходимость построения такого графа для системы состоит в том, что преобразование исходной графовой структуры не должно изменить начального функционального назначения.

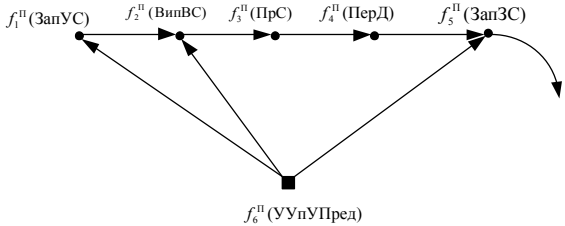
Суть БФР состоит в оптимальном размещении системной функции (совокупности функций) по предполагаемым подсистемам, блокам и т.д.

Определение списков функций на аппроксимирующем графе целесообразно осуществлять с помощью матрицы замещений. В такой матрице число столбцов равно числу вершин исходного графа до его преобразования; число строк равно числу вершин преобразованного графа.

Наименование вершин обеих структур дает название столбцам матрицы. На пересечении строки и столбца ставится 1, если аппроксимирующая вершина покрывает вершину исходного графа; ставится 0, если покрытия нет; размещается обозначение «1\Вх», если некоторая вершина является границей покрытия двумя вершинами и подвергается разрезанию. В связи с последним столбец матрицы обязательно должен содержать обозначения «1\Вх» и «1\Вых». При этом списки функций исходного графа, принадлежавшие аппроксимированным вершинам, переходят полностью в списки функций аппроксимирующих вершин, если на пересечении строки и столбца стоит 1. Если в строке аппроксимирующей вершины размещена информация «1\Вх» и «1\Вых», то к списку функций аппроксимирующей вершины следует отнести часть списка функций аппроксимирующей вершины. Другая часть списка относится к той вершине, строка которой содержит информацию «1\Вх» и «1\Вых» соответственно.

Формализация уровня представления

Составим граф схемы работы уровня в соответствии с описанием выше этого уровня:



Здесь приняты следующие обозначения: $f_1^{\text{П}}(\text{ЗапУС})$ – функция запроса на установление сеанса; $f_2^{\text{П}}(\text{ВипВС})$ – функция соглашения по выбору и повторному выбору синтаксиса; $f_3^{\text{П}}(\text{ПрС})$ – функ-

ция преобразования синтаксиса, включая преобразование данных, форматирование и специальные функции преобразования (сжатие); $f_4^{\text{П}}$(ПерД) – функция передачи данных; $f_5^{\text{П}}$(ЗапЗС) – функция запроса на завершение передачи данных; $f_6^{\text{П}}$(УпУПред) – функция блока управления.

Таким образом, процесс передачи информации на уровне представления можно описать так: блок управления переводит интерфейс в состояние готовности, источник передачи данных инициирует соединение путем запроса на установление сеанса (ЗапУС). Приемник, в случае согласия, пересылает положительный ответ (ВипВСУС), после чего обе стороны «договариваются» о синтаксисе передаваемых сообщений (ПрС). Согласование синтаксисов выполняется посредством связи между логическими объектами уровня представления по запросу прикладных логических объектов. Существует два варианта синтаксиса передачи данных (ПерД): синтаксис, используемый передатчиком, и синтаксис, используемый приемником. Два этих синтаксиса могут являться одним и тем же. Уровень представления включает функции, необходимые для выполнения преобразования между ними. Используемые стандарты могут быть одними и теме же – тогда выполнение преобразований не нужно. После установления единого стандарта передачи данных происходит их передача. Окончив передачу данных, источник пересылает запрос на завершение сеанса (ЗапЗС). На графе рядом с именованными вершинами приведены обозначены их функции $f_i^{\text{П}}$.

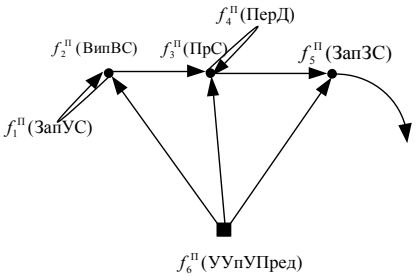
Для оптимизации структуры уровня представления по критерию однородности функциональной представимости осуществим поиск минимального внешне устойчивого множества графа. Построим таблицу замещения (табл. 3.1).

Таблица 3.1

Перераспределение функций на уровне представления

–	$f_1^{\text{П}}$	$f_2^{\text{П}}$	$f_3^{\text{П}}$	$f_4^{\text{П}}$	$f_5^{\text{П}}$	$f_6^{\text{П}}$
$(f_5^{\text{П}})'$					1	
$(f_3^{\text{П}})'$			1	1		
$(f_2^{\text{П}})'$	1			1		
$(f_6^{\text{П}})'$						1

Оптимизированный граф методом БФР имеет вид:



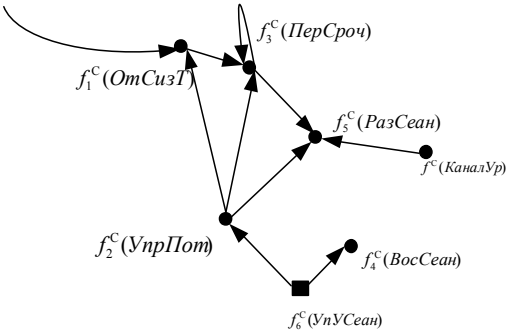
Тогда устойчивое множество оптимизированного графа имеет вид:

$$T_{min} = \{(f_2^{П})', (f_3^{П})', (f_5^{П})'(f_6^{П})'\}$$

Проведем аналогично оптимизацию оставшихся уровней системы представления OSI.

Формализация сеансового уровня

В соответствии с описанием функционирования интерфейса на этом уровне, составим графовую форму функционирования:



Здесь приняты следующие обозначения: f_1^{C}(ОтСизТ) – функция отображения сеансового соединения из транспортного соединения. В любое время между сеансовым и транспортным уровнем существует взаимно-однозначное отображение, но время их существования может отличаться; f_2^{C}(УпрПот) – функция управления потоком данных в сеансовом соединении. Для предотвращения переполнения данными принимающего логического объекта уровня

представления уровень сеансовый воздействует на транспортное соединение; $f_3^{\text{С}}$(ПерСроч) – функция передачи срочных данных; $f_4^{\text{С}}$(ВосСеан) – функция восстановления сеансового соединения; $f_5^{\text{С}}$(РазСеан) – функция разрыва сеансового соединения; $f_6^{\text{С}}$(УпУСеан) – функция административного управления сеансовым соединением. Защита от ошибок и активизация; f^{C}(КаналУр) – функция взаимодействия с канальным уровнем.

Устойчивое множество оптимизированного графа имеет вид:

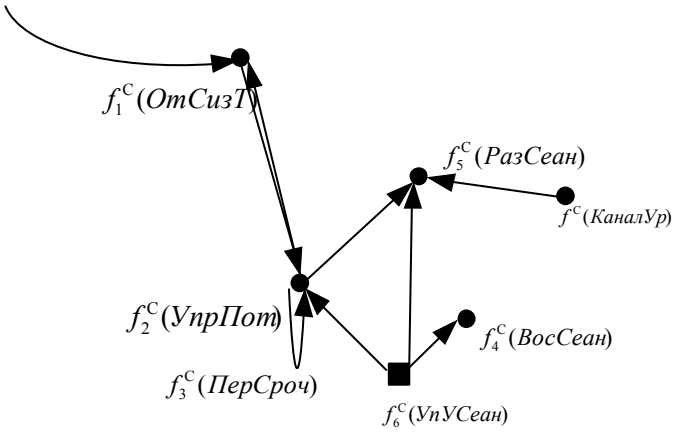
$$T_{min} = \{(f_1^{\text{С}}), (f_5^{\text{С}}), (f_2^{\text{С}})', (f_4^{C}), (f_6^{C}), (f^{C})\}$$

Таблица 3.2

Перераспределение функций на сеансовом уровне

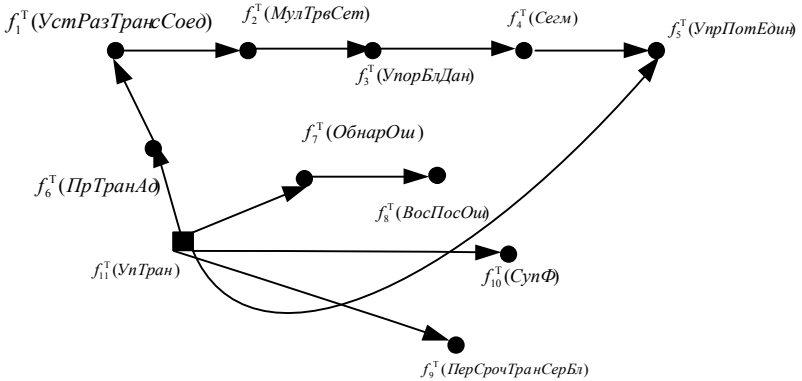
–	$f_1^{\text{С}}$	$f_3^{\text{С}}$	$f_2^{\text{С}}$	$f_5^{\text{С}}$	f^{C}	f_4^{C}	f_6^{C}
$f_1^{\text{С}}$	1						
$(f_2^{\text{С}})'$		1	1				
$f_5^{\text{С}}$				1			
f^{C}					1		
f_4^{C}						1	
f_6^{C}							1

Оптимизированный граф имеет вид:



Формализация транспортного уровня

Используя описание функционирования интерфейса на транспортном уровне, дадим его графовое представление:



Функциональное распределение по вершинам графа обозначено так: $f_1^{\text{Т}}$(УстРазТранСоед) – функция установления транспортных соединений; $f_2^{\text{Т}}$(ПрТранАд) – функция преобразования транспортного адреса в сетевой; $f_3^{\text{Т}}$(УпорБлДан) – функция упорядочивания блоков данных по отдельным соединениям; $f_4^{\text{Т}}$(Сегм) – функция сегментирования, сцепления; $f_5^{\text{Т}}$(УпрПотДанСоед) – функция управления потоком данных по отдельным соединениям; $f_6^{\text{Т}}$(МулТрСет) – функция мультиплексирования транспортных соединений в сетевой; $f_7^{\text{Т}}$(ОбнарОш)– функция обнаружения ошибок и необходимый контроль за качеством услуг; $f_8^{\text{Т}}$(ВосПосОш) – функция восстановления после ошибок; $f_9^{\text{Т}}$(ПерСрочТрансСерБл) – функция передачи срочных сервисных блоков данных; $f_{10}^{\text{Т}}$(СупФ) – супервизорные функции; $f_{11}^{\text{Т}}$(УпТран) – функция управления транспортным уровнем.

Выполним минимизирующее преобразование графа (2.5). Минимальное внешне устойчивое множество запишем следующим образом:

$$T_{min} = \{(f_1^T)', (f_3^T)', (f_7^T)', (f_9^T), (f_{10}^T)\}.$$

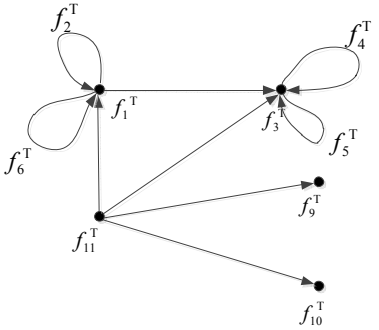
При этом таблица перераспределения функций имеет вид:

Таблица 3.3

Перераспределение функций на транспортном уровне

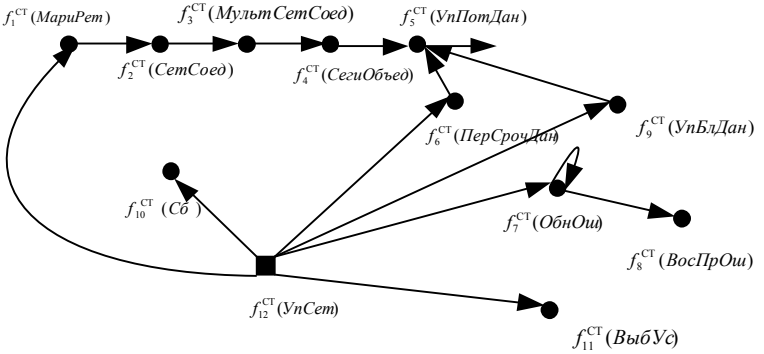
	f_1^T	f_2^T	f_3^T	f_4^T	f_5^T	f_6^T	f_7^T	f_8^T	f_9^T	f_{10}^T	f_{11}^T
$(f_1^T)'$	1	1				1					
$(f_3^T)'$			1	1	1						
$(f_7^T)'$							1	1			
f_9^T									1		
f_{10}^T										1	
f_{11}^T											1

После оптимизации получим следующий граф:



Сетевой уровень

На основании описания работы интерфейса в рамках сетевого уровня составим графовую модель функционирования:



Здесь приняты следующие обозначения: $f_1^{\text{Сет}}(\text{МарИРет})$ – функция маршрутизации и ретрансляции; $f_2^{\text{Сет}}(\text{СетСоед})$ – функция сетевого соединения; $f_3^{\text{Сет}}(\text{МультСетСоед})$ – функция мультиплексирования сетевого соединения; $f_4^{\text{Сет}}(\text{СегмОбъед})$ – функция сегментирования и объединения; $f_5^{\text{Сет}}(\text{ОбнОш})$ – функция обнаружение ошибок; $f_6^{\text{Сет}}(\text{ВосПрОш})$ – функция восстановления при ошибках; $f_7^{\text{Сет}}(\text{УпБлДан})$ – функция упорядочения блоков данных; $f_8^{\text{Сет}}(\text{УпПотДан})$ – функция управления потоком данных; $f_9^{\text{Сет}}(\text{ПерСрочДан})$ – функция передачи срочных данных; $f_{10}^{\text{Сет}}(\text{Сб})$ – функция сброса; $f_{11}^{\text{Сет}}(\text{ВыбУс})$ – функция выбора услуги; $f_{12}^{\text{Сет}}(\text{АдминУпрСетУр})$ – функция административного управления сетевым уровнем.

Осуществим оптимизационные преобразования графа.

В соответствии с этими преобразованиями множество наименьшей внешней устойчивости для графа имеет вид:

$$T_{min} = \{(f_1^{CT})', (f_4^{CT}), (f_5^{CT})', (f_7^{CT})', (f_{10}^{CT}), (f_{11}^{CT}), (f_{12}^{CT})\}.$$

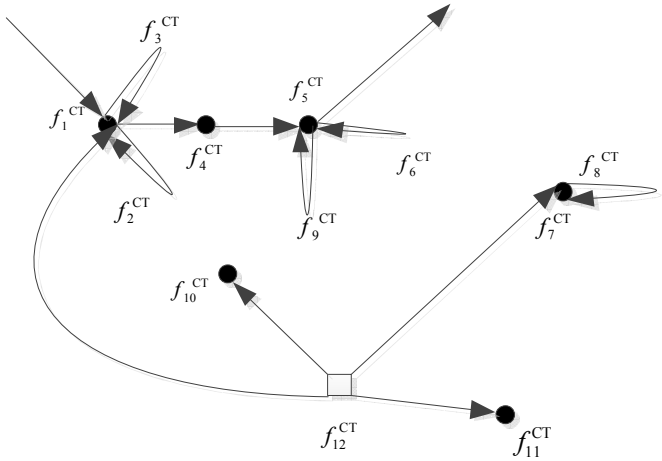
Тогда таблица замещения сетевого уровня имеет вид:

Таблица 3.4

Перераспределение функций на сетевом уровне

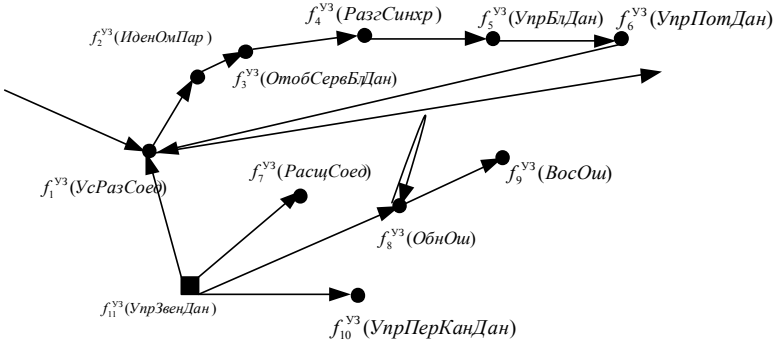
	f_1^{CT}	f_2^{CT}	f_3^{CT}	f_4^{CT}	f_5^{CT}	f_6^{CT}	f_7^{CT}	f_8^{CT}	f_9^{CT}	f_{10}^{CT}	f_{11}^{CT}	f_{12}^{CT}
$(f_1^{CT})'$	1	1	1									
(f_4^{CT})				1								
$(f_5^{CT})'$					1	1			1			
$(f_7^{CT})'$							1	1				
(f_{10}^{CT})										1		
(f_{11}^{CT})											1	
(f_{12}^{CT})												1

В результате получаем граф сетевого уровня с оптимизированной структурой:



Уровень звена данных

В соответствии с описанием функционирования интерфейса в рамках уровня звена данных, можно построить следующую графовую модель уровня:



Здесь приняты следующие обозначения: $f_1^{УЗ}(\text{УсРазСоед})$ – функция установления и разрыва соединения уровня звена данных; $f_2^{УЗ}(\text{ИденОбмПар})$ – функция идентификации и обмена параметрами; $f_3^{УЗ}(\text{ОтобСервБлДан})$ – функция отображения сервисных блоков данных уровня звена данных; $f_4^{УЗ}(\text{РазгСинхр})$ – функ-

ция разграничения и синхронизации; $f_5^{\text{УЗ}}$(УпБлДан) – функция упорядочивания блоков данных; $f_6^{\text{УЗ}}$(УпрПотДан) – функция управления потоком данных; $f_7^{\text{УЗ}}$(РасщСоед) – функция расщепления соединения уровня звена данных; $f_8^{\text{УЗ}}$(ОбнОш) – функция обнаружения ошибок; $f_9^{\text{УЗ}}$(ВосОш) – функция восстановления при ошибках; $f_{10}^{\text{УЗ}}$(УпрПерКанДан) – функция управления переключением каналов данных; $f_{11}^{\text{УЗ}}$(УпрЗвДан) – функция административного управления уровнем звена данных.

После оптимизирующих преобразований получаем множество наименьшей внешней устойчивости:

$$T_{min} = \{(f_1^{\text{УЗ}}), (f_4^{\text{УЗ}}), (f_7^{\text{УЗ}})(f_8^{\text{УЗ}})(f_{10}^{\text{УЗ}}), (f_{11}^{\text{УЗ}})\}.$$

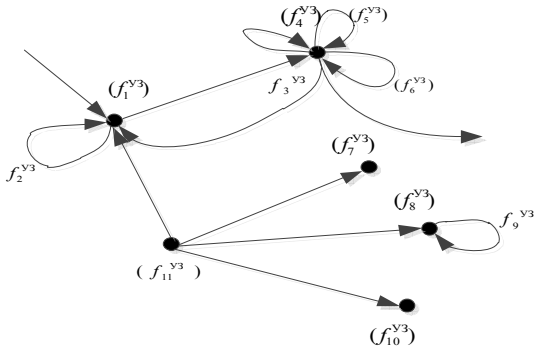
При этом таблица замещения уровня звена данных имеет вид (табл. 3.5):

Таблица 3.5

Перераспределение функций на уровне звена данных

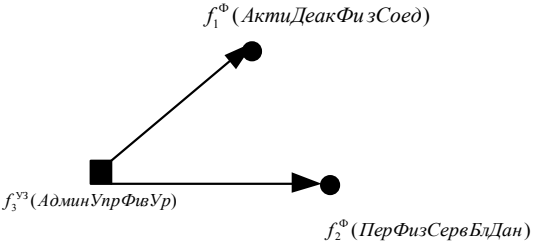
	$f_1^{\text{УЗ}}$	$f_2^{\text{УЗ}}$	$f_3^{\text{УЗ}}$	$f_4^{\text{УЗ}}$	$f_5^{\text{УЗ}}$	$f_6^{\text{УЗ}}$	$f_7^{\text{УЗ}}$	$f_8^{\text{УЗ}}$	$f_9^{\text{УЗ}}$	$f_{10}^{\text{УЗ}}$	$f_{11}^{\text{УЗ}}$
$(f_1^{\text{УЗ}})$	1	1									
$(f_4^{\text{УЗ}})$			1	1	1	1					
$(f_7^{\text{УЗ}})$							1				
$(f_8^{\text{УЗ}})$								1	1		
$(f_{10}^{\text{УЗ}})$										1	
$(f_{11}^{\text{УЗ}})$											1

На основании таблицы замещения строим оптимизированный граф распределения функций уровня звена данных:



Физический уровень

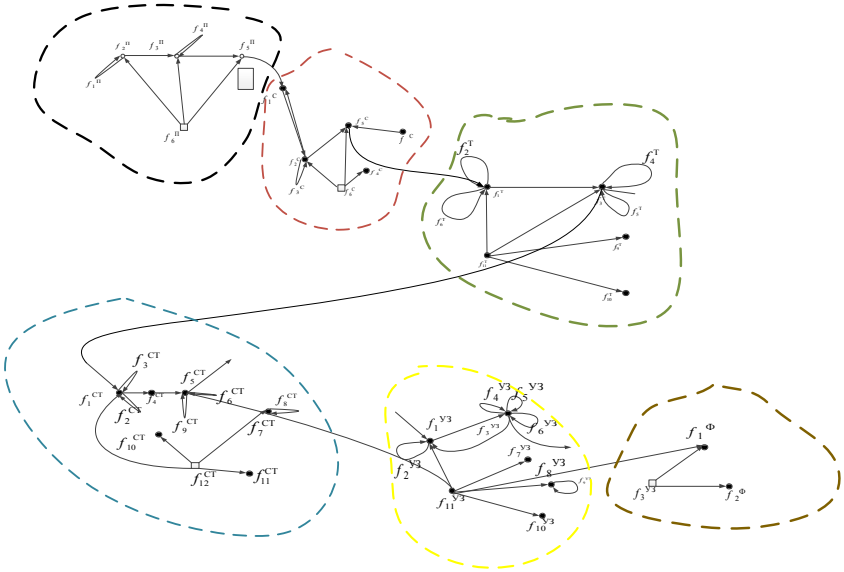
Граф распределения функций на физическом уровне интерфейса в соответствии с описанием имеет вид:



Здесь приняты следующие обозначения:

f_1^{Φ}(АктиДеакФизСоед) – функция активизации и деактивация физического соединения; f_2^{Φ}(ПерФизСервБлДан) – функция передачи физических сервисных блоков данных; f_3^{Φ}(АдмУпрФиз) – функция административного управления физическим уровнем.

На основании полученных уровневых графов распределения функций интерфейса составляем его полный граф:



Таким образом, изображен оптимизированный граф системы OSI, в котором произведено перераспределение функций. Семиуровневая система OSI имеет ряд функций каждого уровня, утвержденных в ГОСТ, однако оптимизация данных уровней и перераспределение функций с сохранением прежнего функционала позволяет сократить время обработки информации и привести ее к нужному виду, а также структурировать предметную область разработки и четко выделить функции ГИИ.

Требуется учесть, что проблематика затрагивает невозможность взаимодействия систем на уровне представления: информация от различных источников не воспринимаема приемниками из-за формы ее представления. Нижележащие уровни системы OSI используют стандартные стеки протоколов, такие как TCP/IP, что в условиях телемедицины, на данный момент, не вносит проблем во взаимодействие источника и приемника информации. Исходя из этого, можно утверждать, что ГИИ для телемедицинских систем необходимо разрабатывать, решая, прежде всего, проблемы нестыковки медицинской техники на уровне представления. При этом подход, который будет разработан для решения этой проблемы, универсален, что дает возможность применения его как на всех уровнях системы OSI, так и при решении задач, не связанных с телемедициной.

3.3. Алгоритм распознавания систем кодирования

Для реализации ГИИ необходимо решить задачу классификации входящего сообщения по системе кодирования информации. Для этого целесообразно воспользоваться теорией распознавания образов. Отличие образов устанавливается в рамках их классификации. Классификация основывается на прецедентах, при этом прецедент – это ранее классифицированный объект, принимаемый как образец, при решении задач классификации. Классификатором, или решающим правилом, называется правило отнесения образа к одному из классов на основании его вектора признаков. Здесь признак – это некоторый количественный результат измере-

ния параметра объекта произвольной природы. Совокупность признаков, относящихся к одному образу, называется вектором признаков. Векторы признаков принимают значения в пространстве признаков. В рамках задачи распознавания считается, что каждому образу ставится в соответствие единственное значение вектора признаков и наоборот: каждому значению вектора признаков соответствует единственный образ.

В зависимости от наличия или отсутствия прецедентной информации различают задачи распознавания с обучением и без обучения. Задача с обучением заключается в распознавании образов на основе имеющегося множества прецедентов. Если имеется множество векторов признаков, полученных для некоторого набора образов, но правильная классификация этих образов неизвестна, возникает задача разделения этих образов на классы по сходству соответствующих векторов признаков. Эта задача называется кластеризацией или распознаванием без обучения.

Учитывая специфику задачи классификации, решаемой ГИИ, заключающуюся в необходимости внесения дополнительных признаков, и наличие гибкости структуры системы, рассмотрим возможные варианты построения анализа входной информации и выберем наиболее подходящий.

Реализация процесса распознавания образов состоит в построении и применении формальных операций над числовыми или символьными отображениями объектов реального или идеального мира, результаты решения которых отражают отношения эквивалентности между этими объектами. При этом отношения эквивалентности выражают принадлежность оцениваемых объектов к каким-либо классам, рассматриваемым как самостоятельные семантические единицы. При «распознавании с учителем» классы эквивалентности могут задаваться исследователем, который пользуется собственными содержательными представлениями или использует внешнюю дополнительную информацию о сходстве и различии объектов в контексте решаемой задачи. При «распознавании без учителя» автоматизированная система решает задачу классификации без привлечения внешней обучающей информации. Для реализации большинства алгоритмов распознавания об-

разов требуется привлечение весьма значительных вычислительных мощностей, которые могут быть обеспечены только высокопроизводительной компьютерной техникой. Для выбора необходимой методики распознавания проведем сравнительный анализ методов, лежащих в их основе.

Методики распознавания образов: преимущества и недостатки

Основная типология методов распознавания образов:

– методы, основанные на принципе разделения;

– статистические методы;

– методы, построенные на основе «потенциальных функций»;

– методы вычисления оценок (голосования);

– методы, основанные на исчислении высказываний, в частности на аппарате алгебры логики.

В основе этой типологии использовано различие в формальных методах распознавания образов и поэтому опущен эвристический подход к распознаванию. Он основан на трудно формализуемых знаниях и интуиции исследователя. При этом исследователь сам определяет, какую информацию и каким образом система должна использовать для достижения требуемого эффекта распознавания.

Подобная типология методов распознавания с той или иной степенью детализации встречается во многих работах по распознаванию. В то же время известные типологии не учитывают очень существенную характеристику, которая отражает специфику способа представления знаний о предметной области с помощью какого-либо формального алгоритма распознавания образов.

В целом выделяют два основных способа представления знаний:

– интенсиональное – в виде схемы связей между атрибутами (признаками);

– экстенсиональное – с помощью конкретных фактов (объекты, примеры).

Интенсиональные представления фиксируют закономерности и связи, которыми объясняется структура данных. Они реализуются посредством операций над значениями атрибутов и не предпола-

гают произведения операций над конкретными информационными фактами (объектами).

В свою очередь, экстенсиональные представления знаний связаны с описанием и фиксацией конкретных объектов из предметной области и реализуются в операциях, элементами которых служат объекты как целостные системы.

В соответствии с этими двумя фундаментальными способами представления знаний можно определить следующую классификацию методов распознавания образов:

– интенсиональные, основанные на операциях с признаками;

– экстенсиональные, основанные на операциях с объектами.

Необходимо особо подчеркнуть, что существования именно этих двух (и только двух) групп методов распознавания, оперирующих с признаками и оперирующих с объектами, недостаточно. С этой точки зрения ни один из этих методов, взятый отдельно от другого, не позволяет сформировать адекватное отражение предметной области. Поэтому между методами существует отношение дополнительности, когда перспективные системы распознавания должны обеспечивать реализацию обоих этих методов, а не только какого-либо одного из них.

Сравнительная характеристика методов приведена в табл. 3.6.

Для решения задачи в рамках построения ГИИ требуется определять систему кодирования входного сообщения по множеству признаков. Следовательно, метод, выбираемый для дальнейшей работы, должен быть интенсиональный. Кроме того, метод должен обладать малой сложностью и высокой ясностью алгоритма. Методы, основанные на предположениях о классе решающих функций, полностью подходят для решения задачи построения ГИИ. Одним из самых разработанных алгоритмов данного класса методов является персептрон. Кроме того, совокупность персептронов позволит легко увеличивать объем решаемой задачи и перестраивать решающее правило. Таким образом, из обзора методов распознавания следует, что в рассматриваемой проблематике наиболее предпочтителен метод персептрона.

Таблица 3.6

Сводная таблица классификации методов распознавания, сравнения их областей применения и ограничений

Классификация методов распознавания		Область применения	Ограничения (недостатки)
Интенсиальные методы распознавания	Методы, основанные на оценках плотностей распределения значений признаков (или сходства и различия объектов)	Задачи с известным распределением, как правило, нормальным, необходимость набора большой статистики	Необходимость перебора всей обучающей выборки при распознавании, высокая чувствительность к непредставительности обучающей выборки и артефактам
	Методы, основанные на предположениях о классе решающих функций	Классы должны быть хорошо разделяемыми, система признаков – ортонормированной	Должен быть заранее известен вид решающей функции. Невозможность учета новых знаний о корреляциях между признаками
	Логические методы	Задачи небольшой размерности пространства признаков	При отборе логических решающих правил (коньюнкций) необходим полный перебор. Высокая вычислительная трудоемкость
	Лингвистические (структурные) методы	Задачи небольшой размерности пространства признаков	Задача восстановления (определения) грамматики по некоторому множеству высказываний (описаний объектов), является трудно формализуемой. Нерешенность теоретических проблем
Экстенсиальные методы распознавания	Метод сравнения с прототипом	Задачи небольшой размерности пространства признаков	Высокая зависимость результатов классификации от меры расстояния (метрики). Неизвестность оптимальной метрики
	Метод k ближайших соседей	Задачи небольшой размерности по количеству классов и признаков	Высокая зависимость результатов классификации от меры расстояния (метрики). Необходимость полного перебора обучающей выбор-

Классификация методов распознавания		Область применения	Ограничения (недостатки)
			ки при распознавании. Вычислительная трудоемкость
	Алгоритмы вычисления оценок (голосования) АВО	Задачи небольшой размерности по количеству классов и признаков	Зависимость результатов классификации от меры расстояния (метрики). Необходимость полного перебора обучающей выборки при распознавании. Высокая техническая сложность метода
	Коллективы решающих правил (КРП)	Задачи небольшой размерности по количеству классов и признаков	Очень высокая техническая сложность метода, нерешенность ряда теоретических проблем, как при определении областей компетенции частных методов, так и в самих частных методах

В соответствии с осуществленным выбором метода персептрона для эффективного его синтеза следует определить вид пороговой функции. С этой целью введем понятия:

1) Поле – единица информационного слова. $X = \{x_1, x_2, \dots, x_n\}$.

2) Признак – уникальная совокупность полей, отличающая данный формат от других форматов. $P = \{p_1, p_2, \dots, p_m\}$.

3) Формат – спецификация структуры данных, записанных в компьютерном файле. $S = \{s_1, s_2, , \dots, s_j\}$.

4) Пороговая функция – функция, задающая значение, необходимое для активации нейрона.

Рассмотрим следующий пример. Пусть в базе знаний существует множество форматов $S = \{s_1, s_2, , \dots, s_j\}$. Определим множество признаков форматов:

$$P = \{p_{11}, p_{21}, , \dots, p_{m1,} p_{12}, p_{22}, , \dots, p_{m2,} p_{1n}, p_{2n}, , \dots, p_{mn,}\},$$

где $s_1 \subseteq \{p_{11}, p_{21}, , \dots, p_{m1,}\}$; $s_2 \subseteq \{p_{12}, p_{22}, , \dots, p_{m2,}\}$; $s_j \subseteq \{p_{1n}, p_{2n}, , \dots, p_{mn,}\}$.

Пусть на вход системы, включающей двухуровневый персептрон (рис. 3.2), приходит информационное слово размером N бит. Каждому биту соответствует вход из множества

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, \dots, x_n\} -$$

элемента входного слоя. Скрытый слой P – это множество признаков. Каждый нейрон p проверяет пороговой функцией f_p наличие и содержимое соответствующего подмножества x_n, характеризующего конкретный признак p [52, 53]. Опишем функционирование нейрона:

$$NET = \sum_1^i w * x,$$

где NET – взвешенная сумма входных сигналов, значение передается на нелинейный элемент, w – вес ребра перехода, x – элемент входного слоя;

$$\text{OUT}=f_{p_i}(NET - \theta),$$

где OUT – выходная функция, θ – пороговый уровень нейрона, f_{p_i} – функция активации.

Функция активации имеет вид:

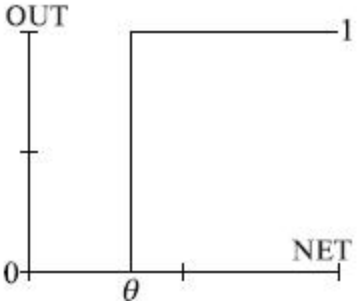


Рис. 3.2. Функция активации нейрона:

$$OUT = \begin{cases} 0, \text{если } NET < \theta \\ 1, \text{если } NET > \theta \end{cases}$$

Скрытый слой S – множество форматов. Каждый нейрон s проверяет наличие и содержимое соответствующего подмножество p_m пороговой функцией f_s.

$$NET = \sum_1^i w * p,$$

где NET – взвешенная сумма входных сигналов, значение передается на нелинейный элемент, w – вес ребра перехода, x – элемент входного слоя;

$$\text{OUT}=f_{s_i}(NET-\theta),$$

где θ – пороговый уровень нейрона, f_{s_i} – функция активации.

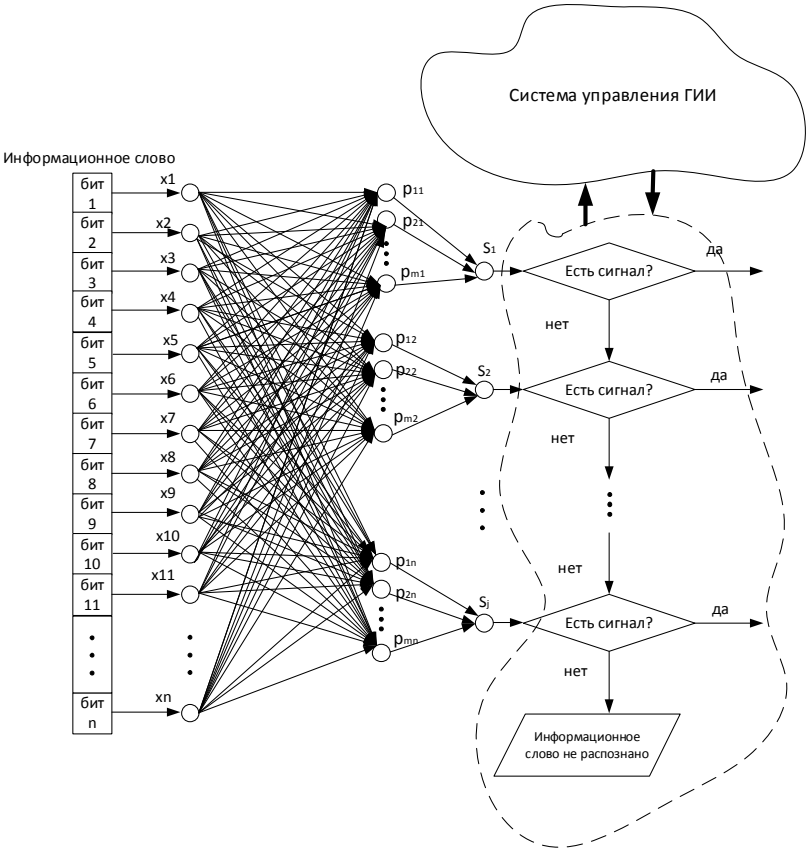


Рис. 3.3. Система распознавания ГИИ

Каскад нейронов позволяет оперативно увеличивать размерность системы анализа в зависимости от потребностей приемника и пере-

датчика, а также корректировать в случае надобности не всю сеть, а лишь отдельный элемент путем удаления или добавления элемента *p*.

Элементы приведенной структуры персептрона определены следующим образом: *S*-элемент – сенсорный элемент, воспринимающий воздействия внешней среды. В нашем случае – это входы нейронной сети [59, 60], воспринимающие цепочку входных символов. S-элемент представлен множеством $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, \dots, x_n\}$ на рис. 3.3.

A-элемент – генератор сигнала с входным и выходными сигналами, на вход которых поступает совокупность сигналов c_j, а на выход $a_j(H)$. В рассматриваемом примере *A*-элемент – это множество

$$P = \{p_{11}, p_{21}, , \dots, p_{m1,} p_{12}, p_{22}, , \dots, p_{m2,} p_{1n}, p_{2n}, , \dots, p_{mn,}\}$$

на рис. 3.1.

R-элемент – имеющий входные связи и выдающий сигнал, поступающий во внешнюю среду. В рассматриваемом примере *R*-элемент – это множество $S = \{s_1, s_2, , \dots, s_j\}$ на рис. 3.1.

$C_j(t)$ – кодовое слово, состоящее из последовательности «0» и «1», рассматриваемых, как признак системы кодирования поступившего сообщения. В рассматриваемом примере это информационное слово.

W_i – множество комбинаций кодовых признаков систем кодирования.

Реакция – *n*-мерный вектор *R*, компонентами которого являются значения $R(w)$ реакции нейронной сети на действия стимулов *S* – признаков систем кодирования.

Классификация – выделение подмножеств реакций, состоящая из однотипных входных последовательностей.

Пространство последовательностей стимулов – совокупность сигналов, поступающих в рамках сеанса связи ГИИ с источником сигнала.

Тогда система подкреплений – метод обучения, при котором величина сигнала подкрепления равна «0», пока текущая реакция нейронной сети верна. При появлении ошибки знак сигнала определяется знаком ошибки.

G-матрица имеет следующий вид:

$$G = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{pmatrix},$$

где n – число стимулов (величина обучаемой выборки, число примеров для запоминания); g_{ij} – коэффициенты обобщения.

Теперь, опираясь на теоремы Розенблатта [24] и используя введенные обозначения, определим необходимые и достаточные условия существования решений при классификации метода кодирования с помощью ГИИ. Приведем теоремы существования решений в задачах классификации.

Теорема 1.

1) Каждый стимул должен возбуждать, по крайней мере, один *A*-элемент.

2) Не должно существовать никакой подпоследовательности стимулов, содержащей, по меньшей мере, по одному стимулу каждого класса, которая приводила бы к одинаковому коэффициенту смещения для каждого *A*-элемента в множестве *A*-элементов, реагирующих на эту подпоследовательность.

Таким образом, для случая работы ГИИ:

1) Для каждой входной кодовой последовательности, поступающей на приемник, должен существовать хотя бы один *A*-элемент, возбуждаемый последовательностью. То есть, каждая входная кодовая последовательность обязана воздействовать на систему.

2) Не должно существовать входных кодовых последовательностей, приводящих к возбуждению одних и тех же *A*-элементов, что вызывает неоднозначность классификации разных классов стимулов.

Теорема 2.

Даны: элементарный персептрон, пространство стимулов W и какая-то классификация $C(W)$. Тогда для существования решения для $C(W)$ необходимо и достаточно, чтобы существовал некоторый вектор u, лежащий в том же ортанте, что и $C(W)$, и некоторый вектор x, такой, что $G_x=u$.

В нашем случае эта теорема полностью применима для системы ГИИ. Из этой теоремы Розенблатт выводит следующие заключения:

1) Если *G*-матрица персептрона особенная, то есть матрица, не имеющая обратной, то может существовать некоторая классификация, которая не имеет решения. В этом случае сходимости при обучении персептрона не будет.

2) Если число стимулов в обучающей выборке больше числа *A*-элементов в элементарном персептроне, то также существует некоторая классификация, которая не имеет решения. То есть, количество элементов второго слоя должно быть не меньше количества входных кодовых последовательностей, на которых происходит обучение.

Опираясь на рассмотренные теоремы и их следствия, возможно установить требования к интеллектуальной части ГИИ, позволяющей классифицировать поступающую информацию. Сформируем теорему сходимости для системы ГИИ, опираясь на теорему сходимости персептрона и признаки существования решений.

Даны: элементарный персептрон, пространство стимулов источника $W_{\text{ист}}$ и пространство стимулов приемника $W_{\text{пр}}$, некоторая классификация $C(W)$, содержащая варианты представления информации медицинской техникой, для которой известно, что решение $C_{\text{ист}}$ и $C_{\text{пр}}$ существует. Учитывая [21], сформулируем теорему сходимости процесса идентификации.

Теорема 3.

Предположим, что все стимулы из $W_{\text{ист}}$ и $W_{\text{пр}}$ появляются в любой последовательности, но при условии, что каждый стимул появляется повторно через некоторый конечный интервал времени. Тогда процесс обучения с коррекцией ошибок (с квантованием или без квантования подкрепления), начинающийся с произвольного исходного состояния, всегда приведет к достижению решения для $C(W)$ в течение конечного промежутка времени.

Таким образом, рассмотренные теоремы существования решения для систем нейронных сетей применимы для ГИИ и полностью задают условия существования решений классификации. Эти условия позволяют определить требования к системе классифика-

ции в ГИИ, что дает возможность выбрать верную архитектуру этой части интерфейса.

3.4. Алгоритм преобразования систем кодирования

Как следует из постановки задачи, ГИИ позволяет: выполнять автоматическую настройку; трансформировать передаваемые данные, ориентируясь на существующие стандарты; приводить передаваемые информационные пакеты к виду, идентифицируемому как источником, так и приемником сигнала.

Поэтому для эффективной работы ГИИ целесообразно учитывать специфику взаимодействия устройств в современных сетях передачи данных, опирающуюся на модель взаимодействия открытых систем – OSI.

В этом случае процесс работы ГИИ можно представить блок-схемой, изображенной на рис. 3.4.

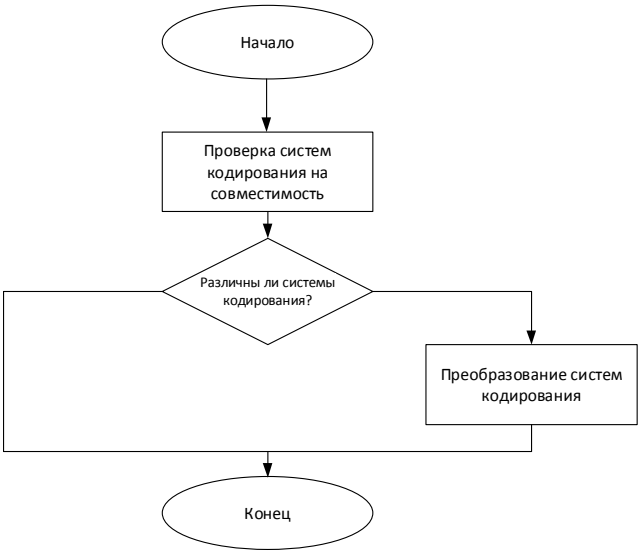


Рис. 3.4. Блок-схема процесса определения и преобразования систем кодирования ГИИ

Выделены этапы работы алгоритма функционирования ГИИ:

1) Этап проверки систем представления информации на совместимость.

На этом этапе ГИИ анализирует при помощи тестовых пакетов, отсылаемых и приемнику, и передатчику системы кодирования информации. После получения тестовой информации происходит сравнение откликов приемника и передатчика и выполняется переход к этапу 2, если адресат и адресант сообщения имеют разные системы представления, в ином случае – окончание работы.

2) Этап преобразования систем кодирования.

В случае различия систем представления информации на передатчике и приемнике система преобразований ГИИ формирует последовательность действий, необходимых для выполнения преобразования информации, и реализует данные преобразования.

Опишем каждый из этапов подробнее.

Проверка систем кодирования на совместимость

На данном шаге выделим действия по анализу формата входного слова и анализу формата на выходе. Для этого требуется сравнить признаки каждого из форматов в базе знаний. Структура подсистемы ГИИ имеет вид, представленный на рис. 3.5.

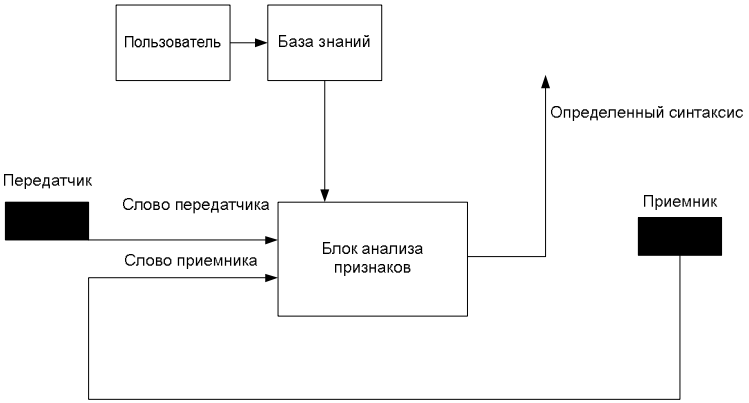


Рис. 3.5. Система анализа пришедшего слова

В соответствии с этой структурой (Рисунок 2.5), последовательность действий, состоит в следующем:

1) Идентифицируется система кодирования по слову передатчика в блоке $F_{\text{зпе}}$.

2) Формируется запрос ГИИ по выбору тестового слова от приемника $F_{\text{зпр}}$.

3) Идентифицируется система кодирования приемника по пришедшему тестовому слову $F_{\text{опр}}$.

4) Осуществляется сравнение систем кодирования: в случае, если системы кодирования совпадают, выполняется переход к $F_{\text{срав}}$, а если нет, выполнение преобразований не требуется.

Таким образом, выполнение всех функций первого шага алгоритма функционирования ГИИ можно записать как:

$$F_{\text{осп}} = F_{\text{срав}}(F_{\text{опр}}(F_{\text{зпр}}(F_{\text{зпе}}(.))))$$

Этап преобразования систем кодирования

Определение последовательности преобразований

Процесс преобразования систем кодирования можно проиллюстрировать с помощью блок-схемы (рис. 3.6), в соответствии с которой последовательность преобразований состоит в следующем.

Блок анализа входного слова осуществляет идентификацию пришедшей информации и определяет систему представления. При этом входное слово имеет некоторую определенную структуру. Она представляет собой совокупность типовых компонентов (следования, циклов, ветвлений), количество и порядок использования которых может меняться, образуя путь на структуре слова. Целесообразной моделью такой структуры является взвешенный граф вида (2.14). Далее блок поиска пути преобразований осуществляет выбор оптимального пути с наименьшей «стоимостью». Стоимости всех путей определены заранее в блоке анализа в зависимости от сложности преобразований.

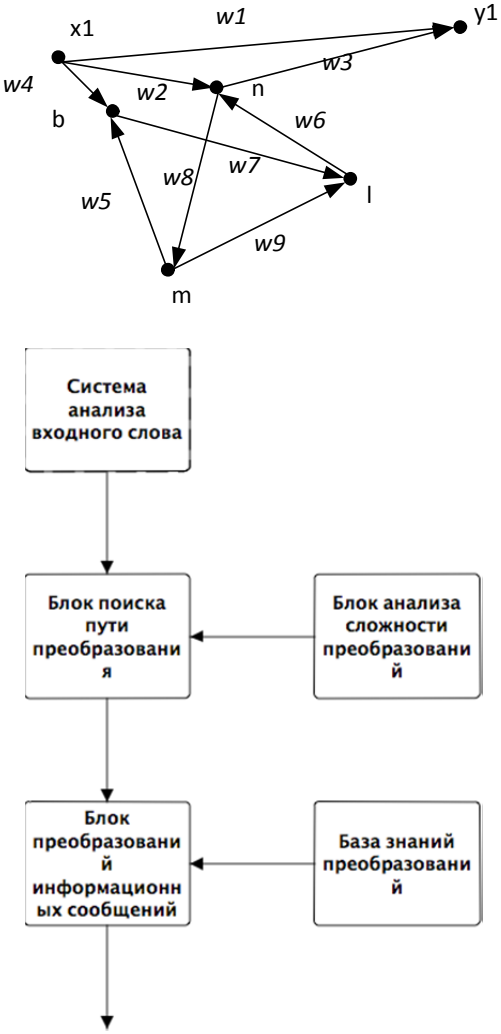


Рис. 3.6. Блок-схема преобразования пришедшего слова

Такая модель позволяет учесть, что эти преобразования должны быть эффективными, то есть в данном случае – наименее сложны-

ми. При выборе конкретного алгоритма преобразований происходит оценка всех возможных путей преобразований и выбирается оптимальный. Для этого предлагается использовать алгоритм Дейкстры [25]. Он позволяет находить кратчайшее расстояние от одной из вершин графа, характеризующих первоначальную форму кодирования, до всех остальных вершин (вариантов кодирования приемника). Каждая возможная форма кодирования слова представляет собой вершину из множества {x1, m, b, n, l, y1}. Ребра, связывающие вершины, – это необходимые преобразования для перевода из одного кодового состояния в другое анализируемого информационного пакета. При этом каждое преобразование имеет свою сложность, а оценка сложности происходит по каждому переходу и учитывается для выбора варианта дальнейших расчетов.

По определенной выбранной последовательности переходов блок преобразований информационных сообщений выполняет пошагово каждое из преобразований, хранимых в базе знаний.

Сложность алгоритма преобразования

Дадим определение сложности алгоритма. Пусть на возможных входах x алгоритма A определена неотрицательная числовая функция $\|x\|$ (размер входа). Пусть также определены целочисленные неотрицательные функции ${C_A}^T(x)$, ${C_A}^S(x)$ временных и пространственных затрат алгоритма A. Тогда временной и пространственной сложностями A называются функции числового аргумента $T_A(n) = \max {C_A}^T(x)$, $S_A(n) = \max {C_A}^S(x)$.

Для получения функции сложности алгоритма, представленного в формальной системе введенной абстрактной машины, необходимо уточнить понятия «элементарных» операций, соотнесенных с языком высокого уровня. В качестве таких «элементарных» операций предлагается использовать следующие:

1) простое присваивание: $a \leftarrow b$;

2) одномерная индексация $a[i]$: (адрес (a)+i*длина элемента);

3) арифметические операции: (*, /, -, +);

4) операции сравнения: $a < b$;

5) логические операции ($l1$) {or, and, not} ($l2$).

После введения элементарных операций анализ трудоемкости основных алгоритмических конструкций в общем виде сводится к следующим положениям:

1. Конструкция «Следование». Трудоемкость конструкции есть сумма трудоемкостей блоков, следующих друг за другом (k – количество блоков).

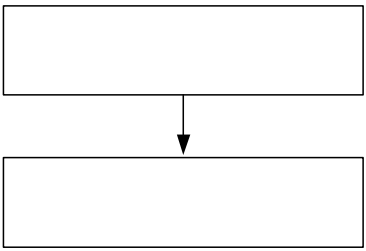


Рис. 3.7. Конструкция «Следование»

$$F_{\text{следование}} = f_1 + \cdots + f_k$$

2.Конструкция «Ветвление».

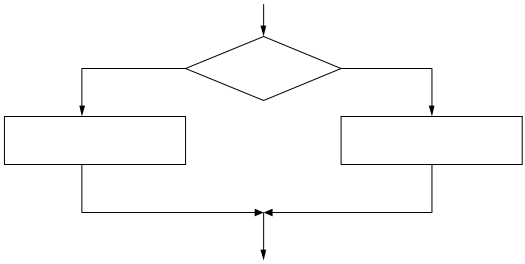


Рис. 3.8. Конструкция «Ветвление»

Общая трудоемкость конструкции «Ветвление» требует анализа вероятности выполнения переходов на блоки «Then» и «Else» и определяется как:

$$F_{\text{ветвление}} = f_{then} * p + f_{else} * (1 - p)$$

3. Конструкция «Цикл».

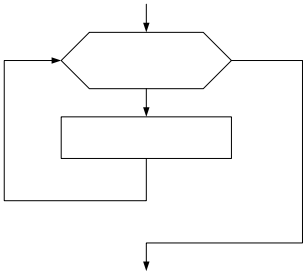


Рис. 3.9. Конструкция «Цикл»

После сведения конструкции к элементарным операциям ее трудоемкость определяется как:

$$F_{\text{цикл}} = 1 + 3 * N + N * f_{\text{тела цикла}}$$

Поиск оптимального преобразования

Дав оценку каждому преобразованию и присвоив ребрам графа соответствующие значения, перейдем к выполнению преобразований, необходимых для трансформации информации приемника $x1$ к виду передатчика $y1$.

Для этого в графе при помощи алгоритма Дейкстры выполним поиск оптимального пути преобразования от вершины $x1$ к вершине $y1$.

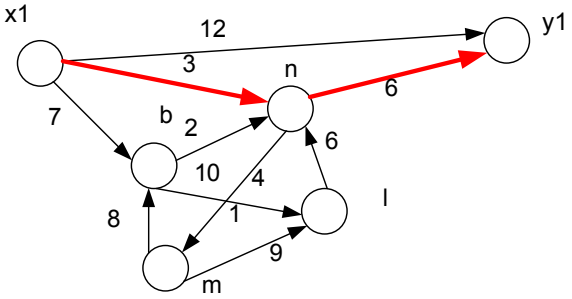
Выполнение преобразование систем кодирования

Путь от вершины x до вершины y является оптимальным по маршруту $x1$->n->y (по оценке алгоритма Дейкстры).

Тогда действия, соответствующие данному пути, будут:

$$\{x1 \rightarrow n,\ n \rightarrow y1\}.$$

После нахождения оптимального пути требуется выполнить соответствующие ему преобразования. Они должны быть внесены заранее в базу знаний преобразований, хранящую все возможные варианты трансформации первоначальной информации к заранее известным формам кодирования.



Таким образом, на втором этапе необходимо выполнить следующие действия:

1) Поиск возможных преобразований по соответствию форматов – $F_{\text{поиск}}$;

2) Выбор оптимального решения при помощи алгоритм Дейкстры – $F_{\text{оптимпр}}$;

3) Выполнение преобразований $F_{\text{преоб}}$.

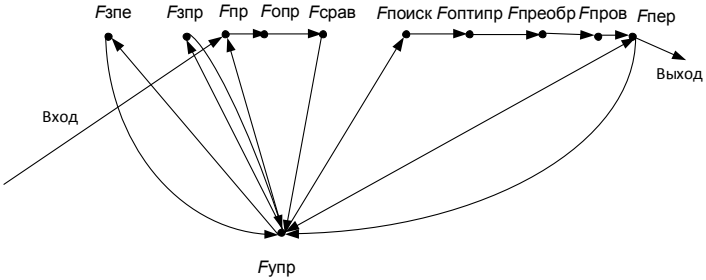
Тогда функция выполнения преобразований систем представления приемника и передатчика имеет вид:

$$F_{\text{пспп}} = (F_{\text{преоб}}(F_{\text{оптимпр}}(F_{\text{поиск}}(.)))).$$

4. СИНТЕЗ ГИБКОГО ИНТЕЛЛЕКТУАЛЬНОГО ПРЕОБРАЗОВАТЕЛЯ

4.1. Синтез структуры ГИИ

В соответствии со схемой алгоритма ГИИ и графовым представлением работы уровня представления системы OSI (2.1), рассмотрим граф, на котором распределены подфункции алгоритма и, используя метод БФР [15], произведем оптимизацию его структуры. Изначальный граф системы имеет вид

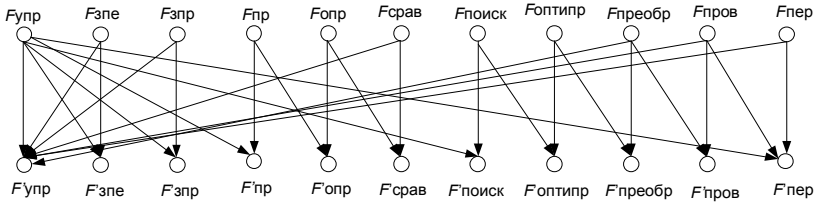


$F_{зпе}$ – запрос установления сеанса и систем кодирования источника;
$F_{зпр}$ – запрос установления сеанса и систем кодирования приемника;
$F_{пр}$ – выполнение приема информации; $F_{опр}$ – определение систем кодирования источника и приемника; $F_{срав}$ – сравнение систем кодирования;
$F_{поиск}$ – поиск алгоритма преобразований систем; $F_{оптипр}$ – выбор оптимального преобразования; $F_{преобр}$ – выполнение оптимального преобразования;
$F_{пров}$ – выполнение проверки информационного сообщения на наличие артефактов; $F_{пер}$ – выполнение передачи информации.

Граф содержит 8 вершин, соответствующих реализации функций ГИИ, описанных в разделе 2. Отдельно внесена вершина $F_{упр}$, соответствующая блоку управления процесса функционирования ГИИ. Блок управления выполняет запросы по функциям получения кодовых слов приемника и передатчика, сравнения систем кодирования, поиска преобразования и проверки информационного

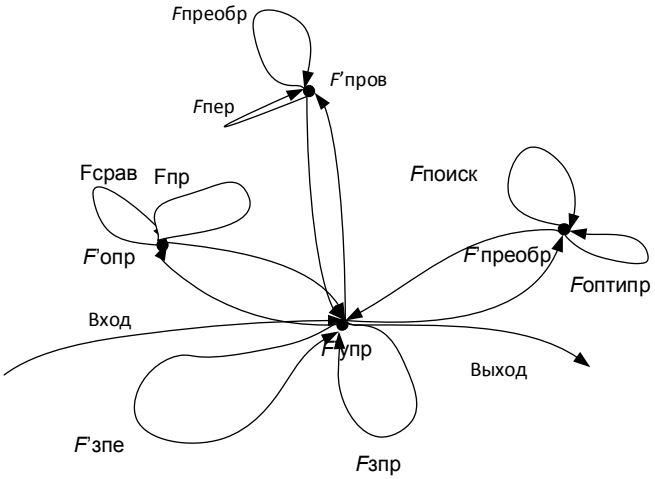
сообщения на наличие артефактов и их исправления. Кроме этого, блок управления принимает, а функции $F_{\text{зпе}}$, $F_{\text{зпр}}$, $F_{\text{срав}}$, $F_{\text{пер}}$ отсылают сигналы об окончании выполнения обозначенных функций.

Воспользуемся методом БФР для оптимизации работы ГИИ. На первом этапе построим двудольный граф систем.



Используя идеологию БФР, рассмотренную во втором разделе, и, опуская промежуточные преобразования графа, отнесем ко множеству минимальной (наименьшей) внешней устойчивости следующие вершины: $T_{min} = \{F'_{\text{упр}}, F'_{\text{опр}}, F'_{\text{преобр}}, F'_{\text{пров}}\}$.

Оптимизированный граф в таком случае будет имеет вид



Матрица замещений при этом выглядит следующим образом:

Таблица 4.1

Матрица финальных замещений

	$F_{упр}$	$F_{зпе}$	$F_{зпр}$	$F_{пр}$	$F_{срав}$	$F_{опр}$	$F_{поиск}$	$F_{оптипр}$	$F_{преобр}$	$F_{пров}$	$F_{пер}$
$F'_{упр}$	1	1	1								
$F'_{опр}$				1	1	1					
$F'_{преобр}$							1	1	1		
$F'_{пров}$										1	1

Таким образом, на основании графа и матрицы (табл. 4.1), блок-схема работы ГИИ может быть представлена на рис. 4.1.

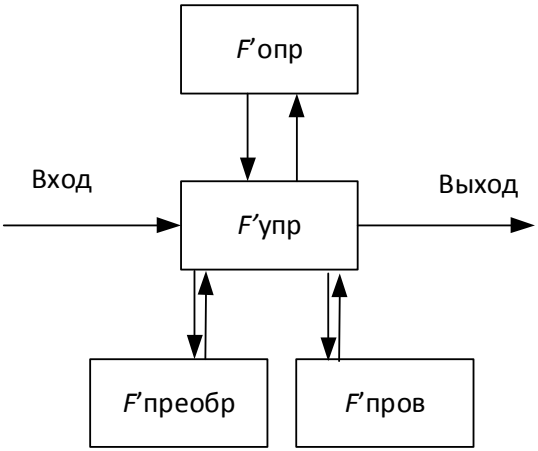


Рис. 4.1. Блок-схема ГИИ

Блок-схема работы состоит из четырех частей: блока определения системы кодирования ($F'_{опр}$), блока управления ($F'_{упр}$), блока преобразования ($F'_{преобр}$), блока проверки преобразованной информации ($F'_{пров}$). Все части обмениваются информацией с блоком управления. Кроме того, блок управления принимает входную информацию и передает преобразованную далее по линиям передачи данных. Опишем функции, исполняемые данными блоками. Блок $F'_{упр}$ организует «челночную» работу ГИИ:

1. Осуществляет прием и запись в буфер входного сообщения в соответствии с функцией $F_{зпр}$.

2. Осуществляет передачу входного сообщения в блок $F'_{опр}$ с целью оценки системы кодового представления входного сообщения путем реализации функции $F'_{упр}$.

3. Передает результаты $F_{ср}$ на блок $F'_{упр.}$

4. Включает блок $F'_{преобр}$ и осуществляется функцию $F_{поиск}$ необходимого результата преобразования. Осуществляет поиск последовательности оптимальных преобразований через $F_{опт}$ и реализует процесс преобразования принятого сообщения с помощью $F_{преобр}$.

4.2. Синтез алгоритма управления ГИИ

Для синтеза алгоритма управления ГИИ рассмотрим подробнее на примере процесс его работы. Предположим, что ГИИ принимает со стороны передатчика информационное слово в формате BMP – формате хранения растровых изображений, разработанный компанией Microsoft. Приемник же принимает информацию только в формате TIFF – формате хранения растровых графических изображений.

С форматом BMP работает большое число программ, так как его поддержка интегрирована в операционные системы Windows и OS/2. При этом файлы формата BMP могут иметь расширения .bmp, .dib и .rle.

Глубина цвета в данном формате представления изображения может составлять 1, 4, 8, 16, 24, 32, 48 бит на пиксель. При этом для глубины цвета меньше 16 бит/пиксель используется палитра с полноцветными компонентами глубиной 24 бита/пиксель. В формате BMP изображения могут храниться в исходном представлении или же с применением некоторых распространённых алгоритмов сжатия. В частности, формат BMP поддерживает RLE-сжатие без потери качества, а современные операционные системы и программное обеспечение позволяют использовать JPEG и PNG (эти форматы встраиваются в BMP как в контейнер).

Структура файлов BMP представляется следующим образом:

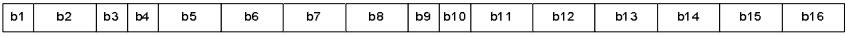


Рис. 4.2. Структура BMP файла

Здесь поля означают:
b1 – Код 4D42h – Буквы ‘BM’;
b2 – размер файла в байтах;
b3 – 0 (Резервное поле);
b4 – 0 (Резервное поле);
b5 – смещение, с которого начинается само изображение;
b6 – размер заголовка BITMAP (в байтах) равно 40;
b7 – ширина изображения в пикселях;
b8 – высота изображения в пикселях;
b9 – число плоскостей;
b10 – бит/пиксел: 1, 4, 8 или 24;
b11 – тип сжатия;
b12 – 0 или размер сжатого изображения в байтах;
b13 – горизонтальное разрешение, пиксел/м;
b14 – вертикальное разрешение, пиксел/м;
b15 – количество используемых цветов;
b16 – количество элементов палитры, необходимых для отображения изображения.

Выделим признаки информационного слова, наличие которых будет характеризовать формат файла BMP (рис. 4.3). Поля b1, b2, b3, b4 могут служить уникальным признаком. Размер, указанный в поле b2, подлежит сравнению с реальным размером файла.

Рис. 4.3. Признаки формата BMP

Перейдем к файлу TIFF.Общий вид файла можно представить так (рис. 4.4).

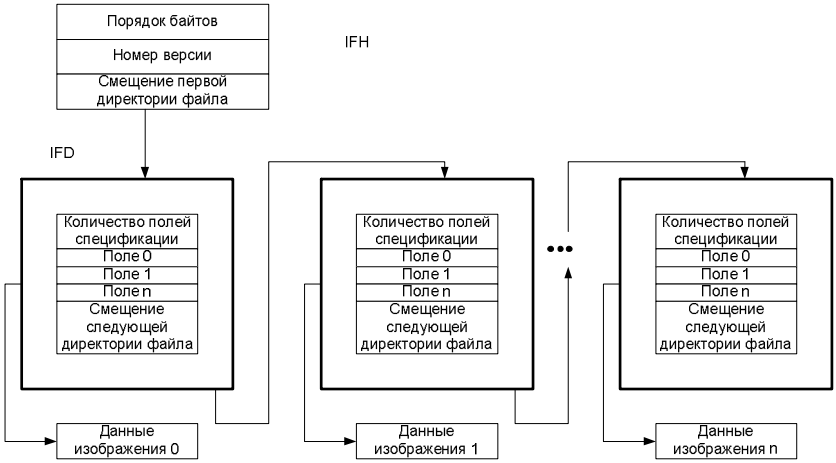


Рис. 4.4. Общая структура файла TIFF

TIFF является сложным форматом, поскольку местоположение каждой директории и ее данных (включая растровые данные) может изменяться. Единственной составной частью файла TIFF, которая имеет постоянное место, является заголовок – он всегда располагается в первых 8 байтах каждого файла TIFF. Все остальные данные файла создаются с использованием информации IFD. Директория файла изображения и связанный с ней растр составляют субфайл TIFF. Ограничений на количество субфайлов в файле TIFF не существует. Каждая директория файла содержит одну или несколько структур данных, называемых полями. Каждое поле является 12-байтовой записью, содержащей определенную информацию о растровых данных. Поле может хранить данные любого типа, и спецификация TIFF определяет свыше 70 типов полей, которые применятся для представления заданной информации. Поля каждой директории объединяются в непрерывные группы. Поля, определенные спецификацией TIFF, называются общедоступными и не могут изменяться в большей мере, чем это предусмотрено спецификацией. Поля, определенные пользователем, называются частными и предназначены для применения в пользовательских целях.

Каждая директория файла изображения связывает все растровые данные файла TIFF. Данные директории можно прочесть либо непосредственно из структуры IFD, либо найдя их по смещению, записанному в IFD. С помощью смещения в файле TIFF можно задать одно из трех перечисленных ниже местоположений. Смещение, указанное в последних четырех байтах заголовка, определяет позицию первой директории. Последние четыре байта каждой директории содержат информацию о смещении следующей директории, а последние четыре байта каждого тега – либо информацию о смещении данных, либо сами данные. Величина смещения всегда определяется количеством байтов от начала файла TIFF.

Опишем обобщенную структуру файла TIFF для дальнейшей работы (рис. 4.5).

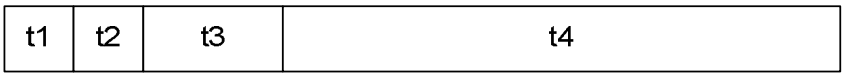


Рис. 4.5. Структура файла TIFF

Здесь поля означают:
t1 – определение порядка байтов;
t2 – идентификатор формата;
t3 – смещение в байтах от начала файла на директорию IFD;
t4 – IFD.
Поля 1 и 2 можно пометить как признаки (рис. 4.6).

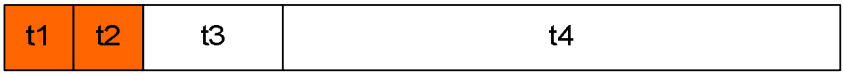


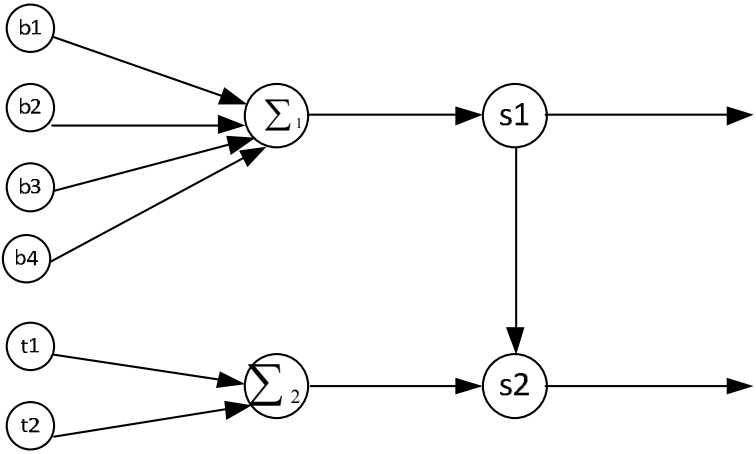
Рис. 4.6. Признаки формата TIFF

Множество {B}={b1,b2,b3,b4} – множество признаков формата BMP. Множество {T} ={t1,t2}– множество признаков формата TIFF.

Определив признаки, построим нейронную сеть с учетом того, что в базе знаний ГИИ указаны всего 2 формата: BMP и TIFF. Составим нейронную сеть, реализующую функцию распознавания пришедшей информации. Сеть будет состоять из двух персептро-

нов, каждый из которых научен распознавать один из форматов BMP или TIFF.

Тогда нейронная сеть, состоящая из двух персептронов, соединенных каскадом, примет вид:



где b1, b2, b3 – признаки формата bmp; а t1, t2 – признаки формата TIFF.

Пусть на вход анализирующего устройства поступает файл, закодированный в формате bmp.

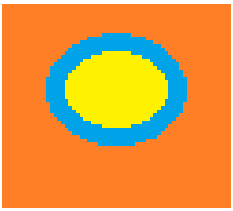
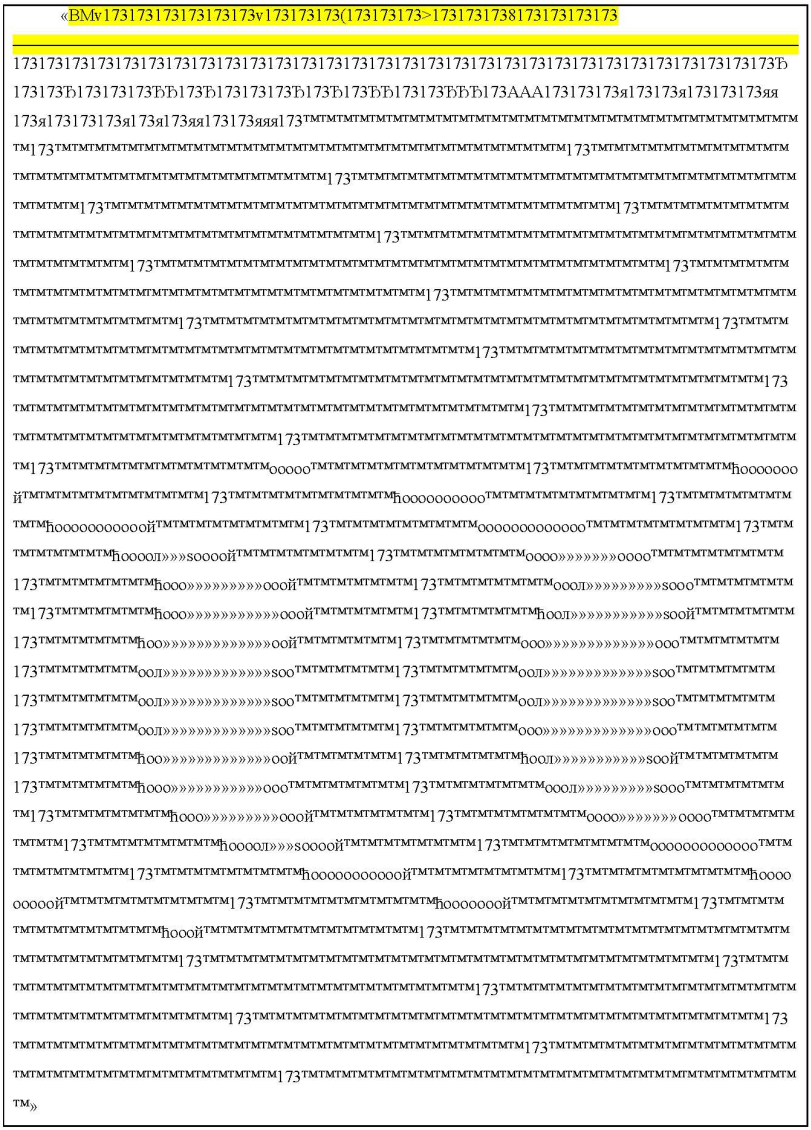


Рис. 4.7. Передаваемый файл

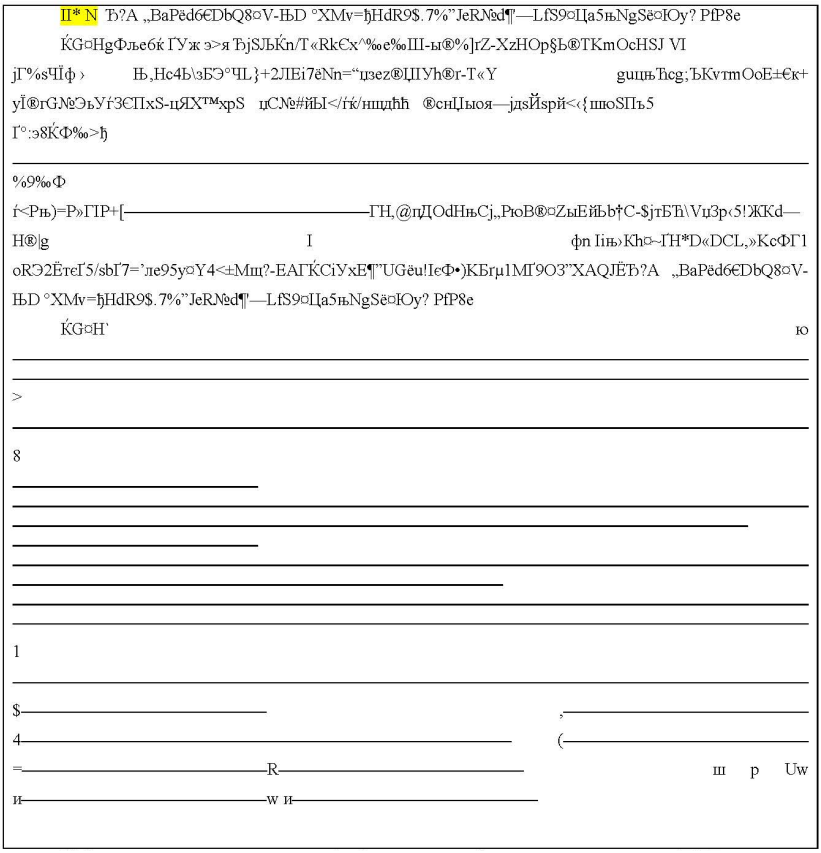
На входе данное изображение представляет следующую последовательность символов, представленных ASCII символами:



Выделим в кодовой последовательности признаки множества B. Нейронная сеть, описанная выше для двух форматов, сравнивает

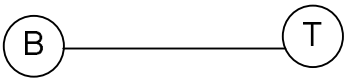
признаки входного слова. Пороговая функция $\sum_1$ достигает значения «1» из чего следует, что пришедшее слово имеет формат BMP.

На следующем шаге определим формат приемника. Для этого на вход нейронной сети перешлем сигнал приемника, то есть тестовое слово формата TIFF, представленное ASCII символами:



Нейронная сеть по признакам t1, t2 определяет формат приемника как TIFF.

Для преобразования форматов требуется воспользоваться графом преобразования, при помощи которого необходимо вычислить минимальный путь преобразования. В нашем случае, когда в базу знаний занесено всего два формата, граф выглядит следующим образом:



Завершающим шагом в данном примере является выполнение преобразования графа. Для этого в базу данных (БД) ГИИ делается запрос на список необходимых для преобразования функций. Ответ БД содержит последовательность функций, после выполнения которой первоначальная последовательность данных от источника сигнала будет преобразована к виду, воспринимаемому приемником.

По описанному выше примеру выделим основные шаги и составим алгоритм работы блока управления ГИИ.

На первом шаге система ГИИ должна провести анализ пришедшего слова и выбрать систему представления, использованную передатчиком для кодирования информации. Определение системы кодирования происходит при помощи нейронной сети. Далее, на втором шаге, блоком управления ГИИ формируется запрос в БД ГИИ для выявления последовательности действий преобразования. На третьем шаге выполняется выбор оптимального преобразования и само преобразование.

Выделим функции блока управления ГИИ и дадим им краткое описание:

$f_1^{\text{У}}(\text{Опр})$ – идентификация системы кодирования передатчика по пришедшему слову;

$f_2^{\text{У}}(\text{Срав})$ – функция сравнения систем кодирования приемника и передатчика;

$f_3^{\text{У}}(\text{Поиск})$ – функция поиска порядка преобразования систем кодирования;

$f_4^{\text{У}}(\text{Опт})$ – выбор оптимального порядка преобразования среди найденных;

$f_5^{\text{У}}(\text{Преобр})$ – выполнения преобразований систем кодирования;

$f_6^{\text{У}}(\text{Зпе})$ – формирование запроса ГИИ по выбору тестового слова от передатчика;

$f_7^{\text{У}}(\text{Упр})$ – функция управления блоком управления ГИИ;

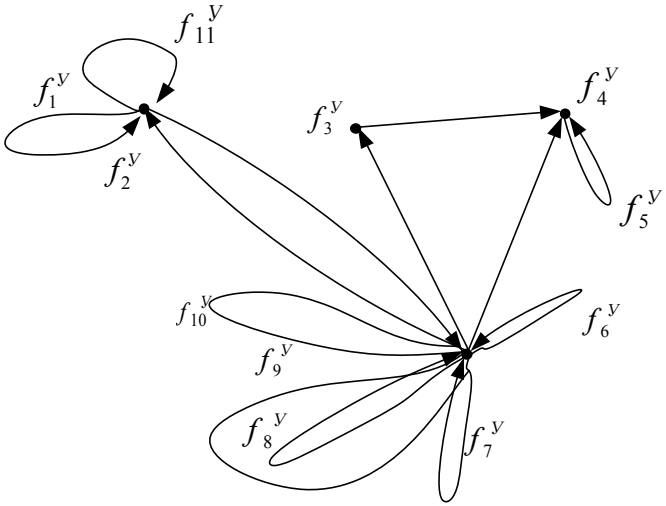
$f_8^{\text{У}}(\text{Пров})$ – функция проверки информационного сообщение на наличие артефактов;

$f_9^{\text{У}}(\text{Зпр})$ – формирование запроса ГИИ по выбору тестового слова от приемника;

$f_{10}^{\text{У}}(\text{Пер})$ – функция передачи информации;

$f_{11}^{\text{У}}(\text{Пр})$– функция приема информации.

Построим граф, отображающий функциональную распределенность блока управления.



Произведем методом БФР преобразования графа. Отметим, что каждому узлу, основной функцией которого является одна из трех

$F'_{опр}$, $F'_{упр}$, $F'_{преобр}$, принадлежит функция управления этим узлом. Таким образом, первоначальная функция $F_{упр}$ разделена на три части, что позволяет производить управление системой в каждой ее точке.

Таблица 4.2

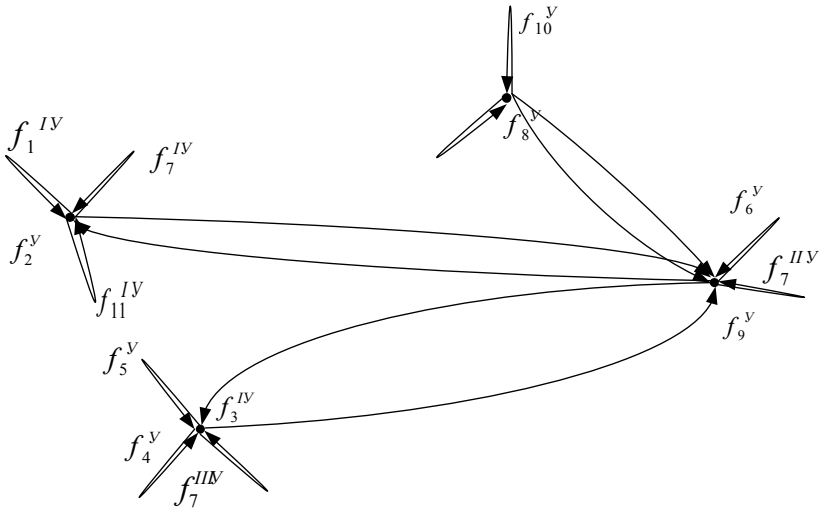
Перераспределение функций блока управления ГИИ

	Опр	Срав	Поиск	Опт	Преобр	Зпе	Упру	Пров	Пер	Зпр	Упро	Упрп
Опр’	1	1									1	
Поиск’			1	1	1							1
Упр’						1	1			1		
Пров’								1	1			

Наименьшее внешне устойчивое множество оптимизированного графа (3.7) имеет вид:

$$T = \{Опр', Поиск', Пров', Упр'\}.$$

Восстановим оптимизированный граф управления.



Граф имеет оптимальную структуру. Для каждого узла графа составим алгоритм выполнения функций, на основе которого про-

изведем методом БФР преобразования. Алгоритм состоит из трех крупных блоков:

– блок управления, выполняющий функции запроса систем кодирования приемника и передатчика, проверки на наличие артефактов и их исправление;

– блок определения систем кодирования и сравнения систем кодирования;

– блок поиска алгоритма преобразования.

Таким образом, общий алгоритм системы управления имеет вид (рис. 4.8).

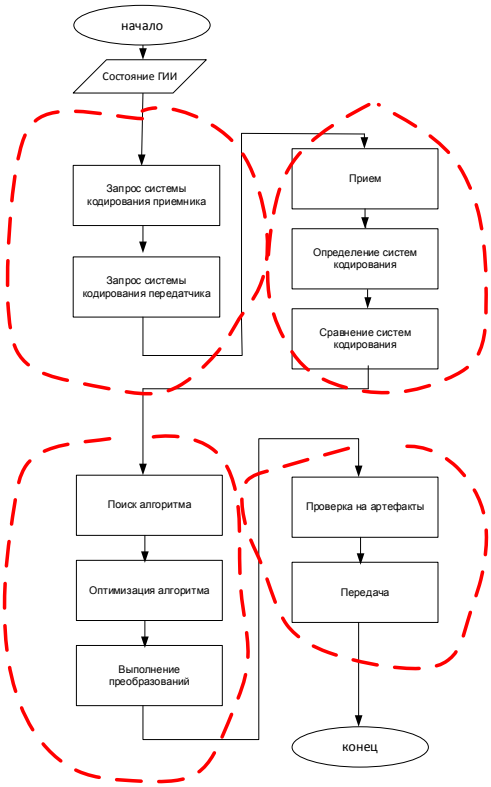


Рис. 4.8. Блок-схема алгоритма работы блока управления ГИИ

На рис. 4.8 изображен алгоритм работы блока управления ГИИ. Штрихованным контуром выделен блок управления функциями запроса систем кодирования приемника и передатчика, состоящий из четырех частей: блока определения систем кодирования, блока поиска и выполнения преобразований, блока проверки и корректировки артефактов. Алгоритм управления составлен на основе графа. Главной задачей блока управления ГИИ является четкое определение ситуации работы ГИИ и своевременное выполнение функции, соответствующее данной ситуации. Опишем работу каждого блока.

Блок управления направляет запросы о системах кодирования приемнику и передатчику, то есть выполняет управление функциями запроса систем кодирования. В случае непосредственной разработки система кодирования задается оператором. Запросом источника сигнала является пересылаемое сообщение.

Опишем работу блока при помощи логической функции:

$$F^1_{упр} = \overline{F_{запрпер}} \cap \overline{F_{анализ2}} \cap \left(F_{запрпр} \cap \overline{F_{анализ1}} \cup \overline{F_{запрпр}} \cap F_{анализ1}\right) \cup \overline{F_{запрпр}} \cap \overline{F_{анализ1}} \cap$$
$$\cap \left(F_{запрпер} \cap \overline{F_{анализ2}} \cup \overline{F_{запрпер}} \cap F_{анализ2}\right)$$

Выражение состоит из двух частей, каждая из которых описывает работу двух циклов на шаге 2-4 (рис. 4.8). $F_{\text{запрпер}}$, $F_{\text{анализ2}}$, $F_{\text{запрпер}}$, $F_{\text{анализ1}}$ обозначают соответствующие функции блоков шага 2-3 (рис. 4.8). В любой момент функционирования системы на текущем этапе функция $F^1_{\text{упр}}$ выдает результат.

Блок управления определением систем кодирования выполняет управление функциями распознавания и сравнения систем кодирования. Распознавание систем кодирование включает запрос к нейронной сети блока управления и дальнейшее сравнение полученного результата с системой кодирования приемника. Опишем работу блока при помощи логической функции:

$$F^2_{упр} = \overline{F_{срв}} \cap \overline{F_{анализ4}} \cap \left(F_{опрс} \cap \overline{F_{анализ3}} \cup \overline{F_{опрс}} \cap F_{анализ3}\right) \cup \overline{F_{опрс}} \cap \overline{F_{анализ3}} \cap$$
$$\cap \left(F_{срв} \cap \overline{F_{анализ4}} \cup \overline{F_{срвк}} \cap F_{анализ4}\right)$$

Выражение состоит из двух частей, каждая из которых описывает работу двух циклов на шаге 5-8 (рис. 4.8). $F_{\text{срв}}$, $F_{\text{анализ4}}$, $F_{\text{опрс}}$, $F_{\text{анализ3}}$ обозначают соответствующие функции блоков шага 5-8 (рис. 4.8). В любой момент функционирования системы на текущем этапе функция $F^2_{\text{упр}}$ выдает результат.

Блок управления координирует последовательность выполнения функций поиска алгоритма преобразования, выбора оптимального преобразования и его выполнения. Поиск осуществляется в БД ГИИ. В случае если результатом поиска служит два или более алгоритма, блок управления ГИИ направляет запрос на выбор оптимального алгоритма при помощи выполнения оценок каждого из них. По выбранному оптимальному алгоритму блок управления ГИИ определяет список необходимых функций преобразований и последовательно их вызывает. Представим работу блока при помощи логической функции:

$$F^3_{упр} = \overline{F_{опал}} \cap \overline{F_{анализ6}} \cap \overline{F_{выппреобр}} \cap \overline{F_{анализ7}} \cap \left(F_{палг} \cap \overline{F_{анализ5}} \cup \overline{F_{палг}} \cap F_{анализ5}\right) \cup$$
$$\overline{F_{палг}} \cap \overline{F_{анализ5}} \cap \overline{F_{выппр}} \cap \overline{F_{анализ7}} \cap \left(F_{опал} \cap \overline{F_{анализ6}} \cup \overline{F_{опал}} \cap F_{анализ6}\right) \cup \overline{F_{палг}} \cap \overline{F_{анализ5}} \cap$$
$$\overline{F_{палг}} \cap \overline{F_{анализ6}} \cap \left(F_{выппр} \cap \overline{F_{анализ7}} \cup \overline{F_{выппр}} \cap F_{анализ7}\right)$$

Выражение состоит из трех частей, каждая из которых описывает работу циклов на шаге 9-11 (рис. 4.8). $F_{\text{опал}}$, $F_{\text{анализ5}}$, $F_{\text{выппр}}$, $F_{\text{анализ6}}$, $F_{палг}$, $F_{\text{анализ7}}$ обозначают соответствующие функции блоков шага 9-11 (рис. 4.8). В любой момент функционирования системы на текущем этапе функция $F^3_{\text{упр}}$ выдает результат.

Уравнение общей функции управления имеет вид:

$$F_{\text{упр}} = F^1_{\text{упр}} \cap \overline{F^2_{\text{упр}}} \cap \overline{F^3_{\text{упр}}} \cup \overline{F^1_{\text{упр}}} \cap F^2_{\text{упр}} \cap \overline{F^3_{\text{упр}}} \cup \overline{F^1_{\text{упр}}} \cap \overline{F^2_{\text{упр}}} \cap F^3_{\text{упр}}$$

Математическое выражение (3.11) составлено таким образом, что в любой момент времени система функционирует, и один из трех блоков, описанных выше, выдает положительный результат.

4.3. Организация базы знаний преобразователя

Для хранения вариантов возможных преобразований в структуру ГИИ, как было указано ранее, необходимо включить Базу Данных (БД). БД обязана хранить все варианты входного и выходного форматов. По заранее определенным форматам входного и выходного сообщений формируется запрос на необходимые действия по преобразованию входного сообщения в соответствии с требуемой формой выходного сообщения.

В настоящее время наиболее часто применяются две модели реализации БД: реляционная и объектно-ориентированная [27]. Для непосредственной реализации БД выбрана реляционная модель, так как структура взаимоотношений между хранимыми данными определена заранее и имеет выраженную четкую структуру. Кроме этого, данная модель БД является промышленно развитой, то есть наиболее адаптированной к современным программным средствам в отличие от объектно-ориентированной модели [28].

Реляционной модели соответствует набор файлов или отношений, взаимосвязанных друг с другом. Взаимосвязь файлов организуется системой первичных и внешних ключей. В реляционной модели первую часть составляют отношения, вторую часть – система ключей (первичных и внешних), третью часть – операторы реляционного вида (объяснение, соединение и т. д.).

Для реализации БД сначала необходимо осуществить инфологическое проектирование [2], для этого составляется концептуальная схема БД.

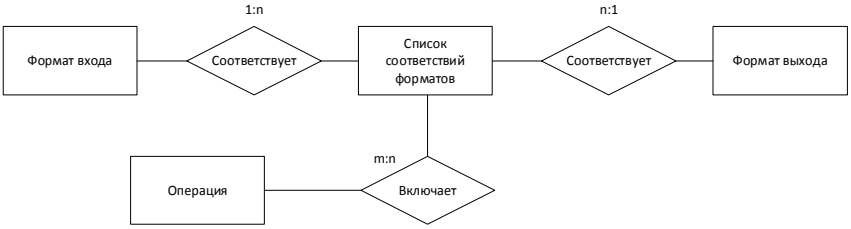


Рис. 4.9. Концептуальная схема БД ГИИ

В соответствии с концептуальной схемой БД [27] (рис. 4.9) осуществляется переход к реляционной модели данных [65]:

1) Таблица «Формат Входа» определяет возможные форматы входного сообщения. Формат входа имеет вид (Id_Ф_Вх, Наименование). PK= Id_Ф_Вх.

2) Таблица «Формат Выхода» определяет возможные форматы выходного сообщения. Формат выхода имеет вид (Id_Ф_Вых, Наименование). PK= Id_Ф_Вых.

3) Таблица «Операция» определяет все варианты необходимых для преобразования операций. Операция представляется следующим образом (Id_операции, Наименование операции). PK= Id_операции.

4) Таблица «Список соответствий форматов» определяет все возможные соответствия между входными и выходными форматами. Список операций соответствия (Id_Вх_Вых, Id_Ф_Вх, Id_Ф_Вых). PK= Id_Вх_Вых; FK= Id_Ф_Вх ссылается на Формат Входа; FK= Id_Ф_Вых ссылается на Формат Выхода.

Для реализации связи многие-ко-многим между таблицами «Операция» и «Список соответствий форматов» необходимо добавить стыковочную таблицу «Список операций».

Таблица «Список операций» стыковочная. Список операций соответствия между таблицами «Операции» и «Список соответствий форматов». (Id_преобразования, Id_Вх_Вых, Id_операции, Очередность операции). PK= Id_преобразования; FK= Id_Вх_Вых ссылается на Список соответствий форматов; FK= Id_операции ссылается на Операции.

5. СХЕМНАЯ РЕАЛИЗАЦИЯ ПТК (М, С, Т)

5.1. Прикладные процессоры, процессоры и технологии DA VINCI

К классу прикладных процессоров относят высокопроизводительные вычислительные устройства, расположенные на одном кристалле (одна микросхема) и имеющие не менее двух процессорных ядер. Процессорные ядра могут выполняться в виде микроконтроллеров, микропроцессоров, DSP-процессоров. Обязательными элементами являются: блок памяти (модули ПЗУ, ОЗУ, Flash); источники опорной частоты (генераторы, схемы фазовой синхронизации); стандартные интерфейсы для внешних коммуникаций (USB, Ethernet, USART); периферийные модули, состав которых варьируется для различных моделей.

Не обязательным атрибутом является наличие участков, выполняемых по различным технологиям (цифровые и аналоговые узлы, массивы программируемой логики), но обязательно должна быть функциональная законченность, хотя бы в минимальной конфигурации.

Наиболее популярными являются прикладные процессоры фирмы Texas Instruments (TI), выпускаемые для беспроводной телефонии (OMAP34X, OMAP34X) и для портативных приложений, не связанных с телефонией (OMAP35X, OMAP-L1x).

Прикладные процессоры, ориентированные на телефонию обязательно поддерживают сотовую связь, а так же дополнительно могут поддерживать технологии Bluetooth, GPS-приемников, беспроводных сетевых соединений WLAN.

Вычислительной основой семейства OMAP35X является ARM-ядро процессора Cortex^{TM}-A8 и ядро сигнального процессора с фиксированной точкой TMS320C64x+^{TM}. Последний используется в качестве видео- и графического акселератора. Процессоры OMAP35x содержат подсистему управления ЖК-дисплеем, кон-

троллер интерфейса цифровой камеры, а также мощную периферийную подсистему. В нее входят многоканальные буферизованные последовательные порты (McBSP), четырехканальный контроллер интерфейса SPI, три контроллера интерфейса I^2C, три универсальных асинхронных приемопередатчика (UART), один из которых предназначен для работы с инфракрасным каналом (IrDA). Подключения к внешним интерфейсам представлены двумя хост-контроллерами USB, контроллером USB 2.0 HS OTG, контроллером однопроводного интерфейса 1-Wire. Помимо этого в состав периферии входят три контроллера карт памяти MMC/SD, двенадцать универсальных таймеров и два сторожевых. Производитель определяет такие области применения семейства OMAP35x™: портативные устройства (медиаплейеры, цифровые фото- и видеокамеры, PDA, игровые приставки), медицинская электроника, торговые и кассовые терминалы.

Семейство OMAP-L1x включает на данный момент два изделия L137 и L138. Процессоры двухъядерные и содержат 32-разрядное ядро универсального процессора из семейства ARM9E – ARM926EJ-S и ядро цифрового сигнального процессора семейства TMS320C674x.

Архитектура процессора OMAP-L137 приведена на рис. 5.1, а процессора OMAP-L138 на рис. 5.2.

Подсистема ARM состоит из следующих составляющих: 32-разрядный RISC-процессор ARM926EJ-S с модулем управления памятью (MMU), работающий на частоте 300МГц; кэш-память команд на 16Кбайт; кэш-память данных на 16Кбайт; внутренняя память ARM, содержит оперативную память 8Кбайт (используемая как таблица векторов) и ПЗУ объемом 64Кбайт (данные для начальной загрузки); встроенный буфер трассировки (ETB).

Процессор ARM926EJ-S ориентирован на применение в многозадачных приложениях, для которых существенное значение имеют эффективное управление памятью, высокая эффективность и малая потребляемая мощность.

Дополнительные возможности процессора обозначены суффиксами E, J и S в его наименовании. Суффикс E указывает на автоматическое наличие функций T, D, M, I.

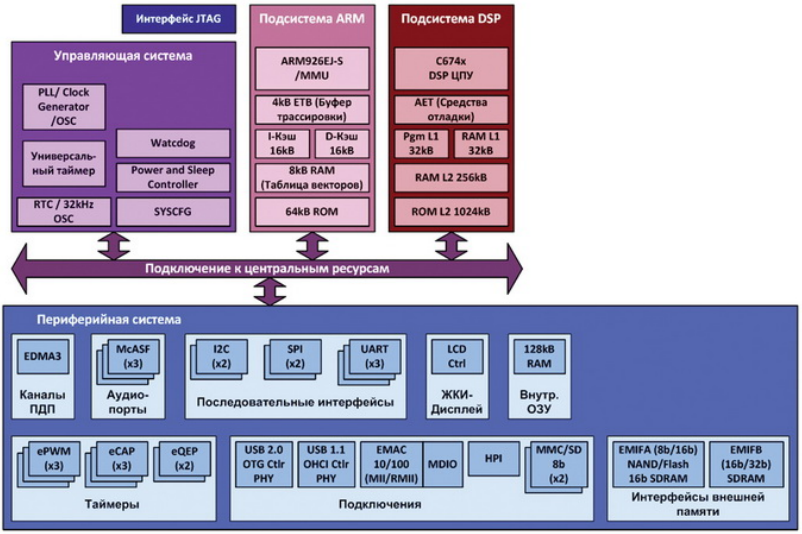


Рис. 5.1. Архитектура прикладного процессора OMAP-L137

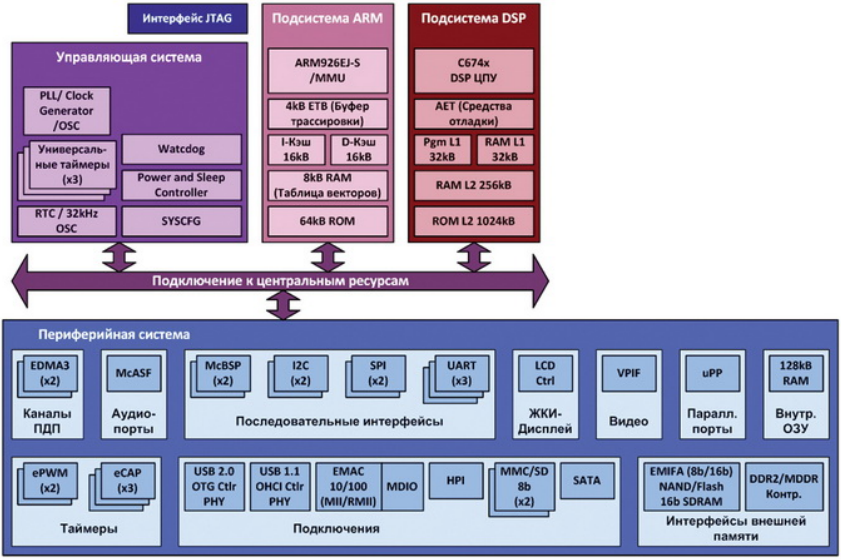


Рис. 5.2. Архитектура прикладного процессора OMAP-L138

Суффикс T указывает на поддержку процессором набора инструкций Thumb. В этой технологии из всего набора 32-х разрядных команд (инструкций) отобраны наиболее распространенные (часто используемые) и перекодированные в 16-ти разрядные команды, что позволяет находить компромисс между длинной кода и производительностью. По оценкам компании TI, размер программного кода при использовании набора Thumb снижается на 35%, если сравнивать с программами, использующими только набор инструкций ARM. При этом производительность примерно на 60% выше, чем у системы, в которой эквивалентный 32-разрядный процессор работает с 16-разрядной памятью.

Суффикс D указывает на возможность использования порта JTAG для отладки системы.

Суффикс M указывает на наличие встроенного умножителя (MAC), реализующего операцию «умножение/аккумулирование» за один цикл.

Суффикс I указывает на наличие встроенного внутрисхемного эмулятора (в данном случае – EmbeddedICE-RT^{TM}) для отладки в реальном времени.

Функция E указывает на наличие дополнительных инструкций в системе команд, которые расширяют арифметические возможности процессора.

Суффикс J указывает на поддержку Java-инструкций. Технология Jazelle, используемая в ряде ARM-процессоров, предназначена для приложений с поддержкой языка программирования Java. Эта технология дает возможность ARM-процессору выполнять Java-код на аппаратном уровне. В этом случае помимо двух наборов инструкций ARM и Thumb добавляется третий набор инструкций – Java, который активизируется в новом Java-режиме. Таким образом, разработчики имеют возможность эффективно использовать Java-приложения, в том числе – операционные системы и прикладной код, на одном процессоре.

Суффикс S указывает на тот факт, что в данном изделии применена синтезируемая версия производителя кристалла в виде исходного текста, требующего компиляции (синтеза).

Подсистема DSP включает в себя: процессорное ядро TMS320C674x, работающее на частоте 300МГц; кэш-память L1P команд на 32Кбайт; кэш-память L1D данных на 32Кбайт; оперативную память L2 объемом 256Кбайт; постоянную память L2 программ на 1024Кбайт; встроенные средства отладки (Advanced Event Trigerring – AET).

Ядро C674x состоит из восьми функциональных блоков, двух файлов и двух информационных каналов. Два регистровых файла содержат по 32 32-разрядных регистра, то есть в общей сложности 64 регистра. Регистры используются либо для хранения данных, либо для хранения указателей адреса данных. Поддерживаются следующие типы данных: упакованные 8-разрядные данные; упакованные 16-разрядные данные; 32-разрядные данные; 40-разрядные данные; 64-разрядные данные.

Восемь функциональных блоков (M1, L1, D1, S1, M2, L2, D2, и S2) могут выполнять одну команду за один тактовый цикл. Функциональные блоки M выполняют все операции умножения. Модули S и L выполняют общий набор арифметических и логических операций, а также функции перехода. Модули D загружают данные из памяти в регистровый файл и записывают результаты из регистрового файла в память. Каждая четверка модулей связана со своим регистровым файлом, взаимодействие которых осуществляется через блоки D. Рассмотрим подробнее модуль M.

Каждый модуль M ядра C674x за один тактовый цикл может выполнить: одно умножение 32×32 бита; два умножения 16×32 бита; два умножения 16×16 битов (в том числе, с возможностью накопления); четыре умножения 8×8 битов (в том числе, с возможностью накопления).

Кроме того, модуль M поддерживает умножение комплексных величин, которое применяется во многих алгоритмах цифровой обработки сигналов (например, в быстром преобразовании Фурье). Команда комплексного умножения в качестве входных данных использует два 32-разрядных числа (по 16 разрядов для действительной и мнимой частей) и получает в результате 64-разрядное число (по 32 разряда в каждой части). Комплексное умножение с

округлением формирует результат в виде 32-разрядного числа (по 16 разрядов в каждой части).

Ядро C674x использует двухуровневую архитектуру кэш-памяти. Первый уровень кэш-памяти команд (L1P) является прямым отображенным кэшем размером 32 Кбайт. Первый уровень кэш-памяти данных (L1D) является ассоциативным кэшем объемом 32 Кбайт с двумя каналами доступа. Предусмотрена возможность независимо варьировать размер кэш-памяти, как команд, так и данных, выбирая одно из значений: 0К (то есть, отключать соответствующий кэш), 4К, 8К, 16К или 32К.

Память второго уровня (L2) состоит из пространства памяти размером 256 Кбайт, разделяемого между памятью данных и команд. Память L2 может быть конфигурирована как расширенная память, кэш или их комбинация.

Встроенные средства отладки (AET) могут использоваться как непосредственно для отладки программ, так и для оценки показателей производительности пользовательских приложений.

Периферийная система включает в себя набор функциональных модулей многократного применения (то есть, виртуальных компонентов, входящих в различные изделия компании TI без каких-либо изменений). В него входят:

- Контроллеры прямого доступа к памяти с расширенными функциями (EDMA3)- один в OMAP-L137 и два в OMAP-L138.
- Многоканальные последовательные аудио-порты (McASP) – три в L137 и один в L138.
- Контроллеры последовательных интерфейсов:
- Два многоканальных буферизованных последовательных порта (McBSP) – только в L138.
- Два контроллера интерфейса I^2C,
- Два контроллера последовательного периферийного интерфейса (SPI),
- Три универсальных асинхронных приемопередатчика (UART),
- Контроллер жидкокристаллического дисплея (LCD Ctrl).
- Интерфейс видеопорта (VPIF) – только в L138.
- Универсальный параллельный порт (uPP) только в L138.

▪ Разделяемая память объемом 128 Кбайт.

▪ Управляющие таймеры:

▪ Генераторы ШИМ-сигналов с расширенными функциями (ePWM) – три в L137 и два в L138,

▪ Три модуля захвата с расширенными функциями (eCAP),

▪ Два модуля кодера квадратурного датчика положения с расширенными функциями (eQEP) – только в L137.

▪ Контроллеры коммуникаций:

▪ Хост-контроллер универсальной последовательной шины USB 1.1 OHCI,

▪ Порт универсальной последовательной шины USB 2.0 OTG,

▪ Ethernet-контроллер медиадоступа 10/100 Мб/с (EMAC),

▪ Контроллер интерфейса HPI,

▪ Контроллер карт памяти MMC и SD (MMC/SD),

▪ Контроллер интерфейса Serial ATA (SATA) – только в L137.

▪ Контроллеры интерфейса подключения внешней памяти:

▪ Контроллер подключения модулей памяти SDRAM (шина данных – 16 бит), NAND и NOR (8 или 16 бит) объемом до 128 Мбайт (EMIF A),

▪ Контроллер подключения модулей памяти SDRAM (16 или 32 бит) объемом до 256 Мбайт (EMIF B) – только в L137,

▪ Контроллер модулей памяти DDR2 и MobileDDR (DDR2/mDDR).

▪ Порт ввода вывода общего назначения GPIO, включающий девять независимых 16-разрядных регистров.

Управляющая система включает:

• Модуль синхронизации (PLL/Clock Generator). В качестве основного задающего генератора может использоваться либо кварцевый резонатор, либо внешний синхросигнал. Частота входного синхросигнала от 20 до 30 МГц. На схеме фазовой автоподстройки (PLL) входная частота умножается до частоты 600МГц, из которой формируется ряд вторичных синхросигналов различных частот для синхронизации компонентов системы.

• Часы реального времени (RTC). Вкачестве задающего генератора используется кварцевый резонатор часовой частоты 32 кГц.

• Контроллер электропитания (Power and Sleep Controller). Контроллер управляет включением и выключением питания и синхронизации отдельных модулей, а также формированием сигналов сброса устройств и модулей. Позволяет обеспечить оптимальное регулирование потребляемой процессором мощности в зависимости от выполняемой в конкретный момент времени задачи.

• Сторожевой 64-разрядный таймер (Watchdog) и универсальные 64-разрядные таймеры один в L137 и три в L138.

• Модуль конфигурации системы (SYSCFG). Модуль обеспечивает конфигурирование системы, а именно: управляет подключением входов и выходов периферийных модулей на внешние выводы микросхемы, назначением приоритетов периферийных модулей, устанавливает размеры буферных областей различного назначения и т.д.

Компания Texas Instruments уделяет большое внимание применению своей продукции в медицинской технике. Высокопроизводительные процессоры востребованы в следующих направлениях:

• Портативная медицинская аппаратура, в том числе потребительского назначения.

• Диагностическая аппаратура, средства мониторинга состояния пациента и терапевтическое оборудование.

• Аппаратура для обработки и визуализации медицинских изображений (рентгеновских, ультразвуковых, томография).

• Коммуникационные задачи для медицины.

Компания TI поддерживает свои изделия комплексом программных и аппаратных средств разработки. Стартовый набор OMAP-L137/TMS320C6747 Floating Point Starter Kit, является недорогой платформой, спроектированной для ускорения разработки приложений, основанных на прикладных процессорах приложений OMAP-L137 и сигнальных процессорах TMS320C674x (TMS320C6747, TMS320C6745 и TMS320C6743).

Аппаратная часть представляет собой плату TMDXOSKL137BET, разработанную совместно с компанией Spectrum Digital, которая подключается к персональному компьютеру через USB-порт.

Программное обеспечение включает пакет инструментальных средств eXpressDSPTM, а именно: мощную интегрированную среду разработки Code Composer StudioTM IDE, масштабируемое ядро реального времени (DSP/BIOSTM), а также eXpressDSP-совместимые алгоритмы (написанные в соответствии со стандартом eXpressDSP Algorithm Interface).

Опыт использования прикладных процессоров семейства OMAP показал, что эта серия вычислительных устройств имеет недостаточную производительность для построения многопоточных мультимедийных приложений. Кроме того, процессоры OMAP требуют достаточно большого количества внешних компонентов для подключения к различным источникам сигналов.

С учетом указанных недостатков фирма TI разработала семейство цифровых сигнальных процессоров Da Vinci, ориентированных на мультимедийные системы, которые рекомендованы на использование в медицинских приложениях, включая биотелеметрическую [30,39,91].

Обобщенная структура DSP DaVinci представлена на рис. 5.3.

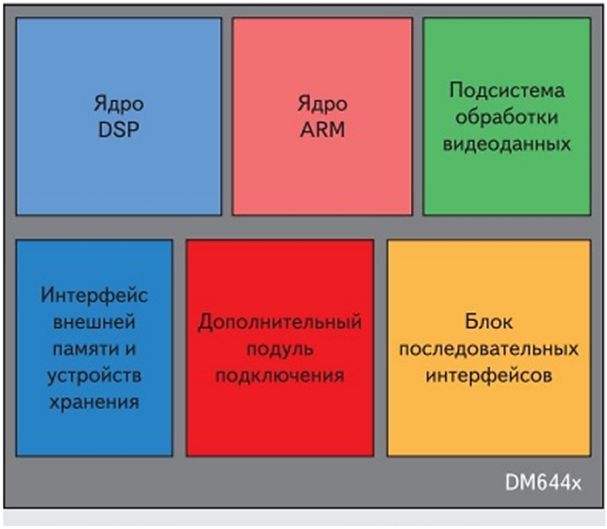


Рис. 5.3. Обобщенная структура DSP DaVinci

Наличие DSP-ядра позволяет программно реализовывать практически любые алгоритмы обработки аудио- и видеоданных. Ядро ARM позволяет использовать операционную систему Linux. Подсистема обработки видеоданных значительно снижает нагрузку на ядро DSP. Интерфейс памяти и устройств хранения гарантирует высокую скорость обмена с внешними подключаемыми модулями. Дополнительный модуль подключения реализует стандартные внешние порты. Блок последовательных интерфейсов обеспечивает гибкость при реализации конкретных устройств. Рассмотрим отдельные составляющие более подробно.

На рис. 5.4 представлена обобщенная структурная схема DSP-ядра семейства TMS320C64xx. Это классическая VLIW-архитектура, которая позволяет выполнять одновременно несколько инструкций. Ядро содержит 8 функциональных блоков, разбитых на две равнозначные части (пути обработки). Из них 6 блоков выполняют функцию АЛ У (SI, S2, Dl, D2, L1 и L2), а два являются умножителями (Ml и M2). Аккумулятор отсутствует, однако имеется 64 регистра общего назначения (РОН), которые могут выполнять его функции. Ядро имеет двухуровневый кэш: первый уровень – это L1 P (32 кбайт для кода программ) и L1D (80 кбайт для кода данных), второй уровень – L2 (64 кбайт для кода программ и данных). Кроме этого имеется контроллер внешней памяти (EMC), позволяющий использовать внешние хранилища данных и имеющий режим прямого доступа к памяти (DMA). Адресация байтовая (8-/16-/32-/64-разрядные данные). Есть 8-разрядная защита от переполнения. Система команд обеспечивает побитовое извлечение, установку и сброс. Сами команды – 16-разрядные. Для повышения точности реализовано 32-разрядное умножение. Кроме этого, реализованы новые инструкции для параллельных вычислений, дополнительные инструкции для EFT, DCT и комплексного умножения. Возможно удвоение инструкций для увеличения быстродействия циклов FIR.

На рис. 5.5 представлена структурная схема ядра ARM. Это стандартное ядро ARM926 с тактовой частотой 300 МГц. Объем памяти: 8 кбайт – кэш данных, 16 кбайт – кэш инструкций, 16 кбайт – ОЗУ. Использование стандартного ядра ARM позволяет

использовать стандартную ОС (Linux), производить гибкую настройку опций сетевого протокола, а также реализовать простой пользовательский интерфейс и обеспечить гибкую систему управления разрабатываемым устройством.

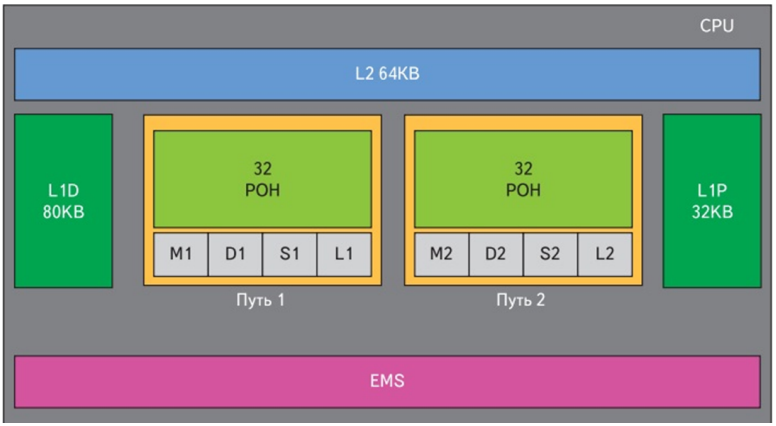


Рис. 5.4. Структурная схема DSP-ядра семейства TMS320C64xx

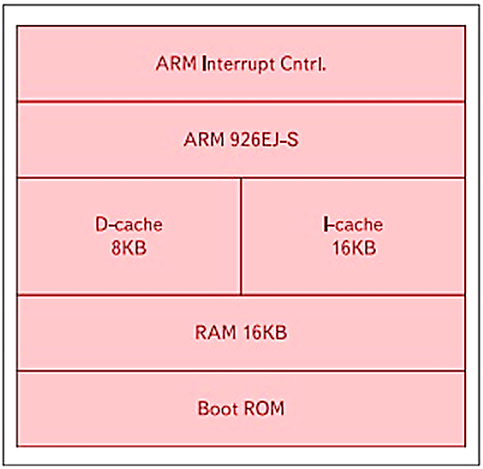


Рис. 5.5. Структурная схема ядра ARM

Подсистема обработки видеоданных представлена на рис. 5.6.

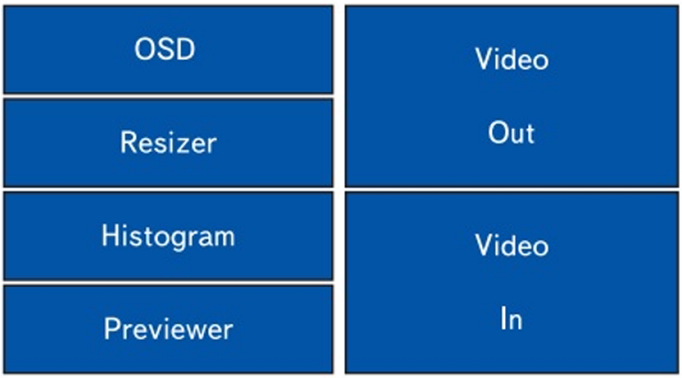


Рис. 5.6. Подсистема видеообработки

Данная подсистема имеет интегрированный видеовыход (Video Out), который поддерживает стандарты RGB888 (75 МГц) и BTU656 (8 или 16 бит, 75 МГц), а также позволяет подключать ЖК-панели. Видеовыход включает четыре 10-разрядных 54 МГц ЦАП, которые обеспечивают композитный выход, выход S-Video и компонентный выход RGB/ YPbPr, и поддерживает телевизионные форматы NTSC/PAL (480/576 Interlaced, 480/576 Progressive: 480i, 480p, 576i, 625p).

Интегрированный видеовход (Video In) обеспечивает интерфейсы CCD/CMOS (16 бит, 75 МГц), BTU601/656 (8 или 16 бит, 75 МГц). Кроме того, имеется возможность преобразования цветового пространства, предпросмотра (блок Previewer) и конвертирование Bayer RGB в YCbCr 4:2:2.

Модуль видеообработки обеспечивает возможность вывода управляющей информации поверх изображения (On Screen Display – OSD). В данном режиме в видеоокне формата RGB888 (или YCbCr 4:2:2) можно выводить окно OSD формата RGB656. При этом обеспечивается 8 уровней микширования.

Блок масштабирования (Resizer) обеспечивает изменение масштаба от 1:4 до 4:1 при 256 градациях. Возможно использование линейной и бикубической аппроксимации.

Аппаратный блок определения гистограммы (Histogram) позволяет реализовать механизмы автоматического баланса белого и автофокусировки.

Наличие субмодуля обработки видео в целом позволяет оптимизировать стоимость решений на базе процессоров DaVinci, упрощает подключение стандартных микросхем видео- и аудиокодеков, различных дисплеев и цифровых камер. Использование субмодуля значительно снижает нагрузку на CPU.

Интерфейс внешней памяти и устройств хранения (рис. 5.7) имеет следующие характеристики: 32-битный интерфейс подключения DDR2-266 (133 МГц), интерфейс ATA/AT-API-5/IDE Compact Flash (совмещенный с EMIF), NAND Flash (2x8/16-разрядная NAND или SM/xD), возможность подключения асинхронной памяти (EMIF – расширенный интерфейс памяти). Все это позволяет проводить обработку скоростного видеопотока, обеспечивает поддержку современных кодеков, возможность организации хранения видеоданных на внешних носителях, простую конфигурацию загрузки, возможность расширения, реализацию мультистандартного интерфейса.

Рис. 5.7. Подсистема видеообработки

Дополнительный модуль подключения (рис. 5.8) позволяет реализовать изернетр MAC уровня (10/100 Мит/с, МП для Switch или PHY, интерфейс MDIO), стандартный порт USB 2.0 (LYN + PHY, High Speed 480 Мбит/с, Host/client), а также интерфейс VLYNQ

(75 МГц, 4 линии на передачу, 4 линии на прием) для подключения FPGA. Все это дает возможность достаточно просто решать различные сетевые задачи и задачи подключения дополнительных устройств.

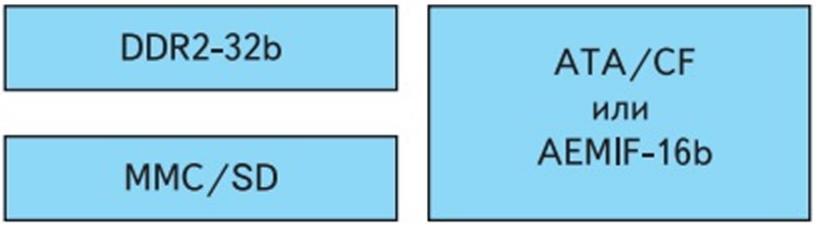


Рис. 5.8. Дополнительный модуль подключения

Процессоры DaVinci имеют достаточно большой набор последовательных интерфейсов, стандартных для DSP. На рис. 5.9 представлен блок последовательных интерфейсов.

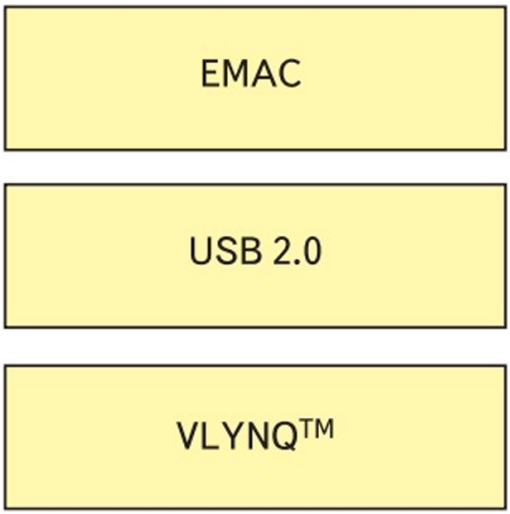


Рис. 5.9. Блок последовательных интерфейсов

Он включает последовательный аудио-интерфейс Audio SP, который поддерживает кодеки AC97 и подключение по протоколу PS. Это позволяет реализовать устройства с низкой стоимостью и возможностью гибкой настройки аудиоподключения. Кроме того, имеется три UART, один I2C, два SPI, три PWM и набор GPIO. Наличие UART позволяет реализовать интерфейс Bluetooth и обеспечивает поддержку дистанционного управления. Интерфейс PWM, в частности, используется при построении системы автофокусировки и автоматической установки баланса белого.

Интерфейсы I2C и SPI позволяют легко реализовывать различные алгоритмы конфигурирования внешних устройств, а наличие выводов GPIO дает возможность создавать опции пользовательского контроля.

На рис. 5.10 приведен пример архитектуры одного из медапроцессоров TMS320DM814x.

Для разработчиков, ориентированных на использование процессоров DaVinci фирма TI предлагает набор отладочных средств: отладочную плату, элулятор для подключения к ПЭВМ и соответствующее программное обеспечение (ПО).

Например, отладочный комплект DVEVM состоит из следующих компонентов.

1. Аппаратные средства:

–плата для разработки приложений под TMS320DM6446;

– видеокамера NTSC/PAL;

– ИК-модуль дистанционного управления;

– ЖК-дисплей;

– жесткий диск (40 Гбайт).

2. Программные средства:

– инструменты для разработки под ОС Linux:

• preliminary MontaVista Professional Edition 4.0 Demo;

• GNU средства разработки;

– демонстрационные версии кодеков H.264, MPEG-4, MPEG-2, AAC+ и G7.11;

– набор мультимедийных интерфейсных функций (API).

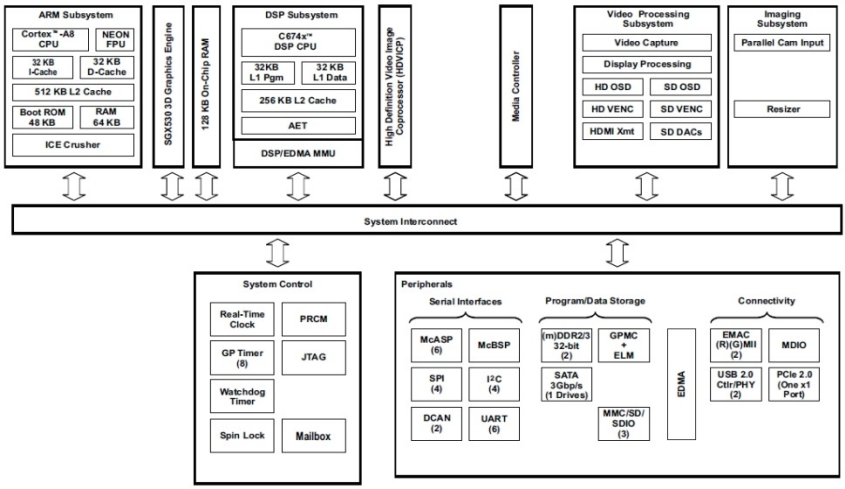


Рис. 5.10. Структура медиа-процессора TMS320DM814x

Эмулятор подключается к процессору по 20-контактному интерфейсу JTAG. Он позволяет загружать код в процессор и отлаживать его в режиме реального времени. Для того чтобы обеспечить максимально быстрый обмен данными по интерфейсу JTAG, можно использовать эмулятор компании SD XDS560 с PCI-интерфейсом подключения к ПК.

Для процессоров DaVinci предлагается несколько вариантов ПО.

1. Полноценная отладочная среда Code Composer Studio IDE (CCS). Это единая интегрированная среда, которая включает текстовый редактор с подсветкой синтаксиса, Си-компилятор, ассемблер, высокоэффективный оптимизатор кода (повышающий быстродействие приложений в сотни раз с помощью одного щелчка мышки), компоновщик, загрузчик программного кода. Кроме этого, CCS обеспечивает поддержку аппаратного отладчика, который позволяет получить доступ в режиме реального времени ко всем модулям DSP. В этой среде можно легко производить профилирование функций и отдельных участков кода. Имеется возможность в реальном времени отображать значение регистров, а также со-

держимое памяти в наиболее удобном виде: осциллограмма, спектрограмма, изображение и т. д.

2. Операционная система Linux — MontaVista. Применение операционной системы позволяет решать огромный круг задач с минимальными затратами ресурсов.

3. Digital Video Software Development Kit (DVSDK), который состоит из следующих программных компонентов:

– eXpressDSP Configuration Kit;

– TMS320DM644x SoC Analyzer;

– версия Linux от MontaVista. Программный пакет eXpressDSP Configuration Kit включает:

• кодеки для видео, изображений, речи и аудио;

• традиционные кодеки, соответствующие стандарту eXpressDSP Digital Media (xDM);

• операционную систему реального времени для DSP — DSP/BIOS.

Уникальное сочетание аппаратных и программных средств, используемых при проектировании устройств на процессорах DaVinci позволило разработчикам говорить о создании высокоэффективной одноименной технологии. Использование этой технологии позволяет разработчику не вникать в технологии проектирования цифровых фильтров, формирования изображений по отдельным пикселям и т.д., а почти полностью сосредоточиться на решении своих прикладных задач.

5.2. Системы контроля за электрофизиологическими параметрами на основе аналоговых интерфейсов AFE

Разработчики современной микроэлектроники поставляют на рынок широкий ассортимент аналоговых интерфейсов подключаемых к различным типам датчиков, которые осуществляют требуемые наборы операций над аналоговыми сигналами (масштабирование, усиление, фильтрацию, коммутацию, контроль качества съема аналоговых сигналов и др.), преобразование сигналов в

цифровой стандартный код и передачу этого кода на дальнейшую обработку. Такие микросхемы в зарубежной литературе относят к классу интерфейсов типа Analog Front End (AFE). Для электрофизиологических исследований на рынок поставляются аналоговые интерфейсы для электрокардиографов, электроэнцефаллографов, измерителей импеданса и т.д. [8, 35, 53, 79, 81].

Рассмотрим подробнее архитектуру наиболее популярных микросхем AFE, используемых при проектировании электрофизиологических приборов.

Применительно к ЭКС фирма Analog Devices разработала специализированную микросхему ADAS 1000 обеспечивающую регистрацию электрокардиосигнала по пяти каналам непосредственно с электродов. Микросхема имеет развитую систему управления позволяющую конструировать интерфейс в зависимости от медико-технических требований.

Входные сигналы могут конфигурироваться внутренним мультиплексором в различные системы отведений (одна микросхема может сформировать до 8 стандартных отведений). Каждый из каналов имеет собственные усилители, фильтры и аналогово-цифровые преобразователи. Предусмотрено дифференциальное подавление синфазных помех, а так же дополнительные схемы подавления помех с помощью подачи сигналов на экраны и удаленный (нога) электрод пациента. Выходной сигнал – последовательный цифровой код стандарта SPI, что позволяет подключать микросхему к различным микроконтроллерам, работающим с тем же протоколом без дополнительного согласования. Микросхема содержит один дополнительный канал для контроля работы системы дыхания.

Перечисленные и ряд других возможностей микросхемы ADAS 1000 позволяют использовать её в медицинских приборах различного назначения, например в системах диагностического назначения, мониторах, в аппаратах для проведения стресс тестов, в дефибрилляторах и т.д. Структурная схема микросхемы ADAS 1000 приведена на рис. 5.11.

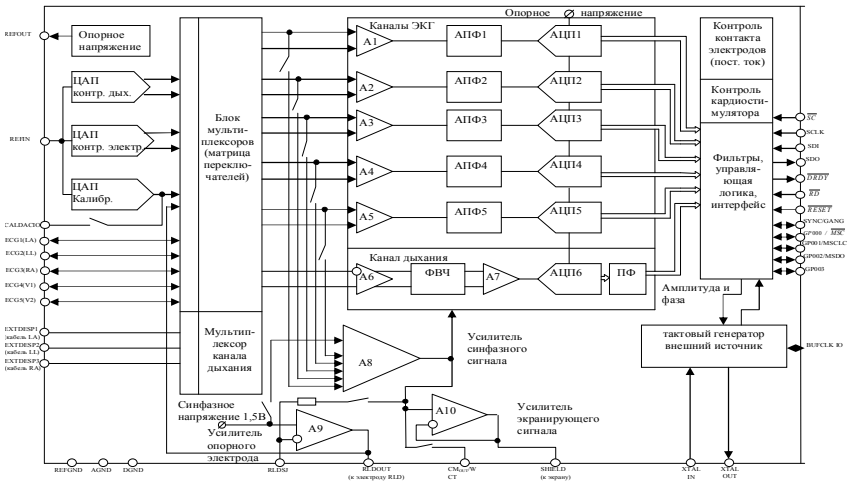


Рис. 5.11. Структурная схема аналогового интерфейса ADAS1000

Для оптимального использования ADAS 1000 в медицинских изделиях необходимо знать её устройство хотя бы на структурно-функциональном уровне, способы её программирования и подключения. Сигналы могут подаваться на входы ECG1,…,ECG5. В некоторых режимах работы на эти электроды могут быть поданы сигналы формируемые интерфейсом ADAS 1000 (при контроле качества прикрепления электродов на постоянном и переменном токе, при контроле дыхания и т.д.), поэтому на рис 5.11 связь мультиплексора с электродами показания двойной стрелкой.

В режиме съема электрокардиосигнала (ЭКС) разность потенциалов с электродов мультиплексором передаётся на дифференциальный усилитель, работающий в диапазоне ±1В относительно синфазной составляющей 1,5В.

Усилитель обеспечивает подавление синфазного сигнала на 110 дБ. Коэффициент усиления может дискретно меняться, но остается небольшим (не более двух). Такой небольшой коэффициент усиления позволят не использовать разделяющих конденсаторов для «подавления» напряжения смещения электродов. Отсутствие емкостей обеспечивает быстрое время восстановления си-

стемы после мощных импульсов дефибрилятора или других электрических перегрузок.

Различные режимы работы входных цепей аналогового интерфейса обеспечиваются системой переключателей блока мультиплексоров (рис. 5.12).

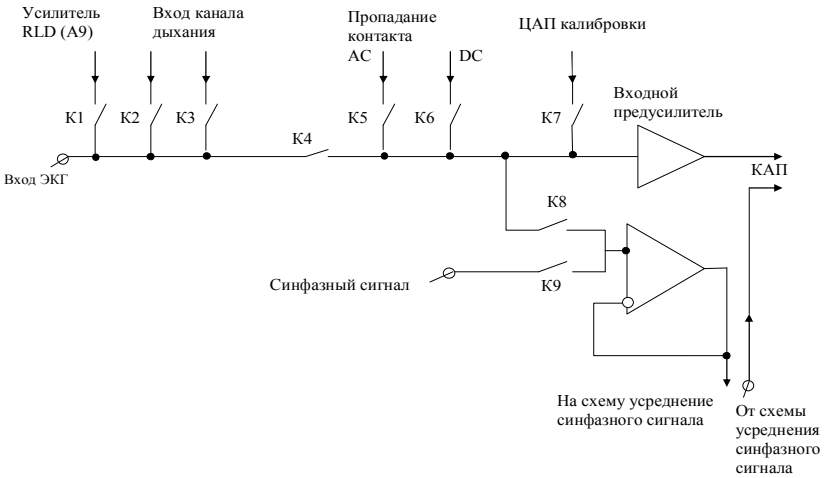


Рис. 5.12. Схема коммутации входных цепей ADAS1000

Например, для организации дополнительного подавления синфазного сигнала сигнал RLD с усилителя опорного электрода A9 должен быть подан на правую ногу (наиболее удаленная от сердца конечность). Для этого размыкаются все контакты кроме К1. При контроле качества контактов на постоянном или переменном токе замыкаются контакты К5 или К6. При включении режима калибровки замыкается контакт К7 остальные размыкаются и т.д.

Усиленный предусилителями A1,…,A5 сигнал подается на вход активных помехоподавляющих фильтров (АПФ1,…,АПФ5, рис. 5.11). Эти фильтры имеют фиксированную частоту среза 65 кГц и обеспечивают дополнительное усиление сигнала с коэффициентом 2,8. Аналогово-цифровое преобразование производится раздельно по каждому из каналов. Аналогово-цифровые преобра-

зователи АЦП1,…,АЦП6 построены по схеме последовательного приближения, что обеспечивает хорошее соотношение сигнал/шум и линейность, а так же низкую мощность рассеивания.

Максимальная скорость работы АЦП обеспечивается на частоте 2 мГц. При этом потребляемая микросхемой мощность максимальна. Имеется возможность программировать скорость преобразования аналог-код.

При снижении скорости преобразования появляется возможность снижать потребляемую мощность так, что ADAS 1000 может быть использована в системах длительного мониторирования работы сердца. Максимальная мощность, потребляемая микросхемой – 22 мВт. В экономном режиме (отключение некоторых функциональных блоков и при низкой производительности) – 11 мВт. Сигналы с выходов АЦП в блоке фильтров, управляющей логики и интерфейса подвергаются усреднению по выборкам, фильтрации и преобразуются в стандартный последовательный код передачи данных по протоколу SPI. Этот протокол физически реализуется четырьмя приводами. В варианте, когда микросхема ADAS 1000 управляется со стороны микроконтроллера (ADAS 1000 имеет подчиненную шину), электрические сигналы шины SPI имеют следующее описание: SDO-выход последовательной передачи данных; SDI-вход последовательного приема данных; SCLK-вход синхронизации приема данных; SC-выбор микросхемы. Чтобы считать данные с ADAS 1000 управляющий микроконтроллер сигналом низкого уровня $\overline{SC}$ выбирает микросхему и посылает ей тактовые импульсы, по входу SCLK одновременно принимая данные с выхода SDO. Микроконтроллер, анализирует фронты сигналов, поступающих с выхода SDO аналогового интерфейса, расшифровывая передаваемые последовательности нулей и единиц. При использовании 16 битных данных частота передачи данных составляет 128 кГц. При 20 битах данных эта частота снижается до 2 кГц.

Сигнал $\overline{DRDY}$ (низкий уровень) сообщает микроконтроллеру о готовности данных в интерфейсе ADAS 1000. Сигнал $\overline{RD}$ (низкий уровень) сообщает о готовности обмена со стороны микроконтроллера.

Группа контактов $GP000/\overline{MSC}$, $GP001/MSCLC$, $GP002/MSD0$, GP003 образуют вторичный интерфейс, управляющий скоростью формирования и передачи данных. Сигналом $\overline{RESET}$ (низкий уровень) обеспечивается общий сброс элементов микросхемы. Выводы REFGND, AGND и DGND предназначены для подключения проводов референтной, аналоговой и цифровой земли соответственно.

Микросхема ADAS 1000 аппаратно поддерживает различные варианты подавления синфазных помех. Усилителем А9 формируется сигнал обратной связи для правой ноги, снимаемый через усилитель А8 с измерительных электродов и подаваемый на соответствующий электрод через вывод RLD OUT. В отличие от схемы рис. 5.12 в микросхеме ADAS 1000 имеется возможность управлять напряжением на электроде подключенном к выходу А9. Здесь предусмотрены такие варианты управления. На электрод RLD можно вывести синфазное напряжение 1,5В с внутреннего источника и компенсирующий сигнал с одного или нескольких электродов (группа переключателей на входах А8). При этом, если контакт на электроде RLD с телом пациента нарушится, то компенсирующий синфазный сигнал перенаправляется на один из измерительных электродов (связь А9 с мультиплексором) поддерживая работоспособность системы.

Усилитель А10 выполняет роль драйвера активного экранирования (аналогично А2 рис. 5.12). Формирование стандартных отведений реализуется вычислительными алгоритмами. Для стандартной пятиэлектродной системы может быть вычислено 8 отведений. Для вычисления 12 стандартных отведений необходимо использовать две микросхемы ADAS 1000.

Вычисления производятся в соответствии с табл. 5.1.

Для проверки качества прикрепления электродов к телу пациента используется два способа: на постоянном и переменном контролирующем токе. При использовании постоянного тока на электрод левой ноги (LL) и измерительные электроды через АЦП контроля электродов и блок мультиплексоров подается небольшое напряжение созда-

ющее текущий от LL ток лежащий в диапазоне 10–70 мкА (устанавливается с шагом 10 мкА), который вызывает небольшое смещение потенциалов на электродах. При нарушении контакта на каком либо из электродов ток, текущий с левой ноги, заряжает соответствующий конденсатор, меняя полярность напряжения на электроде, что фиксируется как отсутствие контакта. Пороги обнаружения положительного и отрицательного уровней полярности могут устанавливаться отдельно по каждому из измерительных каналов.

Таблица 5.1

Вычисление стандартных отведений

Число отведений	Обозначение	Вычислительные формулы
8 отведений, используется одна микросхема	I	LA-RA
	II	LL-RA
	III	LL-LA
	aVR	RA-0,5×(LA+LL)
	aVL	LA-0,5×(LA+RA)
	aVF	LL-0,5×(LA+RA)
	V1’	V1-0,333×(LA+RA+LL)
	V2’	V2-0,333×(LA+RA+LL)
12 отведений, используются две микросхемы	V3’	V3-0,333×(LA+RA+LL)
	V4’	V4-0,333×(LA+RA+LL)
	V5’	V5-0,333×(LA+RA+LL)
	V6’	V6-0,333×(LA+RA+LL)

При использовании переменного тока формируется синусоидальный сигнал с частотой 2кГц, которая лежит выше полосы пропускания усилительных каналов ЭКГ. Качество контакта оценивается по падению напряжения между электродами. Для минимизации общего измерительного тока его величина может отдельно настраиваться для каждого электрода и подаваться на них в противофазе. Использование переменного тока позволяет не только фиксировать отрыв электродов, но и оценивать качество контакта.

При обеих способах оценки качества прикрепления электродов формируются соответствующие флаги, а по интерфейсу SPI передается специальная последовательность двоичного кода.

Одной из опций микросхемы ADAS 1000 является возможность измерения частоты дыхания пациента, которая обеспечивается ка-

налом дыхания с использованием трех выводов EXT DESP1, EXT DESP2 и EXT DESP3 подключаемых к электродам LA, LL и RA. Методика измерения основана на оценке изменения сопротивления грудной клетки в процессе дыхания.

Измерения проводятся на переменном токе силой до 30мкА с частотой 50–56 кГц. Измерительное напряжение формируется с помощью ЦАП контроля дыхания управляемого по входу REFIN. Это напряжение через развязку подается на отведения I, II или III. Таким образом для измерения параметров дыхания не требуется использование дополнительных каналов.

Высокочастотный измерительный сигнал модулированный изменением сопротивления грудной клетки через мультиплексор канала дыхания поступает на вход дифференциального усилителя А6. После усиления с подавлением синфазной помехи низкочастотная (не информативная, включая ЭКГ) составляющая подавляется фильтром высоких частот (ФВЧ) и дополнительно усиливается усилителем А7. Далее модулированный сигнал преобразуется в цифровой код (АЦП6) из которого с помощью цифрового полосового фильтра (ПФ) выделяется информация об амплитуде и фазе дыхательного цикла.

Еще одной дополнительной опцией микросхемы ADAS 1000 является возможность контроля работы кардиостимуляторов, которые могут находиться в теле пациента. Импульсы кардиостимулятора определяются с трех или четырех отведений (I, II, III, VF). Блок контроля кардиостимулятора определяет импульсы амплитудой от 400мкV до 250 mV с длительностью от 100 мк до 4мс. При этом формируются соответствующие кодовые посылки по интерфейсу SPI.

Настроенная под задачи пользователя микросхема ADAS 1000 формирует потоки кодовых сообщений по интерфейсу SPI. Для построения кардиографов эти потоки должны быть соответствующим образом переработаны для принятия диагностических заключений и отображения на экранах графических мониторов. Для этих целей используются микропроцессоры, микроконтроллеры и ПЭВМ.

Например, микросхема ADAS 1000 может быть напрямую подключена к комбинированному RISC-DSP процессору Blackfin.

RISC-ядро может решать задачи управления кардиографом, обработки данных, подготовки для отображения не экранах мониторов. Задачи фильтрации изолинии, подавления шумов, детектирования R-зубца и др. могут решаться сигнальными процессорами.

Для защиты пациентов и усилительных каскадов необходимо использовать специальные схемы, развязывающие усилители, диодные ограничители и т.д.

С учетом сказанного на рис. 5.13 приведена структурная схема электрокардиографа на восемь стандартных отведений (верхняя часть табл. 5.1).

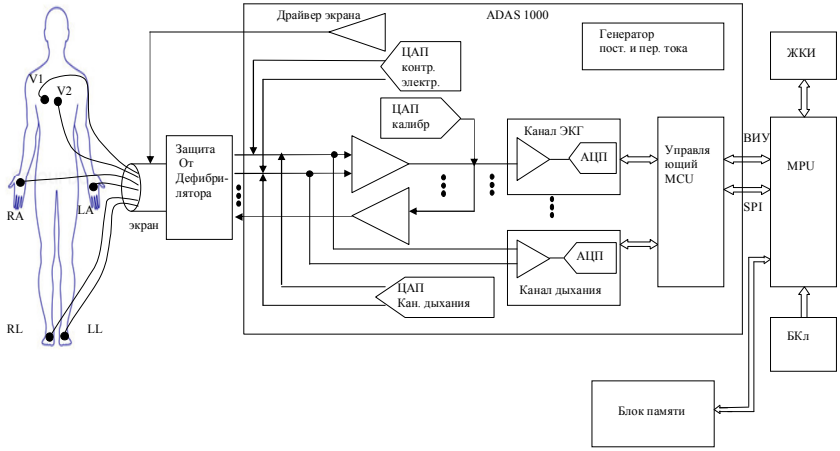


Рис. 5.13. Микропроцессорный кардиограф
с аналоговым интерфейсом ADAS1000

В этой схеме внутри микросхемы ADAS 1000 показана конфигурация соединений обеспечивающая обратную связь на ногу пациента, драйвер экранирования, контроль качества прикрепления электродов и контроль дыхания.

Микросхема ADAS 1000 подключена к микроконтроллеру (MPU) по шине SPI и через вторичный интерфейс управления (ВИУ). Блок памяти обеспечивает хранение фрагментов ЭКС и результаты вычислений. Взаимодействие кардиографа с пользователем обеспечивается блоком клавиатуры (БКл), а отображение всей необходимой информации жидкокристаллическим индикатором (ЖКИ).

Для решения задач мониторинга при малом потреблении энергии фирма Texas Instruments выпускает микросхему, ориентированную на регистрацию электрокардиограмм (ЭКГ) и электроэнцефалограмм (ЭЭГ) (рис. 5.14).

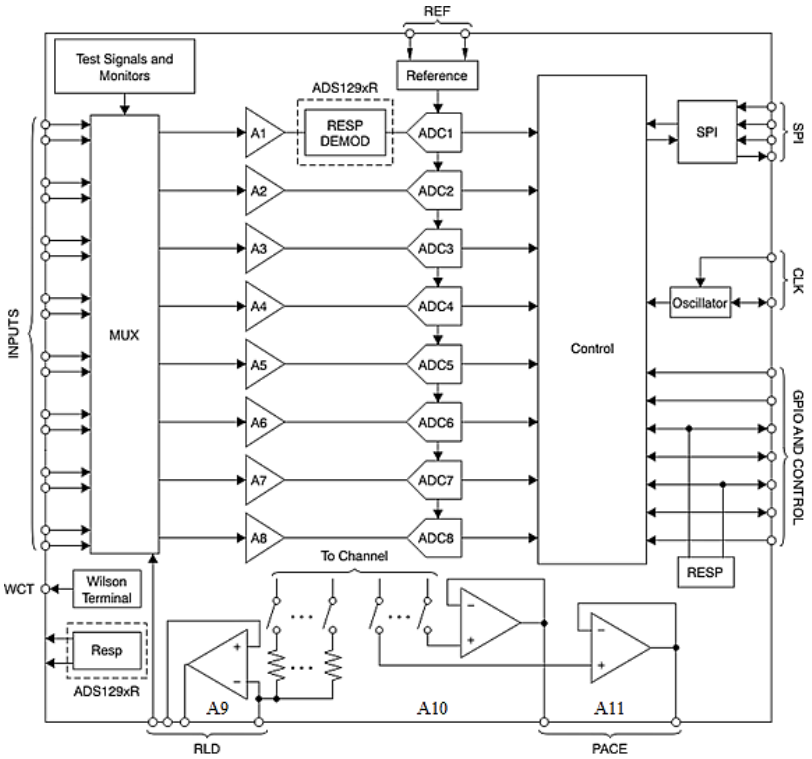


Рис. 5.14. Структура микросхемы типа AFE ADS1298R

В этой микросхеме содержатся: входной мультиплексор MUX, восемь малошумящих усилителей A1,…,A8 с программируемым коэффициентом усиления (в полосе частот 0,5,…, 150 Гц входной шум составляет 4мкВ); восемь параллельно работающих АЦП высокого разрешения; усилитель драйвера правой ноги A9; усилители синфазного сигнала (A10) и драйверы экрана (A11). Управление и формирование стандартного цифрового кода SPI обеспечивается контроллером (Control). Энергопотребление микросхемы ADS1298R составляет 1 мВт на каждый канал.

Микросхема ADS1299 позиционируется как 24-битный 8-канальный постоянно собирающий данные модуль для ЭЭГ, который имеет уровень шума на входе всего в 1 мкВ на частоте 70 Гц. В этой микросхеме имеется 8 малошумящих усилителей с программируемым коэффициентом усиления. Индекс R в названии микросхемы означает наличие дополнительного канала дыхания.

Низкий уровень шумов входных усилителей позволяет регистрировать внутричерепные потенциалы при высоком уровне шумов.

Коэффициент ослабления синфазного сигнала у этой микросхемы 120дБ. Имеется встроенный механизм контроля за качеством прикрепления электродов, тестовые сигналы, усилитель смещения, генератор и источник опорного напряжения.

Специально для 2–3-канальных кардиомониторов фирма Texas-Instruments выпускает микросхемы ADS1293, ADS1294 с очень низким энергопотреблением и требует минимальной обвязки.

Специализированные микросхемы типа AFE могут питаться от литий ионных и литий полимерных аккумуляторов с выходным напряжением 3,7В заряжаемых бесконтактным способом через индукционные зарядные устройства (катушка переменного тока на источнике энергии, приемная катушка с выпрямителем со стороны заряжаемой батареи). Для питания аналоговой части цифрового ядра может быть использовано напряжение 3,3В, что позволяет между аккумулятором и устройством использовать линейные стабилизаторы напряжения, например типа ADP151 фирмы Analog Devices. Для внутренней радиосвязи (например, внутри кардиологического отделения) удобно использовать технологию и инструментарий (микросхемы) Bluetooth для внешней связи и позицио-

нирования удобно пользоваться технологией и оборудованием спутникового позиционирования, например, GPS или GLONASS.

В кардиомониторах собираемая информация используется для расчета ряда параметров (ЧСС, аэробная мощность, физическая работоспособность и др.)

С этой целью используется вычислительное ядро, например, микроконтроллер STM32L который при малом энергопотреблении реализует требуемые вычисления и различные интерфейсы связи.

Дополнительное дублирование регистрируемой информации может осуществляться на твердотельном носителе (например, SD карте).

С учетом сказанного перспективная структура кардиомонитора может иметь вид приведенный на рис. 5.15.

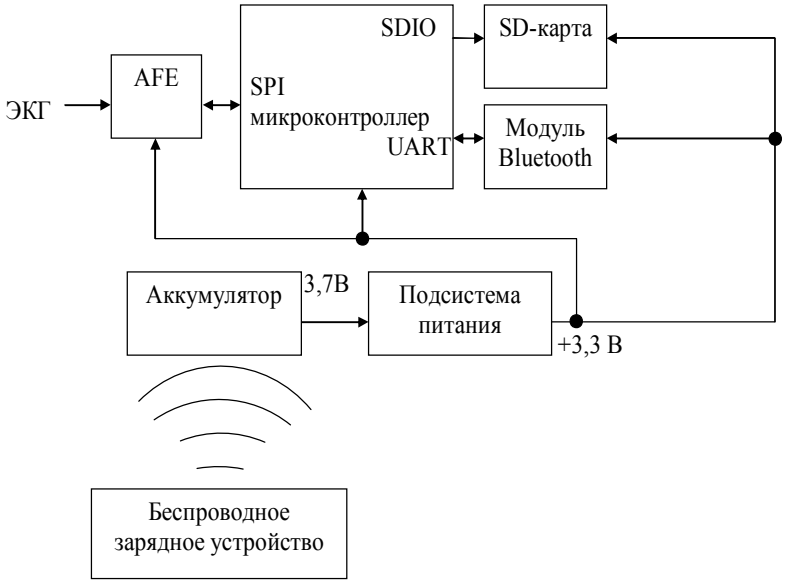


Рис. 5.15. Структурная схема кардиомонитора со специализированной микросхемой AFE

Беспроводные зарядные устройства строятся на основании принципа электромагнитной индукции в соответствии с которым

переменный ток протекающий в передающей катушке наводит индукционный ток в близко расположенной приемной катушке. На рис. 5.16 приведена упрощенная схема беспроводного зарядного устройства.

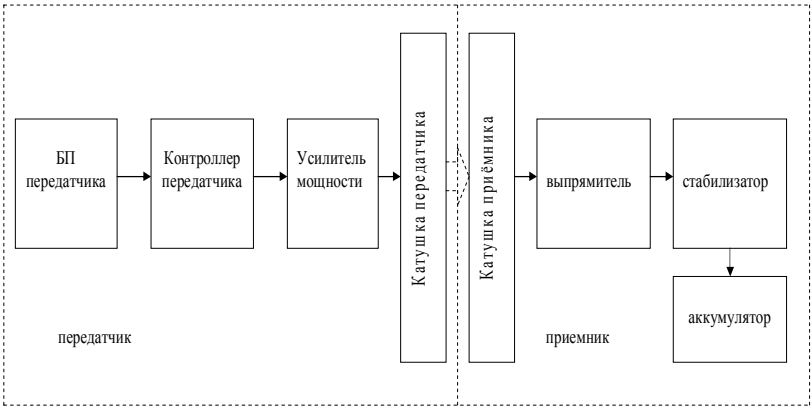


Рис. 5.16. Структурная схема беспроводного зарядного устройства

В этой схеме блок питания передатчика (аккумулятор или сетевой блок питания) обеспечивает работу контроллера и усилителя мощности. Контроллер передатчика через усилитель мощности создает в катушке передатчика переменное напряжение частотой от 90 до 120 Гц. Кроме того контроллер следит за уровнем напряжения и тока в катушке передатчика и за положением катушки приемника. Наведенный в катушке приемника электрический ток выпрямляется, стабилизируется и заряжает аккумулятор. Катушки приемника и передатчика имеют плоскую форму и при зарядке располагается друг над другом.

Для связи с удаленными вычислительными структурами может быть использована система позиционирования и дальней связи глобальных систем GPS и GLONASS. Вариант структурной схемы такой системы приведен на рис. 5.17.

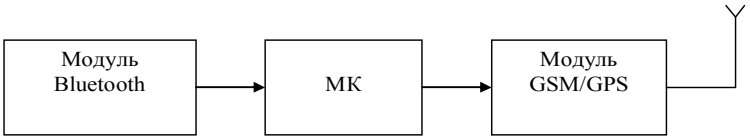


Рис. 5.17. Структурная схема системы позиционирования и дальней связи

В этой схеме в качестве модуля Bluetooth может быть использована микросхема BLE112 со встроенным микроконтроллером 8051. Для увеличения свободной памяти встроенного микроконтроллера используется внешний микроконтроллер МК. Дальняя связь и позиционирование реализуется модулем GSM/GPS (микросхема SIM 968).

5.3. Обобщенные схемы приборов для исследования неэлектрических характеристик организма и окружающей среды

Как и в аппаратуре для электрофизиологических исследований, основным источником информации в приборах, аппаратах и системах, исследующих неэлектрические характеристики организма и среды его обитания, являются датчики, преобразующие измеряемые показатели в нормированные электрические сигналы. В зависимости от типа измеряемой величины примеряются физические и химические датчики. Физические датчики измеряют геометрические, механические, оптические, термические, гидродинамические и т.д. величины. Химические датчики измеряют состав и концентрацию различных веществ. Технология изготовления биомедицинских датчиков в значительной мере базируется на технологиях микроэлектроники, причем наблюдаются тенденции размещения в одной микросистеме датчиков с элементами предварительной обработки информации, включая контроль работоспособности, калибровку, различные типы компенсаций, микроконтроллеры, ка-

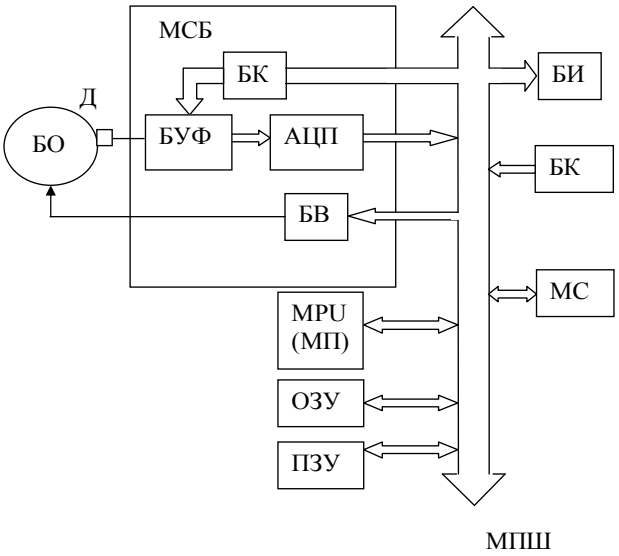
налы коммуникаций и т.д. Примерами промышленных микросистем являются механо-электрические микросистемы, миниатюрные системы комплексного анализа, которые могут оперировать с микропотоками жидкостей и газов, и др.

Подробно техническая реализация датчиков различных типов и назначений рассматривается в соответствующих учебных курсах. Так же, как и в электрофизиологической аппаратуре, часто сигналы с датчиков не могут непосредственно управлять регистрирующей аппаратурой или передаваться на цифровую обработку. В этом случае к ним подключаются усилители, фильтры, аналого-цифровые преобразователи и т.д. Причем этот набор естественно зависит от регистрируемых параметров и методов исследований. Часть методов исследований предполагает наличие воздействий на биообъект и (или) обязательного управления элементами измерительной аппаратуры, что требует введения в состав проектируемой аппаратуры дополнительных узлов и блоков.

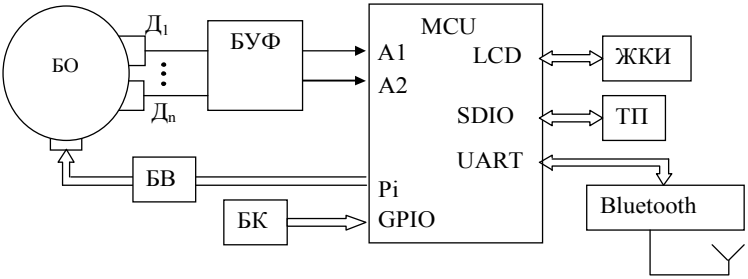
Таким образом, обобщенные структурные схемы измерения электрофизиологических показателей, могут быть использованы и при проектировании приборов для измерения неэлектрических показателей, при условии что датчики (сенсоры, интеллектуальные сенсоры) преобразуют измеряемые физические величины в нормированный электрический сигнал.

С учетом того, что современная медицинская техника предлагает широкий арсенал средств цифровой обработки сигналов, часто с достаточно мощной встраиваемой аналоговой и аналогово-цифровой частью, в дополнение к рис. 2.1, рассмотрим еще несколько типовых вариантов построения исследуемого класса измерительных приборов медицинского и экологического назначений (рис. 5.18).

На рис. 5.18, *а* приведен вариант классической микропроцессорной системы с микропроцессорным ядром (MPU (МП)). Сигнал с датчика (Д) при необходимости усиливается и фильтруется блоком усиления и фильтрации (БУФ). Аналогово-цифровой преобразователь (АЦП) преобразует нормализованный и «очищенный» сигнал в цифровой код.


МСБ
БК
Д
БО
БУФ
АЦП
БВ
БИ
БК
МС
MPU
(МП)
ОЗУ
ПЗУ
МПШ

а)


БО
Д1
Дn
БУФ
БВ
БК
MCU
A1
A2
Pi
GPIO
LCD
SDIO
UART
ЖКИ
ТП
Bluetooth

б)

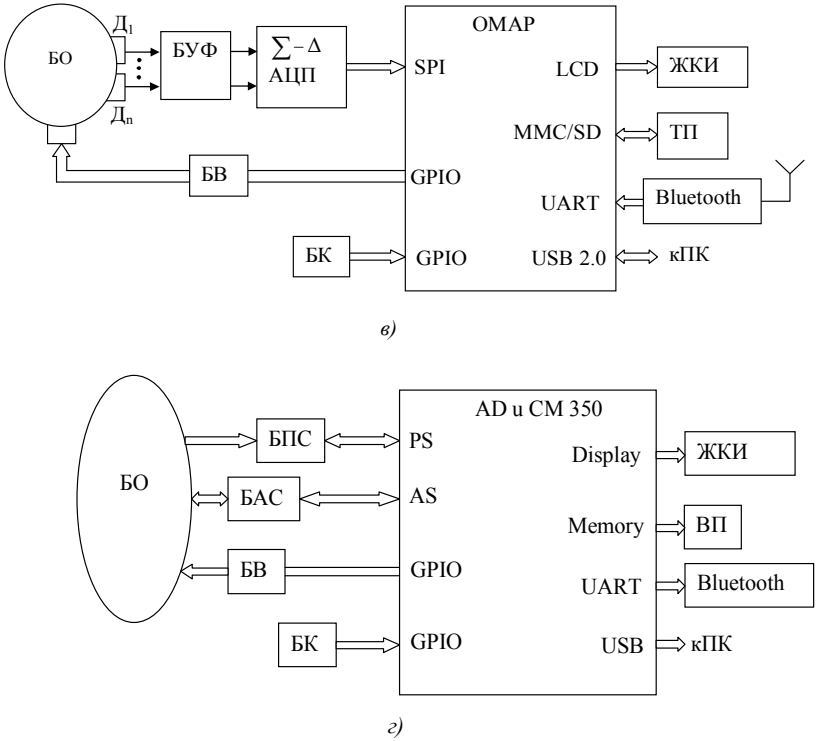


Рис. 5.18. Обобщенные структурные схемы приборов для исследования неэлектрических характеристик биообъектов и окружающей среды

Блок калибровки (БК) (если это требуется по условию работы) обеспечивает калибровку (изменение коэффициентов усиления, смещение нуля усилителей, изменение параметров фильтров) модуля сопряжения с биообъектом (МСБ).

Если технология измерения требует подведения внешних воздействий (например, формирование светового пучка при фотоплетизмографии), то в состав МСБ включается блок воздействия (БВ). Остальные блоки микропроцессорной системы (оперативная память (ОЗУ), постоянная память (ПЗУ), блок индикации (БИ), блок клавиатуры (БК), модуль сопряжения (МС)) выполняют те же

функции, что и в приборах для электрофизиологических исследований.

В настоящее время для различных приложений, включая медицину и экологию, выпускаются микроконтроллеры (MCU) имеющие в своем составе встроенные, достаточно качественные АЦП (STM32F – 3 АЦП ПО 8 коммутируемых аналоговых накалов на каждый из них; PIC 24FJ128GC010 – два АЦП; малоэнергопотребляющий MSP430F5xxx – двенадцатиразрядный АЦП и т.д.). Использование такого типа микроконтроллеров позволяет реализовать приборы со структурной схемой, приведенной на рис. 5.18, *б*. Блок усилителей и фильтров используется при необходимости согласования характеристик датчиков и АПЦ и подавления помех от источника сигнала. Управление блоком воздействия обычно осуществляют через порты микроконтроллера Pi или через программно поразрядно управляемый интерфейс GPIO, который может быть настроен и на восприятие команд и данных с блока клавиатуры. Для управления жидкокристаллическим индикатором (ЖКИ) целесообразно использовать встроенный контроллер LCD, а для использования твердотельной памяти (ТП) – интерфейс типа SDIO.

Радиомодуль Bluetooth подключается к тому типу интерфейса, на который он ориентирован, например UART, SPI.

Удобно при проектировании средств контроля за состоянием биообъектов и окружающей среды использовать прикладные процессоры типа OMAP. Отличительной чертой прикладных процессоров является обязательное наличие не менее двух ядер на одном кристалле. В микросхемах типа OMAP встроено два ядра: центральное микропроцессорное устройство (подсистема ARM) и сигнальный процессор (подсистема DSP).

Такое сочетание позволяет решать задачи обработки данных в сочетании с эффективной цифровой фильтрацией. Типовая структура измерительного прибора с использованием прикладного процессора приведена на рис. 5.18, *в*. Отличительной чертой этой схемы является гораздо более простая реализация БУФ, где вместо высококачественных инструментальных усилителей достаточно использовать операционные усилители с простейшими пассивными фильтрами низкой частоты, улучшающими качество работы $\sum$-Δ АЦП.

Остальная фильтрация осуществляется PSP ядром OMAP. Промышленно-выпускаемые прикладные процессоры, имеют встроенный контроллер протокола USB 2.0,что позволяет «напрямую» подключать их к персональному компьютеру (ПК), не забывая, при необходимости, о гальванической развязке.

Специально для многоканальных систем регистрации разнородной медико-биологической и экологической информации фирма Texas Instruments выпускает прецизионный чип ADuCM 350, который на одном кристалле содержит многофункциональный аналоговый коммутатор, аналоговые и цифровые фильтры, АЦП и ЦАП, разветвленную систему периферийных интерфейсов и достаточно мощный процессор ARM Cortex M3.

Все это позволяет подключать ко входам чипа пассивные датчики (электроды, датчики давлении, температуры и т.д.) и активные датчики (аналоговые интерфейсы типа AFE, интеллектуальные датчики, работающие с протоколами GPIO, I2S, I2C, SPI, UART и т.д.)

Типовая структура измерительного прибора на основе ADuCM 350 приведена на рис. 5.18, *г*.

К пассивным датчикам (блок пассивных сенсоров (БПС), входы PS) ADuCM350 встроенный 16-разрядный АЦП подключается через двухсторонний коммутатор (коммутаторную матрицу), инструментальный усилитель и фильтр низких частот.

Для подключения к активным датчикам (блок активных сенсоров (БАС), входы AS) используются встроенные периферийные интерфейсы. Блок клавиатуры и блок воздействия могут быть подключены через побитно управляемый интерфейс GPIO. Для подключения ЖКИ в чипе предусмотрен интерфейс Display. Дополнительная внешняя память (ВП) подключается через интерфейс Memory. Через интерфейс UART подключается радиомодуль Bluetooth, а через интерфейс USB 2.0 ADuCM 350 может быть подключен к компьютеру.

Для контроля за состоянием окружающей среды выпускаются специализированные аналоговые интерфейсы типа AFE, которые «напрямую» подключаются к соответствующим датчикам по схеме, приведенной на рис. 5.18, *г*.

В биотехнических системах, где велика доля операций, связанных с формированием и обработкой изображений (видеопотоков) целесообразно использовать микросхемы, изготавливаемые с использованием технологии Da Vinci (компания Texas Instruments), ориентированные на обработку видеоизображений. В микросхемах, изготовленных с использованием этой технологии к мощным ARM и DSP процессорам, добавляется процессор видеосигналов (Video Processing Subsystem) и мультимедиа кодеки при сохранении мощной периферии прикладных процессоров с гибкой системой управления внешней памятью.

Структура приборов на основе микросхем, ориентированных на технологию Da Vinci, близка к структуре, приведенной на рис. 5.18, *в*. При реализации иных функциональных систем с повышенными требованиями к обработке видеоинформации целесообразно объединить структуры рис. 5.18, *в* и *г*.

Обобщенные схемы дают общее представление о структуре систем для измерения не электрических характеристик биообъектов и окружающей среды. Конкретное проектирование требует достаточно глубоких знаний по структуре, стандартному программному обеспечению и способах программирования выбранных микроэлектронных устройств.

Специально для пульсоксиметрии, мониторинга сердечного ритма и измерения глюкозы в крови, фирма Texas Instruments разрабатывает аналоговые интерфейсы AFE серии AFE44xx. Промышленность на мировой рынок выпустило две микросхемы AFE4400 – для домашнего применения и AFE4490 для клинического применения. Обе микросхемы имеют похожие структуры, но AFE4490 имеет более широкий динамический диапазон измерений – до 110 дБ обеспечиваемый 22-разрядным Σ-ΔАЦП. У AFE4400 – динамический диапазон уже – 95 дБ.

Структурная схема пульсоксиметра с использованием AFE4490 приведена на рис. 5.19.

В этой схеме программируемый интерфейс с драйвера светодиодов (LED DRIVER) обеспечивает рабочий ток красного и инфракрасного светодиодов (LED) до 75 мА. Программным путем обеспечивается независимое управление светодиодами с программиру-

емым временем их свечения. Фотоприемник (Photodiode) подключается к программируемому малошумящему инструментальному усилителю (TIA).

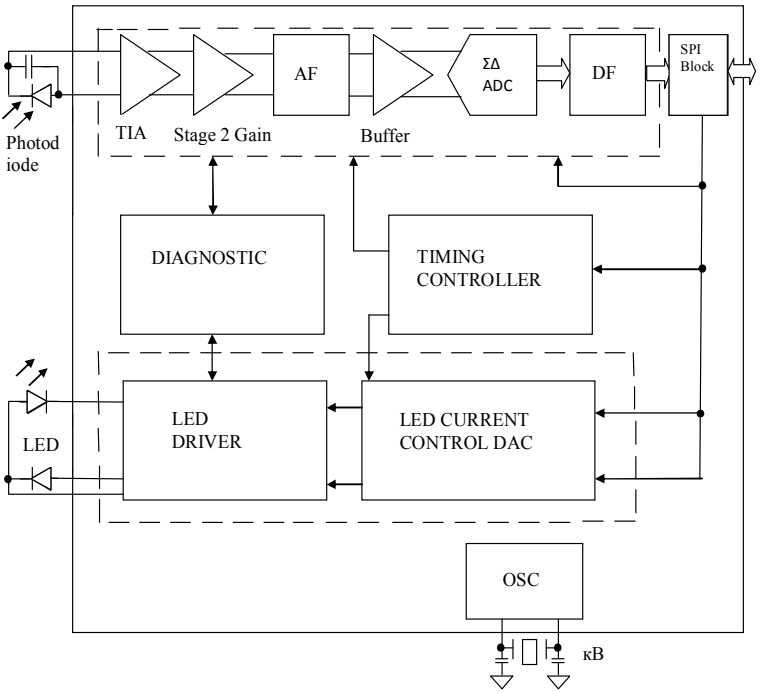


Рис. 5.19. Структурная схема AFE4490, включенного для пульсоксиметрии

Далее включен программируемый (с изменяемым коэффициентом усиления) масштабный усилитель и программируемый аналоговый фильтр низких частот (AF). После буферного усилителя (Buffer) включены 22-разрядный Σ-ΔАЦП и цифровой программируемый фильтр (DF). Связь с другими цифровыми устройствами осуществляется по протоколу SPI (SPI Block). По этому же протоколу через SPI Block осуществляется программирование AFE4490. Блок диагностики (DIAGNOSTIC) позволяет определять ряд неис-

правностей (обрыв или короткое замыкание в цепях фотоприемника и светодиодов, подключение кабеля и т.д.).

Специально для физико-химических анализаторов фирма Texas Instruments (TI) выпустила линейку микросхем серии LMP (см. табл. 5.2).

Эта серия представляет собой AFE с высокой точностью регистрации аналоговых сигналов при низком потреблении энергии. В ней предусмотрена оценка качества работы сигнального тракта либо в ходе измерительного процесса (онлайн) либо на отладочных стендах. Упрощенная структурная схема LMP901XX приведена на рис. 5.20.

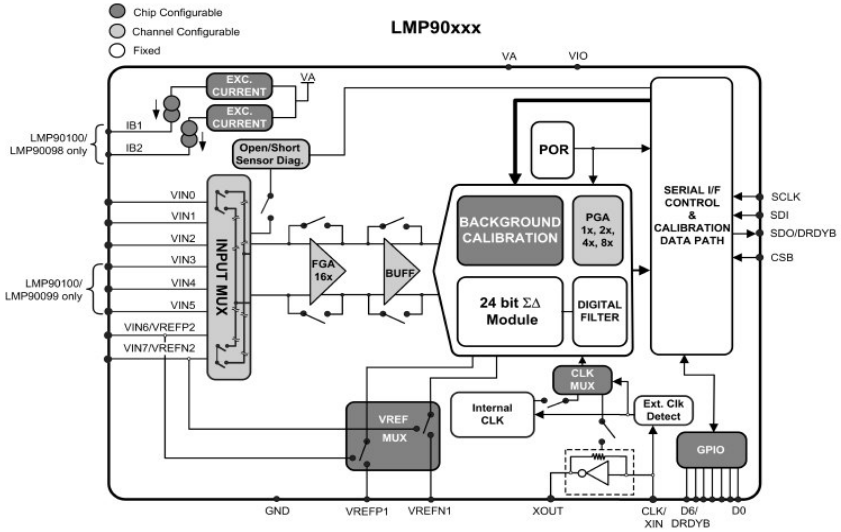


Рис. 5.20. Структурная схема AFE измерителя LMP901XX

Ко входам микросхемы VIN0,…,VIN7 могут быть одновременно подключены восемь датчиков которые через входной мультиплексор (INPUT MUX) в программно определяемом режиме коммутируются со входами малошумящего инструментального усилителя с перестраиваемым коэффициентом усиления (FGA). Далее

через буферный усилитель (BUFF) аналоговый сигнал поступает на вход 24-разрядного $\sum$-ΔАЦП($\sum$-ΔModule). Аналогово-цифровая часть тактируется встроенным (CLK) и (или) внешними (вывод CLK/XIN) генераторами.

Таблица 5.2

Микросхемы фирмы Texas Instruments для физико-химических измерений

Наименование	Описание	Тип датчиков (примерный перечень)
LMP901XX	Семейство многоканальных 16 и 24 битных AFE для датчиков	Датчики температуры RTD (резистивный температурный датчик) NTC (терморезистивные датчики (температуры)) термопары резистивные мосты
LMP9100X	Экономичные AFE для химических датчиков и датчиков газов	Датчики CO_2, CO, O_2, H_2S
LMP9105X	Конфигурируемые AFE для датчиков NDIR, CO_2, CH_4, паров алкоголя	Датчики NDIR (инфракрасный фотометр) CO_2, CH_4, пары алкоголя
LMP91200	Конфигурируемые AFE экологичных приложений на базе химических датчиков	Датчики PH
LMP91300	Индуктивные датчики расстояний	Индуктивный датчик

Отличительной особенностью LMP901XX является блок диагностики состояния датчика (Open/Short Diag.) решающий задачи определения состояния датчиков (обрыв, короткое замыкание, отклонение амплитуды входного сигнала за пределы рабочего диапазона без участия пользователя).

Кроме стандартного интерфейса SPI цифровой узел (SERIALI/F CONTROL & CALIBRATION DATAPATH) реализует побитно управляемый протокол GPI0, позволяя эффективно управлять переключателями, светодиодами, реализовывать гальваническую развязку и т.д.

В старших микросхемах семейства имеется два согласованно работающих программно управляемых источников тока (EXC. CURRENT), для работы с мостовыми (питание моста) и терморезистивными датчиками.

Еще одной отличительной особенностью семейства LMP901XX является наличие механизма калибровки, основной блок реализации которого (BACKGROUND CALIBRATION), встроен в $\sum$-ΔАЦП. Блок калибровки позволяет оценивать и компенсировать погрешности от изменения напряжения смещения и коэффициента усиления аналоговых усилителей возникающие при изменении напряжения питания, температуры и эффекта «старения» от времени. В микросхемах серии LMP901XX реализуется два типа калибровки: системная и фоновая.

При системной калибровке по каждому из каналов определяются ошибки измерения из-за смещения и усиления для всей измерительной системы. Эти ошибки вычитаются как прописанные заранее поправочные коэффициенты из окончательного результата измерений.

Режим фоновой калибровки реализуется автоматически и рассчитан на компенсацию ошибок возникающих внутри микросхемы. Такой тип калибровки практически не оказывает влияния на входной сигнал. В режиме фоновой калибровки на короткое время на входы усилителя FGA подается известный потенциал, например нулевой, и путем перекоммутации прямого и инверстного входов вычисляют напряжение смещения, а при дополнительной перекоммутации цепей обратной связи FGA и BUFF вычисляют поправочный коэффициент для коэффициента усиления. Полученные поправочные значения вычитаются из выходного кода сигнала. Предусмотрено два типа калибровки (коррекции). Производится отслеживание изменения напряжения смещения ($U_{СМ}$) и коэффициента усиления $К_У$. Полученные в результате отслеживания параметры вычитаются из результата измерений. При втором типе калибровки (предсказание) используется последнее измеренное значение ошибки.

Эти значения могут быть взяты из таблицы созданной пользователем и зашитой в ПЗУ, а могут быть вычислены по методу коррекции.

Для различных режимов измерения предусмотрены различные варианты формирования опорного напряжения V_{REF} которое коммутируется с помощью VREF MUX на выходы VIN6 и VIN7 либо от $\sum -\Delta АЦП$ либо с внешнего источника от входов VREFP1 и VREFP2. Очевидно при этом, что датчики подключенные к VIN6 и VIN7 игнорируются.

Подключение кварца к выводам XOUT и CLK/XIN создает дополнительный генератор CLKI.

В качестве примера на рис. 5.21 приведена типовая схема включения LMP-90100.

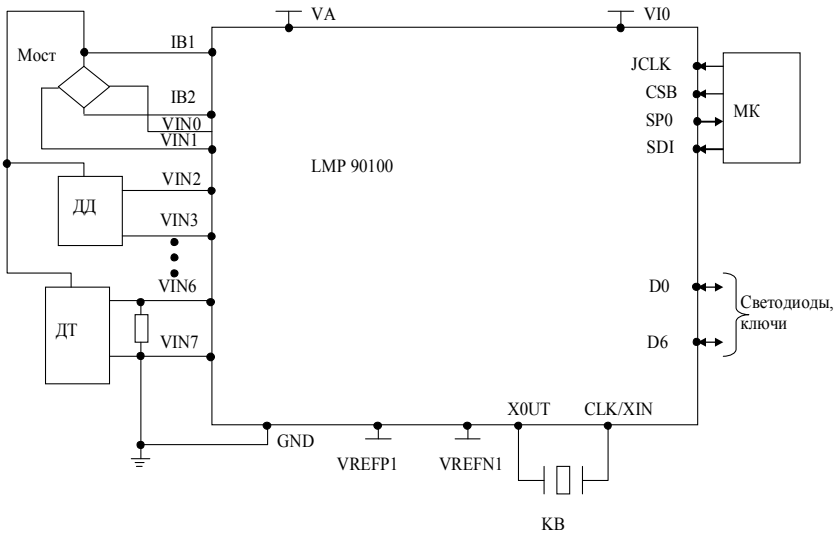


Рис. 5.21. Типовая схема включения LMP-901XX

На схеме рис. 5.21 питание мостового датчика осуществляется от двух внутренних источников тока, один из которых питает датчики давления (ДД) и температуры (ДТ). Микроконтроллер (МК) подключен к микросхеме через порт SPI. Частота работы тактирующего генератора задается внешним кварцем (КВ). К порту типа GPI0 подключен набор светодиодов и управляющих ключей, например компрессором тонометра. Светодиоды могут сигнализировать о выходе частоты сердечных сокращений и артериального давления за рамки нормальных значений.

Для настройки LMP901XX и любых других микросхем данного семейства фирма TI предлагает среду разработки WEBENCH Sensor AFE представляющей собой веб-приложение.

Для удобства работы веб-приложения на начальной странице по каждой из микросхем формирует блок-диаграмму аналогичную тому как это изображено на рис. 5.20, причем каждый из узлов микросхемы окрашен в определенный цвет определяющий назначение и доступность для пользователя. На этой же странице изображен путеводитель (Help bar) который помогает в работе с приложением в случае возникновения тех или иных затруднений.

Если «кликнуть» выбранный узел, то появляется информация о его возможных настройках. Для удобства пользователя в приложении есть база данных на различные типы промышленных датчиков которая может быть расширена пользователем на новые датчики.

Работа с приложением начинается с выбора датчика (Select Sensor). Датчик можно выбрать из предложенного в виде стандартного окна списка или ввести его параметры вручную. После этого шага приложение предложит пользователю схему подключения датчиков в виде рисунка где справа изображена схема аналогичная рис. 5.20, а слева изображаются датчики с выбранными ножками для подключения. Если подключение пользователя не устраивает, выбранные контакты связи могут быть переназначены. При выборе каналов подключения следует иметь ввиду, что подключение может быть дифференциальным ($U_{ВЫХ}=U^{+}_{ВХ}-U^{-}_{ВХ}$) и несимметричным, когда один из входов заземлен. При дифференциаль-

ном включении $U_{ВХ}^{+}$ и $U_{ВХ}^{-}$ могут быть сформированы из любых каналов. При несимметричном подключении возможна организация семи каналов, но один из каналов должен быть обязательно заземлен. На следующем этапе производится установка токов (Source IB1/IB2).

Значение тока может быть установлено в диапазоне 0,1,…,1,0 мА с шагом 100 мкА. Величина тока естественно зависит от тока датчика. Выбор значений тока осуществляется с использованием окна, вариант конфигурации которого приведен на рис. 5.22, *а*.

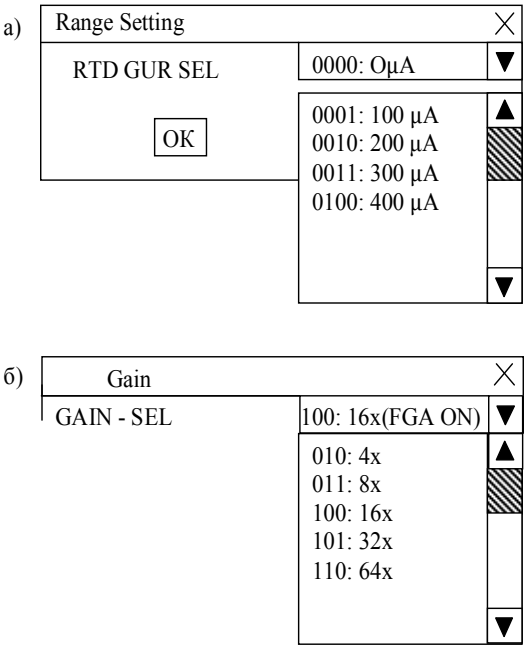


Рис. 5.22. Варианты окон настройки параметров узлов LMP901XX

Пример окна настройки коэффициентов усиления FGA приведен на рис. 5.22, *б* и т.д.

В ходе настройки можно увидеть как меняются параметры работы измерительной системы (ток потребления, уровень шума, вносимая погрешность измерения и др.).

Для настройки измерительной системы фирма TI выпускает отладочную плату LMP90100EB подключаемую к компьютеру.

С описанием других электронных средств используемых для физико-химических анализаторов можно ознакомиться в соответствующей литературе, включая интернет-издания.

5.4. Использование микромеханических систем, выполненных по технологиям МЭМС, в телемедицинских системах

Широкое распространение при проектировании медицинских приборов различных типов и назначений находят микроэлектромеханические системы (МЭМС) – интегрированные микро устройства или системы, сочетающие электрические и механические компоненты, изготовленные по технологиям, совместимым с технологией интегральных схем и имеющие размеры от микрометров до миллиметров.

Микромеханические сенсоры давления применяются для измерения давления крови в кардиохирургии и пластической хирургии при сложных операциях на сосудах, во внутриматочных сенсорах давления и датчиках-преобразователях, контролирующих давления вокруг головы младенца в течение всего периода беременности и родов.

Микромеханические датчики давления также широко применяются в насосах вливания лекарственных средств, которые наиболее востребованы людьми больными диабетом, в респираторах и системах искусственного дыхания, в эндоскопии, а также в почечных механизмах диализа и других медицинских устройствах [47, 94, 95].

Достаточно широкое применение МЭМС находят при изготовлении хирургических инструментов, что позволяет хирургу более

точно контролировать работу хирургического инструмента, получать информацию о плотности ткани, ее температуре и составе.

Это позволяет лучшими методами производить подготовку ткани к операции, а также быстро осуществлять извлечение образцов ткани и жидкостей для исследования. МЭМС микрочипы используют в качестве датчиков потока и скорости крови (УЗ-датчики), которые встраиваются в наконечники сверхмалых катетеров. В наконечник катетера микросистемы может быть встроена матрица термисторов, используемых для считывания распределения скорости крови по кровеносному сосуду. Это важно знать для определения частичной закупорки артерии. Разрабатываются датчики для катетеров, которые используются в предродовых хирургических операциях на сердце еще не родившегося ребенка (операции на сердце плода) [47].

Высокоточные химические сенсоры используются для точного оперативного определения химического состава крови или выдыхаемого человеком воздуха, как в домашних условиях, так и в условиях оказания экстренной медицинской помощи. Данные сенсоры применяются в автономных медицинских устройствах и в аппаратах искусственного дыхания и кровообращения.

Еще одна биомедицинская МЭМС-технология – это так называемая «лаборатория на чипе» (Lab-on-a-Chip Technology) или биологический микрочип (биочип). Основой первых биочипов, созданных в Институте молекулярной биологии им. В.А. Энгельгарда РАН, является матрица полусферических гидрогелиевых ячеек (диаметром менее 100 мкм), каждая из которых содержит зонд, специфичный к одной из множества биологических молекул (например, последовательностям ДНК или РНК, белкам). На 1 $см^2$ может быть размещено до 1000 ячеек биочипа для анализа мутаций, определяющих предрасположенность к различным наследственным и онкологическим заболеваниям, для обнаружения патогенных бактерий и вирусов и выявления их лекарственно-устойчивых форм. На сегодняшний день технология биочипов является наиболее перспективным направлением в медицинской диагностике [47].

МЭМС нашли свое применение в бижелудочных электрокардиостимуляторах, отличающихся от первых электрокардиостимуляторов своими миниатюрными размерами и принципом работы. Электрокардиостимуляторы первого поколения подавали электрические импульсы только левому желудочку, в то время как бижелудочные электрокардиостимуляторы, имеющие две стимуляционные камеры, уменьшают нагрузку на сердце и имеют лучшее параметры синхронизации. Большинство современных электрокардиостимуляторов имеют встроенный дефибриллятор, способный посредством шоковой терапии замедлять неестественно быстрый ритм работы сердца и осуществлять сердечную стимуляцию.

В настоящее время ведутся разработки по созданию нейростимуляторов на базе МЭМС, которые способны электрическими импульсами воздействовать на нервную систему человека. Внедренные в мозг нервные электростимуляторы могут блокировать нервную передачу признаков эпилепсии, тремора конечностей при болезни Паркинсона, Альцгеймера, звона в ушах и др. Эти устройства также могут изменять поведение, связанное с депрессией, повышенным беспокойством и т.д. Помимо вышеперечисленных специализированных микро электромеханических компонентов в медицинской электронике применяются и микро электромеханические компоненты универсального назначения. Они используются не только в приборах функциональной диагностики, лабораторной технике, стерилизационном оборудовании, наркознодыхательной аппаратуре, физиотерапевтических аппаратах, но также широко распространены и в современных цифровых рентгеновских, ультразвуковых и магниторезонансных системах [47].

Рассмотрим более подробно типовые структуры МЭМС используемые в медицинском приборостроении.

Для исследования механической активности биообъектов широкое применение находят МЭМС акселерометры позволяющие с высокой точностью измерять такой параметр движения как ускорение.

Основными элементами акселерометров являются тела сенсоров, пружины и инерционные массы (ИМ). Когда скорость тела акселерометра меняется, ИМ через пружину так же начинает дви-

гаться, пружина изгибается и расстояние между телом и ИМ изменяется пропорционально ускорению тела акселерометра.

Рабочие принципы сенсоров различаются в зависимости о того, как реализуется движение между телом и ИМ. В емкостном сенсоре тело и ИМ изолированы друг от друга и их емкость или емкостной заряд являются предметом измерения. Когда дистанция между ними уменьшается, емкость увеличивается и электрический ток идет по направлению к сенсору. В случае, когда расстояние увеличивается, наблюдается обратная ситуация. Сенсор преобразует ускорение тела в электрический ток, заряд или напряжение. Хорошие характеристики рассматриваемых датчиков основаны на технологии емкостного измерения и хорошо подходят для определения малых изменений в движении. Чувствительный элемент для определения ускорения сделан из монокристального кремния и стекла. Это обеспечивает сенсору исключительную надежность, высокую точность и устойчивость показаний по отношению к воздействию времени и температуры. Как правило, чувствительный элемент датчика с диапазоном измерений ±lg выдерживает как минимум 50000g ускорений (lg = ускорение, вызванное силой тяжести Земли). Датчик измеряет ускорение, как в положительном, так и в отрицательном направлении, и чувствителен к статическому ускорению и вибрации. «Сердцем» акселерометра является симметричный чувствительный элемент (ЧЭ), изготовленный по технологиям объемной микромеханики, у которого есть два чувствительных конденсатора. Симметрия ЧЭ уменьшает зависимость от температуры и чувствительности по оси и улучшает линейность. Герметичность датчика обеспечивается за счет анодного соединения пластин друг с другом. Это облегчает корпусирование элементов, повышает надежность и позволяет использовать газовое затухание в сенсорном элементе.

Все необходимые вычислительные операции производит микросхема ASIC контактирующая с микромеханическим блоком.

Различные производители используют различные технологии изготовления микромеханических блоков. Например, при создании трехосевых акселерометров используют технологию Chip-on-MEMS (CoM) которая включает в себя следующие шаги:

– перераспределение и изоляция слоев на МЭМС пластине,
– нанесение 300 микронных шариков припоя,
– установка на МЭМС-пластину микросхем,
– пассивация зазоров между микросхемами и МЭМС,
– тестирование пластины с МЭМС-устройствами,
– резка пластины,
– финальное тестирование и калибровка сенсоров после резки.

Технология CoM позволяет получать МЭМС акселерометры размером 4×2×1 мм.

Фирмой STM для производства МЭМС акселерометров в настоящее время используется несколько технологий [3].

На кварцевую пластину наносится шаблон, затем вытравливаются ненужные части и остаются только необходимые механические детали устройства. Поверхностная микрообработка – это технология 2D MEMS, при которой на пластину кварца наносятся тонкие слои пленок, после чего проводится процесс травления.

В технологии Bulk Micromachining используется объемная микрообработка с сухим плазменным травлением. Наиболее прецизионные датчики для специальных целей изготавливаются с помощью этой технологии (3D DRIE). Травление в объеме позволяет повысить чувствительность, а также соотношение сигнал-шум. Однако технология Bulk Micromachining не получила широкого распространения при выпуске потребительских изделий. Более сложная технология Surface Micromachining предусматривает послойное построение механического устройства. На кремниевую пластину наносятся промежуточные слои из вспомогательного материала (обычно диоксида кремния). Затем необходимая конструкция вытравливается слой за слоем. При этом прежде всего удаляется жертвенный слой. Обе технологии применяются при изготовлении объемных деталей.

Разница заключается в самом процессе технологии.

В качестве примера на рис. 5.23 показан механический блок МЭМС – акселерометра, выполненный по технологии Surface Micromachining [3].

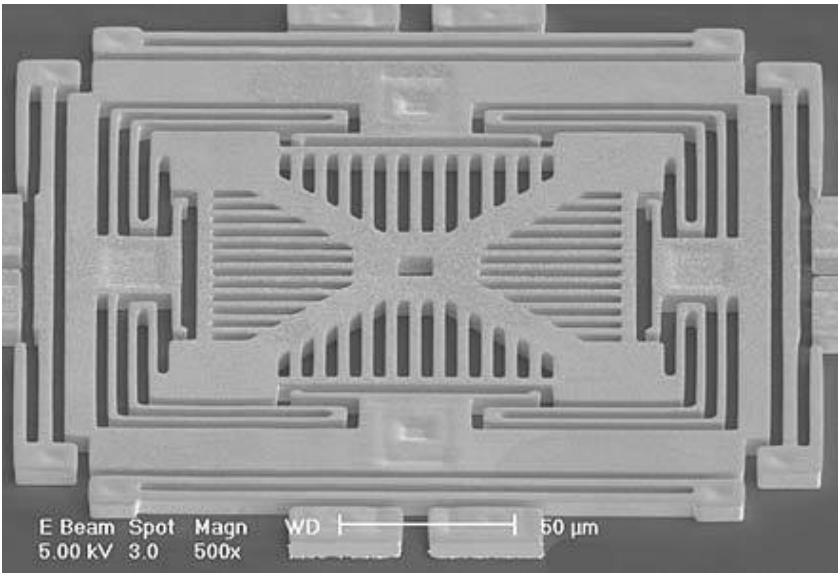


Рис. 5.23. Механическая структура МЭМС-акселерометра

В тех случаях, когда необходимы очень сложные прецизионные конструкции, применяется метод LIGA, базирующийся на объемной литографии с применением жесткого гамма-излучения. Производственные мощности и использование пластин большого диаметра (300 и 450 мм) дают возможность выпускать сотни тысяч датчиков на одной кремниевой пластине за один производственный цикл. На следующем этапе пластина просто разрезается на отдельные датчики. Поскольку при создании механической и электронной частей МЭМС-датчиков используются в принципе одни и те же технологии, то появляется возможность изготовления обеих частей на одной кремниевой пластине.

Структурная схема МЭМС акселерометра STM показана на рис. 5.25.

МЭМС-акселерометр STM состоит из механической подвижной части, электрическая емкость которой изменяется в зависимости от ускорения, и электронной части, фиксирующей эти изменения.

В механической части акселерометра подвижная часть, закрепленная в нескольких точках, может свободно перемещаться в направлении ускорения.

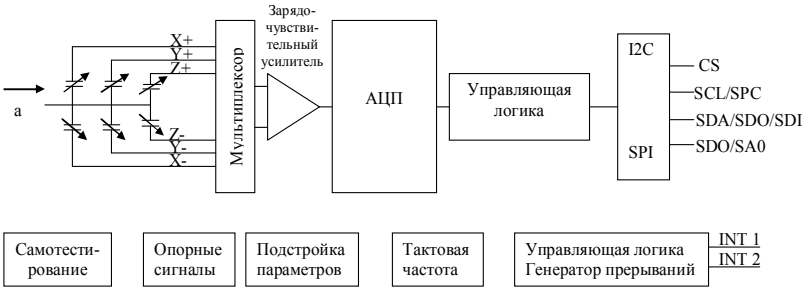


Рис. 5.25. Структурная схема МЭМС-акселерометра STM

Для того чтобы предотвратить блокировку подвижной части в процессе корпусирования, она закрывается специальной защитной крышкой. При наличии ускорения изменяется емкость механической части акселерометра от единиц до сотен пикофарад. Высокоточный усилитель фиксирует заряд этих емкостей и передает сигнал на мультиплексор, связанный с блоком обработки сигнала. Акселерометры STM классифицируются на пороговые и аналоговые (или аналогоподобные цифровые) датчики. В одном случае задается пороговое значение срабатывания. В другом случае акселерометр реагирует на любое движение и на свободное падение. Пороги и времена срабатывания устанавливаются пользователем самостоятельно. Кроме того, акселерометр можно запрограммировать таким образом, чтобы команды прерывания вырабатывались при определенной последовательности событий.

Специально для медицинских приложений фирмой Analog Devices выпускается AFE акселерометр с цифровым выходом ADXL362 с ускорений, выполненный по технологии IMEMS. В микродатчиках ускорения механическая часть датчика содержит поликремневую пластинку, соединяемую с подложкой при помощи упругих элементов подвеса. По краям пластины вытравлива-

ются балки, образующие дифференциальную систему большого числа ячеек парных емкостей. В отсутствии ускорений емкости ячеек почти одинаковы. При возникновении ускорения пластины смещаются, нарушая баланс емкостей. Таким образом, технология IMEMS в качестве датчиков ускорения использует датчики емкостного тока.

Структурная схема AFE типа ADXL 362 приведена на рис. 5.26.

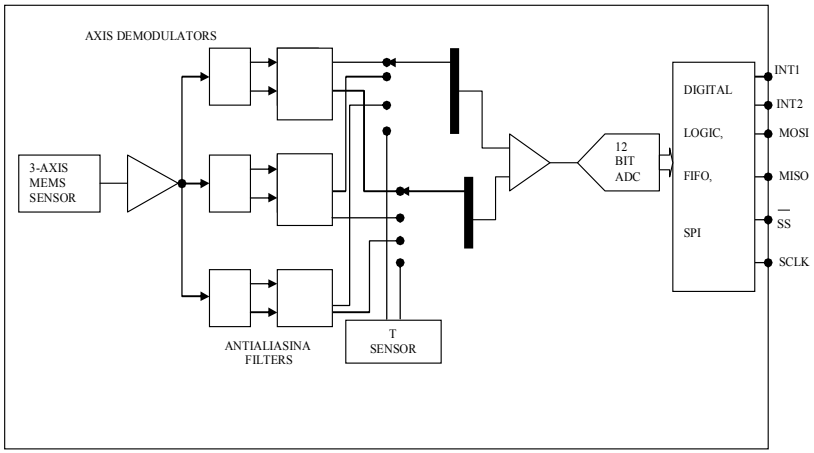


Рис. 5.26. Структура акселерометра ADXL 362

В этой схеме использован трехкоординатный (трехосевой) датчик ускорений MEMS (3-AXIS MEMS SENSOR) сигнал с которого после ускорения подвергается демодуляции с выделением огибающей ускорения (AXIS DEMODULATORS) и подвергается фильтрации (ANTIALIASING FILTERS). После дополнительного усиления сигналы ускорений с трех координат оцифровываются 12-разрядным АЦП (12-BIT ADS), подвергаются соответствующей цифровой обработке и передаются на выход микросхемы в стандартном цифровом коде SPI.

Буфер FIFO может быть использован для временного хранения данных (разгрузка хост-процессора).

Для температурной стабилизации используется информация с температурного датчика (TSENSOR).

Акселерометр ADXL362 в режиме покоя потребляет ток всего в 2 мкА. После пробуждения по движению потребляемый ток возрастает до 270 мА при частоте обновления данных в 100 Гц, напряжение питания микросхемы может варьироваться в диапазоне от 1,6 В до 3,5 В. Диапазоны измеряемых ускорений: ±2 g; ±4g; ±8g. Разрешение в диапазоне ±2g составляет $1*10^{-3}$ g/LSB.

В медицинских приложениях чувствительности ADXL хватает для применения в слуховых аппаратах.

Для бесконтактной регистрации баллистокардиаграммы и дыхательной активности используют акселерометр типа ADXL350. Структурная схема этой микросхемы приведена на рис. 5.27.

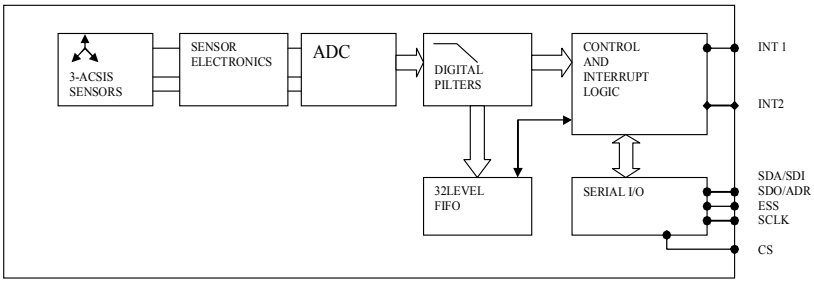


Рис. 5.27. Структурная схема акселерометра ADXL350

В этом акселерометре использован 13 разрядный АЦП (ADC) с возможностью выбора диапазона измерений: ±1 g; ±2 g; ±4g; ±8g. Передача данных в микроконтроллеры производится по протоколу SPI.

В качестве второй группы приборов использующих технологии МЭМС рассмотрим принцип действия МЭМС гироскопов.

В отличие от акселерометра, гироскоп реагирует на изменение углов ориентации относительно инерциальной системы отсчета. Принцип их действия основан на использовании эффекта силы Кориолиса.

Микроэлектромеханические гироскопы также содержат две части – механический и электронный блоки. Механическая часть

относится к классу вибрационных устройств, которые сохраняют направление своих колебаний при повороте основания.

Вариант механической части простейшего одноосевого (курсового) гироскопа фирмы STM схематически показан на рис. 5.28.

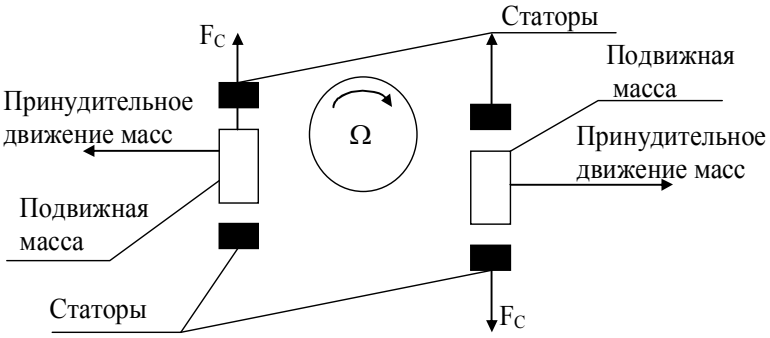


Рис. 5.28. Схема курсового МЭМС гироскопа

Работает гироскоп следующим образом. Специальным блоком (вибратором) с заданной частотой две подвижные массы заставляют двигаться в противоположном направлении.

Как только произойдет изменение угловой скорости Ω, сила Кориолиса F_C будет действовать в перпендикулярном направлении, и вызовет смещение масс, пропорциональное величине угловой скорости. Так как движущиеся электроды (роторы) сенсорной части датчика расположены рядом с фиксированными электродами (статоры), то любое смещение будет вызывать изменение электрической емкости конденсаторов, образованных статорами и роторами. Таким образом, осуществляется преобразование угловой скорости гироскопа в электрический параметр, величина которого детектируется специализированной схемой.

По сравнению с другими гироскопами дифференциальный характер используемых STMicroelectronics камертонов делает систему нечувствительной к нежелательному линейному ускорению и случайной вибрации, воздействующей на сенсор. При наличии та-

ких воздействий обе массы датчика будут смещаться в одном направлении, в результате чего за счет дифференциального включения будет регистрироваться нулевое результирующее изменение общей емкости.

Схему преобразования и нормирования сигнала, используемую в гироскопе, можно рассматривать как сочетание секции управления двигателем и воспринимающей части акселерометра (секция регистрации) (рис. 5.29).

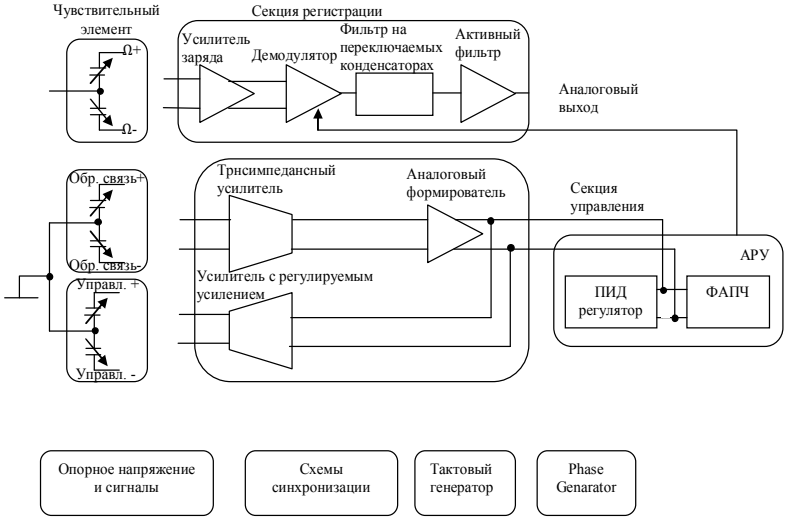


Рис. 5.29. Блок-схема одноосевого курсового (yaw) гироскопа

– Секция управления предназначена для возбуждения механического элемента, вызывающего его колебания вперед и назад с помощью электростатического привода (актуатора).

– Секция регистрации определяет смещение масс, вызываемого силой Кориолиса, посредством измерения емкости. Схема датчика формирует аналоговый или цифровой выходной сигнал, пропорциональный угловой скорости приобретенной датчиком.

Усовершенствованные функции снижения потребляемой мощности, встроенные в схему управления гироскопом, позволяют от-

ключать датчик при бездействии. С другой стороны, датчик может находиться в спящем режиме, когда общее потребление гироскопа значительно уменьшается по сравнению с нормальным режимом работы, и, в тоже время, по команде пользователя немедленно переключаться в активный режим для измерения угловых скоростей.

Для регистрации углов поворотов в трех направлениях фирма STM разработала миниатюрный сверхстабильный датчик L3G4200D со следующими техническими характеристиками.

– Три выбираемые полные шкалы (250/500/2000 dps).

– Цифровые выходные интерфейсы I2C и SPI.

– 16-битные выходные данные.

– Встроенный сенсор температуры, с которого можно считать 8-битные данные.

– Два дополнительных выходных сигнала (прерывание и готовность
данных).

– Встроенные фильтры ФНЧ и ФВЧ, полосу пропускания которых можно выбрать.

– Высокая стабильность показаний по температуре и времени.

– Широкий диапазон питающих напряжений 2.4V .. 3.6V.

– Входы и выходы (I/O), совместимые с низковольтной логикой (1.8V).

– Встроенные режимы отключения (power-down) и сна (sleep).

– Встроенный FIFO.

– Высокая устойчивость к ударам и вибрациям.

– Расширенный рабочий температурный диапазон (–40°C ... +85°C).

– Миниатюрный пластмассовый корпус LGA-16 (4x4xl.l mm).

В ряде конструкций вместо двух вибрирующих масс используют вибрирующее с частотой 22 кГц в двух взаимоперпендикулярных направлениях кольцо (рис. 5.30).

Если гироскоп подвергается воздействию угловой скорости, то на кольцо действуют силы Кориолиса: по касательной к периметру кольца относительно главных осей. Эти силы деформируют кольцо, что вызывает радиальное движение вторичных снимающих преобразователей. Данное движение, определяемое на вторичных

снимающих преобразователях, пропорционально прилагаемой угловой скорости. При этом двухполосный сжатый передающий сигнал демодулируется с учетом основного движения.

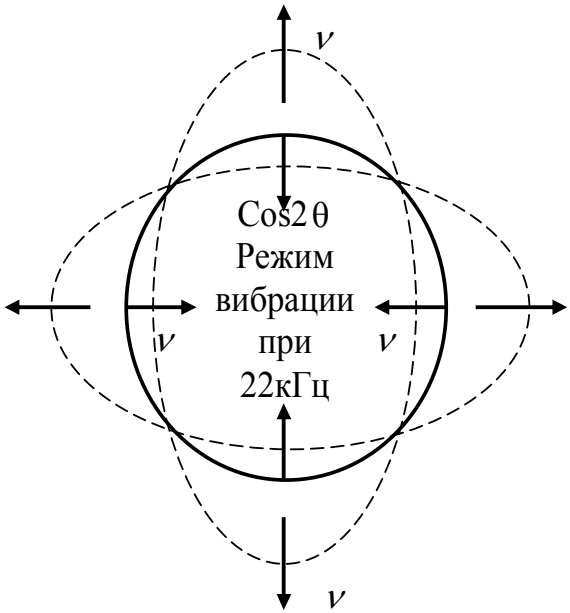


Рис. 5.30. Схема работы кольцевого датчика МЭМС гироскопа

В итоге получается низкочастотный компонент, который пропорционален угловой скорости.

Успехи в развитии технологий МЭМС привели к миниатюризации и радикальному удешевлению инерциальных датчиков, что, в свою очередь, вызвало лавинообразный рост спроса на них во множестве отраслей – от промышленности, транспорта и военной техники до бытовых приборов и индустрии развлечений. Многие производители, следуя потребностям рынка, начали предлагать комплексные решения – инерциальные измерительные модули, содержащие три оси гироскопа, три оси акселерометра, а также имеющие возможность интеграции в этот же блок электронного

компаса — магнитометра, барометра и др. В подавляющем большинстве приложений заказчику достаточно информации о линейной и угловой скорости в трехмерном пространстве.

Решению таких задач в наибольшей степени соответствует концепция измерительного модуля с 6 степенями свободы, реализованная фирмой Silicon Sensing в модели DMU02. Эта относительно недорогая сборка в защищенном корпусе размером 25x25x25 мм построена на трех гироскопах CRG20 и 3-осевом акселерометре, имеет цифровой интерфейс SPI и диапазон измерений ±300°C для канала гироскопа и ±6g для канала акселерометра. Для обмена данными с модулем на этапе разработки системы производитель предлагает использовать отладочный набор USB01-00-0100.

5.5. Радиоинтерфейс BLUETOOTH

Общие сведения о принципах передачи информации

Разработчики BLUETOOTH определили этот термин как технологию передачи данных по радиоканалу на небольшие расстояния (при мощности передатчика 1,…,2,5 мВт – на расстояния до 10 м и при мощности передатчика до 100мВт – до 100 м).

Физически BLUETOOTH устройства представляют собой микросхемы, обеспечивающие радиосвязь на частоте близкой к 2,4 гГц. Этот диапазон в большинстве стран Европы и США относят к полосе ISM для приборов промышленного, научного и медицинского назначений. Работа в этом частотном диапазоне не требует специальных разрешений и лицензирования. Технология BLUETOOTH полностью открыта. Любая компания, подписавшая лицензионное соглашение может создать свои предметы с использованием этой технологии. Скорость передачи данных в BLUETOOTH-устройствах с базовыми спецификациями (1.1,1.2 и 2.0) составляет 1 мб/с, а в устройствах поддерживающих спецификацию 2.0 и расширение EDR, скорость передачи, в зависимости от способа модуляции составляет 2 или 3 мб/с.

Основополагающим принципом построения систем BLUETOOTH является использование метода расширения спектра при скачкообразным измерении частоты FHSS (Frequency Hop Spread Spectrum). Суть этого принципа заключается в следующем. Каждое из BLUETOOTH устройств может скачкообразно по псевдослучайному закону выбрать несущую частоту в диапазоне 2,402,…, 2,480 гГц. Причем несущая частота $F_к$ выбирается дискретно в соответствии с формулой

$$F_K = 2402 + k \text{ (мГц), где k=0,1,...,78}$$

Таким образом, BLUETOOTH устройства могут работать на 79 несущих частотах (подканалах связи). Полоса каждого подканала составляет 1 мГц, а частотный «разнос» каналов составляет 140,…,175 кГц. Для физического канала время разделяется на такты (интервалы) называемые слотами продолжительностью 625 мкс. В течение каждого слота передача осуществляется по одному подканалу (на одной несущей частоте).

Данные между устройствами BLUETOOTH передаются пакетами с одной и той же несущей частотой. Пакет может составлять из одного или нескольких (до пяти) слотов (тактов). После передачи пакета несущая частота изменяется.

Постоянное чередование частот позволяет выбрать свободную (не занятую другими устройствами) частоту и «уйти» от, подканалов, где есть помехи от различных «шумящих» источников.

Рисунок 5.91 иллюстрирует переход BLUETOOTH устройства на работу с различной несущей частотой от пакета к пакету.

В соответствии с примером приведенном на рис. 5.31 устройство выбрало начальную частоту передачи с несущей $F_1 \approx 2403$ МГц и по нему передан пакет длиной 2 слота.

После передачи первого пакета устройство по своему внутреннему алгоритму выбрало несущую частоту F_6=2408 МГц и передало второй пакет длиной в 4 слота.

Алгоритм перехода от одной несущей частоты к другой, хотя и реализуется по псевдослучайному закону, но с использованием информации о значении своих часов и номера ведущего устройства.

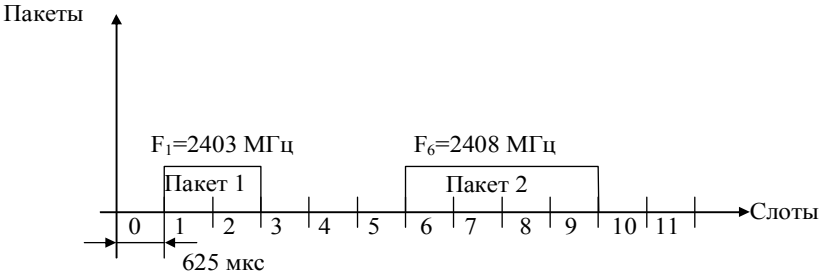


Рис. 5.31. Иллюстрация пакетной передачи данных с адаптивной сменой несущей частоты.

Информация внутри пакета передается двоичным кодом с простым способом кодирования. Единица кодируется частотой превышающей несущую частоту на 160 КГц. Ноль кодируется частотой меньшей на 160 кГц, чем несущая частота (рис. 5.32).

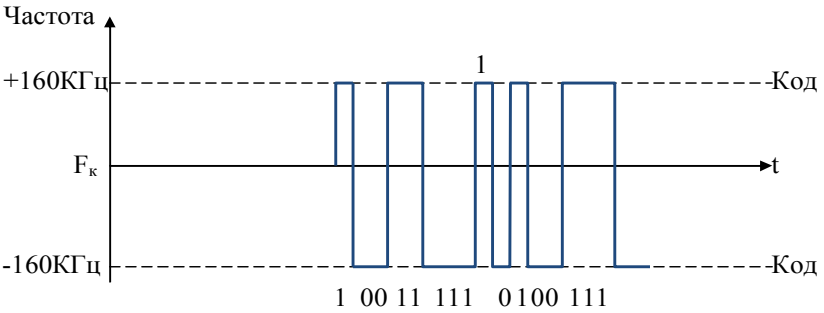


Рис. 5.32. Частотная модуляция для кодирования информации в BLUETOOTH

Технология BLUETOOTH использует дуплексный способ передачи информации с временным разделением (мультиплексированием). Ведущее устройство предает пакеты в нечетные интервалы времени (слоты), а ведомое устройство в четные (рис. 5.33).

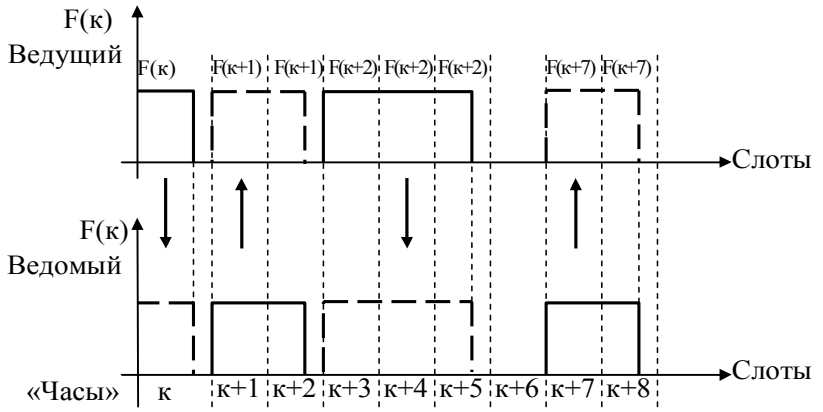


Рис. 5.33. Пакетный обмен между ведущим и ведомым BLUETOOTH-устройствами

На этом рисунке стрелки между верхней и нижней диаграммами показывают направления передачи пакетов. Сплошной линией изображен пакет, который передается, пунктиром – который принимается. Для ведущего устройства краткость частоты (к) выбирается нечетной.

Структура пакета и сетевое обеспечение BLUETOOTH связи

В протоколах BLUETOOTH пакет определяется как формат организованных в массиве битов передаваемых по физическому каналу. Пакет состоит из трех частей: кода доступа, заголовка и информации пользователя (рис. 5.34).

Код доступа 72 бита	Заголовок 64 бита	Информация пользователя 0,…,2475

Рис. 5.34. Структура пакета

Код доступа позволяет определять принадлежность передаваемых пакетов к группе устройств, которые синхронизированы на общие часы и общую последовательность смены частот. Эта груп-

па устройств называется пикосетью. Код доступа (рис. 5.35) содержит в своем составе преамбулу (4 бита), синхрослово (64 бита) и концевик (4 бита) содержащий контрольную сумму.

Преамбула 4 бита	Синхрослово 64 бита	Концевик 4 бита

Рис. 5.35. Структура поля кода доступа

Заголовок состоит из шести полей (рис. 5.36) и содержит информацию для управления связью

Адрес 3 бита	Тип 4 бита	FLOW 1бит	ARQ 1 бит	SEQN 1 бит	Контрольная сумма HEC 8битов

Рис. 5.36. Структура заголовка

В поле адреса содержится код активного элемента. В поле тип определяется код типа данных. Бит FLOW показывает готовность устройства к приему информации. Бит ARQ подтверждает правильность приема информации. Бит SEQN служит для определения последовательности пакетов, контрольная сумма заголовка формируется поле HEC.

Блок информация пользователя состоит из трех сегментов (рис. 5.37): Заголовок пользователя, собственно информация пользователя и контрольная сумма CRC.

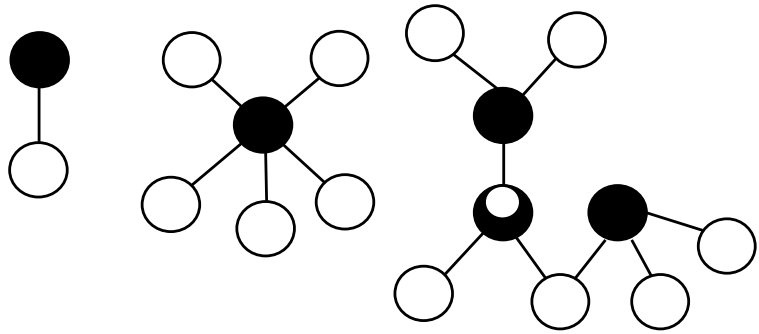
Заголовок 8 бит	Информация	CRC 16 бит

Рис. 5.37. Структура блока информации пользователя

Заголовок определяет логический канал и управление потоками данных на уровне этого канала, а также содержит указатель длины информации пользователя. Собственно передаваемые данные располагаются в поле информации. Контроль правильности передачи данных осуществляется с помощью 16-разрядного циклического

контрольного кода CRC, который после соответствующих вычислений добавляется к полю информации.

При взаимодействии BLUETOOTH устройств объединяемых в пикосеть одно из них (как правило, наиболее мощное) обязательно выполняет функции ведущего устройства, остальные функции ведомых устройств. Ведущее устройство формирует пикосеть и формирует общие для этой сети сигналы синхронизации. Пикосети относят к категории сетей “Ad-HOG”. Они могут включать в себя от 1 до 7 активных ведомых устройств (рис. 5.38).



а) б) в)

Рис. 5.38. Варианты пикосетей: а – с одним ведомым устройством; б – с несколькими ведомыми устройствами; в – скаттернет

Кроме того, в зоне уверенного приема (окресности) ведущего устройства могут находится неактивные (припаркованные) ведомые устройства, которые синхронизированы и имеют общие часы с ведущим устройством, но не могут обмениваться данными пока ведущее устройство их не активирует.

Для определенности на рис. 5.38 ведущие устройства показаны как зачерненные круги, а ведомые как не зачерненные. В аппаратной и программной частях ведущие и ведомые устройства имеют один и тот же состав. Любое из них может быть как тем, так и другим. Однако только ведущее устройство формирует пикосеть и контролирует весь трафик. Наиболее часто в качестве ведущего

устройства выступает персональный компьютер (host, master) или плата CPU мини-ЭВМ. Ведомые устройства могут посылать сообщения только в интервале «ведомые – ведущему» после того, как к ним обратился предшествующий слот «ведомый – ведущему». Если в этом интервале у ведущего нет никакой информации для ведомого, то он передает информацию только с кодом доступа и заголовок.

Для формирования единых часов пикосети каждый BLUETOOTH контроллер имеет 28 разрядный счетчик изменяющийся каждые 312,5 мск. Значение кода этого счетчика (часы) определяет нумерацию слота и синхронизацию в различных физических каналах.

Множество пикосетей могут взаимодействовать друг с другом формируя распределенную сеть. Это может обеспечиваться через устройства, которые являются элементом более чем одной сети (PMP – устройства). Устройство – участник нескольких пикосетей чередует свою работу с каждой из пикосетей. Такое взаимодействие через PMP устройство не имеет общей синхронизации и называется скаттернетом (рис 5.28, *в*).

При одновременной работе нескольких BLUETOOTH-модулей в пикосети каждому модулю в переделах каждого слота (такта) назначается соответствующий частотный канал и режим адресации или приема (рис. 5.39).

Архитектурная иерархия системы BLUETOOTH содержит физический и ряд логических уровней определяемых как ядро BLUETOOTH.

Физический и первый (наиболее близкий к физическому) логический уровни ядра BLUETOOTH группируются в подсистему называемую контроллером BLUETOOTH. В контроллер BLUETOOTH (рис. 5.40) на физическом уровне входит приемопередатчик (RF) а на логическом уровне – контроллер соединений (Linc-Controller), уровень управления ресурсами (Baseband), уровень управления соединениями (Linc manager), уровень управления устройством (Device manager).

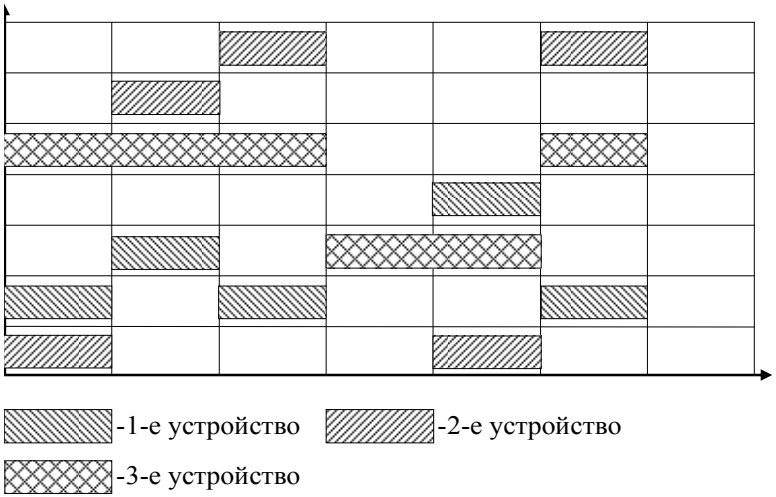


Рис. 5.39. Частотно-временная диаграмма работы модулей BLUETOOTH

Наиболее высокий логический уровень обеспечивается хост BLUETOOTH реализующим логический уровень L2CAP и ряд сервисов. В качестве хост BLUETOOTH может выступать компьютер, специализированное вычислительное устройство, периферийное устройство, мобильный телефон, точка доступа к локальной сети и т.д.

Хост Bluetooth и контроллер Bluetooth взаимодействуют между собой, используя специальный интерфейс HCI.

Все устройства Bluetooth взаимодействуют между собой, используя ряд протоколов (заштрихованные двойные стрелки на рис. 5.39): протокол физического уровня RF (Radio); протокол контроллера соединений (LC); протокол управления соединениями (LMP); адаптированный протокол управления логическими связями (L2CAP). Кроме того существует протокол обнаружения услуг (SDP) реализующий приложения BLUETOOTH.

Протокол LMP отвечает за соединение устройств BLUETOOTH между собой, а так же решает вопросы безопасности (генерирова-

ние ключей шифрования и подключения, обмен ключами и их проверка).

Протокол L2CAP адаптирует протоколы верхних уровней к протоколам Baseband. Многие протоколы и службы более высокого уровня используют L2CAP как транспортный протокол. Одним из самых важных протоколов использующих L2CAP как транспортный протокол является прокол обнаружения услуг. Используя SDP можно запросить информацию об устройстве, его услугах и характеристиках.

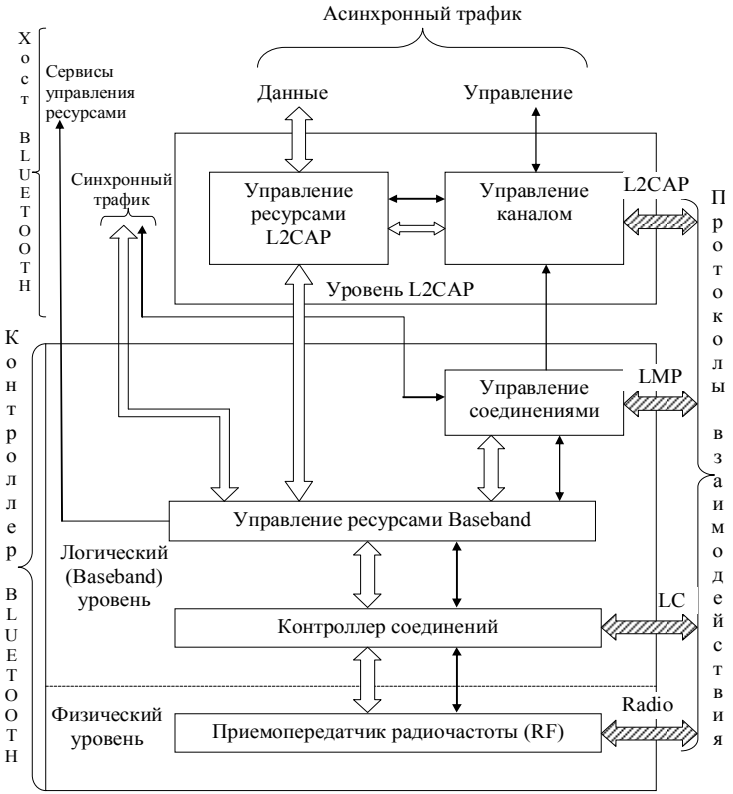


Рис. 5.40. Иерархия уровней BLUETOOTH

Кроме протоколов обмена естественными путями обмена между блоками и уровнями BLUETOOTH являются потоки управления (одинарные стрелки на рис. 5.40) и потоки данных (двойные стрелки на рис. 5.40).

Доступ к среде и процедурам физического уровня обеспечивается логическим уровнем Baseband. Этот уровень обеспечивает обмен потоками данных и звуковой информацией в режиме реального времени между устройствами BLUETOOTH входящими в сеть в синхронном и асинхронном режимах.

Синхронную передачу осуществляют обычно между устройствами (связь типа «точка-точка») для передачи оцифрованных звуковых сообщений. При таком типе передачи ведущее устройство резервирует для каждой связи определенное время (в слотах) взаимодействия. Связи организуются симметричные двусторонние. Повторные передачи пакетов в случае ошибок не реализуются.

Асинхронные передачи используют коммутацию пакетов по схеме «точка множество точек». Ведущее устройство связывается с любым ведомым устройством пикосети, требуя от него ответ. Ведомое устройство имеет право, на передачу только получив обращенный к нему запрос. Для большинства типов предусматривается повторная передача в случае обнаружения ошибок

Профили BLUETOOTH

Общие механизмы (протоколы и функции) через которые устройства BLUETOOTH взаимодействуют друг с другом называют **профилями** BLUETOOTH. Профили определяют области возможного применения устройства BLUETOOTH. Устройства различных производителей соответствующих одному профилю могут взаимодействовать друг с другом. При этом для всех используемых моделей необходимо обязательное наличие профиля общего доступа (GAP). Остальные профили добавляются в зависимости от назначения устройства и выполняемых им функций.

Профиль GAP (Generic Access Profile) отвечает за поддержание связи между устройствами, выявления других доступных профилей, а так же за безопасность соединений. В этот профиль вхо-

дят все функции необходимые для работы всех основных протоколов BLUETOOTH, однако сам по себе он недостаточен для практических приложений.

Для всех моделей связанных с передачей данных кроме GAP применяют еще три профиля общего назначения:

– профиль последовательного порта SPP;

– профиль приложения обнаружения услуг SDAP;

– профиль общего обмена объектами GOEP.

Профиль SPP осуществляет обмен данными путем эмуляции интерфейсов персональных компьютеров типа RS 232 или USB.

Профиль SDAP необходим для того, чтобы любое приложение могло узнать, какие услуги (сервисы) BLUETOOTH доступны на любом из доступных устройств с которым оно соединено.

Профиль GOEP используется для организации непосредственного обмена объектами между двумя устройствами. При этом объекты могут иметь любые типы (изображения, документы, визитные карточки и т.д.).

В зависимости от назначения устройств кроме перечисленных профилей к ним могут добавляться до девяти возможных профилей:

DUP – профиль дозвона до сети;

FAX – профиль факсимильной связи;

HSP – профиль гарнитуры;

CAP – профиль доступа к сети;

FTR – профиль передачи файлов;

OPP – профиль помещения объектов в стек;

SYNC – профиль синхронизации.

Эти тринадцать профилей являются основными. К ним могут быть добавлены дополнительные профили типа профиля замены кабеля твердой копии, основной профиль принтера, профили для аудио и видео аппаратуры.

Технические профили поддерживаются модулями собираемыми из одной или нескольких микросхем, с соответствующим программным обеспечением, например профиль SPP поддерживается модулем BTM-112, модуль BTM-130/140 поддерживает профили HSD (профиль гарнитуры), A2DP (расширенный профиль распро-

странения аудио), AVRCP (профиль дистанционного управления аудио и видео аппаратурой) и др.

В качестве примера на рис. 5.41 приведена структура модуля ROK 101107 производства ERICSSON MICROELECTRONICS, который может быть использован как базовый модуль для осуществления BLUETOOTH связи. В состав модуля входят Baseband контроллер обеспечивающий формирование несущей частоты и взаимодействие с другими устройствами через три последовательные стандартные шины RS232 (линии связи TxD, RxD, RTS и CTS), USB (линии связи D^+,D^-, DATA CH, WAKE UP), PCM (линии связи IN, OUT, SYNC, CLC), флеш память, кварцевый генератор (КГ) и радиомодуль PBA31301112. Baseband контроллер построен на основе процессора ARM7. Одной из основных функций этого контроллера является чтение и знание информации во внутренние регистры радиоконтроллера ASIC радиомодуля. Эти регистры используются для установления частоты, настройки и управления, радиоконтроллер работает либо от своего кварцевого генератора (13МГц), либо тактируется внешней частотой (линия такт).

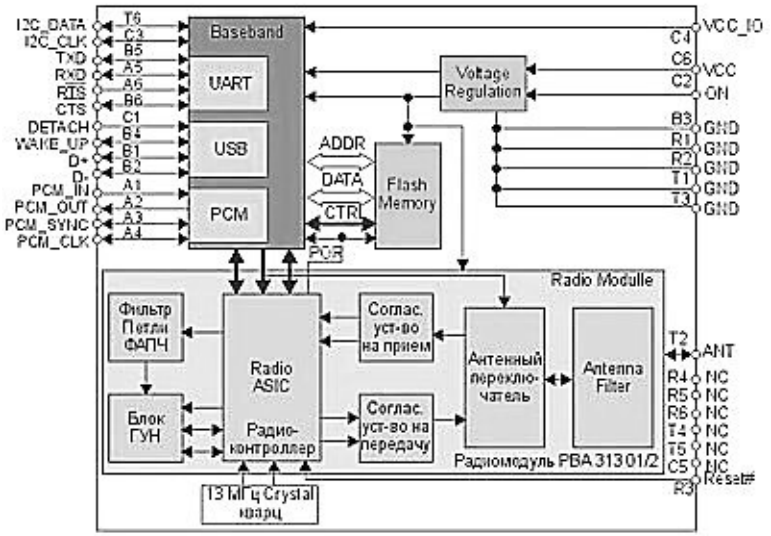


Рис. 5.41. Структурная схема модуля Bluetooth ROK 101107

Для управления частотой обмена информацией используют фильтры автоматической подстройки частоты (ФАПЧ) и генератор управляемый напряжением (ГУН). В оконечных каскадах радиомодуля используются схемы управления (СУ) приема и передачи, антенный переключатель (АП) и антенный фильтр (АФ), подключенный к шаровым разъёмам типа BGA.

Для полноценного функционирования радиомодуль должен быть подключен либо к Baseband-контроллеру, либо к прибору, который может его эмулировать. Питание модуля осуществляется от источника 3,3 В, из которого выделяются для напряжения 2,8В (для питания ГУН) и 2,2В для питания остальных блоков. Все источники развязаны по низкой частоте.

Конструктивно плата модуля состоит из 6 слоев совместно обжигаемой керамики (технология LTCC) и семи металлических слоев. Все дискретные элементы смонтированы на одной стороне платы. Размеры модуля 32,8×16,8×2,75мм (длина модуля почти в два раза меньше длины обычной спички).

Модуль может быть вмонтирован в любое радиоэлектронное устройство.

Информационное взаимодействие BLUETOOTH на примере двух хостов

Существует достаточно большой арсенал программно-технических средств различного типа, использующих Bluetooth технологию обмена информацией.

В качестве примера на рис 5.42 приведена структурная схема Bluetooth связи между двумя хостами (ПЭВМ1и ПЭВМ2), построенной по принципу точка-точка. Каждый Bluetooth модуль содержит формирующую и приемо-передающую аппаратуру и встроенное (“зашитое”) программное обеспечение. Программное обеспечение Bluetooth модуля состоит из интерфейса хост-контроллера (HCI), менеджера связи и контроллера несущей частоты (Baseband). Связь модуля с ПЭВМ (хостом) на физическом и канальном уровнях осуществляется по стандартным интерфейсам с помощью встроенного программного обеспечения.

К физическому уровню связи относится также радиолиния. При использовании интерфейсов модель Bluetooth является USB ведомым прибором, поэтому ресурсов ПЭВМ не требуется.

Интерфейс хост-контроллера модуля (HCI) является командным интерфейсом. Через этот интерфейс хост (ПЭВМ) направляет команды и в ответ от модуля принимает сообщение об их использовании. Менеджер связи устанавливает необходимую конфигурацию для HCI.

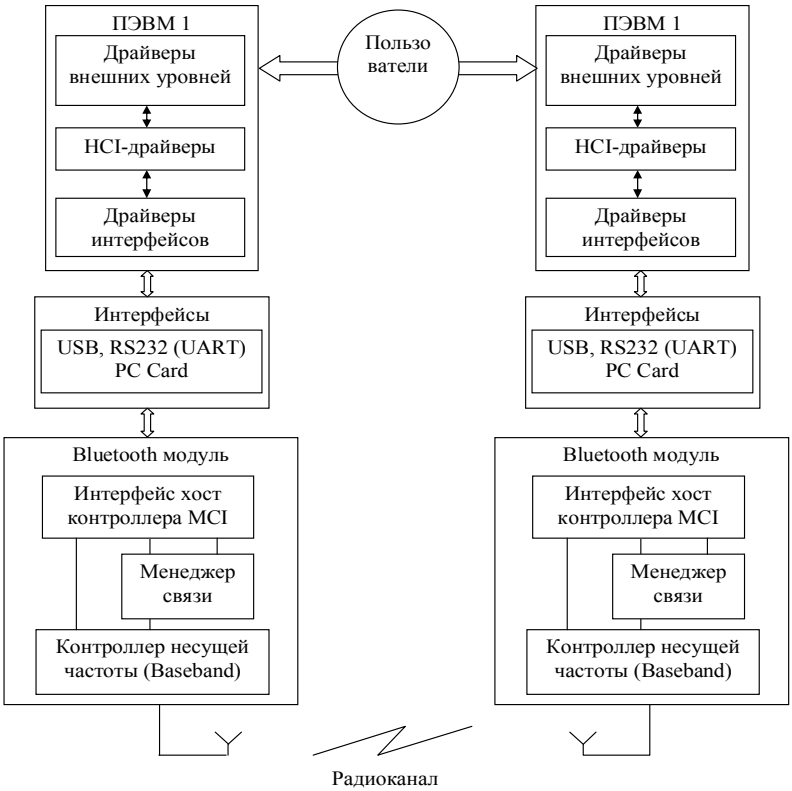


Рис. 5.42. Bluetooth-связь между хостами – программное обеспечение; – встроенное программное обеспечение, – аппаратура.

Использование BLUETOOTH-технологий в медицинских приложениях

Использование беспроводных каналов связи в медицинском приборостроении позволяет избавиться от множества кабелей и разъемов, что естественно создает дополнительные удобства при работе с соответствующими приборами, системами и комплексами. В медицинских приложениях используются различные Bluetooth модули, в частотности модули WT41 фирмы Bluegiga. Модули WT41 способны взаимодействовать между собой на расстоянии до 1000 метров при скорости обмена данными 3 Мбит/с.

В состав аппаратной части WT41 входят трансивер, контроллер связи (Baseband) и управляющее устройство, реализующее верхние уровни Bluetooth протокола.

На плате модуля размещены: базовый чип “Bluetooth chip CSR Blue Core-04”; высокочастотный приемник с чувствительностью – 90 дБ; встроенный RISC – контроллер; память RAM на 8 Мбит и Флеш-память на 2 Мбит; кварцевый генератор 16МГц; маломощный предварительный усилитель; усилитель мощности; радиочастотные фильтры и антенный переключатель. Флеш-память модуля предназначена для хранения протоколов Bluetooth и прикладных программных блоков. Память RAM используют для временного хранения приложений пользователя. Основными внешними интерфейсами модуля являются полные версии UART и USB. Установки UART могут меняться пользователем. Интерфейс USB работает со скоростью передачи данных до 12 Мбит/с. Модуль можно подключать напрямую к другим USB устройствам. При работе с USB модуль WT41 работает как периферийное устройство, отвечающее мастеру (например, ПЭВМ). Модуль WT41 выпускается в различных модификациях, отличающихся типом антенны и вариантами прошивки встроенного программного обеспечения.

Для разработки и настройки оборудования на базе WT41 фирма Bluegiga выпускает отладочный комплект EBWT41. Модули фирмы Bluegiga выпускаются с двумя базовыми прошивками программного обеспечения iWRAP (для решения достаточно простых задач) и HCI (для более сложных приложений).

Программное обеспечение iWRAP разработано для работы со встроенным RISC-процессором с использованием простых ASCII команд (стандартные коды обмена информацией), аналогично тому, как осуществляется работа с модемами через AT команды.

Интерфейс HCI (Host Controller Interface) позволяет управлять портами через UART или USB. Этот интерфейс полезен в тех случаях, когда нужно получить максимально большие скорости и выполнять задачи не предусмотренные ASCII кодами.

Для медицинских приложений используются в основном два профиля, подстраиваемого программного обеспечения iWRAP – 3/4; профиль “Serial Port Profull” (SPP) и профиль Bluetooth Health Device Profull (HDP).

Профиль SPP предназначен для соединения по Bluetooth-каналу двух устройств, имеющих последовательные порты. В этом случае одно устройство («ведущий»), например WT41-1, берет на себя инициативу установить связь с другим устройством WT41-2. Второе устройство выступает в качестве «ведомого» и пассивно ожидает вызова.

Для эмуляции последовательного порта используется транспортный протокол RFCOMM, который является адаптацией для Bluetooth-канала известного протокола GSM TS 07.10. По профилю SPP может быть установлено до семи одновременных соединений. При этом скорость передачи зависит от количества задействованных устройств. Для одновременной передачи данных от одного устройства нескольким другим используется режим MUX mode.

Характерной особенностью медицинского применения беспроводных технологий является наличие различных типов датчиков и устройств, разрабатываемых различными фирмами и странами, имеющими несогласованные разнотипные интерфейсы. Для устранения проблем согласований различных устройств и решения задач взаимозаменяемости была сформирована международная рабочая группа “Medical Device Working”, которая разработала профиль HDP.

В результате работы этой группы были созданы и в 2008 г. одобрены два основных нормативных документа: Multi-channel Adaptation Protocol (MCAP) и Bluetooth Health Device Profile

(HDP). Сейчас эти документы приняты к исполнению подавляющим большинством ведущих мировых производителей Bluetooth-оборудования и закреплены в стандартах Application-level Interoperability IEEE 11073-xxxxx Personal Health Devices.

Профиль HDP предназначен как для медицинских приборов общего пользования, так и для профессионального оборудования. При взаимодействии двух Bluetooth-устройств этот профиль обеспечивает установление контрольного, а также наиболее надежного канала для передачи данных. Он также поддерживает процедуру идентификации взаимодействующих устройств и синхронизирует их работу. В схеме работы HDP одно из устройств рассматривается в качестве получателя (Sink), а другие определяются как источники данных. Такой подход позволяет выводить информацию от нескольких измерительных блоков на единый диспетчерский пульт, что значительно облегчает работу врачей при массовых обследованиях пациентов. Данные с измерительного устройства можно вводить в ПЭВМ любого типа и передавать на центральный диспетчерский пульт по GSM/GPRS-каналу. Подобное оборудование используется в телемедицине, когда результаты диагностики пациента могут быть переданы для консультации в любую клинику мира и в операциях участвуют ведущие мировые специалисты.

Благодаря HDP согласованные беспроводные диагностические системы позволяют врачу не выходя из своего рабочего кабинета проводить экспресс-диагностику состояния больного непосредственно у него дома, используя как бытовую, так и профессиональную медицинскую технику. Внедрение этого профиля дало также толчок к развитию нового поколения медицинского беспроводного оборудования. В качестве примера можно привести точную синхронизацию во времени работы беспроводных датчиков в современных компьютерных системах кардиодиагностики.

Профиль HDP обеспечивает строгое соответствие взаимодействующего оборудования протоколу обмена ISO/IEEE 11073-20601 Personal Health Data Exchange Protocol. Для контроля доступа в беспроводные сети медицинского оборудования используется протокол Bluetooth Service Discovery Protocol (SDP). Совместимость различных устройств на потребительском уровне гарантируется

стандартами “ISO/IEEE 11073-xxxxx”. Оптимизированный протокол обмена IEEE 11073-20601 предназначен для бытовых медицинских приборов, использующих Bluetooth-канал. Этот протокол регламентирует правила доступа к информации, процессы передачи результатов измерений, а также описывает базовые методики обработки результатов измерений и их визуализации. Кроме того, этот протокол содержит основные требования, предъявляемые к различного рода датчикам бытового медицинского оборудования. В разделе Device Data Specializations (DDS) этого протокола содержится подробное описание того, какие именно требования предъявляются к конкретному медицинскому оборудованию. В качестве примера приборов, на которые распространяется IEEE 11073-20601, можно назвать различные измерители температуры и веса тела, тонометры, фиксаторы граничной частоты сердцебиений (пульса), глюкометры, бытовые кардиографы и д.р.

6. РАЗРАБОТКА МАШИННОЙ МОДЕЛИ ГИИ

6.1. Описание модели преобразователя систем кодирования

Программная модель состоит из трех частей: передатчика (клиент), ГИИ (сервер) и приемника. Передатчик имитирует работу источника сигнала, то есть медицинскую технику, передающую полученную ею информацию от пациента. ГИИ получает и обрабатывает информацию в соответствии с требованиями приемника. Приемник устанавливает необходимый формат данных, к виду которого приводится передаваемая информация.

6.2. Описание модели передатчика

Передача информации заключается в последовательном исполнении двух действий: выборе файла и его пересылки. По нажатию кнопки «Отправить файл приемнику» (рис. 6.1) открывается форма «Выбор передаваемого файла» (рис. 6.2).

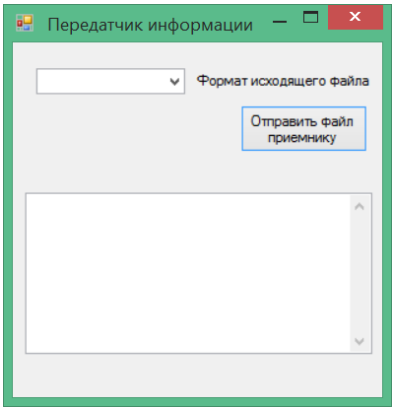


Рис. 6.1. Основное меню Передатчика

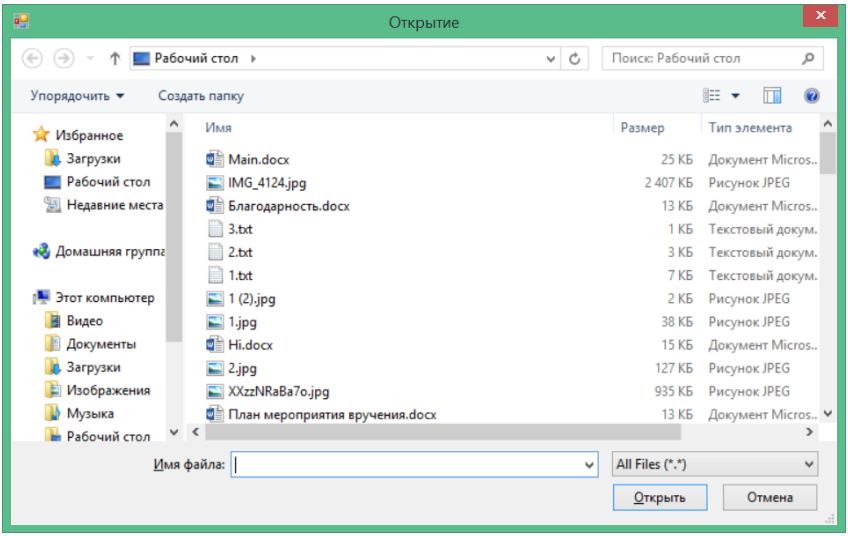


Рис. 6.2. Выбор передаваемого файла

После выбора файла происходит отправка информации по TCP/IP протоколу ГИИ. Пересылка происходит посредством вызова функции StartSend (). Листинг данной функции имеет следующий вид:

```
    private void StartSend(string fName)
        {

            txtLog.Text += "Sending file information\r\n";
            FileInfo fInfo = new FileInfo(fName);
            // Data buffer for incoming data.
            byte[] bytes = new byte[1024]; // Буффер для отправкии частей
            string pathFile = fInfo.FullName;
            string extension = fInfo.Extension;

              // Соединяемся с ГИИ
              try
              {
                 IPHostEntry ipHostInfo =
Dns.Resolve(Dns.GetHostName());
```

```
                IPAddress ipAddress = ipHostInfo.AddressList[0];
                IPEndPoint remoteEP = new IPEndPoint(ipAddress,
11000);

                // Добавляем a TCP/IP  socket.
                Socket sender1 = new Socket(AddressFamily.InterNetwork,
                try
                {
                   sender1.Connect(remoteEP);

                   txtLog.Text += "Socket connected to {0}" +
                       sender1.RemoteEndPoint.ToString();
                   // Отправляем размер
                   FileStream fs = new FileStream(pathFile, File-
Mode.Open, FileAccess.Read); //считывание файла
                   byte[] bytes1 = Encod-
ing.UTF8.GetBytes(Convert.ToString(fs.Length)+"_"+extension);
                   int bytesSent1 = sender1.Send(bytes1);
                   System.Threading.Thread.Sleep(10);
                   System.Threading.Thread.Sleep(10);
                   // Отправляем файл
                   byte[] bytes2 = new byte[fs.Length]; //массив с данны-
ми файла
                   for (int i = 0; i < fs.Length; i++)
                      bytes2[i] = Convert.ToByte(fs.ReadByte());
                   //отправляем кусками
                   int totalLenth = 0;

                   using (FileStream fs1 =
File.Create(@"d:\OtClienta.jpeg", 1024))
                   {
                      while (totalLenth < fs.Length)
                      {
                         int bytesSent2;
                         if (totalLenth + 1024 < fs.Length)
                         {
                            Array.Copy(bytes2, totalLenth, bytes, 0, 1024);
//остался не 1 пакет
                            fs1.Write(bytes, 0, bytes.Length);
```

```
                        bytesSent2 = sender1.Send(bytes);

                    }
                    else
                    {
                        Array.Copy(bytes2, totalLenth, bytes, 0,
bytes2.Length - totalLenth); //остался 1 пакет
                        fs1.Write(bytes, 0, bytes2.Length - totalLenth);
                        bytesSent2 = sender1.Send(bytes);
                    }
                    totalLenth += 1024;
                }

            }
            string needtype = null;
            if (comboBox1.SelectedItem == "jpeg")
            {
                // входной файл в bmp, сохраняем его в jpg
                needtype = "1";

            }
            if (comboBox1.SelectedItem == "bmp")
            {
                // входной файл в jpg, сохраняем его в bmp
                needtype = "2";
            }
            if (comboBox1.SelectedItem == "png")
            {
                // входной файл в jpg, сохраняем его в png
                needtype = "3";
            }
            if (comboBox1.SelectedItem == "gif")
            {
                // входной файл в jpg, сохраняем его в gif
                needtype = "4";

            }
            if (comboBox1.SelectedItem == "tiff")
            {
```

```
                        // входной файл в jpg, сохраняем его в tiff
                        needtype = "5";

                    }

                    System.Threading.Thread.Sleep(10);
                    byte[] formatnow = Encoding.UTF8.GetBytes(needtype);
//отправляем тип
                    sender1.Send(formatnow);
                    System.Threading.Thread.Sleep(10);
                   // Закрываем сокет
                    sender1.Shutdown(SocketShutdown.Both);
                    sender1.Close();
                }
                catch (ArgumentNullException ane)
                {
                    txtLog.Text += "ArgumentNullException : {0}" +
ane.ToString();
                }
                catch (SocketException se)
                {
                    txtLog.Text += "SocketException : {0}" + se.ToString();
                }
                catch (Exception e1)
                {
                    txtLog.Text += "Unexpected exception : {0}" +
e1.ToString();
                }
            }
            catch (Exception e2)
            {
                txtLog.Text += e2.ToString();
            }
        }
```

6.3. Описание работы ГИИ

ГИИ принимает информацию от передатчика, сохраняет ее и преобразовывает, а затем пересылает приемнику. ГИИ имеет два

режима работы: режим обучения и режим функционирования. Преобразование систем представления выполняется путем выбора в меню режима «Функционирование». Предварительно нейронная сеть должна быть обучена при помощи режима «Обучение». Оба режима выбираются в меню программы в подменю «Выбор режима работы» (рис. 6.3). При активации подменю «Выбор режима работы», пользователю предлагается выбрать один из двух режимов: обучения или функционирования, а также возможность выхода.

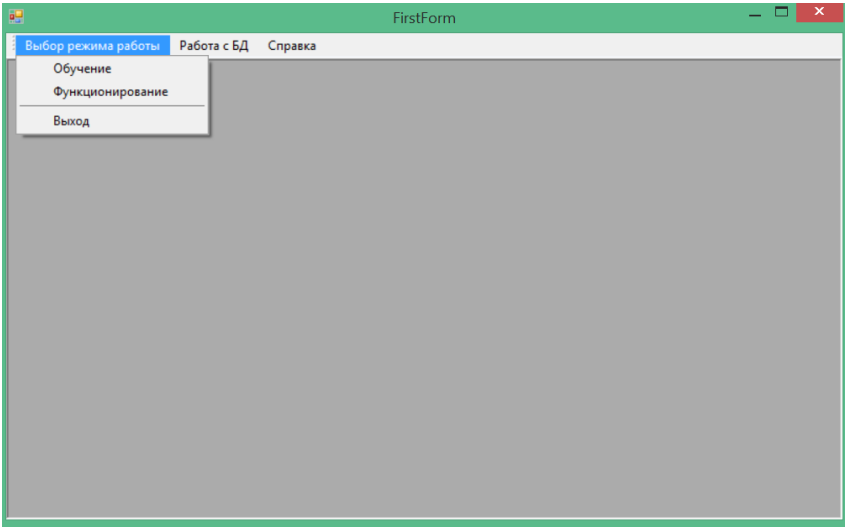


Рис. 6.3. Подменю «Выбор режима»

Блок-схема работы программы в этом случае представлена на рис. 6.4.

В начале работы программы выполняется проверка текущего состояния, то есть находится ли программа в режиме функционирования или в режиме обучения. Далее пользователь должен выбрать режим работы программы. Режим обучения предоставляет пользователю возможность обучить нейронную сеть распознавать по заранее заготовленным эталонным сообщениям различные форматы кодирования. Этот режим обязательно должен быть вы-

зван ранее режима функционирования. В ином случае машинная модель ГИИ не сможет выполнить распознавание пришедшего от передатчика сообщения.

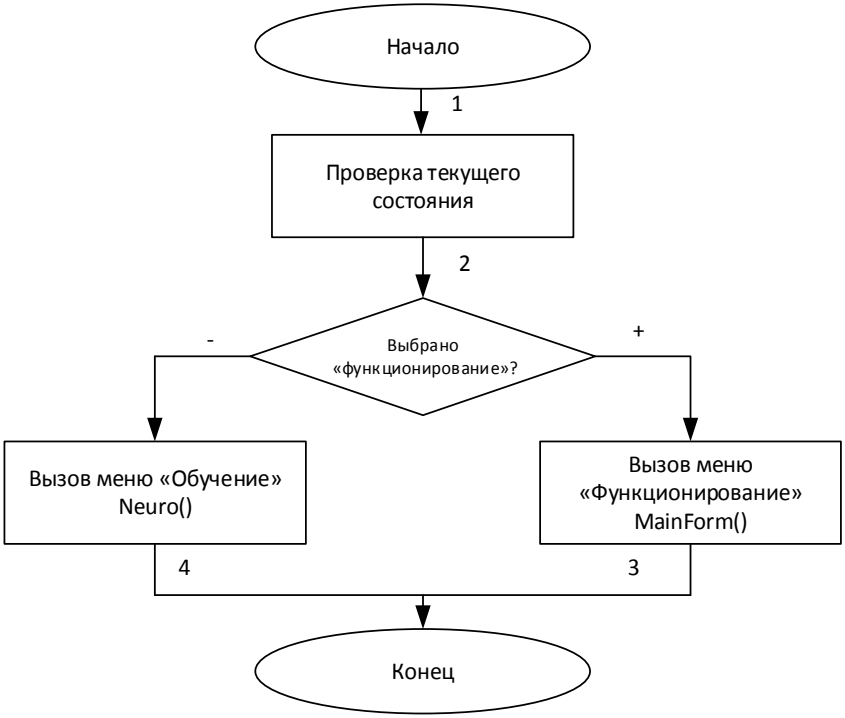


Рис. 6.4. Блок-схема подменю «Выбор режима работы»

Режим функционирования выполняет все функции ГИИ: распознает пришедшую информацию, выполняет сравнения систем представления информации приемника и передатчика и дальнейшее преобразование этих систем.

В случае любого ранее вызванного режима работы программа блокирует в подменю «Выбор режима работы» возможность его повторного вызова. При закрытии выполняемого режима в подменю снова появляется возможность вызова этого режима (рис. 6.5).

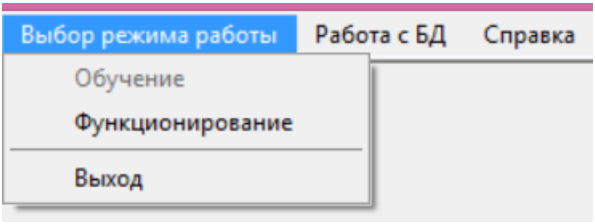


Рис. 6.5. Блокирование режима обучения при уже вызванном режиме работы

При выборе пункта «Выход», пользователю выдается сообщение о завершении работы (рис. 6.6). В случае положительного ответа программа завершает свою работу.

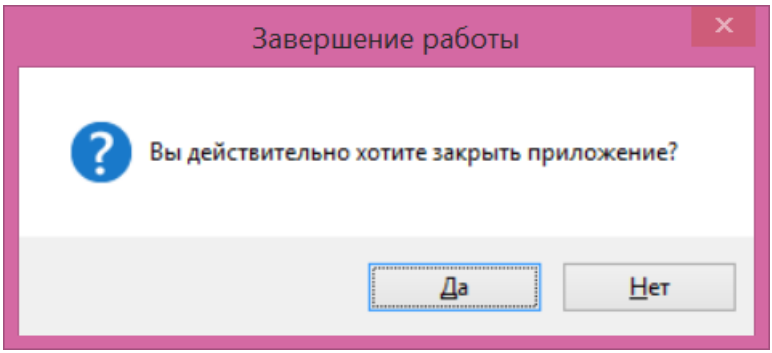


Рис. 6.6. Графическая форма завершения работы

6.4. Режим функционирования

В режиме функционирования выполняются функции, описанные в Разделе 3. Программа выполняет определение системы кодирования входного сообщения, сравнения систем кодирования приемника и передатчика, необходимые преобразования. Прием информации осуществляется при запуске приложения при помощи функции StartReceiving() по нажатию кнопки «Принять от передатчика». Листинг данной функции имеет следующий вид:

```
    private void StartReceiving()
        {
            // Буферы данных
            byte[] bytes1 = new Byte[1024];
            byte[] bytes2 = new Byte[1024];
            string data1 = null;
            string data2 = null;
            string format1 = null;
            string format2 = null;
            string dataName = null;
            int i1 = 0;
            int i2 = 0;
            int totalLenth = 0;
            IPHostEntry ipHostInfo = Dns.Resolve(Dns.GetHostName());
            IPAddress ipAddress = ipHostInfo.AddressList[0];
            IPEndPoint localEndPoint = new IPEndPoint(ipAddress, 11000);

            // Создание TCP/IP сокета.
            Socket listener = new Socket(AddressFamily.InterNetwork,
                SocketType.Stream, ProtocolType.Tcp);
            // Слушаем сокет
            try
            {
                listener.Bind(localEndPoint);
                listener.Listen(10);

                // Ждем подключения
                while (true)
                {
                    Console.WriteLine("Waiting for a connection...");
                    while (true)
                    {

                        Socket handler = listener.Accept();
                        int bytesRec1 = handler.Receive(bytes1); //размер
                        data1 += Encoding.ASCII.GetString(bytes1, 0, bytesRec1);
//Принимаем размер + формат принимаемого

                        //разбиваем строку с размером и названием формата
                        String[] words = data1.Split(new char[] { '_' }, StringSplitOp-
tions.RemoveEmptyEntries);
                        _inputFileName = @"d:\Server1" + words[1];
                        using (FileStream fs1 = File.Create(_inputFileName))
                        {
```

```
                        i1 = Convert.ToInt32(words[0]) % 1024;
                        i2 = Convert.ToInt32(words[0]) / 1024;
                        if (i1 != 0)
                           i2 = i2 + 1; //кол-во итераций считывания
                        for (int i = 0; i < i2; i++)
                        {
                           System.Threading.Thread.Sleep(10);
                           if (totalLenth + 1024 < Convert.ToInt32(words[0]))
//Принимаем файл
                           {
                              int bytesRec2 = handler.Receive(bytes2); // информация
                              data2 += Encoding.ASCII.GetString(bytes2, 0,
bytesRec2); //полностью
                              fs1.Write(bytes2, 0, bytes2.Length);
                              //   i = 5;
                           }
                           else
                           {
                              int bytesRec2 = handler.Receive(bytes2); // информация
                              data2 += Encoding.ASCII.GetString(bytes2, 0, Con-
vert.ToInt32(words[0]) - totalLenth); //последние байты
                              fs1.Write(bytes2, 0, Convert.ToInt32(words[0]) - totalLenth);
                           }
                           totalLenth += 1024;

                        }
                        bytes2 = Encoding.ASCII.GetBytes(data2);
                        // Приняли последовательность от клиента, записываем файл
                        int bytesName = handler.Receive(bytes1); //тип
                        format2 += Encoding.ASCII.GetString(bytes1, bytesName - 1, 1);
                     }
                     if (format2 == "1")
                     {
                        // входной файл в bmp, сохраняем его в jpg
                        format2 = "jpeg";
                     }
                     if (format2 == "2")
                     {
                        // входной файл в jpg, сохраняем его в bmp
                        format2 = "bmp";
                     }
                     if (format2 == "3")
                     {
                        // входной файл в jpg, сохраняем его в png
```

```
                    format2 = "png";
                }
                if (format2 == "4")
                {
                    // входной файл в jpg, сохраняем его в gif
                    format2 = "gif";

                }
                if (format2 == "5")
                {
                    // входной файл в jpg, сохраняем его в tiff
                    format2 = "tiff";

                }
                textBox1.Text = format2;
                break;
            }
            break;
        }

    }
    catch (Exception e1)
    {
        Console.WriteLine(e1.ToString());
    }
}
```

После приема информации главная форма отображает путь к принятому файлу. Его изображение в окне предпросмотра (рис. 6.7).

Общий вид блок-схемы режима функционирования ГИИ имеет вид (рис. 6.8).

После принятия входного файла от передатчика (медицинской техники) происходит анализ входного сообщения при помощи нейронной сети, результатом которого является тип формата, входного сообщения. Для этого последовательно исполняются функции: load_weigh(), save_weigh(), MassFormaration() и Load(). Данные функции загружают нейронную сеть, формируют массив принятого файла с ключевыми полями и далее анализируют файл соответственно. Листинг load_weigh() функции имеет следующий вид (рис. 6.9).

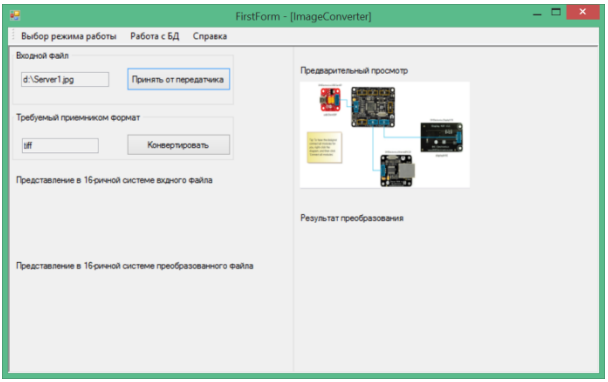


Рис. 6.7. Меню режима «Функционирование»

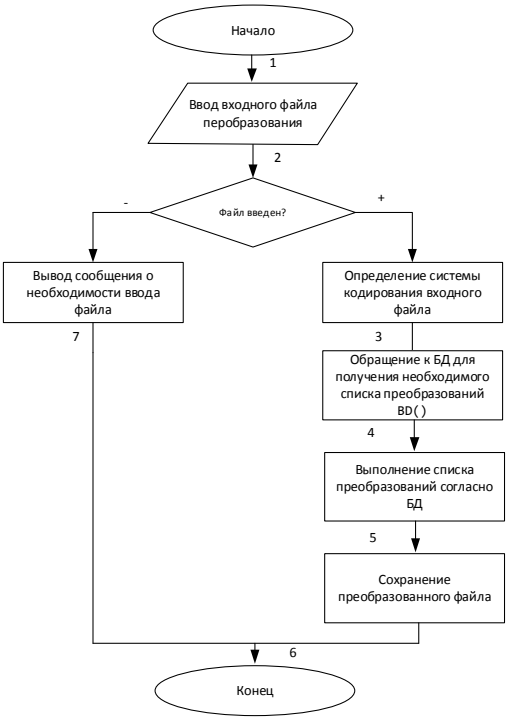


Рис. 6.8. Блок-схема режима функционирования

```
    private string load_weigh(int number)
        {
            string res = "D:\\TextAnalyze
одномерный\\TextAnalyze\\TextAnalyze\\bin\\Debug\\w_"
              + number.ToString()
             + ".txt";  // Открываем соответствующий файл весов

            return res;
        }
```

Листинг функции save_weigh() имеет следующий вид:

```
    private void save_weigh(int number)
        {
            //Запись
            string s = "";
            string[] s1 = new string[5];
            string st = "w_" + selNeur.ToString() + ".txt";
            System.IO.File.Delete("w_" + selNeur.ToString() + ".txt");
            FileStream FS = new FileStream("w_" + selNeur.ToString() + ".txt", File-
Mode.OpenOrCreate);
            StreamWriter SW = new StreamWriter(FS);

            for (int y = 0; y <= 4; y++)
            {
               s = Convert.ToString(NW[selNeur].weight[y]);
               s1[y] = s;
               SW.WriteLine(s);
            }
            SW.Close();
        }
```

Листинг функции MassFormaration() имеет следующий вид:

```
    private void MassFormaration() //Формирование массива с текстом из файла для
анализа
        {

            byte[] array;
            int[,] MassOfByte = new int[5, 8];
            int[] NumberOfSign = { 0, 1, 2, 3, 4 }; //массив, включающий номера
всех признаков
            array = File.ReadAllBytes("jpgfile.jpg"); // массив хранит все байты файла
            string file = "";
            string s = Convert.ToString(array[1], 16);
            string s1 = Convert.ToString(array[1], 2);
```

```
            for (int i = 0; i < NumberOfSign.Length; i++)
            {
               for (int j = 0; j < s1.Length; j++)
               {
                  MassOfByte[i, j] = Convert.ToInt32(s1[j].ToString()); //считывание
бит из байт-признаков

               }
            }
            for (int i = 0; i < array.Length; i++)
            {
               file = file + Convert.ToString(array[i], 16);

            }
            // label1.Text = file;
            //label1.ForeColor = System.Drawing.Color.Red;
         }
```

Листинг функции Load() имеет следующий вид:

```
         private void Load() //загрузка нейронов
         {
            for (int j = 0; j < NW.Length; j++)  // Заполняем массив номерами
            {
               NW[j] = new Web(5, input);

               string s = load_weigh(j); // вызывем необходимый файл весов
               StreamReader sr = File.OpenText(s);
               string line;
               string[] s1;
               int k = 0;
               while ((line = sr.ReadLine()) != null)
               {

                  s1 = line.Split(' ');

                  for (int i = 0; i < s1.Length; i++)
                  {
                     //    listBox1.Items.Add("");

                     NW[j].weight[k] = Convert.ToInt32(s1[i]);
                     //listBox1.Items[k] += Convert.ToString(NW[j].weight[i]);
                  }
```

```
                k++;

            }
            sr.Close();
        }

    }
```

После определения формата машинная модель ГИИ делает запрос в базу данных (БД) ГИИ для выяснения необходимых преобразований. Ответом БД ГИИ послужит упорядоченная последовательность преобразований, которую выполнит ГИИ на следующем этапе. Запрос к БД происходит при помощи функции BD(). Листинг данной функции имеет следующий вид:

```
    private string BD(string nameofconversionIN, string nameofconversionOUT)
        {
            char[] charsToTrim = { '.' };

            nameofconversionIN = nameofconversionIN.Trim(charsToTrim);
            if (nameofconversionIN == "jpg")
                nameofconversionIN = "jpeg";
            // TODO: This line of code loads data into the 'гИ-
ИИDataSet.Список_операции_упорядоченный' table. You can move, or remove it, as
needed.
            //
this.список_операции_упорядоченныйTableAdapter.Fill(this.гИИDataSet.Список_о
перации_упорядоченный);
            // TODO: This line of code loads data into the 'гИИDataSet.Операции' ta-
ble. You can move, or remove it, as needed.
            //  this.операцииTableAdapter.Fill(this.гИИDataSet.Операции);

            string str = "";
            // string queryString = "SELECT * FROM Операции";
            string queryString = "SELECT [Формат Вход].Наименование, [Формат
Выход].Наименование, Вх_Вых.Id_Вх_Вых FROM [Формат Выход] INNER JOIN
([Формат Вход] INNER JOIN Вх_Вых ON [Формат Вход].Id_Ф_Вх =
Вх_Вых.Id_Ф_Вх) ON [Формат Выход].Id_Ф_Вых = Вх_Вых.Id_Ф_Вых WHERE
((([Формат Вход].Наименование)=\"" + nameofconversionIN + "\") AND (([Формат
Выход].Наименование)=\"" + nameofconversionOUT + "\"));";

            using (OleDbConnection connection = new OleDbConnec-
tion("Provider=Microsoft.Ace.OLEDB.12.0;Data Source = ГИИ.mdb"))
```

```
            {
                OleDbCommand command = new OleDbCommand(queryString, connec-
tion);
                connection.Open();
                OleDbDataReader reader = command.ExecuteReader();
                while (reader.Read())
                {
                    str = Convert.ToString(reader.GetInt32(2));
                }
                reader.Close();
            }
            string str1 = "";
            string queryString1 = "SELECT Вх_Вых.Id_Вх_Вых, [Список
операций].Id_операции, Операции.[Наименование операции] FROM Операции
INNER JOIN (Вх_Вых INNER JOIN [Список операций] ON Вх_Вых.Id_Вх_Вых =
[Список операций].Id_Вх_Вых) ON Операции.Id_операции = [Список
операций].Id_операции WHERE (((Вх_Вых.Id_Вх_Вых)=" + str + ")) ORDER BY
[Список операций].[Очередность операции];";
            using (OleDbConnection connection1 = new OleDbConnec-
tion("Provider=Microsoft.Ace.OLEDB.12.0;Data Source = ГИИ.mdb"))
            { //Provider=Microsoft.Jet.OLEDB.4.0
                OleDbCommand command1 = new OleDbCommand(queryString1, con-
nection1);
                connection1.Open();
                OleDbDataReader reader1 = command1.ExecuteReader();
                while (reader1.Read())
                {
                    str1 = str1 + "; " + reader1.GetString(2);
                }
                reader1.Close();
            }
            return str1;
        }
```

Путь ко входному файлу отображается в специальной форме путем указания его адреса на ПК (рис. 6.8). Требуемый выходной формат задается пользователем путем выбора на графической форме (рис. 6.9).

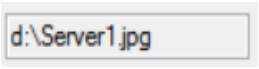


Рис. 6.8. Адрес входного файла

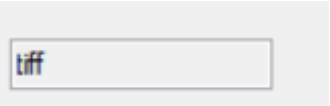


Рис. 6.9. Требуемый приемником формат

После принятия входного файла на форме (рис. 6.10) отображается входной файл. Здесь в качестве примера дан образ набора микропроцессорной платы. Для дальнейшего преобразование требуется нажать кнопку «Конвертировать».

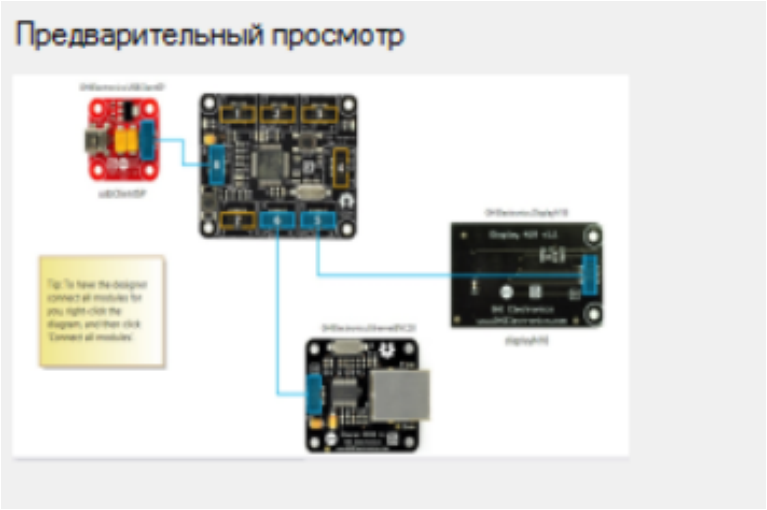


Рис. 6.10. Окно отображения входного файла

После нажатия кнопки «Конвертировать» происходит анализ программной моделью ГИИ входного файла, запрос в БД ГИИ на необходимые преобразование, а далее – сама последовательность преобразований. Результатом выполнения является сконвертированный файл. В специальных окнах на форме режима функционирования отображается 16-тиричное представление первоначального файла и его сконвертированного аналога (рис. 6.11).

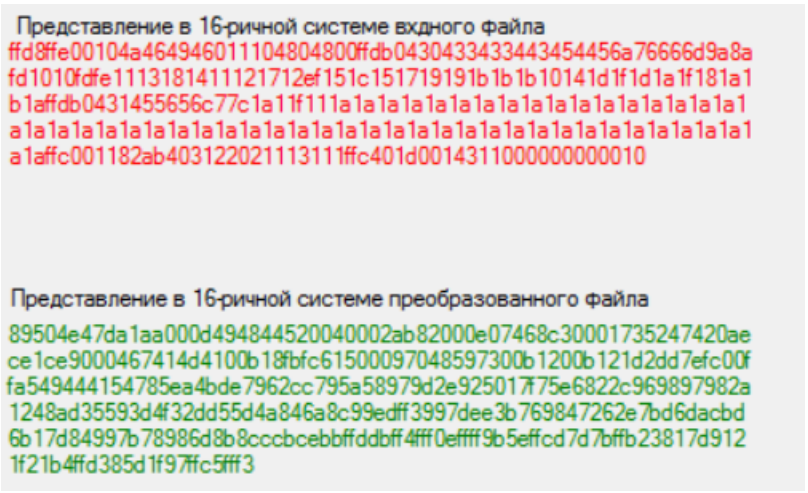


Рис. 6.11. Представление входного и выходных файлов в 16-тиричной системе представления

Здесь в качестве примера дан образ набора микропроцессорной платы. Конвертация происходит при помощи исполнения функции StartButton_Click()

Листинг данной функции имеет следующий вид:

```
private void StartButton_Click(object sender, EventArgs e)
      {
         if (string.IsNullOrEmpty(InputFileName))
            MessageBox.Show("Не задан входной файл", "ImageConverter", Message-
BoxButtons.OK, MessageBoxIcon.Error);
         else
         {
            // все параметры заданы - можно конвертировать изображение
            Image sourceImage = Image.FromFile(InputFileName);
            string inputFileExtension = Path.GetExtension(InputFileName).ToLower();

            byte[] array;
            array = File.ReadAllBytes(InputFileName); // массив хранит все байты
файла
            string file = "";
            for (int i = 0; i < 200; i++)
            {
```

```
                file = file + Convert.ToString(array[i], 16);

            }
            label2.Text = file;
            label2.ForeColor = System.Drawing.Color.Red;
            if (textBox1.Text == "jpeg")
            {
                // входной файл в bmp, сохраняем его в jpg
                sourceImage.Save(@"d:\конвертированый1.jpg", Sys-
tem.Drawing.Imaging.ImageFormat.Jpeg);
                restruct(@"d:\конвертированый1.jpg");
                //  label4.Text = BD(inputFileExtension, "jpeg");
                PostviewPictureBox.Image = Im-
age.FromFile(@"d:\конвертированый1.jpg");
            }
            if (textBox1.Text == "bmp")
            {
                // входной файл в jpg, сохраняем его в bmp
                sourceImage.Save("конвертированый2.bmp",
System.Drawing.Imaging.ImageFormat.Bmp);
                restruct("конвертированый2.bmp");
                //  label4.Text = BD(inputFileExtension, "bmp");
                PostviewPictureBox.Image = Im-
age.FromFile("конвертированый2.bmp");
            }
            if (textBox1.Text == "png")
            {
                // входной файл в jpg, сохраняем его в png
                sourceImage.Save(@"d:\конвертированый3.png", Sys-
tem.Drawing.Imaging.ImageFormat.Png);
                restruct(@"d:\конвертированый3.png");
                //      label4.Text = BD(inputFileExtension, "png");
                PostviewPictureBox.Image = Im-
age.FromFile(@"d:\конвертированый3.png");
            }
            if (textBox1.Text == "gif")
            {
                // входной файл в jpg, сохраняем его в gif
                sourceImage.Save(@"d:\конвертированый4.gif", Sys-
tem.Drawing.Imaging.ImageFormat.Gif);
                restruct(@"d:\конвертированый4.gif");
                //     label4.Text = BD(inputFileExtension, "gif");
                PostviewPictureBox.Image = Im-
age.FromFile(@"d:\конвертированый4.gif");
```

```
            }
            if (textBox1.Text == "tiff")
            {
                // входной файл в jpg, сохраняем его в tiff
                sourceImage.Save(@"d:\конвертированый5.tiff", Sys-
tem.Drawing.Imaging.ImageFormat.Tiff);
                restruct(@"d:\конвертированый5.tiff");
                //  label4.Text = BD(inputFileExtension, "tiff");
                PostviewPictureBox.Image = Im-
age.FromFile(@"d:\конвертированый5.tiff");
            }
            MessageBox.Show("Изображение сконвертировано", "ImageConverter",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
```

По окончанию работы пользователю выдается соответствующее сообщение (рис. 6.12).

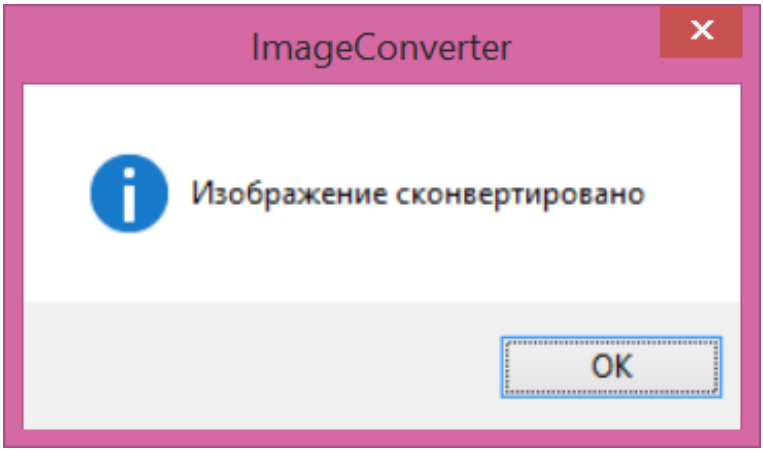


Рис. 6.12. Сообщение об окончании выполнения преобразования

В специальном окне на главной форме режима функционирования отображается результат преобразования файлов в графическом представлении (рис. 6.13).

Рис. 6.13. Окно отображения преобразованного файла

После конвертации выполняется проверка преобразованного изображения с первоначально полученным. Это происходит посредством алгоритма перцептивного хэша [30].

7. МЕТРОЛОГИЧЕСКИЙ АНАЛИЗ ГИИ

7.1. Погрешность сложных информационных пакетов в процессе преобразования

Причинами возникновения погрешностей при передаче, преобразовании информации ГИИ могут служить:

1) технические проблемы (неисправность аппаратной части или неверные условия эксплуатации);

2) ошибки программного обеспечения ГИИ (неверное исполнение алгоритмов обработки данных)

3) нехватка системных ресурсов (переполнение памяти в связи с объемом обрабатываемых данных)

4) погрешности датирования (несовпадение выполняемых действий в связи с временными задержками и последующая ресинхронизация).

В зависимости от причин возникновения погрешности ГИИ ошибка в работе ГИИ может произойти на любом из описанных в Разделе 2 шагов согласно схеме работы. Блоками, подверженными возникновению ошибок, являются следующие структурные элементы:

1) Блок запроса приемника ($R_{\text{зпр}}$).

2) Блок запроса передатчика ($R_{\text{зпе}}$).

3) Блок приема ($R_{\text{пр}}$).

4) Блок определения систем кодирования источника и приемника ($R_{\text{опр}}$).

5) Блок сравнения ($R_{\text{срав}}$).

6) Блок поиска ($R_{\text{поиск}}$).

7) Блок оптимизации ($R_{\text{опт}}$).

8) Блок преобразования ($R_{\text{преобр}}$).

9) Блок проверки ($R_{\text{пров}}$).

10) Блок передачи ($R_{\text{пер}}$).

Шаги R_i реализуются на базе электронных компонентов, составляющих конкретные структуры микроконтроллеров, образующих их электронный состав. Таким образом, функционально R_i – это программные свертки тех конкретных микропрограмм, которые участвуют в работе по «перестройке» информационных пакетов из принятого в передаваемый на фоне ограничений по разрядной сетке и межразрядных передач, которые являются реальной причиной инструментального влияния на формирование результата кодовых преобразований.

В соответствии с этим, составим уравнение интерфейсной процедуры, включающее в себя функции по приему, распознаванию, преобразованию и передачи информации. На основании [84]

$$\boldsymbol{\lambda}^{*}_{\text{вых}}(t) = \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R_{\text{срав}}R_{\text{опр}}R_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вход}}(t),$$

где l=1, n=4, m=1, k=4 – значения индексов задается необходимым количеством повторений в соответствии с алгоритмом работы ГИИ; $\gamma_{j\text{вход}}(t)$ – сложный информационный пакет (например, некоторое изображение), поставленный на блок $R_{\text{зпе}}$.

Определим уравнение полной погрешности результата ГИИ-преобразования в соответствии с методикой операционального представления сложных функциональных зависимостей, изложенной в [83].

$$\boldsymbol{\lambda}^{*}_{\text{вых}}(t) = \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R_{\text{срав}}R_{\text{опр}}R_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вход}}(t) -$$
$$\left(R^{\Gamma}_{\text{пер}}R^{\Gamma}_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R^{\Gamma}_{\text{преобр}}R^{\Gamma}_{\text{опт}}R^{\Gamma}_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вход}}(t)$$

Здесь R_i^{Γ} – гипотетическая операция (“Г”), преобразования с помощью i-го блока.

Разложение полной погрешности результата ГИИ-преобразования осуществляется в соответствии с методом [85].

1. Функция погрешности от операции приема $R_{\text{пр}}$:

$$\Delta_{\text{пр}}\boldsymbol{\lambda}^{*}_{\text{вых}}(t) = \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{Упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{Упр}|_1^m}\left(R_{\text{срав}}R_{\text{опр}}R_{\text{пр}}\right)^{\text{Упр}|_1^k} - \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{Упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{Упр}|_1^m}\left(R_{\text{срав}}R_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{Упр}|_1^k}\gamma_{j\text{вх}}(t)$$

2. Функция погрешности от операции определения передатчика $R_{\text{опр}}$:

$$\Delta_{\text{опр}}\boldsymbol{\lambda}^{*}_{\text{вых}}(t)$$
$$= \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R_{\text{срав}}R_{\text{опр}}R_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$
$$- \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$

3. Функция погрешности от операции сравнения передатчика $R_{\text{ср}}$:

$$\Delta_{\text{ср}}\boldsymbol{\lambda}^{*}_{\text{вых}}(t)$$
$$= \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$
$$- \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$

4. Функция погрешности от операции поиска передатчика $R_{\text{поиск}}$:

$$\Delta_{\text{поиск}}\boldsymbol{\lambda}^{*}_{\text{вых}}(t)$$
$$= \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}$$
$$- \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R^{\Gamma}_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$

5. Функция погрешности от операции оптимизации передатчика $R_{\text{опт}}$:

$$\Delta_{\text{опт}}\boldsymbol{\lambda}^{*}_{\text{вых}}(t)$$
$$= \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R^{\Gamma}_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$
$$- \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R^{\Gamma}_{\text{опт}}R^{\Gamma}_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$

6. Функция погрешности от операции преобразования $R_{\text{преобр}}$:

$$\Delta_{\text{преобр}}\boldsymbol{\lambda}^{*}_{\text{вых}}(t)$$
$$= \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R^{\Gamma}_{\text{опт}}R^{\Gamma}_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$
$$- \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R^{\Gamma}_{\text{преобр}}R^{\Gamma}_{\text{опт}}R^{\Gamma}_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$

7. Функция погрешности от операции преобразования $R_{\text{пров}}$:

$$\Delta_{\text{пров}}\boldsymbol{\lambda}^{*}_{\text{вых}}(t)$$
$$= \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R^{\Gamma}_{\text{преобр}}R^{\Gamma}_{\text{опт}}R^{\Gamma}_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$
$$- \left(R_{\text{пер}}R^{\Gamma}_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R^{\Gamma}_{\text{преобр}}R^{\Gamma}_{\text{опт}}R^{\Gamma}_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$

8. Функция погрешности от операции передачи $R_{\text{пер}}$:

$$\Delta_{\text{пер}}\boldsymbol{\lambda}^{*}_{\text{вых}}(t)$$
$$= \left(R_{\text{пер}}R^{\Gamma}_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R^{\Gamma}_{\text{преобр}}R^{\Gamma}_{\text{опт}}R^{\Gamma}_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$
$$-\left(R^{\Gamma}_{\text{пер}}R^{\Gamma}_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R^{\Gamma}_{\text{преобр}}R^{\Gamma}_{\text{опт}}R^{\Gamma}_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R^{\Gamma}_{\text{срав}}R^{\Gamma}_{\text{опр}}R^{\Gamma}_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{j\text{вх}}(t)$$

ГИИ – это блок, принципиально работающий в динамическом режиме. Поэтому погрешности результата кодовых преобразований являются принципиально динамическими. Процесс преобразований на 90% является алгоритмическим. Таким образом, и динамический процесс кодовых преобразований носит методический характер. Поэтому основной погрешностью результата кодовых преобразований является динамическая методическая погрешность.

$$\boldsymbol{\lambda}^{*}_{\text{вых}}(t)$$
$$= \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R_{\text{срав}}R_{\text{опр}}R_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{\text{вход}}(t)$$

Определим погрешность датирования:

$\gamma_{\text{вход}}(t)$ – числовая форма «образа»

$$\gamma_j(t+\Delta t_2)$$
$$\rightarrow \left(R_{\text{срав}}R_{\text{опр}}R_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{\text{вход}}\left(t+\Delta t_a-\sum_1^k\Delta t_{\text{пр}}-\sum_1^k\Delta t_{\text{опр}}--\sum_1^k\Delta t_{\text{срав}}\right)$$
$$\rightarrow \left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R_{\text{срав}}R_{\text{опр}}R_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{\text{вход}}(t)\left(t+\Delta t_a-\sum_1^k\Delta t_{\text{пр}}\right.$$
$$\left.-\sum_1^k\Delta t_{\text{опр}}-\sum_1^k\Delta t_{\text{срав}}-\sum_1^m\Delta t_{\text{поиск}}-\sum_1^m\Delta t_{\text{опт}}-\sum_1^m\Delta t_{\text{преобр}}\right)$$
$$\rightarrow \left(R_{\text{пер}}R_{\text{пров}}\right)^{\text{упр}|_1^n}\left(R_{\text{преобр}}R_{\text{опт}}R_{\text{поиск}}\right)^{\text{упр}|_1^m}\left(R_{\text{срав}}R_{\text{опр}}R_{\text{пр}}\right)^{\text{упр}|_1^k}\gamma_{\text{вход}}(t)\left(t+\Delta t_a\right.$$
$$-\sum_1^k\Delta t_{\text{пр}}-\sum_1^k\Delta t_{\text{опр}}-\sum_1^k\Delta t_{\text{срав}}-\sum_1^m\Delta t_{\text{поиск}}-\sum_1^m\Delta t_{\text{опт}}-\sum_1^m\Delta t_{\text{преобр}}$$
$$\left.-\sum_1^n\Delta t_{\text{пров}}-\sum_1^n\Delta t_{\text{пер}}\right)$$

Здесь t – момент начала процесса работы ГИИ; $\Delta t_{\text{пр}}$ – время исполнения функции приема; $\Delta t_{\text{опр}}$ – время исполнения функции определения; $\Delta t_{\text{срав}}$ – время исполнения функции сравнения; $\Delta t_{\text{поиск}}$ – время исполнения функции поиска; $\Delta t_{\text{опт}}$ – время исполнения функции оптимизации; $\Delta t_{\text{преобр}}$ – время исполнения функции преобразования; $\Delta t_{\text{пров}}$ – время исполнения функции проверки; $\Delta t_{\text{пер}}$ – время исполнения функции передачи.

Таким образом, погрешность датирования может быть записана следующим образом [86]:

$$\boldsymbol{\lambda}_j^*(t) = \boldsymbol{\lambda}_{j\text{вых}}(t + \Delta t_a) - \boldsymbol{\lambda}_{j\text{вых}}(t + \Delta t_a + \sum_1^k \Delta t_{\text{пр}} + \sum_1^k \Delta t_{\text{опр}} + \sum_1^k \Delta t_{\text{срав}} + \\ + \sum_1^m \Delta t_{\text{поиск}} \sum_1^m \Delta t_{\text{опт}} + \sum_1^m \Delta t_{\text{преоб}} \\ + \sum_1^n \Delta t_{\text{пров}} + \sum_1^n \Delta t_{\text{пер}})$$

Для оценки погрешностей преобразований R_i,целесообразно выполнить экспериментальный анализ ГИИ, и оценить основные характеристики его работы – быстродействие, динамическую погрешность и помехоустойчивость. Под быстродействием будем понимать скорость выполнения ГИИ совокупности действий по преобразованию входной информации к требуемому приемником виду. Оценку быстродействия будем производить при помощи измерения времени исполнения ГИИ действий с помощью программных средств.

Под помехоустойчивостью будем понимать способность системы передачи информации противостоять искажающему действию помех. Оценку помехоустойчивости будем проводить при помощи сравнения файла первоначальной информации с файлом ее преобразованной версии.

7.2. Экспериментальный анализ процесса преобразования систем кодирования

В результате проведенных исследований на базе программных и технических средств создан опытный образец Гибкого интеллектуального интерфейса, реализующего функции приема/передачи, проверки и преобразования передаваемой информации в распределенных вычислительно-измерительных сетях. В соответствии с функциональной наполненностью ГИИ составим схему проведения эксперимента. В эксперименте будем использовать програмную модель ГИИ.

Как известно [78], общая структура эксперимента может быть представлена следующим образом (рис. 7.1).

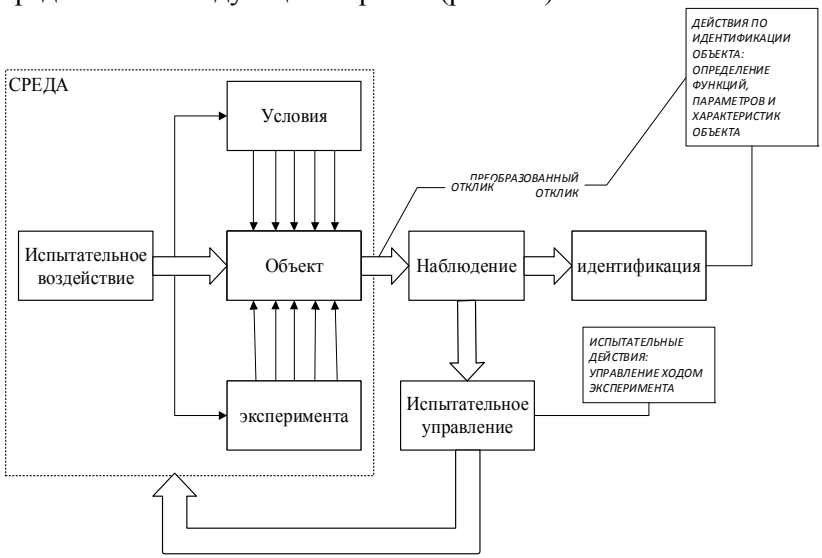


Рис. 7.1. Структура эксперимента

Эксперимент – это совокупность испытательных действий над объектами с целью наблюдений и определения его функций, характеристик и параметров. Структура физического эксперимента

определяет структуру экспериментальной установки и схему экспериментальных испытаний.

В ходе эксперимента необходимо разработать экспериментальную установку, позволяющую выполнять все действия ГИИ (прием информации от передатчика, преобразование информации со всеми сопутствующими действиями, проверку и пересылку информации приемнику) и при этом определять характеристики ГИИ (быстродействие, надежность и погрешности, возникающие в процессе работы ГИИ).

Изобразим (рис. 7.2) общую схему эксперимента ГИИ.

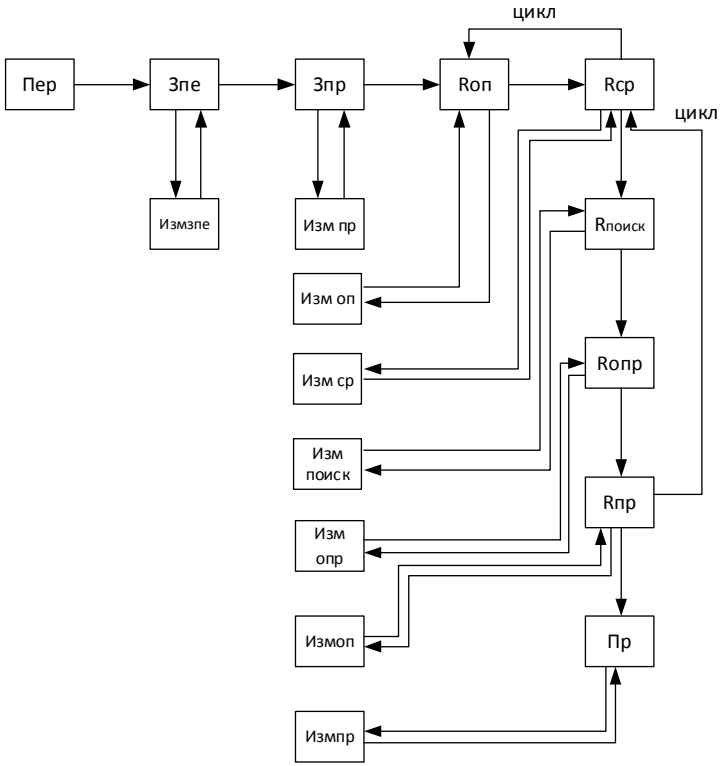


Рис. 7.2. Общая схема эксперимента ГИИ

Экспериментальная установка, реализованная на основе машинной модели, включает:

1) Блок-передатчик информации. Источник информации, которую необходимо преобразовать для дальнейшей обработки.

2) Блок-ГИИ, занимающейся преобразованием информации.

3) Блок-приемник преобразованной информации.

4) Блок измерительной структуры экспериментальной установки. Измеряет быстродействия системы преобразования, погрешности и надежность эксперимента. Является составной частью блока ГИИ для увеличения точности измерений.

Предварительные условия

1) Система находится в покое.

2) Запущены программные приложения блока-передатчика информации, блока-ГИИ и блока-приемника информации.

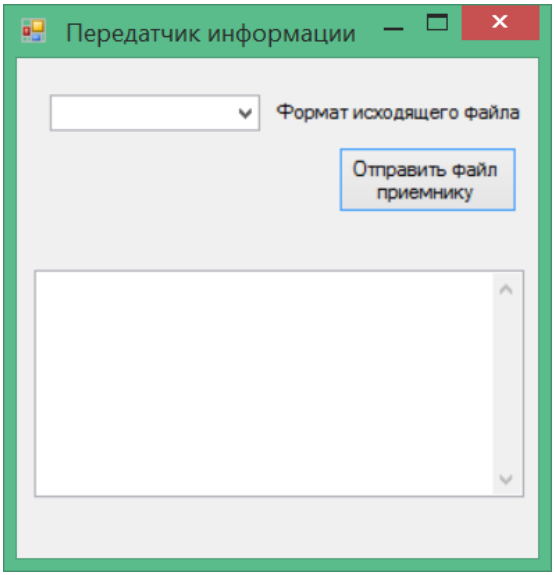


Рис. 7.3. Блок передатчика ГИИ

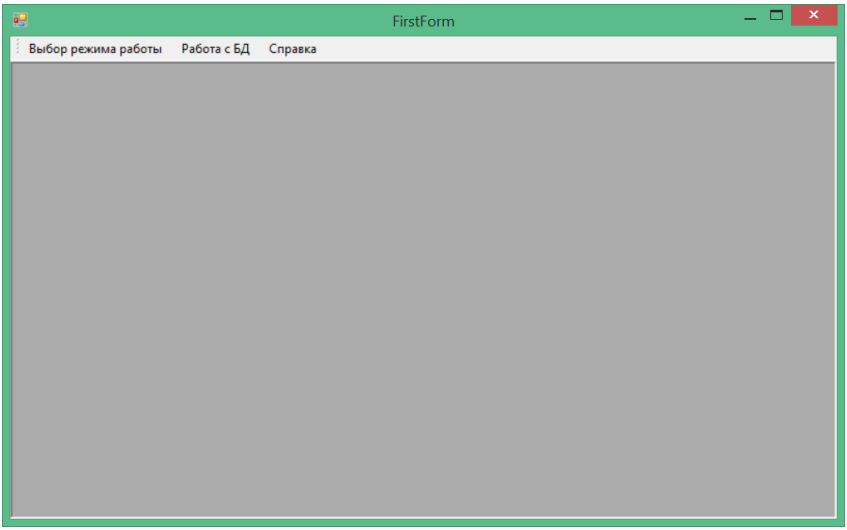


Рис. 7.4. Блок ГИИ

3) В меню выбора режима установлен «Режим функционирования».

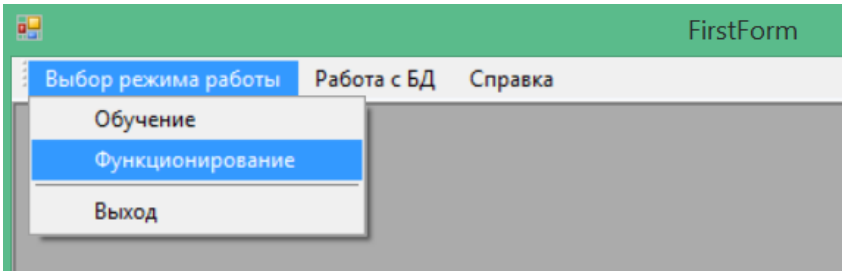


Рис. 7.5. Выбор режима функционирования ГИИ

4) ГИИ находится в режиме ожидания запроса на передачу информации от передатчика.

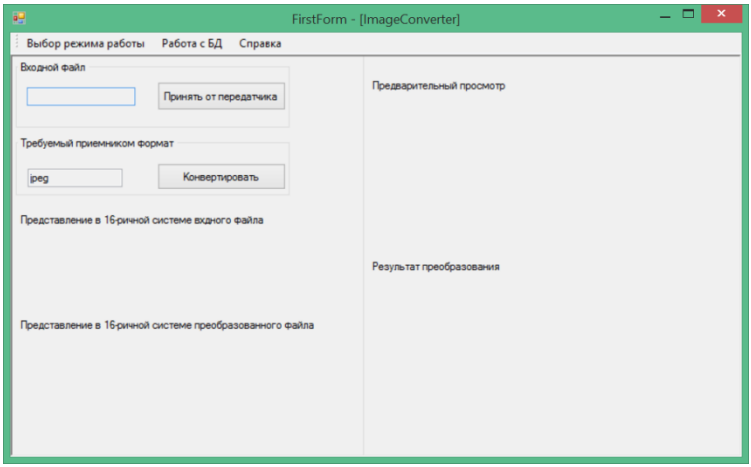


Рис. 7.6. Режим ожидания запроса на принятие

Регистрируемые сигналы и параметры

В процессе эксперимента фиксации подвергаются:

1) время выполнения операций преобразования информации;

2) количество отказов и ошибок, допущенных в процессе преобразования.

В протоколе испытаний необходимо указать код выполненной операции в соответствии с Приложением 1 (коды выполненных операций). По факту выполненных операций сделать отметки об отклонениях в работе ГИИ, составных частей и исполнительных блоков с указанием даты, времени и условий проведения регистрации.

Порядок проведения эксперимента и регистрации

В соответствии с рис. 7.1 опишем схему проведения эксперимента:

1) Передатчик отправляет запрос на пересылку информации. В соответствии с этим шагом оператор системы выбирает на гра-

фической форме блока передатчика информации кнопку пересылки информации блоку ГИИ (рис. 7.7).

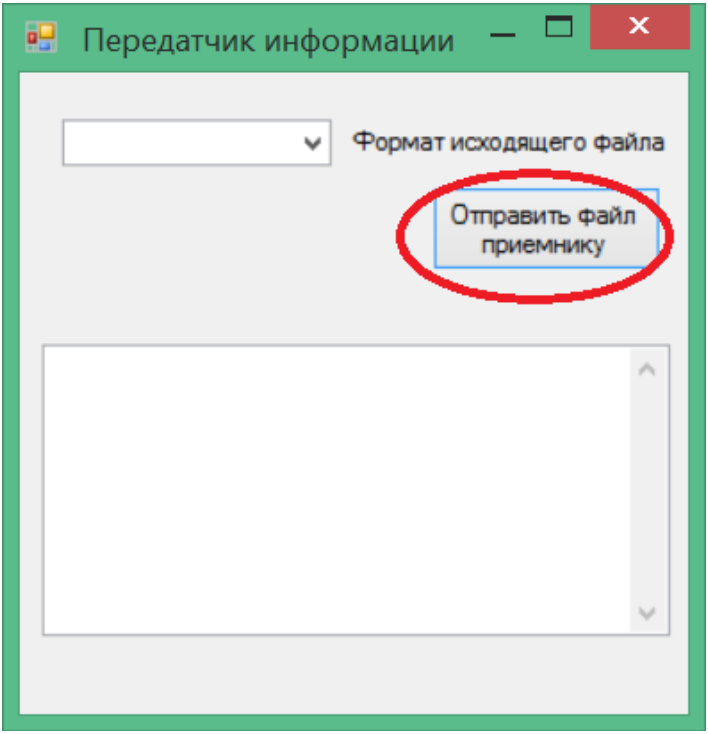


Рис. 7.7. Пересылка изображение блоком передатчика

После нажатия кнопки «Отправить файл приемнику» происходит выполнение функции StartSend(), исходный код которой можно увидеть в Приложении 2.

2) ГИИ подтверждает запрос и принимает информацию.

В соответствии с этим шагом оператор системы выбирает на графической форме блока ГИИ кнопку принять информацию от передатчика.

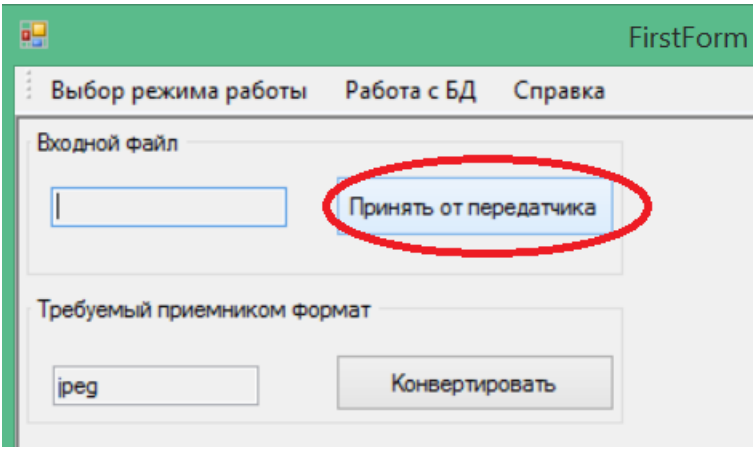


Рис. 7.8. Принятие информации от передатчика

После нажатия кнопки «Принять от передатчика» происходит выполнение функции control(), исходный код которой можно увидеть в Приложении 2.

3) ГИИ анализирует полученную информацию на необходимость преобразования в соответствии с видом принимаемой информации на стороне приемника. Анализ выполняется функцией determination(). На графической форме блока ГИИ должна быть установлена форма принимаемой приемником информации в разделе «Требуемый приемником формат» (рис. 7.9).

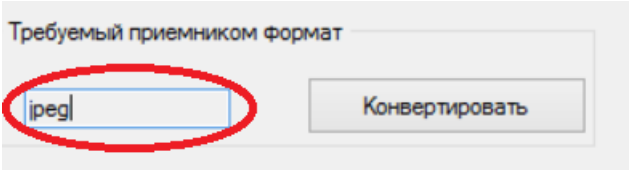


Рис. 7.9. Требуемый приемником формат

4) В случае отсутствия необходимости преобразования ГИИ ретранслирует полученную информацию приемнику.

В случае необходимости преобразования ГИИ производит необходимые действия по видоизменению информации и ее дальнейшую пересылку приемнику информации. В соответствии с этим шагом оператор системы выбирает на графической форме блока ГИИ кнопку преобразования информации «Конвертировать» (рис. 7.10).

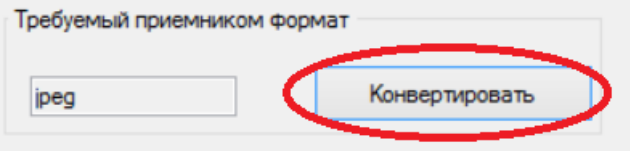


Рис. 7.10. Кнопка преобразования информации

После нажатия «Конвертировать» начинается исполнение функции conversion(). По окончанию процедуры преобразования пользователю выдается системное сообщение о завершении работы ГИИ (рис. 7.11).

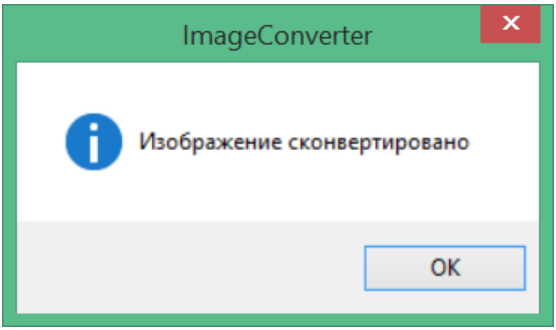


Рис. 7.11. Сообщение о завершении работы ГИИ

Критерии завершения эксперимента и регистрации

Эксперимент и регистрация работы ГИИ считаются проведенными успешно, если:

1) По завершению приема информации от блока-передатчика на экранной форме блока ГИИ наблюдается сообщение об успешном принятии изображения (рис. 7.11). Принятое изображение соответствует пересланному. Сравнение происходит как пользователем визуальным сравнением первоначального изображение с сконвертированным изображением (рис. 7.12), так и при помощи программного сравнение по алгоритму перцептивного хэша (рис. 7.13).

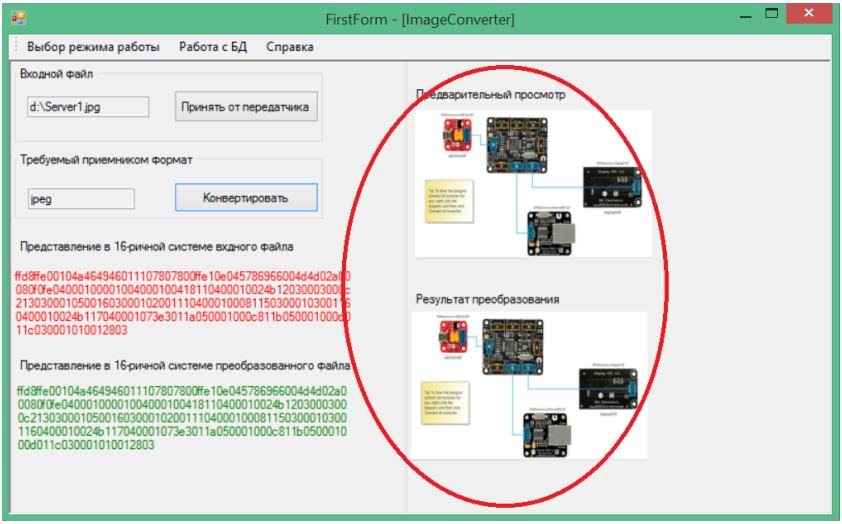


Рис. 7.12. Визуальное сравнение на графической форме блока ГИИ

При программном сравнении преобразованного и преобразуемого изображения результат может отличаться от визуального в связи с ухудшением качества изображения, которое не заметно пользователю, как и показано в примере.

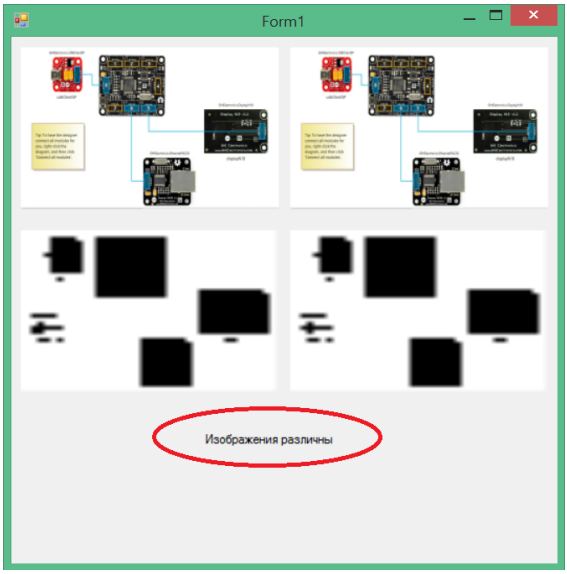


Рис. 7.13. Программное сравнение на графической форме блока ГИИ

2) По завершению приема информации от блока ГИИ, на экранной форме блока-приемника наблюдается сообщение об успешном принятии изображения.

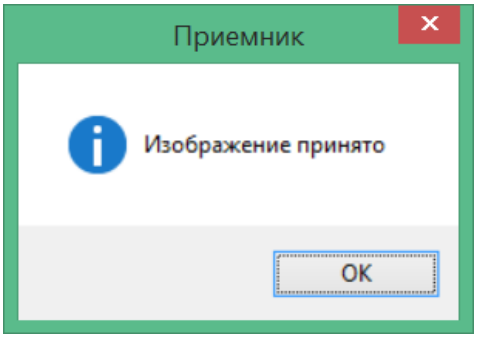


Рис. 7.14. Сообщение блока-приемника о принятии изображения

3) На экранной форме ГИИ наблюдается сообщение об успешном завершении расчета измеряемых величин и окончании эксперимента.

В соответствии с вышеописанным ходом эксперимента составим табл. 7.1.

Таблица 7.1

Воздействия на систему

№ шага	Воздействие	Блок воздействия	Отметка о выполнении
1	Выбор файла для передачи	Блок-передатчик	
2	Передача выбранной информации по команде оператора системы	Блок-передатчик	
3	Прием выбранной информации по команде оператора системы	Блок-ГИИ	
4	Преобразование информации по команде оператора системы	Блок-ГИИ	
5	Передача преобразованной информации блоку приемника информации по команде оператора системы	Блок-ГИИ	
6	Прием преобразованной информации по команде оператора системы	Блок-приемника	

В ходе работы были проведено 20 итераций эксперимента. Результаты эксперимента отражены в Приложении 1. При внешнем сходстве результатов, переданная информация отличается не только форматом сообщения, но и качеством. Изображение, используемое при эксперименте передачи информации ГИИ, представляет собой программируемый модуль фирмы GHI.

В проводимых экспериментах использована передача изображения, содержащего микроконтроллерную схему (рис. 7.15).

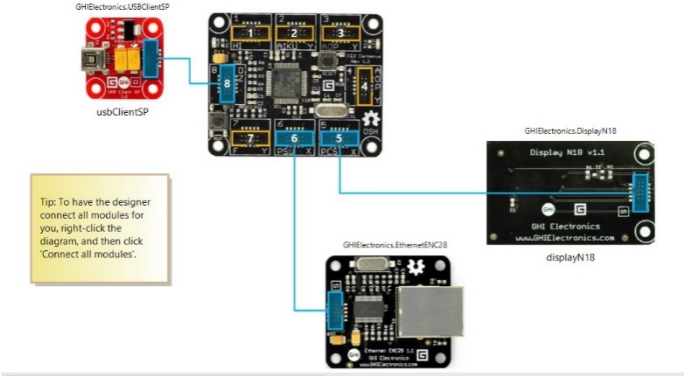


Рис. 7.15. Микроконтроллерная схема

Таким образом, проведены 20 экспериментов, учитывающих все возможные комбинации принимаемых/отсылаемых информационных сообщений. Так как одним из важнейших свойств ГИИ является достоверность преобразуемой им информации, необходимо оценить с этой точки зрения все проведенные эксперименты [82]. Для этого дадим определение достоверности информации. Достоверность информации – соответствие принятого сообщения переданному. Количественное ее определение основывается на вероятности возникновения ошибок при передаче информации. Достигается: обозначением времени свершения событий, сведения о которых передаются; тщательным изучением и сопоставлением данных, полученных из различных источников; своевременным вскрытием дезинформации; исключением искаженной информации и др. В нашем случае под достоверностью информации будем понимать соответствие принятого изображения передаваемому с заданной точностью. Под заданной точностью будем понимать точность алгоритма сравнения перцептивной хэш-функцией.

Согласно Приложению 2 все проведенные эксперименты кроме экспериментов № 4, 8, 12 являются достоверными. В экспериментах № 4, 8, 12 наблюдается несоответствие принимаемой и передаваемой после преобразования информации в связи с появлением артефактов на изображении. Данные артефакты вызваны примене-

ние сжатия с потерями первоначального изображения, что ведет к получению зернистости на изображении (рис. 7.16, 7.17).

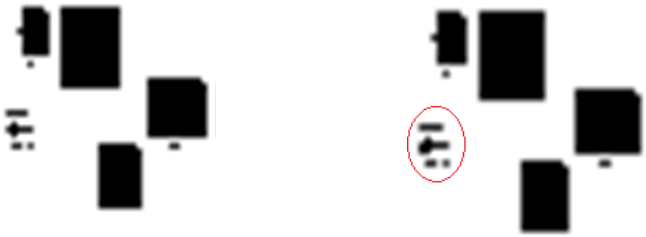


Рис. 7.16. Появление артефакта после обработки перцептивным хэшем

Далее при сравнении перцептивным хэшем, алгоритм которого предполагает уменьшение изображения (рис. 7.16) и перевода его в черно-белую палитру, явно видны различия передаваемого и принимаемого сообщений, что при попиксельном сравнении выявляет программный комплекс экспериментальной установки ГИИ. Качество изображения при этом методе прямо пропорционально зависит от степени сжатия изображения согласно алгоритму [26].

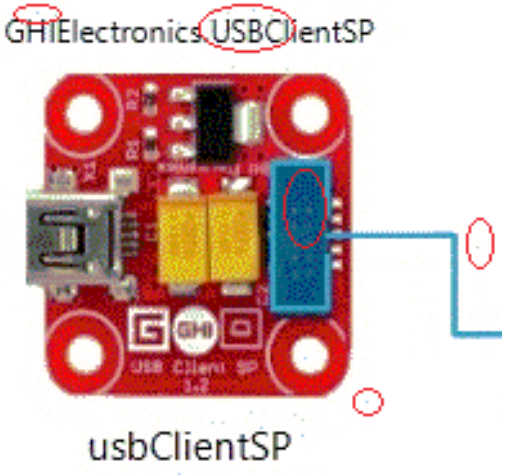


Рис. 7.17. Артефакты изображения

В случае если первоначальными условиями заданы низкие требования к достоверности, например, при передаче графиков кардиограмм, данный эксперимент пройдет успешно в связи с тем, что артефакты перцептивным хэшем будут не распознаны (эксперименты № 1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15, 17, 18, 19, 20).

Погрешности в таком случае можно оценить, как процентное отличие принятых и полученных сообщений при сравнении алгоритмом перцептивного хэша. Так для таблицы (Приложение) соответствующая погрешность указана в табл. 7.2. При каждой итерации подсчитан процент отличия изображений (рис. 7.16).

Таблица 7.2

Процентное несовпадение изображений

№	Процент несовпадения первоначального и преобразованных изображений, %
1.	0
2.	0
3.	0
4.	12
5.	0
6.	0
7.	0
8.	9
9.	0
10.	0
11.	0
12.	11
13.	0
14.	0
15.	0
16.	12
17.	0
18.	0
19.	0
20.	0

Размер отличия изображений может влиять на постановку конечного диагноза врачом.

7.3. Определение оценок ГИИ

В ходе эксперимента были проведены два типа исследований:

1. Расчет погрешности, определение времени выполнения функций ГИИ;

2. Сравнение принятой и преобразованной информации на идентичность при помощи алгоритма перцептивной хэш-функции.

Эксперимент первого типа осуществлен при выполнении функций, возложенных на ГИИ, и состоит в измерении времени реализация как отдельных функций $\Delta t_{\text{сумм}}$ так и измерении общего времени $\Delta t_{\text{реалсум}}$ отдельным таймером с дальнейшим сравнением $\Delta t_{\text{разн}}$.

В ходе эксперимента было проведено 140 реализаций процесса функционирования системы ГИИ, состоящего из последовательности операций передачи информации, ее приема, распознавания, конвертации и дальнейшего сравнения результата сообщений по всем форматам. Сообщение представляет собой изображение определенного формата. В соответствии с этой последовательностью была построена Таблица 5, содержащая следующие столбцы:

1) «Прием» – содержит информацию о длительности времени приема данных от передатчика к ГИИ ($\Delta t_{\text{пр}}$);

2) «Распознавание» – содержит информацию о длительности времени выполнения процедуры распознавания данных ГИИ ($\Delta t_{\text{расп}}$);

3) «Конвертация» – содержит информацию о длительности времени выполнения процедуры конвертирования данных ГИИ ($\Delta t_{\text{конв}}$);

4) «Сравнение» – содержит информацию о длительности времени выполнения процедуры сравнения данных ГИИ ($\Delta t_{\text{ср}}$);

5) «Передача» – содержит информацию о длительности времени выполнения процедуры передачи данных от ГИИ к приемнику ($\Delta t_{\text{пер}}$);

6) «Время суммированное» – содержит алгебраическую сумму времени выполнения операций пунктов 1–5 ($\Delta t_{\text{сумм}}$).

7) «Сумма реальная» – содержит информацию о длительности времени выполнения процедур 1–5, замеренную отдельным программным таймером ($\Delta t_{\text{реалсум}}$).

8) «Разница в измерениях» – содержит разность между длительностями времен, записанных в столбцах 6 и 7 $\Delta t_{\text{разн}}$.

9) «Погрешность» – содержит погрешность в процентном отношении между столбцами 6 и 7 $\Delta t_{\text{погр}}$.

10) «Объем входного файла» – содержит размер файла, передаваемого на ГИИ, выраженного в Кб.

11) «Размер входного файла» – содержит размер файла, передаваемого на ГИИ, выраженного в пикселях.

Замер времени делался программным таймером с точностью до одиннадцатого знака после запятой.

Эксперимент проводился на файле с одним и тем же изображением, сохраненным в пяти различных форматах. Кроме этого, в каждом из форматов изображение сохранялось в семи различных разрешениях, что влияет на качество сохраняемого изображения и, следовательно, на его размер. В ходе эксперимента каждый файл каждого из форматов и размеров поочередно преобразовывался во все оставшиеся форматы.

Каждая из процедур, указанных таблице (Приложение 3) имеет свое уникальное время исполнения. Наиболее быстро исполняемой функцией является функция конвертации данных из формата передатчика в формат приемника, так как используются отлаженные алгоритмы преобразования операционной системы. Наиболее затратными по времени являются функции приема и передачи информации ГИИ, значения которых сильно зависят от объема передаваемых данных и алгоритма передачи информации.

Результаты эксперимента показывают, что в зависимости от размера обрабатываемого файла изменяется время выполнения каждой операции, что влияет на суммарное время. Вне зависимости от размеров обрабатываемого файла значение алгебраической суммы и реальной суммы отличаются.

Это доказывает, что в процессе обработки информации при помощи ГИИ присутствует задержка: алгебраическая сумма времени

исполнения функций по отдельности (время суммированное) меньше времени исполнения всей программы, измеренного при помощи счетчика (время общее). То есть можно говорить о наличии погрешности датирования.

Погрешность датирования возникает за счет появления при проведении эксперимента задержек на исполнение не только основных функций программной модели ГИИ, но также на работу операционной системы, обработку вне программных функций ПК. Значения погрешности датирования изменяются от 2 до 50% в зависимости от размера и формата передаваемого файла.

Однако остается неясным, насколько влияют на погрешности плавающие параметры (размер файла и формат). Для завершения эксперимента на вычисление погрешностей при передаче информации при помощи ГИИ, проведем 100 итераций того же эксперимента, что и ранее, но с фиксированными размерами и форматом преобразования: выберем преобразование из формата JPEG в BMP файла размером 2000 x 3000 пикселей и объемом 1680 Кб. Результаты эксперимента приведены в Таблице (Приложение 4).

Таким образом, после проведения анализа экспериментальных данных, можно говорить, о том, что несмотря на фиксацию параметров файла, погрешности при передаче данных ГИИ сохраняются и отличаются между собой в каждой из проведенных 100 итераций. Из этого следует, что на возникновение погрешности влияет не только внутренняя среда ГИИ, но и внешняя: каждый из экспериментов обладает уникальным временем исполнения операций ГИИ.

В ходе проведения второго типа из вышеперечисленных экспериментов, эксперимента сравнения принятой и преобразованной информации на идентичность, было проведено 84 итерации (табл. 7.3). Суть эксперимента заключается в том, что исследуется зависимость устанавливаемой пользователем точности к конвертированному и первоначальному изображениям в пикселях. Данный размер влияет на точность сравнения первоначального и преобразованного изображения при использовании алгоритма перцептивного хэша.

В ходе данного эксперимента были рассмотрены два случая:

1) сравнение преобразованного изображения в формате gif с первоначальным изображением в формате jpeg;

2) сравнение преобразованного изображения в формате png с первоначальным изображением в формате jpeg.

Данные случаи выбраны с учетом, что при преобразовании изображения в формат gif качество изображение сильно ухудшается, а в формат png остается прежним. Эти случаи являются наиболее показательными и закономерности, выявленные в ходе данного эксперимента применимы и для других случаев.

В табл. 7.3 отображены результаты эксперимента, она содержит следующие столбцы:

1) «Номер эксперимента» – отображает порядковый номер итерации;

2) «Формат сравнения» – содержит формат преобразованного изображения;

3) «Эталонный формат» – содержит формат эталонного изображения;

4) «Первоначальный размер изображения» – содержит размер первоначального изображения в пикселях;

5) «Точность» – указывает размер изображения в пикселях при сравнении методом перцептивного хэша;

6) «Результат сравнения» – содержит результат сравнения алгоритмом перцептивного хэша.

Таблица 7.3

Результаты эксперимента «Сравнение принятой и преобразованной информации на идентичность»

№	Формат сравнения	Эталонный формат	Первоначальный размер изображения (пикс х пикс)	Заданная точность (пикс)	Результат сравнения
1.	gif	jpeg	100 х 150	2	одинаковы
2.	gif	jpeg	100 х 150	4	одинаковы
3.	gif	jpeg	100 х 150	7	различны
4.	gif	jpeg	100 х 150	10	различны
5.	gif	jpeg	100 х 150	13	различны
6.	gif	jpeg	100 х 150	15	различны
7.	gif	jpeg	200 х 300	2	одинаковы

№	Формат сравнения	Эталонный формат	Первоначальный размер изображения (пикс х пикс)	Заданная точность (пикс)	Результат сравнения
8.	gif	jpeg	200 x 300	4	одинаковы
9.	gif	jpeg	200 x 300	7	различны
10.	gif	jpeg	200 x 300	10	различны
11.	gif	jpeg	200 x 300	13	различны
12.	gif	jpeg	200 x 300	15	различны
13.	gif	jpeg	500 x 750	2	одинаковы
14.	gif	jpeg	500 x 750	4	одинаковы
15.	gif	jpeg	500 x 750	7	различны
16.	gif	jpeg	500 x 750	10	различны
17.	gif	jpeg	500 x 750	13	различны
18.	gif	jpeg	500 x 750	15	различны
19.	gif	jpeg	1000 x 1500	2	одинаковы
20.	gif	jpeg	1000 x 1500	4	одинаковы
21.	gif	jpeg	1000 x 1500	7	одинаковы
22.	gif	jpeg	1000 x 1500	10	различны
23.	gif	jpeg	1000 x 1500	13	различны
24.	gif	jpeg	1000 x 1500	15	различны
25.	gif	jpeg	2000 x 3000	2	одинаковы
26.	gif	jpeg	2000 x 3000	4	одинаковы
27.	gif	jpeg	2000 x 3000	7	одинаковы
28.	gif	jpeg	2000 x 3000	10	одинаковы
29.	gif	jpeg	2000 x 3000	13	различны
30.	gif	jpeg	2000 x 3000	15	различны
31.	gif	jpeg	3000 x 4500	2	одинаковы
32.	gif	jpeg	3000 x 4500	4	одинаковы
33.	gif	jpeg	3000 x 4500	7	одинаковы
34.	gif	jpeg	3000 x 4500	10	одинаковы
35.	gif	jpeg	3000 x 4500	13	одинаковы
36.	gif	jpeg	3000 x 4500	15	различны
37.	gif	jpeg	4000 x 6000	2	одинаковы
38.	gif	jpeg	4000 x 6000	4	одинаковы
39.	gif	jpeg	4000 x 6000	7	одинаковы
40.	gif	jpeg	4000 x 6000	10	одинаковы
41.	gif	jpeg	4000 x 6000	13	одинаковы
42.	gif	jpeg	4000 x 6000	15	одинаковы
43.	png	jpeg	100 x 150	2	одинаковы
44.	png	jpeg	100 x 150	4	одинаковы
45.	png	jpeg	100 x 150	7	одинаковы
46.	png	jpeg	100 x 150	10	одинаковы

№	Формат сравнения	Эталонный формат	Первоначальный размер изображения (пикс х пикс)	Заданная точность (пикс)	Результат сравнения
47.	png	jpeg	100 x 150	13	одинаковы
48.	png	jpeg	100 x 150	15	одинаковы
49.	png	jpeg	200 x 300	2	одинаковы
50.	png	jpeg	200 x 300	4	одинаковы
51.	png	jpeg	200 x 300	7	одинаковы
52.	png	jpeg	200 x 300	10	одинаковы
53.	png	jpeg	200 x 300	13	одинаковы
54.	png	jpeg	200 x 300	15	одинаковы
55.	png	jpeg	500 x 750	2	одинаковы
56.	png	jpeg	500 x 750	4	одинаковы
57.	png	jpeg	500 x 750	7	одинаковы
58.	png	jpeg	500 x 750	10	одинаковы
59.	png	jpeg	500 x 750	13	одинаковы
60.	png	jpeg	500 x 750	15	одинаковы
61.	png	jpeg	1000 x 1500	2	одинаковы
62.	png	jpeg	1000 x 1500	4	одинаковы
63.	png	jpeg	1000 x 1500	7	одинаковы
64.	png	jpeg	1000 x 1500	10	одинаковы
65.	png	jpeg	1000 x 1500	13	одинаковы
66.	png	jpeg	1000 x 1500	15	одинаковы
67.	png	jpeg	2000 x 3000	2	одинаковы
68.	png	jpeg	2000 x 3000	4	одинаковы
69.	png	jpeg	2000 x 3000	7	одинаковы
70.	png	jpeg	2000 x 3000	10	одинаковы
71.	png	jpeg	2000 x 3000	13	одинаковы
72.	png	jpeg	2000 x 3000	15	одинаковы
73.	png	jpeg	3000 x 4500	2	одинаковы
74.	png	jpeg	3000 x 4500	4	одинаковы
75.	png	jpeg	3000 x 4500	7	одинаковы
76.	png	jpeg	3000 x 4500	10	одинаковы
77.	png	jpeg	3000 x 4500	13	одинаковы
78.	png	jpeg	3000 x 4500	15	одинаковы
79.	png	jpeg	4000 x 6000	2	одинаковы
80.	png	jpeg	4000 x 6000	4	одинаковы
81.	png	jpeg	4000 x 6000	7	одинаковы
82.	png	jpeg	4000 x 6000	10	одинаковы
83.	png	jpeg	4000 x 6000	13	одинаковы
84.	png	jpeg	4000 x 6000	15	одинаковы

В процессе сравнения было выявлено, что преобразованное изображение в формат без потери качества изображения (png) при любой точности изображения содержит первоначальную информацию.

Преобразованное изображение с потерей качества, в зависимости от заданной точности сравнения, может отличаться от первоначального. Это зависит от размера первоначального изображения: чем выше разрешение первоначального изображения, тем с большей точностью сравнения распознаются первоначальное и преобразованное изображения как одинаковые. Чем ниже качество первоначального изображения, тем ниже должна быть заданная точность сравнения изображений как одинаковых.

Это хорошо видно на первых 42 экспериментах: при постепенном увеличении первоначального изображения постепенно увеличивается и точность сравнения изображений для достижения результата сравнения «одинаковы». Так при разрешении 4000 х 6000 пикселей формата gif точность изображения, при которой результат сравнения имеет значение «одинаковы», равно 15 пикселям (эксперимент № 42) (рис. 7.18).

а – эталонное изображение, формат JPEG

б – преобразованное изображение, формат GIF

Рис. 7.18. Сравнение изображений при точности 15 пикселей при первоначальном разрешении 4000 х 6000

При этой же точности сравнения, но меньшем первоначальном изображении 3000 х 4500 результат сравнения имеет значение «различны» (рис. 7.19).

а – эталонное изображение, формат JPEG

б – преобразованное изображение, формат GIF

Рис. 7.19. Сравнение изображений при точности 15 пикселей при первоначальном разрешении 3000 х 4500

Подводя итоги метрологического анализа ГИИ, следует отметить, что:

1) несовпадение при сравнении входного и выходного изображений (Приложение 1 и Приложение 2) характеризует абсолютную статическую погрешность [86], для которой роль эталонной играет входное изображение, и она не зависит от временного параметра;

2) Приложение 3, Приложение 4 содержат информацию о быстродействии всех этапов преобразований: $R_{\text{пр}}$, $R_{\text{пер}}$, $R_{\text{зпе}}$, $R_{\text{зпр}}$, $R_{\text{срав}}$, $R_{\text{опт}}$, $R_{\text{пр}}$.

3) Приложение 3, Приложение 4 позволяют оценить динамическую погрешность в форме погрешности датирования $\Delta t_{\text{дат}}$. Она имеет случайный характер, с нормальным законом распределения, что следует из большого числа конструктивных причин, независимых друг от друга, приводящих к случайности процесса датирования. Характеристики можно определить следующим образом:

$\mathrm{M}\{\Delta t_{\text{дат}}\} = \lim_{N\to\infty} \Delta t_{\text{дат}}/\mathrm{N}$

$\mathrm{M}[\Delta t_{\text{дат}}] = \frac{\Delta t_{\text{дат}1}+\Delta t_{\text{дат}2}+\cdots+\Delta t_{\text{дат}100}}{100} = 0{,}075474$

$\mathrm{D}\{\Delta t_{\text{дат}}\} = \lim_{N\to\infty} \sum_1^N \frac{(\Delta t_{\text{дат}i}-M[\Delta t_{\text{дат}}])^2}{N}$

$\mathrm{D}[\Delta t_{\text{дат}}] = \frac{(\Delta t_{\text{дат}1}-M[\Delta t_{\text{дат}}])^2+\cdots+(100-M[\Delta t_{\text{дат}}])^2}{100} = 1{,}31624\text{E-}06$

4) Погрешность, связанная с разрешением изображения (табл. 7.3), является статической, так как не зависит от времени и инструментальной, так как определяется процессом формирования цифрового файла.

5) Погрешности от конвертации – методические, так как характеризуют все процессы формирования файла в зависимости от используемых алгоритмов сжатия.

6) Погрешности всех шагов R_i – мультипликативные, так как зависят от входных величин, что проявляется неизбежно при каждом формировании конкретного цифрового файла в рамках каждого шага R_i и в зависимости от системы кодирования, то есть алгоритмов сжатия и восстановления конвертируемых изображений.

7.4. Технические характеристики ГИИ

Эксперименты проводились на программной модели на персональной ЭВМ с установленными программными пакетам: ОС Windows 7 Professional, а также MS Visual Studio 2012.

Непосредственная реализация планируется к выполнению на программируемом микроконтроллерном модуле GHI Electronics G400-D (рис. 7.20).

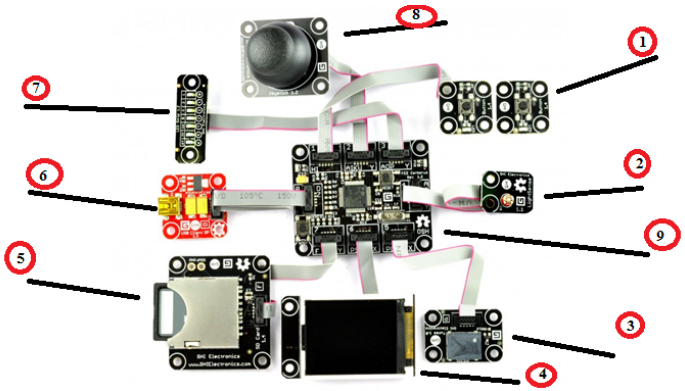


Рис. 7.20. Микроконтроллер модуле GHI Electronics G400:
1 – Модуль кнопок для GHI Electronics G400; 2 – Кнопка GHI Electronics G400;
3 – Звуковой модуль GHI Electronics G400; 4 – Экран GHI Electronics G400;
5 – Модуль для flash-карты GHI Electronics G400; 6 – Интерфейс для связи
с программатором; 7 – Модуль светодиодов для GHI Electronics G400;
8 – Джойстик; 9 – Микроконтроллерный блок GHI Electronics G400

Технические характеристики данного модуля (табл. 7.4).

Таблица 7.4

Технические характеристики
микроконтроллерного модуля GHI Electronics G400

Процессор	Atmel 400 МГц MPU
Оперативная память	128MB DDR SDRAM
Постоянная память	4 Мб (8 Мб желанию) последовательной флэш-памяти
Интерфейс программирования	USB Host / Device
Интерфейсы взаимодействия	сетевой (Ethernet, Wi-Fi), SQLite базы данных, и графика; Сетевые технологии - TCP / IP с Ethernet, Wi-Fi поддержка
Размер	1261.81mil x 2657.08mil (32.05 x 67.49mm);
Дополнительно	Поддерживает следующие технологии: файловая система FAT, TCP/IP, графику и .NET API. Кроме того, встроенное программное обеспечение включает в себя .NET Micro Framework библиотеки, которые добавляют поддержку важных функций, таких как Wi-Fi, USB Host, SQLite.

Для реализации данной программной разработки на микроконтроллере используется программная среда .NET Micro Framework. Даная среда – реализация методологии разработки программного обеспечения .NET, предназначенная для 32-разрадных микроконтроллеров. Основная идея .NET MF – предоставить разработчикам компактных и даже карманных устройств возможность использовать современный язык программирования C# совместно с мощной библиотекой классов, предоставляющей широкие возможности работы с различными устройствами, файловыми системами, аппаратными интерфейсами, сетью и многим другим.

Выбранный микроконтроллер позволяет реализовать все функции ГИИ.

8. ПРИМЕРЫ ПРАКТИЧЕСКОГО ПРИМЕНЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНО-АППАРАТНЫХ КОМПЛЕКСОВ

8.1. Модуль обмена видеоинформацией в телемедицине

Модуль обмена видеонформацией (МОВ) отвечает за механизм работы с видео, механизм захвата и редактирования изображений. Обеспечивает функционирование обмена видеоинформацией. Определяет способ и устройство видеозахвата, преобразование, калибровку и передачу сигнала через модуль сетевого взаимодействия, а также прием и воспроизведение видеоинформации на клиентских станциях назначения. Данный модуль включает в себя программный компонент по работе с видео, который реализует функции захвата, корректировки и масштабирования изображений, а также включает в себя программный компонент обеспечения интерфейса работы с медицинскими приборами, позволяющий получать и редактировать изображения в формате DICOM.[12].

Модуль обмена видеоинформацией является универсальным структурным блоком системы. Он используется как в клиентской, так и в серверной части АИС ТМК [12].

DICOM (Digital Imaging and COmmunications in Medicine, цифровые изображения и обмен ими в медицине) – это индустриальный стандарт для передачи радиологических изображений и другой медицинской информации между компьютерами, опирающийся на стандарт Open Systera Interconnection (OSI), разработанный Международной организацией по стандартам (International Standards Organization, ISO). Стандарт DICOM описывает «паспортные» данные пациента, условия проведения исследования, положение пациента в момент получения изображения и т.п., для того чтобы впоследствии было возможно провести медицинскую интерпретацию данного изображения. Первая версия данного

стандарта была разработана Американским Колледжем Радиологии (American College of Radiology, ACR) и Национальной ассоциацией производителей электронного оборудования (National Electrical Manufacturers Association, NEMA) в 1985 году.

Стандарт позволяет организовать цифровую связь между различным диагностическим и терапевтическим оборудованием, использующимся в системах различных производителей. Рабочие станции, компьютерные (КТ) и магнитно-резонансные Томографы (МРТ), микроскопы, УЗ-сканеры, общие архивы и хост – компьютеры от разных производителей, расположенные в одном городе или нескольких городах, могут «общаться» друг с другом на основе DICOM с использованием открытых сетей по стандартным протоколам, например TCP/IP.

С использованием DICOM' а можно проводить различные медицинские исследования в территориально-распределенных диагностических центрах с возможностью сбора и обработки информации в нужном месте.

Стандарт DICOM версии 3.0 (окончательно выпущена в 1993 г.) предназначен для передачи медицинских изображений, получаемых с помощью различных методов лучевой и иной диагностики, в его описании перечислено 29 диагностических методов. Данный стандарт получил широкое распространение в США, Японии, Германии и других странах.

К началу 80-х гг. проблема интеграции сложных цифровых устройств лучевой диагностики стояла уже достаточно остро, что побудило Американский институт радиологии (ACR) и Национальную ассоциацию производителей электрооборудования CfflA (NEMA) заняться разработкой стандарта передачи цифровых медицинских изображений. В 1983 г. ими был создан объединенный комитет, в задачи которого входила разработка стандарта. Стандарт обеспечивает передачу цифровых медицинских изображений, не зависящую от производителей диагностического оборудования, и способствующую:

– развитию систем архивирования и передачи изображений (PACS);

– обеспечению их взаимодействия с автоматизированными больничными информационными системами;

– созданию баз данных, содержащих диагностическую информацию, получаемую с помощью большого числа удаленных друг от друга устройств различных типов.

Первая версия этого стандарта была опубликована в 1985 г., а в 1988 г. вторая. Новая, третья версия стандарта была выпущена в 1991 г. и перерабатывалась до 1993 г. Версия получила название DICOM 3.0 (английская аббревиатура от Digital Imaging and Comminications in Medicine (Standard) и явилась существенным шагом вперед, поскольку ее можно было использовать в сетевых средах с использованием стандартных протоколов, например TCP/IP; в ней были описаны уровни совместимости со стандартом, семантика программных команд и ассоциированных с ними данных; структура стандарта была приведена в соответствие с директивами Международной организации по стандартам (ISO). Все это привело к тому, что такие крупнейшие производители диагностического оборудования, как General Electric, Philips, Siemens и др., начали планировать в составе своих систем возможность обмена изображениями в стандарте DICOM 3.0.

Для описания и интерпретации медицинских изображений недостаточно хранить их в универсальных форматах (таких как JPG, Tiff, Bmp). Для интерпретации таких изображений необходимо, чтобы была еще и сопутствующая информация, с помощью которой специалист может получить полную картину представленного перед ним клинического случая. При этом необходимо кроме собственно снимка, иметь достаточно подробные сведения об условиях, в которых он был сделан, например о расстоянии до фотографируемого объекта, его пространственном положении, освещенности, состоянии предмета в момент получения снимка (к примеру, движение или покой), а также об экспозиции, фокусном расстоянии объектива и размере диафрагмы, характеристиках использовавшегося светофильтра и т.д. При интерпретации нередко приходится сравнивать серии снимков одного и того же объекта, полученные в разное время и в разных условиях

Поскольку медицинские изображения нужны, прежде всего, для профессиональной интерпретации, то каждому такому изображению также должно сопутствовать описание условий, в которых они были сделаны, а также сведения о состоянии пациента в момент получения изображения. Поэтому, основной целью создания стандарта является формализация описания условий получения и хранения изображений, а также сведений о состоянии пациента.

С расширением парка высокотехнологичных медицинских диагностических устройств лучевой диагностики, называемых также устройствами медицинской визуализации (цифровые рентгеновские системы, компьютерные, магнитно-резонансные, позитронно-эмиссионные томографы, системы ультразвуковой диагностики и т.д.), проблема электронного обмена медицинскими цифровыми изображениями становится все более и более актуальной. Электронный обмен позволяет обеспечить дистанционный доступ нескольких специалистов к одному и тому же медицинскому изображению, что бывает важно, например, при подготовке к консилиуму. Электронная передача медицинских изображений из одного лечебного учреждения в другое, например, для проведения консультации, может выполняться за минуты или в крайнем случае за десятки минут, в то время как пересылка снимков обычными способами нередко занимает несколько дней. Наличие изображения в электронной форме позволяет выполнять достаточно сложную компьютерную обработку, к примеру, наложение изображений компьютерного томографа и магнитно-резонансного томографа и т.д., значительно улучшающую возможность постановки диагноза и принятия решений о лечении пациента. В целом электронная передача медицинских изображений лучевой диагностики способна существенно ускорить процесс диагностики заболеваний и лечения пациентов и повысить обоснованность принимаемых медицинских решений. Однако реализация электронной передачи медицинских изображений наталкивается на многие трудности. Одно из основных препятствий связано с тем, что как устройства медицинской визуализации, так и их эксплуатация стоят очень дорого. Поэтому даже в крупнейших медицинских центрах они приобретаются постепенно, зачастую у разных поставщиков, предлагаю-

щих несовместимые аппаратные и программные средства. Не спасает положения и закупка этих устройств у одного поставщика, поскольку с течением времени менялись модели встраиваемых в них компьютеров, версии системного и прикладного программного обеспечения, форматы хранения и передачи данных. Все это существенно затрудняет задачу интеграции устройств медицинской визуализации в единой сети лечебного учреждения.

В настоящее время существуют разные способы решения задачи интеграции устройств медицинской визуализации. Производители могут предлагать собственные, уникальные решения. Например, знаменитая фирма Sieraens, поставляющая широкий спектр медицинского диагностических устройств, в дополнение к ним предлагает систему SIENET, обеспечивающую передачу различного вида изображений на специальным образом оснащенные (и достаточно дорогие) рабочие станции, которые могут быть расположены в любом месте лечебного учреждения. Для этой передачи используются собственные протоколы обмена, поэтому система SIENET допускает прием и обработку изображений, полученных в основном на диагностических устройствах производства именно фирмы Sieraens, скажем, цифровых радиографах SIREGRAPH, FLUOROSPOT, компьютерных томографах SOMATOM, магнитно-резонансных томографах MAGNETOM. Однако в нескольких американских больницах, например в Методистском госпитале Индианы, система SIENET используется в сочетании с томографами фирмы General Electric. Чтобы обеспечить большую привлекательность и сделать систему более открытой, ее разработчики включили в нее возможность передачи циркулирующих в ней изображений во внешние информационные системы и сети по стандарту DICOM. Однако эта возможность является скорее довеском к системе, а не ее основой. По информации фирмы Siemens, система SIENET в различной комплектации установлена в 180 учреждениях по всему миру; общее число ее рабочих станций превысило 600 единиц.

Другой подход предлагает российская фирма ТАНА. Она использует достаточно универсальную систему перехвата изображений непосредственно на входе видеомониторов компьютеров диа-

гностических устройств. Затем эти изображения могут быть обработаны с помощью предлагаемого этой фирмой специализированного (по методам исследований) программного обеспечения и выведены на пленку или бумагу либо переданы по компьютерной сети на любую соответствующим образом оснащенную сетевую рабочую станцию. Этот подход дешевле описанного выше и позволяет относительно легко подключаться к диагностическим устройствам разных фирм-производителей, что очень важно для большинства медицинских центров. Однако при этом не обеспечиваются автоматическое получение и дальнейшая передача информации об условиях получения исходных изображений. В своих технических решениях фирма TANA ориентируется в основном на компьютеры фирмы Silicon Graphics, но предлагает также решения для компьютеров Pentium. В настоящее время насчитывается уже несколько десятков установок разработанного ею программного обеспечения.

Третий подход предлагает, например, германская фирма Optiware, разработавшая программный пакет Hipax, в котором основная ставка делается на получение и передачу медицинских изображений в стандарте DICOM. Ссылки на DICOM делают и другие поставщики аналогичных пакетов, например фирма Line Imaging, предлагающая пакет WinRad. Для этих пакетов характерны достаточно скромные требования к используемым компьютерам, скажем Pentium с шиной PCI, оперативной памятью 24 Мбайт, видеоадаптером и монитором, обеспечивающими режим SVGA с разрешением 1024x768 и палитрой TrueColor. Похожий подход предлагается также российской фирмой «Федеральное медицинское агентство».

Стандарт DICOM 3 распространяется на передачу растровых медицинских изображений, получаемых с помощью различных методов лучевой диагностики (рентгенография, ультразвуковая диагностика, эндоскопия, компьютерная и магнитно-резонансная томография и др., всего в нем перечислены 29 методов). Он получил широкое признание не только в США, но и во многих других странах, в том числе европейских. Стандарт DICOM был взят за основу разработки европейского стандарта MEDICOM, работа над

которым велась рабочей группой WG4 технического комитета TC 251 Европейского института стандартизации CEN.

Стандарт состоит из 13 частей, из которых в текущей версии (DICOM 3.0) представлены первые [12].

Часть 1. Введение

Описываются история разработки стандарта, его назначение и структура.

Часть 2. Соответствие стандарту

Указываются структура сертификата соответствия стандарту и критерии, которым должен удовлетворять производитель диагностического оборудования, чтобы иметь право объявить его совместимым со стандартом DICOM.

Часть 3. Определение информационных объектов. Специфицируются используемые в стандарте информационные объекты. Предлагается информационная модель «реального мира», описывающая взаимоотношения между нормализованными объектами (пациент, устройство) и составными (исследования, изображения и др., наследующими некоторые атрибуты нормализованных объектов).

Часть 4. Спецификации классов операций.

Специфицируются классы действий или операций, которые могут выполняться над информационными объектами. Вводится понятие операция-объект SOP (service-object pair). Разработчики стандарта исходили из того, что применение операции к объекту может быть ограничено его свойствами, поэтому есть нужда в отдельном описании классов SOP. Например, для класса операций Хранение выделяются отдельные стандартные классы операций-объектов Хранение изображений цифровой радиографии, Хранение ультразвуковых изображений, Хранение наложений, Хранение таблицы преобразования пикселов и т.д. Аналогичные пары выделяются для операции Запрос/извлечение и пр.

Часть 5. Структура и семантика данных.

Описываются типы данных и правила кодирования, используемые при передаче данных из одной системы в другую. Специфицируются форматы передачи изображений. Стандарт допускает передачу исходных и уплотненных изображений; особо описывается синтаксис передачи при использовании неискажающих и ис-

кажающих алгоритмов уплотнения JPEG. Допускаются другие, не специфицируемые в стандарте алгоритмы уплотнения.

Часть 6. Словарь данных.

Приводится полный список элементов данных, описанных в стандарте DICOM. Каждый элемент данных идентифицируется парой целых чисел, например пара (0018,5100) идентифицирует описание положения пациента по отношению к устройству в момент проведения исследования. Кроме идентификатора, приводятся имя элемента, характеристика его значения (строка символов, число и т.д.) и допустимое число повторений элемента в сообщении.

Часть 7. Обмен сообщениями.

Описывается структура команд и протокола обмена сообщениями в стандарте DICOM.

Часть 8. Обеспечение обмена сообщениями в сетевых средах.

Определяются все необходимые компоненты системы обмена сообщениями в стандарте DICOM в сетевых средах, использующих протокол TCP/IP. Изложение этой части существенно опирается на соответствующие стандарты Модели взаимодействия открытых систем OSI (ISO 8222 и ISO 8649).

Часть 9. Обеспечение обмена сообщениями при прямой связи абонентов (point-to-point).

Приводится подробное описание прямого взаимодействия двух устройств, включая назначение каждой ножки 50-контактного разъема, уровня передаваемых сигналов, их временные характеристики и т.д. Оно напоминает описание параллельного дуплексного интерфейса миникомпьютера, и, похоже, в основном переписано из других стандартов, ранее разработанных ассоциацией NEMA.

На стадии разработки и утверждения находятся еще 4 части стандарта:

Часть 10. Носители данных и форматы файлов.

Описываются теоретические основы хранения медицинских изображений на различных внешних носителях данных.

Часть 11. Прикладные характеристики хранения данных на внешних носителях.

Описываются требования к данным, которые должны храниться на внешних носителях. Описания имеют клиническую направлен-

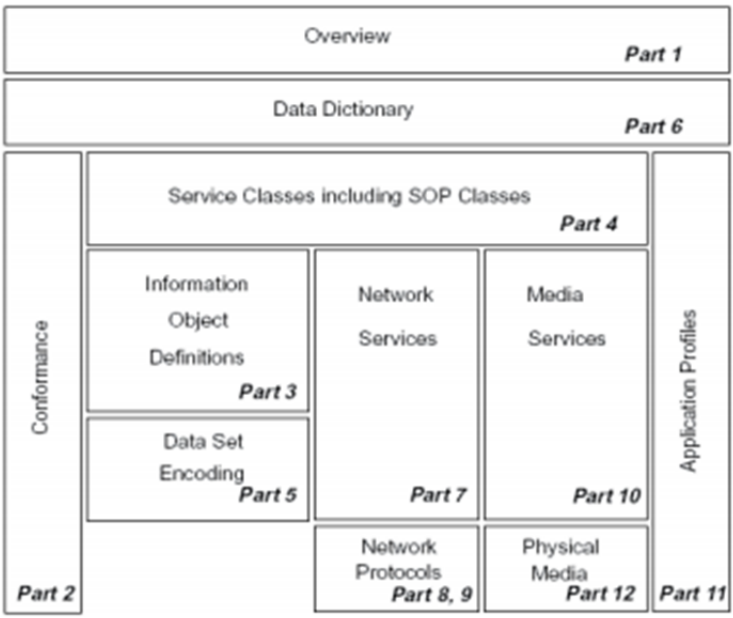
ность, например, задают, какие данные должны храниться на внешних носителях при проведении ангиографии.

Часть 12. Форматы носителей и физическая среда хранения данных.

Специфицируются различные носители данных, которые могут использоваться для хранения медицинских изображений, например, дискеты 3.5", компакт-диски CD ROM, магнитооптические диски и т.д.

Часть 13. Управление выводом на печатающие устройства при прямом соединении

Описываются протоколы и операции, необходимые для вывода изображения на печатающее устройство. Вывод осуществляется системой-исполнителем, имеющей прямое соединение с системой-инициатором вывода.



Достаточно одного взгляда на представленное выше оглавление, чтобы заметить определенную формализованность изложения стандарта DICOM. И действительно, структура этого документа приведена в соответствие с директивами организации ISO [ISO/IEC Directives, 1989 Part3: Drafting and Presentation of International Standards], что существенно отличает DICOM от стандарта электронной передачи текстовых медицинских документов HL7. Надо сказать, что это не способствует лучшему восприятию материала. Например, здесь явно не хватает примеров реальных сообщений. Тем не менее, чтение стандарта DICOM окажется достаточно поучительном для многих специалистов по медицинской информатике, в том числе даже тех из них, кто не занимается обработкой медицинских изображений [12].

DICOM File (DICOM Файл) представляет собой объектно-ориентированный файл с теговой организацией. Информационная модель Стандарта DICOM для DICOM файла 4-ступенчатая:

Пациент (Patient) ==> Исследование (Study) ==> Серия (Series) ==> Изображение (кадр или серия кадров) (Image).

Проиллюстрируем стиль стандарта на простом примере. Термин «медицинское изображение» нередко вводит нас в заблуждение из-за устойчивости восприятия понятия «изображение». Для определения термина «сложное изображение» в стандарте используется BR-диахрамма, показанная на рис. 8.1 [12].

Из нее видно, что сложное изображение представляет собой элемент серии изображений, полученной при исследовании одного и того же пациента. Кроме элементарного изображения, компонентами сложного изображения могут быть несколько кривых (например, задающих область интереса на обзорном снимке), наложений (растровых текстов с параметрами исследования, фамилией врача-диагноста и т.д.), несколько таблиц (или формул) преобразования пикселов изображения при его визуализации, таблица или формула преобразования пикселов изображения из внутренней формы хранения, используемой производителем данного оборудования, в универсальную внешнюю форму. Допускаются отдельные кривые или их наложение, не связанное ни с каким двумерным изображением, что теоретически позволяет обеспечить

передачу не только изображений лучевой диагностики, но и, допустим, кардиограмм. Однако такой подход не является ни эффективным, ни целесообразным.

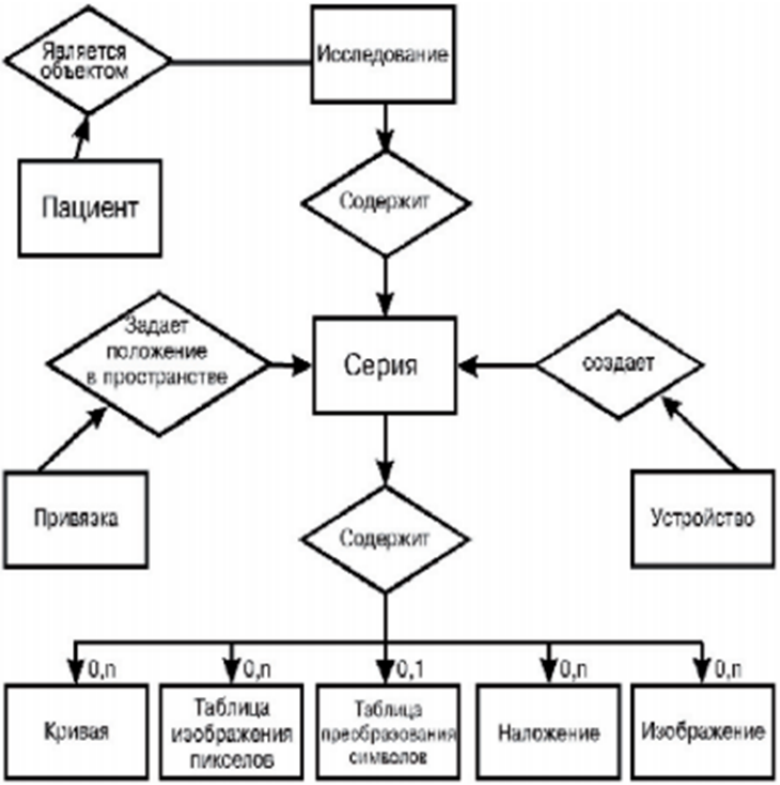


Рис. 8.1. ER-диаграмма файла стандарта DTCOM 3.0

В разрабатываемой АИС ТМК, согласно ТЗ, должна быть предусмотрена возможность измерения линейных размеров на изображениях формата DICOM. Для реализации этой задачи было сделано следующее:[5].

1. При заполнении истории болезни пациента (проведение телемедицинской консультации) специалист имеет возможность

прикреплять к ней медицинские изображения формата DICOM, которые будут храниться в БД системы. При проведении консультации врач-консультант, таким образом, имеет возможность доступа к приложенным изображениям (имеющих формат DICOM) исследуемых органов человека, полученных в результате томографического обследования, УЗИ и др. процедур.

2. При просмотре изображения специалист может измерить линейные размеры заинтересовавших его участков изображения. Для этого используется бесплатный программный продукт «K-PACS V1.6.0 DICOM Viewing» [http://www.k-pacs.net/], при помощи которого специалист может извлечь и обработать интересующее изображение.

В настоящее время идет активное взаимодействие с разработчиками данного продукта с целью встраивания ряда его функций в состав АИС ТМК. В ходе доработки системы функция измерения линейных размеров изображений в формате Dicom будет полностью интегрирована в АИС ТМК и будет представлять собой отдельный модель системы. Это позволит добиться полной целостности телемедицинской системы и исключить её зависимость от сторонних продуктов.

На рис. 8.2 представлен вид окна, в котором происходит измерение линейных размеров области снимка, которая заинтересовала специалиста. Снимок взят из примеров, предлагаемых на сайте разработчиков программного пакета «K-PACS V1.6.0 DICOM Viewing».

Модуль обмена видеоинформацией выполняет следующие функции:

– подготовку интерфейса захвата и редактирования видеоизображения;

– захват и анализ видеоизображения;

– запись видеоизображения в формат хранения данных в БД;

– трансляция видеоизображения во время проведения телемедицинских консультаций;

– воспроизведение видеоизображения при телеконсультациях или по требованию клиента системы.

Подготовка интерфейса захвата и редактирования видеоизображения заключается в создании окна-формы захвата и редактирования, формирования списка устройств видеозахвата, настройка вспомогательных средств для подключения к устройству захвата. В окне захвата создается меню устройств захвата, из которых пользователь выбирает источник изображения. С помощью интерфейса DirectX DirectDraw на форму выводится изображение с устройства захвата, а также инструменты корректировки изображения - регулировка яркости, контрастности, резкости, масштаба, разрешения. Схема алгоритма представлена на рис. 8.3.

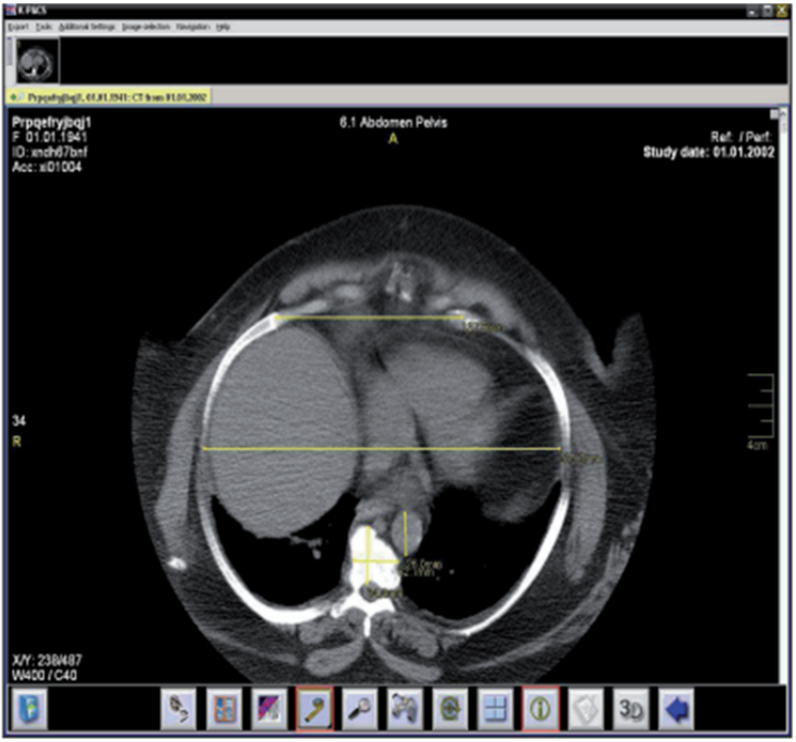


Рис. 8.2. Измерение линейных размеров изображения формата DICOM

Рис. 8.3. Блок-схема алгоритма формирования окна видеозахвата

В зависимости от цели, пользователь АИС ТМК может записывать видеоизображение для сохранения в БД либо для трансляции по сети. В любом случае, изображение считывается с устройства захвата по кадрам, каждый кадр анализируется и сегментируется, затем сохраняется в поток памяти для дальнейшей работы. Процесс анализа и сегментации изображения происходит следующим образом.

Поступивший (очередной) кадр анализируется сначала целиком. Сначала изображение преобразуется в битовую матрицу (мас-

сив) RGB (красный-зеленый-синий) каналов, 8 бит на канал, а также преобразуется в градиентное распределение (градации серого – распределение по яркости).

Затем, по каждому каналу и по яркостному распределению вычисляется среднее значение и дисперсия.

Значения сравниваются с соответствующими характеристиками эталонного кадра (эталонный кадр модифицируется по мере работы, самым первым эталоном выбирается поступивший первый кадр, он считается базовым и записывается без анализа, лишь подвергается сжатию).

Если характеристики разнятся значительно (10–15%), значит изображение изменилось. В этом случае эталонный кадр заменяется текущим, текущий сохраняется в поток памяти и алгоритм анализа завершается.

Если характеристики разнятся незначительно (менее 10%), изображение сегментируется. Затем каждый сегмент подвергается такому же анализу. Сегменты, отличающиеся от эталонных сегментов более 10% добавляются в поток, остальные игнорируются.

Все изображения и их участки, помещаемые в поток, маркируются соответствующим образом, и подвергаются сжатию алгоритмом JPEG.

Таким образом, составляется видеофильм не целиком из кадров, а из изменившихся участков изображения, что значительно экономит объем памяти, и не нагружает ресурсы компьютера конвертацией. Схематически, механизм анализа можно представить на рис. 8.4.

При проведении телемедицинских консультаций, видеоизображение, считываемое с источника захвата, сначала передается серверу, затем сервер ретранслирует его на другие клиентские машины, где оно воспроизводится. Опишем более подробно эти процессы.

После того, как пользователь присоединяется к проводящейся телемедицинской консультации, открывается форма видеозахвата и форма отображения видеоинформации. Если пользователь выберет устройство видеозахвата из списка, начнется трансляция.

Рис. 8.4. Механизм анализа изображения

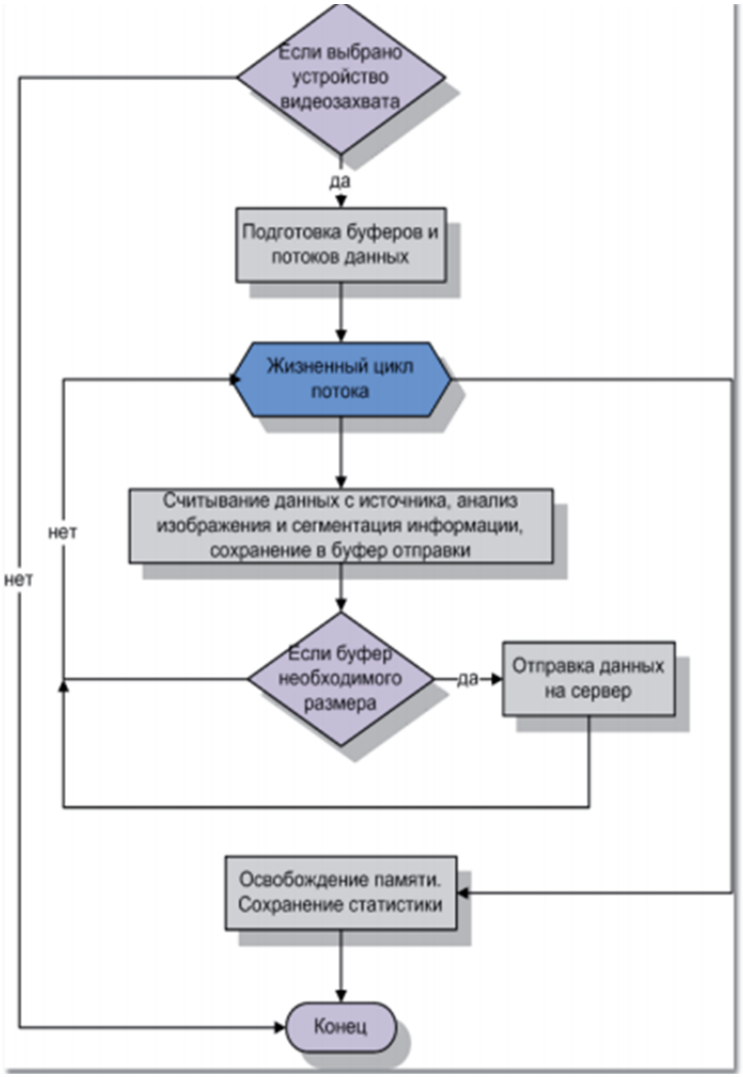


Рис. 8.5. Блок-схема алгоритма трансляции видеоизображения

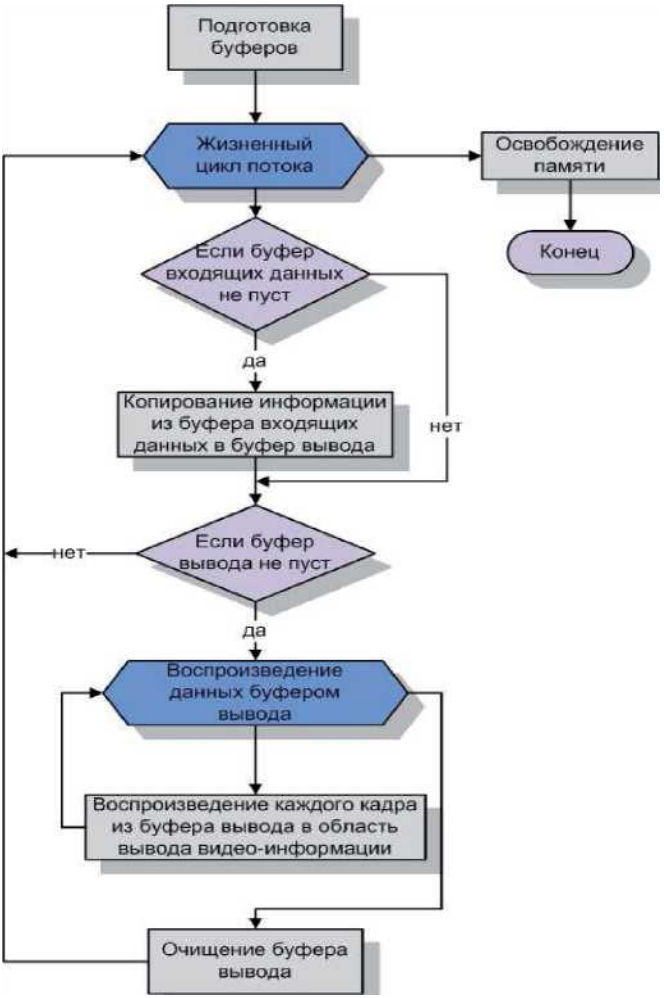


Рис. 8.6. Процесс воспроизведения

Для трансляции выделяется отдельный программный поток, который, с определенной периодичностью, считывает кадр изображения с устройства видеозахвата. После этого полученное изображение анализируется и сегментируется, и помещается в поток

памяти – буфер отправки. Как только буфер занимает определенный объем информации, он отправляется на сервер. Схематически алгоритм представлен на рис. 8.5.

Принятые видеоданные на сервере помещаются в буфер для анализа и ретрансляции. Сервер выясняет, какие из клиентов ожидают видеоизображения от данного пользователя, а затем передает им видеоданные, попутно анализируя необходимость сохранения данных в истории телеконсультации.

На клиентских машинах, параллельно потоку передачи видеоданных запускается поток приложения воспроизведения видео. Поступившие от сервера данные на воспроизведение помещаются в буфер входящих данных. Поток воспроизведения проверяет буфер и если данные поступили, копирует их в буфер вывода. В буфере вывода таким образом создается очередь видеоданных, требующих отображения. Поток при каждой итерации выводит кадр или определенную последовательность сегментов кадров в форму вывода изображений. Схематически процесс изображен на рис. 8.6.

8.2. Дистанционная идентификация личности человека

Биометрические технологии основаны на биометрии, измерении уникальных характеристик отдельно взятого человека. Это могут быть как уникальные признаки, полученные им с рождения, например: ДНК, отпечатки пальцев, радужная оболочка глаза; так и характеристики, приобретённые со временем или же способные меняться с возрастом или внешним воздействием, например: почерк, голос или походка.

Принцип работы. Все биометрические системы работают практически по одинаковой схеме. Во-первых, система запоминает образец биометрической характеристики (это и называется процессом записи). Во время записи некоторые биометрические системы могут попросить сделать несколько образцов для того, чтобы составить наиболее точное изображение биометрической характеристики. Затем полученная информация обрабатывается и преобразовывается в математический код. Кроме того, система может по-

просить произвести ещё некоторые действия для того, чтобы «приписать» биометрический образец к определённому человеку. Например, персональный идентификационный номер (PIN) прикрепляется к определённому образцу, либо смарт-карта, содержащая образец, вставляется в считывающее устройство. В таком случае, снова делается образец биометрической характеристики и сравнивается с представленным образцом (рис. 8.7).

Рис. 8.7. Способы идентификации

Таблица 8.1

Классификация систем идентификации личности

Системы идентификации	
врожденные	приобретенные
по отпечаткам пальцев (дактилоскопия)	по характеристикам речи
по радужной оболочке глаза	по изображению лица
по геометрии ладони руки	по подписи

Биометрия – технология идентификации личности, использующая физиологические параметры субъекта (отпечатки пальцев, радужная оболочка глаза и т.д.). Основные сведения Биометрические параметры – статические отпечатки пальцев, геометрия руки, сетчатка глаза и т.п. – динамические параметры: динамика воспроизведения подписи или рукописного ключевого слова, голос и т.п.

По отпечаткам пальцев. Этот метод является самым старым и распространенным среди способов идентификации, использующих физиологические особенности организма. Применяется во всех странах органами охраны правопорядка для идентификации преступников.

Рис. 8.8. Системы идентификации по отпечаткам пальцев

Дактилоскопия (распознавание отпечатков пальцев) – наиболее разработанный на сегодняшний день биометрический метод идентификации личности. Катализатором развития метода послужило его широкое использование в криминалистике XX века.

Каждый человек имеет уникальный папиллярный узор отпечатков пальцев, благодаря чему и возможна идентификация. Обычно алгоритмы используют характерные точки на отпечатках пальцев: окончание линии узора, разветвлении линии, одиночные точки. Дополнительно привлекается информация о морфологической структуре отпечатка пальца: относительное положение замкнутых линий папиллярного узора, «арочных» и спиральных линий. Особенности папиллярного узора преобразовываются в уникальный код, который сохраняет информативность изображения отпечатка. И именно «коды отпечатков пальцев» хранятся в базе данных, используемой для поиска и сравнения. Время перевода изображения отпечатка пальца в код и его идентификация обычно не превышает 1 с, в зависимости от размера базы. Время, затраченное на поднесение руки – не учитывается.

Системы идентификации по характеристикам речи. Это одна из старейших биометрических технологий. В настоящее время ее развитие активизировалось, ей прочится большое будущее и широкое использование в построении «интеллектуальных зданий». Существует достаточно много способов построения кода идентификации по голосу, как правило, это различные сочетания частотных и статистических характеристик голоса.

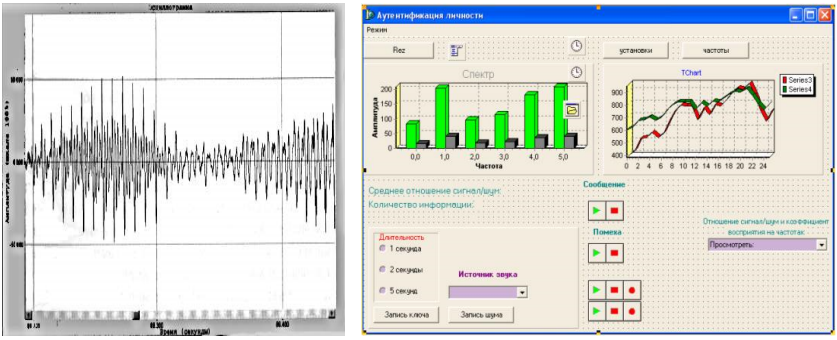


Рис. 8.9. Системы идентификации по характеристикам речи

Системы идентификации по радужной оболочке глаза. Это более простой и удобный способ, но и менее надежный. Радужная

оболочка глаза является уникальной характеристикой человека. Рисунок радужки формируется на восьмом месяце внутриутробного развития, окончательно стабилизируется в возрасте около двух лет и практически не изменяется в течение жизни, кроме как в результате сильных травм или резких патологий. Метод является одним из наиболее точных среди биометрических методов.

Система идентификации личности по радужной оболочке логически делится на две части: устройство захвата изображения, его первичной обработки и передачи вычислителю и вычислитель, производящий сравнение изображения с изображениями в базе данных, передающий команду о допуске исполнительному устройству.

Время первичной обработки изображения в современных системах примерно 300–500 мс, скорость сравнения полученного изображения с базой имеет уровень 50 000–150 000 сравнений в секунду на обычном ПК. Такая скорость сравнения не накладывает ограничений на применения метода в больших организациях при использовании в системах доступа.

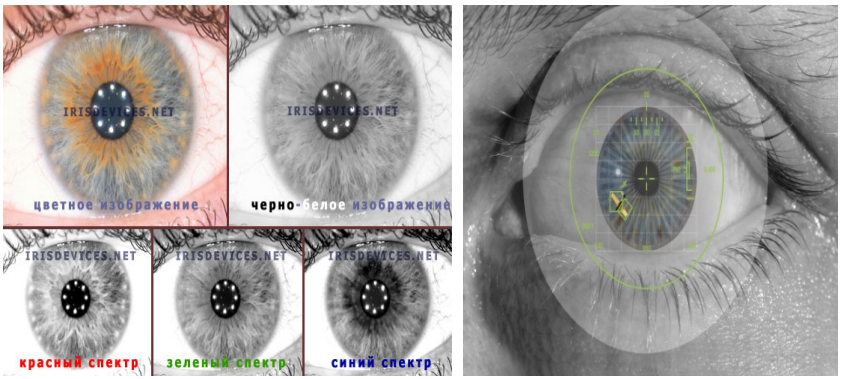


Рис. 8.10. Система идентификации по радужной оболочке глаза

Системы идентификации по изображению лица. В данном методе идентификации строится трехмерный образ лица человека. На лице выделяются контуры бровей, глаз, носа, губ и т.д., вычис-

ляется расстояние между ними и строится не просто образ, а еще множество его вариантов на случаи поворота лица, наклона, изменения выражения. Количество образов варьируется в зависимости от целей использования данного способа (для аутентификации, верификации, удаленного поиска на больших территориях и т.д.).

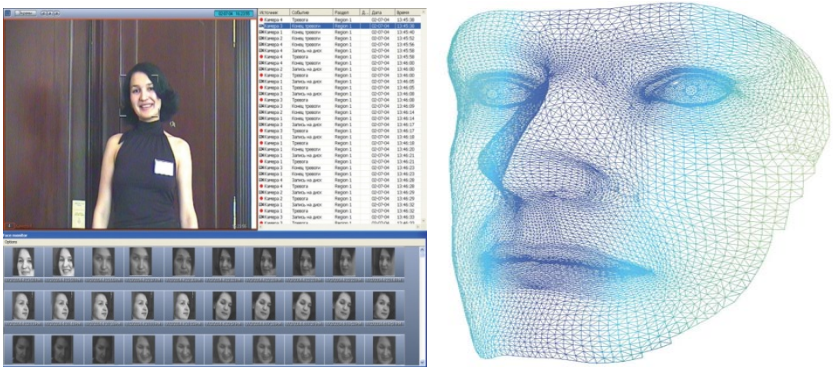


Рис. 8.11. Системы идентификации по изображению лица

Системы идентификации по геометрии ладони руки. Метод идентификации пользователей по геометрии руки по своей технологической структуре и уровню надежности вполне сопоставим с методом идентификации личности по отпечатку пальца.

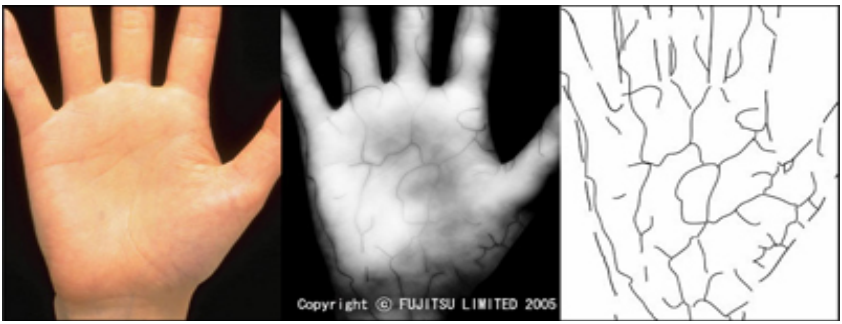


Рис. 8.12. Система идентификации по геометрии ладони руки

Системы идентификации по подписи. Как правило, для этого вида идентификации человека используется его роспись (иногда написание кодового слова). Цифровой код идентификации формируется, в зависимости от необходимой степени защиты и наличия оборудования (графический планшет, экран карманного компьютера Palm и т.д.), двух типов:

– По самой росписи (для идентификации используется просто степень совпадения двух картинок);

– По росписи и динамическим характеристикам написания (для идентификации строится свертка, в которую входит информация по росписи, временным характеристикам нанесения росписи и статистическим характеристикам динамики нажима на поверхность).

Рис. 8.13. Системы идентификации по подписи

Применение. Биометрические технологии активно применяются во многих областях связанных с обеспечением безопасности доступа к информации и материальным объектам, а также в задачах уникальной идентификации личности, в целях:

– контроля доступа;

– защиты информации;

– идентификации клиентов.

На рис. 8.14 представлена примерная статистика по практическому использованию биометрических систем идентификации личности в Российской Федерации в настоящее время.

Таблица 8.2

Сравнительная таблица стоимости систем идентификации

Системы идентификации	Стоимость от, руб.
по отпечаткам пальцев (дактилоскопия)	*676,62*
по радужной оболочке глаза	16 599
по геометрии ладони руки	45 430
по характеристикам речи	786,02
по изображению лица	24 800

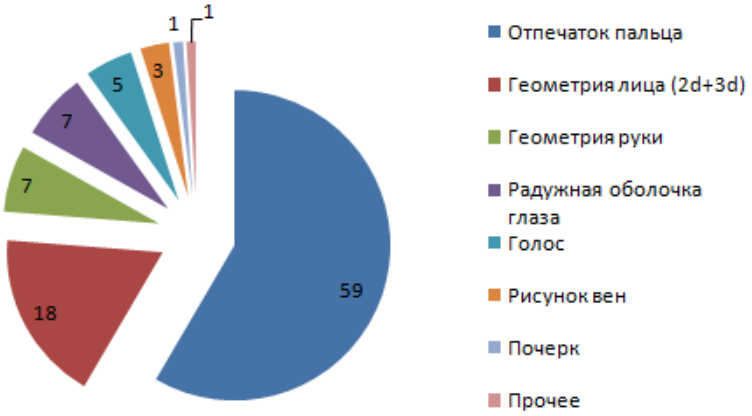


Рис. 8.14. Статистика практического использованию биометрических систем идентификации личности

На рис. 8.15 приведена обобщенная блок-схема универсальной распределенной интеллектуальной системы идентификации личности человека (УРИСИЛЧ), которая содержит датчики информации (геометрии ладони руки, тактильный, видеодатчик (отпечатки пальцев), подписи человека, лазерный инфракрасный, звуковой, запаха и 3D цветная видеокамера общего вида, 3D цветная видеокамера радужной оболочки глаза), интерфейс, блок процессов, пульт управления, блок памяти, монитор и исполнительный блок. Датчики информации воспринимают полную информацию о личности человека(изображение лица, радужной оболочки глаз, голоса, отпечатки пальцев) и передают ее через интерфейс в блок процессов.

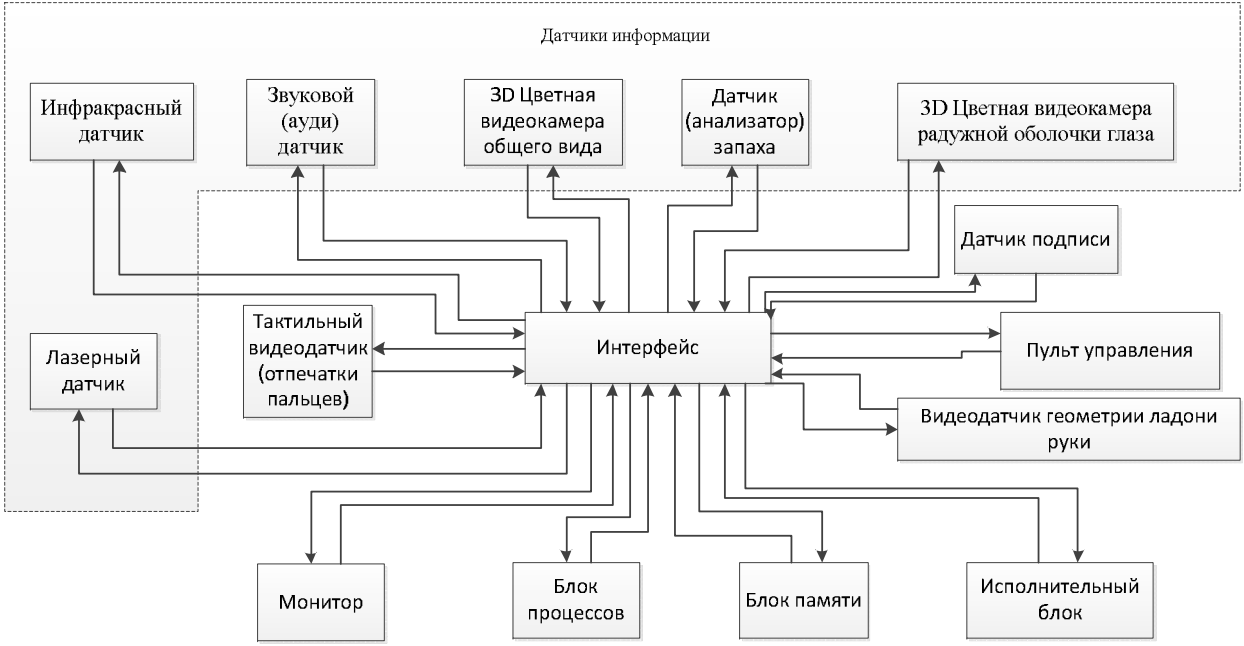


Рис. 8.15. Обобщенная блок-схема универсальной распределенной интеллектуальной системы идентификации личности человека

Кроме этого, воспринимается информация о походке, одежде и запахе человека, по которой можно идентифицировать человека. В блоке памяти записываются эталонные изображения(ЭИ) людей (изображение лиц, отпечатки пальцев и др.) и алгоритмы распознавания (корреляционный, структурно-перестраиваемый, вейвлет-анализа, нейро-нечеткий). В блоке процессоров сравниваются ЭИ с текущей информацией(изображениями ТИ) и принимается решение об идентификации личности человека. Информация о принятом решении, а также ТИ и ЭИ отображаются на экране монитора. С помощью исполнительного блока обеспечивается необходимые исполнение(открывает или закрывает дверь, шлагбаум для пропуска человека и др.) Оператор управляет работой УРИ-СИЛЧ (в автоматизированном режиме).

Примеры биометрической защиты информации:

Сканирующий модуль легко устанавливается и органично вписывается в интерьер салона автомобиля

ЛИТЕРАТУРА

1. Шидловский С.В. Автоматическое управление. Перестраиваемые структуры в системах с распределенными параметрами. Томск: ТГУ, 2007. 192 с.
2. Сырямкин В.И., Титов В.С., Якушенков Ю.Г. Системы технического зрения: справочник. Томск: МГП «РАСКО», 1992. 367 с.
3. Козлов В.Н. Элементы математической теории зрительноговосприятия. М.: МГУ, 2001. 122 с.
4. Герман Кругль. Профессиональное видеонаблюдение. Практика и технологии аналогового и цифрового CCTV. М.: Секьюрити Фокус, 2010. 672 с.
5. Титов В.С., Панищев В.С., Яковлева В.С. Адаптивные видеодатчики на базе КМОП приемников излучения с активными пикселями. Курск: Изд-во Курск. гос. техн. ун-та, 2008. 100 с.
6. Апенко М.И., Дубовик А.С. Прикладная оптика. М.: Наука, Гл. ред. физ.-мат. лит-ры, 1971. 392 с.
7. Бугаенко Е.И., Труфанов М.И. Способ автоматического определения и коррекции радиальной дисторсии на цифровых изображениях // Изв. вузов. Приборостроение. 2008. № 2. С. 16–22.
8. Стрелкова А.Н., Степченко А.А. Устройство повышения качества изображения при эндоскопии // Изв. вузов. Приборостроение. 2008. № 2. С. 54–58.
9. Лазаренко В.А., Мишустин В.Н., Титов В.С., Труфанов М.И. Прогнозирование тромбоэмболических осложнений при травме. Курск: КГМУ, 2004. 148 с.
10. Гридин В.Н., Титов В.С., Труфанов М.И. Принципы функционирования бинокулярного оптико-электронного устройства для диагностики отклонений глазодвигательного аппарата // Изв. вузов. Приборостроение. 2008. № 2. С. 48–53.
11. Гридин В.Н., Мишустин В.Н., Тарасова О.Б., Титов В.С., Труфанов М.И. Автоматизированный аппаратно-программный комплекс диагноситки тромбоэмболических осложнений // Информационные технологии в проектировании и производстве. 2008. № 2. С. 79–82.
12. Прэтт У. Цифровая обработка изображений / пер. с англ. М.: Мир, 1982. Кн. 1. 312 с.; Кн. 2. 493 с.
12. Дуда Р., Харт П. Распознавание образов и сцен / пер. с англ. Г.Г. Вайнштейна и А.М. Васьковского; под ред. В.Л. Стефанюка. М.: Мир, 1976. 511 с.

14. Методы компьютерной обработки изображений / под ред. В.А. Сойфера. М.: Физматлит, 2003. 784 с.
15. Гридин В.Н., Михайлов В.Б., Слюсаренко А.С. Управление вычислительными процессами в автоматизированных системах // Информационные технологии в проектировании и производстве. ВИМИ, 1997. Вып. 2. С. 32–36.
16. Гридин В.Н., Гребнев А.К., Дмитриев В.П. Оптоэлектронные элементы и устройства. М.: Радио и связь, 1998. 366с.
17. Приборы с зарядовой связью / под ред. М. Хоувза и Д. Моргана; пер. с англ. М.: Атомиздат, 1981. 376 с.
18. Гридин В.Н., Рыжиков И.В. Физические основы микроэлектроники. М., 2002. 181 с.
19. Стемпковский А., Шилин В. КМОП-фотодиодные СБИС – перспективная элементная база однокристальных систем приема и обработки информации // Электроника: наука, технология, бизнес. 2003. № 2. С. 14–20.
20. Егорова С.Д., Колесник В.А. Оптико-электронное цифровое преобразование изображений. М.: Радио и связь, 1991. 208 с.
21. Путятин Е.П., Аверин С.И. Обработка изображений в робототехнике. М.: Машиностроение, 1990. 320 с.
22. De Haan G. Progress in motion estimation for video format conversion // IEEETrans.onConsumerElectron. Aug 2000. Vol. 46, № 3. P. 449–450.
23. Кубасов Д., Ватолин Д. Обзор методов компенсации движения // Графика и мультимедиа. 2005. № 2. URL: http://cgm.graphicon.ru/content/view/76/28/.
24. Заблоцкий В.Р. Повышение контрастности космических сканерных изображений анроландшафта цифровыми методами // Исследование Земли из космоса. 1997. № 6. С. 30–34.
25. Фурман Я.А. Основы теории обработки контуров изображений: учеб. пособие. Йошкар-Ола: Изд-во МарГТУ, 1997. 255 с.
26. Фишер Р.Б. От поверхностей к объектам. Машинное зрение и анализ трехмерных сцен / пер. с англ. под ред. Д.А. Денисова. М.: Радио и связь, 1993. 287 с.
27. Быстрые алгоритмы в цифровой обработке изображений / Т.С. Хуанг, Дж. О. Эклунд. Г. Дж. Нусбаумер и др.; под ред. Т.С. Хуанга; пер. с англ. под ред. Л.П. Ярославского. М.: Радиоисвязь, 1984. 221 с.
28. Stapor K.Avectorizedthinningalgorithmforhandwrittinsymbolsrecognition // MachineGraphics&Vision. 1999. Vol. 8, № 3. P. 341–352.

29. Ratai A.Finding outlines of objects in raster images // Machine Graphics&Vision. 1999. Vol. 8, № 3. P. 295–312.
30. Fung G., Yung N., Pang G. Camera calibration from road lane markings // Opt. Eng. 2003. Vol. 42, № 10. P. 2967–2977.
31. Гридин В.Н., Курносов А.И. Разработка архитектуры спецвычислителя, реализующего заданный алгоритм // Интеллектуальные интегрированные САПР РЭА и БИС. М.: Наука, 1990. С. 60–66.
32. Ширабакина Т.А., Труфанов М.И. Нечеткая математическая модель автофокусирующейся системы обработки изображений // Изв. вузов. Приборостроение. 2003. Т. 46, № 11. С. 12–16.
33. Пат. №2337501РФ, МКИ H04N 1/409, G06K9/46, G06T9/20. Способ компенсации размытости изображения движущихся объектов и устройство для его реализации / С.В. Дегтярев, В.С. Титов, М.И. Труфанов, В.С. Яковлева. № 2006146536/09. Заявл. 27.12.2006; опубл. 27.10.08. 40 с.
34. Гридин В.Н. Вербицкий А.А., Солодовников В.И., Солодовников И.В. Использование нейронных сетей в задаче тестирования // Информационные технологии. Машиностроение. 2007. № 9. С. 21–26.
35. Пат. № 2199150 РФ, МКИ G06K9/32. Устройство калибровки оптико-электронной системы / С.В. Дегтярев, В.С. Титов, М.И. Труфанов. № 2001103097/09. Заявлено 02.02.2001; опубл. 20.02.2003. Бюл. № 5. 6 с.
36. Титов В.С., Труфанов М.И. Метод автоматической калибровки бинокулярной оптико-электронной системы // Датчики и системы. 2003. № 8. С. 10–12.
37. Якушенков Ю.Г. Теория и расчет оптико-электронных приборов: учеб. для студентов вузов. М.: Логос, 1999. 480 с.
38. Афанасьев В.Н., Колмановский В.Б., Носов В.Р. Математическая теория конструирования систем управления. М.: Высш. шк., 1989. 447 с.
39. Семушин И.В. Адаптивные схемы идентификации и контроля при обработке случайных сигналов. Саратов: Изд-во Саратов. ун-та, 1985. 180 с.
40. Titov V.S., Yakovleva V.S., Degtyarev S.V. Structure of a tunable adaptive videodetector // Pattern Recognition and Image Analysis. 2004. Vol. 14, № 3. P. 463–465.
41. Гридин В.Н. Автоматизация схемотехнического проектирования аналоговых микросхем: современное состояние и перспективы развития // Информационные технологии в проектировании и производстве (ФГУП ВИМИ). 2007. № 1. С. 80–84.

42. Titov V.S., Tevs S.S., Shirabakina T.A., Cinha M.K. Photon systems for speed mode inspection of transport means // SPIE Proc. 1999. Vol. 3901. P. 65–68.
43. Патрик Э. Основы теории распознавания образов / пер. с англ.; под ред. Б.Р. Левина. М.: Сов. радио, 1980. 408 с.
44. Гридин В.Н., Сперанский С.Н. Система реляционно-логического программирования «концепт» // Техника, информатика, экономика. Сер. Автоматизацияпроектирования. 1989. Вып. 1. С. 14–19.
45. Titov V.S. Tevs S.S. A fuzzy mathematical model for recognition in optoelectronic systems.Pattern recognition and image analysis. 2001. Vol. 11. № 1. P. 105–107.
46. Титов В.С., Тевс С.С. Оптико-электронная система распознавания и нормализации объектов // Датчики и системы. 2000. № 3. С. 30–32.
47. Труфанов М.И., Козлов П.С., Мелик-Гайказян В.И. Оптико-электронная система измерения поверхностного натяжения жидкости и оценки флотационной активности веществ // Изв. вузов. Приборостроение. 2008. № 2. С. 59–64.
48. Мелик-Гайказян В.И., Емельянова Н.П., Козлов П.С., Труфанов М.И. Релаксационные кривые, методика их снятия и значение их для понимания процесса пенной флотации и управления им // Изв. вузов. Цветная металлургия. 2008. № 2. С. 6–15.
49. Гридин В.Н., Тарасова О.Б. Построение интеллектуальных диагностических систем в медицинских приложениях // Информационные технологии. 2003. № 7. С. 54–58.
50. Урмахер Л.С. Справочник по офтальмологической оптике и приборам. М.: Медицина, 1971. 179 с.
51. Гридин В.Н. и др. Принцип действия и математическая модель взаимодействия волоконно-оптического анализатора с оптической системой глаза // Информационные технологии в проектировании и производстве. 2008. (ФГУП ВИМИ). № 2. С.69–74.
52. Гридин В.Н., Дегтярев С.В., Мирошниченко С.Ю. Метод распознавания глаз человека на цифровом изображении для систем биометрического контроля доступа // Информационные технологии и математическое моделирование систем: материалы междунар. науч.-технич. конф. М.: Радиотехника, 2008. С. 162–164.
53. Бондарчук А.С., Боровик В.С., Гуцул В.И., Евтушенко В.Г., Клестов С.А., Кузнецов Д.Н., Марченко В.В., Марченко М.В., Пешкичев Р.Ю., Попков А.И., Пыхтырев В.С., Сахапов Д.А., Судакова Е.С., Сырямкин В.И., Сырямкин М.В., Шатравин В.В., Шумилов В.Н. Ин-

теллектуальные робототехнические и мехатронные системы : учеб. пособие / под ред. В.И. Сырямкина. Томск, 2017. (Сер. Интеллектуальные технические системы).
54. Кузнецов Д.Н., Кузнецова А.С., Сырямкин В.И. Применение системы технического зрения в телемедицине // Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации. Распознавание-2017: сборник материалов XIII Международной научно-технической конференции. Томск, 2017. С. 210–212.
55. Сырямкин В.И., Сырямкин М.В., Гуцул В.И., Кузнецов Д.Н. Группа интеллектуальных многофункциональных транспортных роботов для мониторинга территорий // Материалы 10-й Всероссийской мультиконференции по проблемам управления (МКПУ-2017): в 3 т. / отв. ред. И.А. Каляев. 2017. С. 322–324.
56. Кузнецов Д.Н., Кузнецова А.С., Сырямкин В.И. Современное состояние развития системы телемедицины // ИННОВАТИКА-2016: сборник материалов XII Международной школы-конференции студентов, аспирантов и молодых ученых / под ред. А.Н. Солдатова, С.Л. Минькова. Томск, 2016. С. 340–344.
57. Kuznetsov D. Development of Telemedicine System // MATEC Web of Conferences 7th Scientific Conference with International Participation "Information-Measuring Equipment and Technologies", IME and T 2016. 2016. P. 01074.
58. Matveev I., Siemens E., Yurchenko A., Kuznetsov D. Development and Experimental Investigations of Motion Detection Module for Smart Lighting System // IOP Conference Series: Materials Science and Engineering (см. в книгах). 2016. V. 132. P. 1.
59. Кузнецов Д.Н., Сырямкин В.И. Обзор состояния разработки телемедицины // ИННОВАТИКА-2015: сборник материалов XI Международной школы-конференции студентов, аспирантов и молодых ученых / под ред. А.Н. Солдатова, С.Л. Минькова. Томск, 2015. С. 123–129.
60. Кузнецов Д.Н., Сырямкин В.И. Принципы построения системы телемедицины // ИННОВАТИКА-2015 : сборник материалов XI Международной школы-конференции студентов, аспирантов и молодых ученых / под ред. А.Н. Солдатова, С.Л. Минькова. Томск, 2015. С. 129–135.
61. Кузнецов Д.Н., Сырямкин В.И. Проектирование телемедицинской системы // ИННОВАТИКА-2015: сборник материалов XI Междуна-

родной школы-конференции студентов, аспирантов и молодых ученых / под ред. А.Н. Солдатова, С.Л. Минькова. Томск, 2015. С. 453–458.
62. Kuznetsov D.N., Syryamkin V.I. Robotics in Medicine // AIP Conference Proceedings 5. New Operational Technologies, NEWOT 2015: Proceedings of the 5th International Scientific Conference "New Operational Technologies". 2015. P. 040004.
63. Кузнецов Д.Н., Сырямкин В.И. Обзор телемедицинских технологий // Телекоммуникации. 2015. № 8. С. 44–48.
64. Кузнецов Д.Н., Осипов А.Ю., Сырямкин В.И. Основы телемедицины: учеб. пособие. Томск, 2016.
65. Удалённая идентификация. URL: https://rb.ru/longread/remoteauthentication
66. Язов Ю.К. и др. Нейросетевая защита персональных биометрических данных / Ю.К. Язов (редактор и автор), соавторы В.И. Волчихин, А.И. Иванов, В.А. Фунтиков, И.Г. Назаров. М.: Радиотехника, 2012. 157 с.
67. Ахметов Б.С., Иванов А.И., Фунтиков В.А., Безяев А.В., Малыгина Е.А. Технология использования больших нейронных сетей для преобразования нечетких биометрических данных в код ключа доступа. Алматы: LEM, 2014. 144 с., свободный доступ URL: http://portal.kazntu.kz/files/publicate/2014-06-27-11940.pdf
68. URL: http://elibrary.ru/item.asp?id=24846416
69. Теоретические основы цифровой обработки изображений в встраеваемых системах технического зрения / Ю.С. Бехтин, С.А. Клестов, М.С. Куцов, В.И. Сырямкин, Д.В. Титов; под ред. В.И. Сырямкина, Д.В. Титова. Томск: STT, 2016. 406 с.
70. Интеллектуальные робототехнические и мехатронные системы: учебное пособие / под ред. В.И. Сырямкина. Томск: STT, 2017. 256 с.
70. Основы телемедицины / под ред. В.И. Сырямкина. Lap Lambert Academic Publishing, 2015. 233 с.

ОГЛАВЛЕНИЕ

СПИСОК СОКРАЩЕНИЙ 3

ОПРЕДЕЛЕНИЯ 4

ПРЕДИСЛОВИЕ 5

1. ОБЗОР СОВРЕМЕННЫХ РАСПРЕДЕЛЕННЫХ ИЗМЕРИТЕЛЬНО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ 6
- 1.1. Описание предметной области 6
- 1.2. Система передачи данных медицинской информации 7
- 1.3. Классификация телемедицинских систем 9
- 1.4. Варианты обобщенных структур телемедицинских систем 14
- 1.5. Проблемы построения медицинских информационных систем 24
- 1.6. Типовые схемы включения микропроцессоров и микроконтроллеров в аппаратуре контроля за состоянием пациентов 63

2. ПРОЕКТИРОВАНИЕ УЗЛОВ ЦИФРОВОЙ ОБРАБОТКИ С ИСПОЛЬЗОВАНИЕМ СИГНАЛЬНЫХ ПРОЦЕССОРОВ 76
- 2.1. Аналогово-цифровое преобразование 77
- 2.2. Цифровая фильтрация 95
- 2.3. Аппаратная реализация DSP процессоров 105
- 2.4. Взаимодействие DSP процессоров с внешними устройствами 116
- 2.5. Особенности проектирования аппаратуры ЦОС 123

3. ТЕОРИЯ УНИВЕРСАЛЬНОГО ПРЕОБРАЗОВАНИЯ СИСТЕМ КОДИРОВАНИЯ 142
- 3.1. Многоуровневое представление средств сетевого взаимодействия 142
- 3.2. Формализация спецификации ГИИ 149
- 3.3. Алгоритм распознавания систем кодирования 161
- 3.4. Алгоритм преобразования систем кодирования 172

4. СИНТЕЗ ГИБКОГО ИНТЕЛЛЕКТУАЛЬНОГО ПРЕОБРАЗОВАТЕЛЯ 180
4.1. Синтез структуры ГИИ 180
4.2. Синтез алгоритма управления ГИИ 183
4.3. Организация базы знаний преобразователя 196

5. СХЕМНАЯ РЕАЛИЗАЦИЯ ПТК (М, С, Т) 198
5.1. Прикладные процессоры, процессоры и технологии DA VINCI 198
5.2. Системы контроля за электрофизиологическими параметрами на основе аналоговых интерфейсов AFE 214
5.3. Обобщенные схемы приборов для исследования неэлектрических характеристик организма и окружающей среды 227
5.4. Использование микромеханических систем, выполненных по технологиям МЭМС, в телемедицинских системах 241
5.5. Радиоинтерфейс BLUETOOTH 254

6. РАЗРАБОТКА МАШИННОЙ МОДЕЛИ ГИИ 272
6.1. Описание модели преобразователя систем кодирования 272
6.2. Описание модели передатчика 272
6.3. Описание работы ГИИ 276
6.4. Режим функционирования приемника 279

7. МЕТРОЛОГИЧЕСКИЙ АНАЛИЗ ГИИ 293
7.1. Погрешность сложных информационных пакетов в процессе преобразования 293
7.2. Экспериментальный анализ процесса преобразования систем кодирования 298
7.3. Определение оценок ГИИ 312
7.4. Технические характеристики ГИИ 320

8. ПРИМЕРЫ ПРАКТИЧЕСКОГО ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНО-АППАРАТНЫХ КОМПЛЕКСОВ ПРИМЕНЕНИЯ 323
8.1. Модуль обмена видеоинформацией в телемедицине 323
8.2. Дистанционная идентификация личности человека 341

Литература 352

Научное издание

КОРОЛЕВ Артем Дмитриевич,
КОРЕНЕВСКИЙ Николай Алексеевич,
КУЗНЕЦОВ Денис Николаевич, НГУЕН Тхе Кыонг,
МУХА Юрий Петрович, СЫРЯМКИН Владимир Иванович,
ТИТОВ Дмитрий Витальевич

ИНТЕЛЛЕКТУАЛЬНЫЕ ПРОГРАММНО-АППАРАТНЫЕ КОМПЛЕКСЫ ПЕРЕДАЧИ ИНФОРМАЦИИ В ТЕЛЕМЕДИЦИНСКИХ СЕТЯХ

2-е издание, исправленное и дополненное

Издание подготовлено в авторской редакции
Оригинал-макет А.И. Лелоюр
Дизайн обложки Л.Д. Кривцовой

Подписано к печати 25.03.2019 г. Формат $60\times84^{1}/_{16}$.
Бумага для офисной техники. Гарнитура Times.
Печ. л. 22,5. Усл. печ. л. 20,9.
Тираж 500 экз. Заказ № 3711.

Отпечатано на оборудовании
Издательского Дома
Томского государственного университета
634050, г. Томск, пр. Ленина, 36
Тел. 8+(382-2)–52-98-49
Сайт: http://publish.tsu.ru
E-mail: rio.tsu@mail.ru

ISBN 978-5-94621-806-1