

61:03-5/1782-1

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ ЭЛЕКТРОННОЙ
ТЕХНИКИ (ТУ)

На правах рукописи



**Нагин
Владимир Александрович**

**РАСПРЕДЕЛЕННАЯ КОМПЬЮТЕРНАЯ СИСТЕМА СБОРА И
МАТЕМАТИЧЕСКОЙ ОБРАБОТКИ ЭЛЕКТРОФИЗИОЛОГИЧЕСКИХ
СИГНАЛОВ**

05.13.01- Системный анализ, управление и обработка
информации

Диссертация на соискание ученой степени кандидата
технических наук

НАУЧНЫЙ РУКОВОДИТЕЛЬ

доктор физико-математических наук,
профессор
С.В. Селищев

Москва-2002

Содержание

СПИСОК ИСПОЛЪЗУЕМЫХ СОКРАЩЕНИЙ	4
ВВЕДЕНИЕ	6
ГЛАВА 1. ОСНОВНЫЕ ИСХОДНЫЕ ПОЛОЖЕНИЯ	12
1.1. БИОМЕДИЦИНСКИЕ СИСТЕМЫ РЕГИСТРАЦИИ И МАТЕМАТИЧЕСКОЙ ОБРАБОТКИ ЭЛЕКТРОФИЗИОЛОГИЧЕСКОЙ ИНФОРМАЦИИ	12
1.1.1 Понятие о биомедицинских системах регистрации и математической обработки физиологических сигналов	12
1.1.2 Основные направления стандартизации в области аппаратно-программных систем регистрации и обработки биологических сигналов	17
1.1.3 Расширенные требования, предъявляемые к компьютерным биомедицинским системам	22
1.2. ПРОЕКТИРОВАНИЕ КОМПЬЮТЕРНЫХ БИОМЕДИЦИНСКИХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ	24
1.2.1 Основные понятия и общая терминология	24
1.2.2 Разработка биомедицинских программных систем на основе компонентной программной архитектуры	26
1.2.3 Применение архитектур промежуточного ПО при проектировании программных биомедицинских систем	31
ГЛАВА 2. РАСПРЕДЕЛЕННАЯ КОМПОНЕНТНО-ОРИЕНТИРОВАННАЯ ПРОГРАММНАЯ АРХИТЕКТУРА ДЛЯ КОМПЬЮТЕРНЫХ БИОМЕДИЦИНСКИХ СИСТЕМ РЕГИСТРАЦИИ ЭЛЕКТРОФИЗИОЛОГИЧЕСКОЙ ИНФОРМАЦИИ	35
2.1. ОСНОВНЫЕ ЭТАПЫ ИССЛЕДОВАНИЯ	35
2.2. РАЗРАБОТКА КОНЦЕПЦИИ РЕШЕНИЯ ЗАДАЧ	38
2.2.1 Структуризация биомедицинских систем регистрации электрофизиологической информации	38
2.2.2 Показатели и критерии эффективности	41
2.3. РАЗРАБОТКА ОСНОВНЫХ МОДЕЛЕЙ КОМПОНЕНТНОЙ АРХИТЕКТУРЫ БМС	42
2.3.1 Модель организации вычислительных процессов	42
2.3.2 Структурная модель данных	50
2.3.3 Имитационная модель подсистемы распределения и обработки данных	53
ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНАЯ ИНТЕГРИРОВАННАЯ КОМПЬЮТЕРНАЯ СИСТЕМА РЕГИСТРАЦИИ И МАТЕМАТИЧЕСКОЙ ОБРАБОТКИ ЭЛЕКТРОФИЗИОЛОГИЧЕСКИХ СИГНАЛОВ "INTELLICARD"	62
3.1. ОСНОВНЫЕ ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА	62
3.1.1 Краткая характеристика и назначение компьютерной БМС "IntelliCard"	62
3.1.2 Состав и структура программного комплекса	68
3.2. ОСНОВНЫЕ ПРОГРАММНЫЕ КОМПОНЕНТЫ СИСТЕМЫ "INTELLICARD"	73
3.2.1 Подсистема записи данных с устройств регистрации электрофизиологических сигналов и распределения данных	73
3.2.2 Подсистема визуализации и печати биомедицинских данных	80
3.2.3 Подсистема хранения и накопления биомедицинских данных	83
3.2.4 Подсистема внешних коммуникационных интерфейсов	84
3.3. АНАЛИЗ ЭФФЕКТИВНОСТИ РЕАЛИЗАЦИИ ЭЛЕМЕНТОВ СИСТЕМЫ	92
3.3.1 Планирование внутренних задач системы	92
3.3.2 Исследование производительности программной реализации на основе СОМ+	94
3.3.3 Оценка затрат на реализацию дополнительных требований программной архитектуры	95
ГЛАВА 4. ПРИКЛАДНОЕ МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНОЙ БИОМЕДИЦИНСКОЙ СИСТЕМЫ "INTELLICARD"	96
4.1. ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ЭКГ РЕАЛЬНОГО ВРЕМЕНИ	97
4.1.1 Характеристики и классификация шумов, присутствующих в электрокардиограмме	97
4.1.2 Методы подавления ряда инструментальных шумов	101
4.1.3 Реализация подсистемы подавления шумов ЭКГ методами цифровой фильтрации	104
4.2. ВЫДЕЛЕНИЕ И АНАЛИЗ QRS-КОМПЛЕКСОВ В ЭКГ-СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ	112

4.2.1	<i>Классификация методик выделения QRS-комплексов</i>	113
4.2.2	<i>Алгоритм детектирования QRS-комплексов</i>	116
4.2.3	<i>Определение параметров Q-волны</i>	122
4.2.4	<i>Коррекция дрейфа изолинии</i>	126
4.2.5	<i>Определение параметров S-волны и окончания QRS-комплекса</i>	127
4.2.6	<i>Особенности реализации QRS-детектора</i>	128
4.2.7	<i>Реализация алгоритма детектирования QRS- комплексов на базе микроконтроллера</i>	131
ЗАКЛЮЧЕНИЕ		133
ЛИТЕРАТУРА		135
ПРИЛОЖЕНИЯ		141
П. 1.	ОСНОВНЫЕ СВОЙСТВА КОМПОНЕНТНОЙ АРХИТЕКТУРЫ SOM+	141
П. 2.	ОБЗОР ХАРАКТЕРИСТИК НАИБОЛЕЕ ЗНАЧИМЫХ МЕТОДОВ СЖАТИЯ ЭКГ	146
П. 3.	ИМИТАЦИОННАЯ МОДЕЛЬ QRS-ДЕТЕКТОРА	148
П. 4.	IDL-СПЕЦИФИКАЦИИ ОСНОВНЫХ ИНТЕРФЕЙСОВ ПРОГРАММНОЙ АРХИТЕКТУРЫ	152
П. 5.	ПОКОМПОНЕНТНАЯ СТРУКТУРА РАЗРАБОТАННОГО ПО	156
П. 6.	АКТ ВНЕДРЕНИЯ РЕЗУЛЬТАТОВ РАБОТЫ	157

СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

АРМ – автоматизированное рабочее место
АЦП – аналого-цифровой преобразователь
АЧХ – амплитудно-частотная характеристика
БД – база данных
БИХ – бесконечная импульсная характеристика
БМИ – биомедицинская информация
БМС – биомедицинская система
БОЗ – брокер объектных запросов
БПФ – быстрое преобразование Фурье
ВМУ – виртуальное медицинское устройство
ВСП – вариабельность сердечного ритма
ВЧ – высокочастотный
ДПФ – дискретное преобразование Фурье
ИПП – интерфейс прикладного программирования
КБМС – компьютерная биомедицинская система
КИХ – конечная импульсная характеристика
МНК – метод наименьших квадратов
МО – математическое ожидание
МОРВ – массив отсчетов реального времени
МС – медицинская система
НЧ – низкочастотный
ОС – операционная система
ООП – объектно-ориентированное проектирование
ПА – программная архитектура
ПК – персональный компьютер
ПП – программная платформа
ПО – программное обеспечение
ППО – промежуточное программное обеспечение
ПТ – плавающая точка
ПЭВМ – персональная ЭВМ
РУП – рациональный унифицированный процесс
СУБД – система управления БД
СЭ – структурный элемент
ФВЧ – фильтр верхних частот
ФНЧ – фильтр нижних частот
ФТ – фиксированная точка
ЦОС – цифровая обработка сигналов
ЦП – центральный процессор
ЧСС – частота сердечных сокращений
ЭВМ – электронная вычислительная машина

ЭКГ – электрокардиограмма

ЭОГ – электроокулограмма

ЭЭГ – электроэнцефалограмма

ЭМГ – электромиограмма

COM – компонентная объектная модель (Component object model)

CORBA – общая архитектура брокера объектных запросов (Common object request broker architecture)

DCOM – распределенный COM (Distributed COM)

FIFO – буфер «первым вошел, первым вышел» (First in-first out)

GUI – графический интерфейс пользователя (Graphical user interface)

IDL – язык описания интерфейсов (Interface description language)

MIB – медицинская информационная магистраль (Medical information bus)

RMI – удаленный вызов процедур (Remote method invocation)

UML – унифицированный язык моделирования (Unified modeling language)

USB – универсальная последовательная шина (Universal serial bus)

ВВЕДЕНИЕ

Сбор, обработка и автоматизированный анализ физиологической информации человека являются важнейшей составной частью многих диагностических методов современной медицины. Компьютерные системы сбора и математической обработки электрофизиологических сигналов являются сложными аппаратно-программными комплексами, состоящими из множества программных компонент, выполняющих функции регистрации биомедицинской информации (БМИ), ее обработки и системного анализа, а также диагностические и сервисные операции. Основным подходом к проектированию подобных комплексных систем долгое время была реализация монолитной программной архитектуры с заранее определенной функциональностью, обеспечиваемой жестко связанными друг с другом программными компонентами. Функциональные свойства подобных систем практически невозможно было расширить, так как они были способны выполнять лишь те функции, которые были заложены на этапе проектирования. Однако современные требования, предъявляемые к данным системам, в значительной мере, связаны с возможностями постоянного расширения и наращивания их функциональных свойств. Важными проблемами являются также универсализация биомедицинского программного обеспечения, под которой, прежде всего, понимается проблема повторного использования кода, и преодоление имеющихся препятствий на пути интеграции разнородных компьютерных биомедицинских систем (БМС).

За последнее десятилетие достигнут значительный прогресс в области проектирования сложных программных систем, который в корне изменил подход к их разработке и моделированию. Однако отсутствие соответствующих стандартов не позволяет полностью воспользоваться преимуществом нового подхода. Успехи процесса стандартизации, в значительной степени, состоят в разработке документов рекомендательного характера, регламентирующих инфраструктуру низшего звена – протоколов обмена, форматов файлов данных, медицинских записей и сообщений, а также концептуальные модели взаимодействия систем. Вне рассмотрения остается, так называемое, промежуточное программное обеспечение (ПО) (англ. middleware), под которым понимается определенный функционально законченный набор программных средств, интегрированных в рамках выбранной операционной системы (ОС), обеспечивающий прозрачную работу программ в неоднородной среде. Неоднородными средами, с точки зрения информатики, являются системы (локальные или глобальные), состоящие из компонент, не совместимых друг с другом с точки зрения программного окружения.

Таким образом, для реального обеспечения взаимодействия разнородных программных и аппаратных систем необходима выработка единых спецификаций программных интерфейсов ПО промежуточного звена. В данный момент эта работа имеет высокий приоритет у ведущих мировых учреждений по стандартизации, хотя и далека до завершения.

С этой точки зрения актуальным является исследование общих свойств компьютерных биомедицинских систем и разработка с учетом требований современных стандартов единых информационных моделей их функционирования. Реализация на этой основе универсальной объектно-ориентированной инфраструктуры, под которой понимается множество программных компонент и интерфейсов с четкой регламентацией возможностей их использования, дает возможность перейти от монолитной программной архитектуры к компонентно-ориентированной распределенной, решить проблемы универсализации программного обеспечения (ПО) и значительно повысить экономическую эффективность разработок. Существенная разнородность и различная функциональная направленность биомедицинских систем делает задачу в общем смысле практически невыполнимой. Однако существует относительно широкий спектр программных биомедицинских систем, для которых подобная задача может быть успешно решена и, прежде всего, для компьютерных биомедицинской систем сбора и математической обработки электрофизиологической информации.

Цель и задачи исследования

Целью исследования являлась разработка распределенной компонентно-ориентированной компьютерной биомедицинской системы регистрации и математической обработки электрофизиологической информации на основе единых информационных моделей, исследование их свойств и определение характеристик отдельных элементов, в том числе:

- 1) анализ существующих направлений стандартизации данных биомедицинских систем, изучение возможностей интеграции разнородных систем в гетерогенной среде;
- 2) построение единых информационных моделей функционирования рассматриваемых систем, включая модель распределения и представления данных, формирование единой программной архитектуры;
- 3) решение задачи функционирования исходной системы путем введения ряда независимых процессов, допускающих параллельное выполнение в режиме разделения времени, и решение проблемы их планирования;
- 4) разработка программного обеспечения распределенной компьютерной системы сбора и математической обработки электрофизиологической информации;

- 5) реализация эффективных методов цифровой обработки данных в реальном масштабе времени, в том числе, методов предварительной обработки сигналов, выделения из электрокардиограммы QRS-комплексов и их анализа, сжатия и отображения биомедицинской информации;
- 6) оценка эффективности разработанного программного обеспечения на основе выбранных критериев и показателей.

Основные результаты работы:

- Разработана и реализована компонентно-ориентированная программная архитектура (ПА) для компьютерных биомедицинских систем регистрации и обработки физиологических сигналов, основанная на компонентной архитектуре COM+. Реализация ПА включает спецификации интерфейсов взаимодействия, реализованных средствами языка IDL и механизмы интеграции на основе ведущих стандартов, представленные UML-моделями.
- На основе разработанной ПА построен экспериментальный электрокардиографический переносной компьютерный комплекс «Intellicard» (см. Рис. 1), обеспечивающий функции регистрации, хранения, обработки, анализа ряда физиологических сигналов: электрокардиограмм, степени насыщения крови кислородом, артериального давления. ПО программного комплекса полностью реализовано средствами языка C++ интегрированной среды разработки Visual Studio.

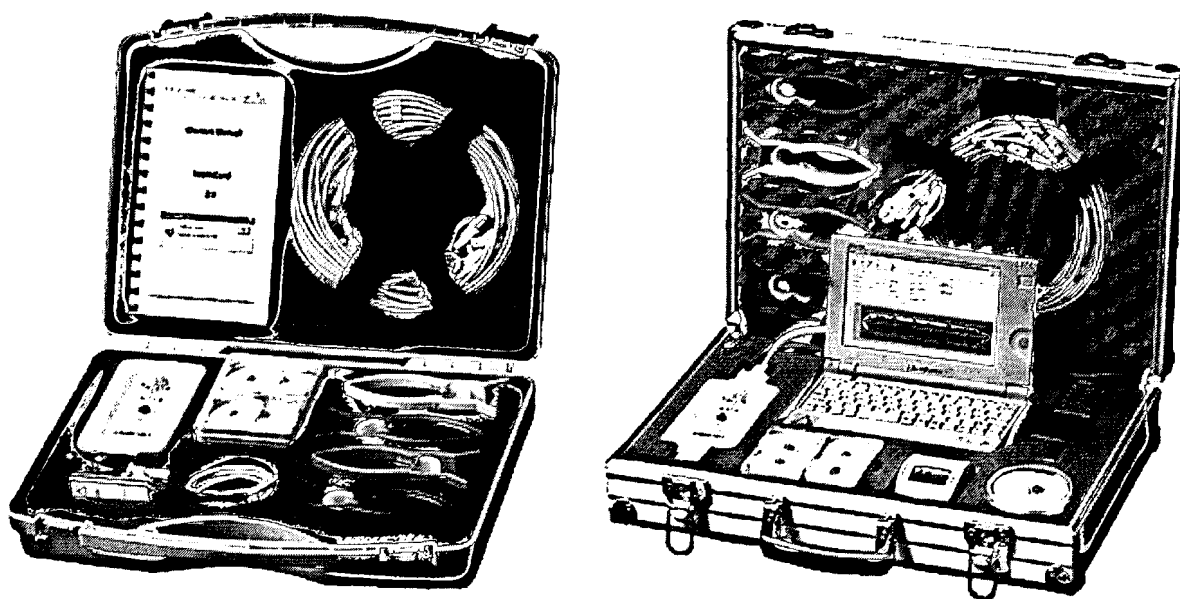


Рис. 1. Компьютерная БМС «Intellicard» в полной комплектации.

В состав комплекса "IntelliCard" могут входить следующие аппаратные компоненты:

- портативная ЭВМ;
 - генератор тестовых сигналов ЭКГ;
 - сотовый телефон с поддержкой функций модема через инфракрасный порт IrDa;
 - пульсовой оксиметр, управляемый через интерфейс последовательного порта;
 - монитор артериального давления;
 - компьютерный электрокардиограф «КАРДИ», разработанный на основе сигма-дельта АЦП;
 - прочие устройства регистрации электрофизиологической информации.
- Разработана система математических и имитационных моделей отдельных элементов компьютерных биомедицинских систем регистрации и обработки электрофизиологической информации, позволяющих подойти к разработке данных систем с единых позиций. Созданы модели распределения и физического представления данных, решена задача выделения параллельно выполняемых процессов и их планирования. Имитационные модели выполнены средствами визуальной среды Simulink, входящий в пакет Matlab.
 - Созданы с использованием языка Visual C++, разработанного инструментария и вспомогательного программного обеспечения отладки и профилирования программные модули универсального представления, печати и хранения БМИ, интерфейсы прикладного программирования для работы с телекоммуникационными интерфейсами.
 - Реализованы в виде универсальных программных компонент методики предварительной обработки сигналов, сжатия БМИ, выделения и анализа QRS-комплексов.
 - Разработана программная библиотека поддержки динамически подключаемых модулей, основанная на технологиях COM и ActiveX, и инструментарий разработчика данных модулей.
 - Создан программный слой динамического подключения устройств регистрации физиологических данных и соответствующие библиотеки прикладного программирования для WindowsNT\2000\XP. В настоящее время реализована подсистема управления несколькими устройствами съема, реализовано несколько модулей, позволяющих вести запись 12-ти, 6-ти и 3-канальной ЭКГ, а также съем данных с монитора артериального давления и пульсового оксиметра. Аппаратная

часть системы поддерживает съем с фиксированной частотой дискретизации, равной 500 Гц, с 22-битным аналого-цифровым преобразованием на основе сигма-дельта АЦП [1], что обеспечивает достаточную точность для дальнейшего анализа.

- На базе комплекса «Intellicard» разработан встраиваемый модуль ЭКГ, построенный на базе промышленного компьютера, представляющего в качестве интерфейса управления несколько функциональных клавиш, расположенных по периметру ЖК-экрана, и манипулятор. Программное обеспечение встраиваемого модуля построено из компонент, используемых в системе «IntelliCard».

Достоверность полученных результатов обусловлена использованием общепринятых математических методов обработки сигналов, методов разработки и верификации программного обеспечения, российских и международных стандартов на программно-аппаратные комплексы съема и обработки биомедицинских сигналов.

Верификация разработанного программного обеспечения заключалась в:

- 1) проверке эффективности разработанного программного обеспечения поддержки компонентно-ориентированной архитектуры на основе ПО промежуточного уровня СОМ+[2];
- 2) отладке и оценке работоспособности системы «Intellicard», как в целом, так и отдельных ее компонент, реализующих различные функции, в том числе, сжатия и хранения БМИ, предварительной обработки, выделения и анализа элементов ЭКГ;
- 3) опытном применении разработанного программного обеспечения.

Апробация работы

Основные положения диссертации докладывались и обсуждались на следующих научно-технических конференциях и семинарах:

- Электроника и информатика – 97. Всероссийская научно-техническая конференция с международным участием, Москва, МИЭТ, 1997г.
- Биомедприбор – 98. Международная конференция по биомедицинскому приборостроению, Москва, ВНИИМП РАМН, 1998 г.
- Микроэлектроника и информатика – 98. Межвузовская научно-техническая конференция, Москва, МИЭТ, 1998 г
- Микроэлектроника и информатика – 99. Межвузовская научно-техническая конференция, Москва, МИЭТ, 1999 г.
- Компьютерная электрокардиография на рубеже столетий XX–XXI. Международный симпозиум, Москва, 1999 г.
- Микроэлектроника и информатика – 2000. Межвузовская научно-техническая конференция, Москва, МИЭТ, 2000 г.

- Физика и радиоэлектроника в медицине и экологии – 2000. Международная научно-техническая конференция, Владимир, ВлГУ, 2000 г.
- Биомедприбор – 2000. Международная конференция по биомедицинскому приборостроению, Москва, ВНИИМП-РАМН, 2000 г.
- Электроника и информатика – 2000. Международная научно-техническая конференция, Москва, МИЭТ, 2000 г.
- Микроэлектроника и информатика – 2001. Межвузовская научно-техническая конференция, Москва, МИЭТ, 2001 г.
- Медицинская физика-2001. 1-й Евразийский конгресс по медицинской физике и инженерии, Москва, МГУ, 2001 г.
- Радиоэлектроника в медицинской диагностике – 2001. Международная научно-техническая конференция, Москва, ИРЭ РАН, 2001 г.
- 23rd Annual international conference of the IEEE Engineering in Medicine and Biology Society. Istanbul, 2001 г.
- Микроэлектроника и информатика – 2002. Межвузовская научно-техническая конференция, Москва, МИЭТ, 2002 г.
- The 17th international congress of European Federation for Medical Informatics. Budapest, 2002 г.
- Электроника и информатика – 2002. Международная научно-техническая конференция, Москва, МИЭТ, 2002 г.

Глава 1. ОСНОВНЫЕ ИСХОДНЫЕ ПОЛОЖЕНИЯ

1.1. Биомедицинские системы регистрации и математической обработки электрофизиологической информации

1.1.1 Понятие о биомедицинских системах регистрации и математической обработки физиологических сигналов

Биологические системы являются наиболее сложными системами, с которыми когда-либо приходилось сталкиваться человеку. Это, прежде всего, объясняется тем, что исследование подобных систем осложняется наличием ряда ограничений, наиболее важным из которых является невозможность изолировать анализируемую систему. Любая подобная система испытывает сильное влияние со стороны значительного числа смежных систем, без учета воздействия которых получение достоверных результатов невозможно. В настоящее время одним из наиболее важных инструментов при проведении подобных исследований служат компьютерные биомедицинские системы (БМС) регистрации и обработки электрофизиологических сигналов, обеспечивающие функции записи, обработки и анализа биомедицинских сигналов различной природы.

Биологические сигналы допускают несколько возможных классификаций, наиболее важными из которых являются разделение по физической природе источника сигнала, по предметной области применения и, наконец, по характеристикам самого сигнала.

По физической природе источника регистрируемые биомедицинские сигналы можно разделить на несколько групп [3]:

- 1) *биоэлектрические* – связаны с активностью нервных и мышечных клеток;
- 2) *импедансные* – состоят в изменении импеданса тканей при пропускании переменных токов, с частотами до 1 МГц;
- 3) *биоакустические* – создаются различными физиологическими процессами человека: дыхание, движение и т.д.;
- 4) *биомагнитные* – связаны с излучением человеком чрезвычайно слабых магнитных полей;
- 5) *биомеханические* – включают все виды механических функций человека;
- 6) *биохимические* – являются результатом измерений химических процессов, происходящих в тканях;
- 7) *оптические* – связаны с регистрацией электромагнитных излучений в оптической области спектра, обусловленных функциональной деятельностью биологических систем или наведенных извне.

Основные биологические сигналы, с которыми наиболее часто работают компьютерные БМС, приведены в Табл. 1.1.

Табл. 1.1. Характеристики основных физиологических сигналов человека.

Предметная область	Сигнал	Частотный диапазон	Динамический диапазон
Нейрофизиологическая	Электроэнцефалограмма (ЭЭГ)	0.5 – 100 Гц	2-100 мкВ
	Электроокулограмма (ЭОГ)	0 – 100 Гц	10мкВ – 5мВ
	Электромиограмма (ЭМГ)	500 – 10 кГц	1-10мВ
	Электроретинограмма (ЭРГ)	0.2 – 200 Гц	0.5мкВ – 1мВ
	Электrokортикограмма	100 – 5 кГц	0.5мкВ– 10мкВ
Сердечно-сосудистая	Электрокардиограмма	0.05 – 100 Гц	1-10мВ
	Артериальное давление	до 100 Гц	40-200 мм.рт.ст
	Частота сердечных сокращений	до 3 Гц	0-200 уд.
Респираторная	Степень насыщения кислородом	0-1 Гц	
	PO ₂ , PCO ₂	0-1 Гц	
	Шум дыхания	до 5 кГц	
	Температура тела	до 1 Гц	0-42С

Классификация сигналов по предметной области основывается на рассмотрении спектра биомедицинских задач, таких как диагностика, мониторинг, решение которых строится путем регистрации, математической обработки и анализа тех или иных электрофизиологических сигналов.

Классификация по физическим характеристикам предусматривает разделение множества сигналов на группы, которые в свою очередь, также делятся на подгруппы в соответствии с такими свойствами, как периодичность, стационарность, детерминированность и т.д.

Различные по своему назначению, компьютерные биомедицинские системы имеют одну общую особенность - источником данных является сам человек. Немалую роль в развитии БМС играет прогресс в области вычислительных средств и вычислительных методов анализа. Возможности комплексов увеличиваются, совершенствуются методики и алгоритмы

анализа данных, вычислительная нагрузка растет и перекладывается полностью с аппаратного обеспечения на программные средства. Ядром таких программных комплексов выступает подсистема цифровой обработки сигналов (ЦОС), от эффективности работы которой и зависит общая производительность системы, ее надежность и потребительские свойства комплекса в целом, в то время как качество биомедицинской информации в значительной мере определяется реализацией аналого-цифрового преобразования и схемотехникой аналоговых входных схем [4] (Рис. 1.1).

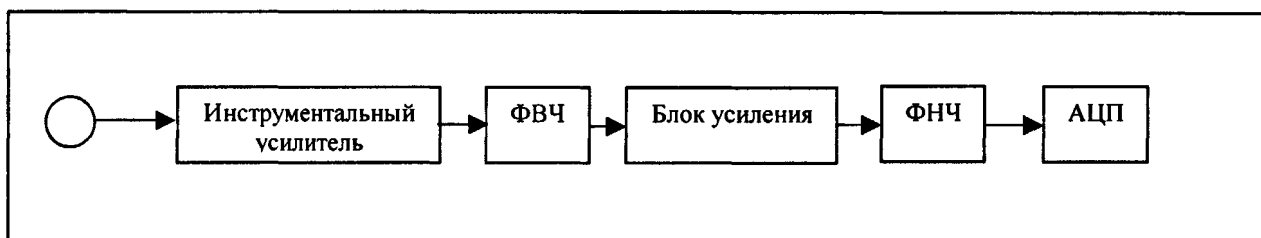


Рис. 1.1. Схема аналоговой части системы сбора электрофизиологической информации.

Одним из наиболее распространенных подходов при разработке биомедицинских систем является проектирование комплекса на базе аппаратной части ПК или одноплатных ЭВМ, содержащих необходимый набор периферийных устройств со встроенным программным обеспечением. Это дает возможность использовать стандартные ОС, обслуживающие ресурсы системы и сосредоточиться на реализации специфичной для биомедицинской системы функциональности (Рис. 1.2). При этом большое значение получает выбор типа интерфейса ПК или одноплатной ЭВМ с проектируемыми периферийными устройствами. Подобные системы принято называть компьютерными.

В процессе проектирования компьютерных БМС часто выделяется два типа основных работ:

- 1) проектирование и реализация аппаратной части устройств регистрации биомедицинской информации (БМИ);
- 2) проектирование и разработка программного обеспечения, в том числе встраиваемого.

При этом, разработка программного обеспечения (ПО) может быть разделена на несколько функциональных этапов:

- 1) выбор программной платформы, под которой понимается ОС и набор предоставляемых ею инструментальных средств;

- 2) разработка архитектуры программного обеспечения системы на основании технического задания и исследование ее свойств на основе методов моделирования;
- 3) реализация отдельных функциональных компонент системы;
- 4) интеграция аппаратных средств регистрации БМИ;
- 5) выбор и реализация коммуникационных каналов для интеграции системы в информационное пространство.

Одним из старейших методов обработки и анализа электрофизиологической информации человека является электрографический метод. Этот метод регистрации и анализа биоэлектрических процессов человека и животных, нашел весьма широкое применение в клинической практике, физиологическом эксперименте, авиационной и космической медицине, исследованиях по физиологии труда и спорта. Столь широкое применение электрографического метода объясняется тем, что он позволяет получить ценную информацию о деятельности тканей, органов и систем. Электрическое возбуждение распространяется в определенных направлениях и последовательности, создавая на поверхности тела электрическое поле. Поскольку электропроводные свойства тканей неоднородны, электрическое поле асимметрично с разностью потенциалов между отдельными участками тела. Это свойство положено в основу метода электрокардиографии, который регистрирует разность потенциалов путем различных отведений от поверхности тела, что достигается с помощью приборов - электрокардиографов[5].

Компьютерные системы ЭКГ-диагностики берут свое начало в начале 70-х годов, когда появились достаточно дешевые и компактные аппаратные средства для их реализации. Первые программные комплексы являлись попыткой автоматизировать процесс получения из ЭКГ диагностической информации с помощью компьютера [6,7]. Ожидалось, что кроме тех преимуществ, которые дает автоматизация, удастся также повысить диагностическую точность анализа, благодаря применению статистических методов, которые не могут быть использованы при обработке вручную. Результатом развития различных компьютерных систем явился ЭКГ-анализ, позволивший существенно повысить эффективность диагностики, уменьшить относительную частоту диагностических ошибок, сократить время электрокардиографического обследования, освободить квалифицированных врачей от выполнения трудоемких и рутинных процедур измерения и анализа электрокардиограмм. До начала 70-х годов ЭКГ-отведения записывались последовательно одно за другим. Лишь позже появились трехканальные электрокардиографы, позволявшие вести запись сразу трех отведений. Современная регистрирующая аппаратура может одновременно фиксировать до 12 отведений, а также больше, если это необходимо.

Логика развития современных методов медицинской диагностики заключается в увеличении объема БМИ, необходимой в клинической практике. Поэтому для современных компьютерных БМС регистрации электрофизиологических сигналов характерна поддержка записи и обработки нескольких различных биоэлектрических сигналов. После ЭКГ наиболее значимыми клинически являются ЭЭГ и ЭМГ-сигналы, также велико значение таких физиологических параметров, как артериальное давление, степень насыщения крови кислородом, температура тела. Приведенная обобщенная схема на Рис. 1.2. является характерной для реализаций систем на основе компьютерных устройств.

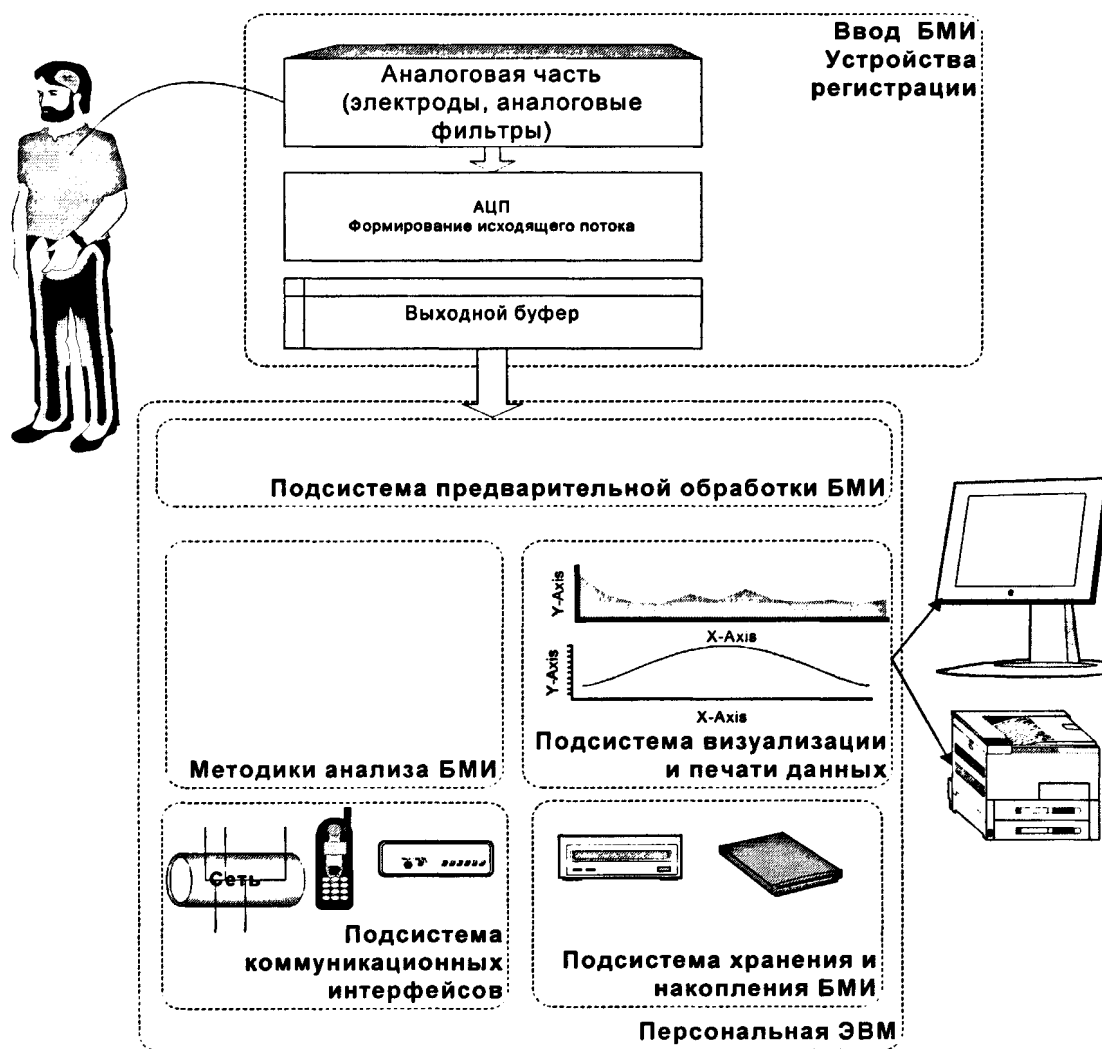


Рис. 1.2. Обобщенная схема компьютерной БМС.

Аппаратная часть представляет собой устройство регистрации и предварительной обработки сигнала и обычно осуществляет аналого-цифровое преобразование. Буферизация данных на выходе является обычным для систем реального времени с целью синхронизации или

упаковки передаваемых данных. Интерфейсной частью со стороны программной системы выступает системное приложение - драйвер, обеспечивающий взаимодействие с аппаратной частью на низком уровне. Средства визуализации и печати обеспечивают представление данных в удобном для анализа формате. Все больший вес приобретает подсистема коммуникационных интерфейсов, поэтому она в данном представлении выделена в отдельный блок.

1.1.2 Основные направления стандартизации в области аппаратно-программных систем регистрации и обработки биологических сигналов

Бурное развитие и внедрение информационных систем в клиническую практику привело к необходимости их стандартизации. Потребность в разработке регламентирующих норм наступает, когда развитие какой-либо отрасли достигает определенного состояния, которое характеризуется окончанием процесса предварительного накопления знаний. К этому моменту в эту сферу деятельности обычно вовлекается достаточно широкий круг специалистов и значительное количество финансовых средств. Массовое производство и спрос порождает широкое конкурентное поле, на котором появляются многочисленные производители. Необходимость повышения качества изделий, обеспечения их интеграции и получения единых методов оценки качества требует разработки единых стандартов. При этом стандартизация начинается с разрешения каких-либо частных специфических вопросов, где противоречия назрели достаточно остро. В дальнейшем, развитие этого процесса приводит к логичному обобщению и появлению широких общепринятых норм, опирающихся на существующую технологическую базу и разработанных на основе современных методов документирования.

Стандартизация в области медицинской информатики в целом, биомедицинских аппаратов и систем в частности, наталкивается на ряд трудностей, вызванных, прежде всего, сложностью в формализации задач и построении единых моделей подобных систем. Однако, несмотря на то, что развитие отрасли биомедицинских систем характеризуется, в значительной мере, противоречивостью и непоследовательностью, стандарты обмена медицинской информацией успешно развиваются и внедряются на уровне крупных и средних информационных систем. Задачей разработки информационных норм является построение на основе моделирования разнообразных биомедицинских данных единой номенклатуры, универсальных методов представления и кодирования данных, а также системы сообщений, позволяющей интегрировать подсистемы различных производителей в рамках единой информационной среды медицинских учреждений. Рост стоимости организаций медицинского обслуживания

привел разработчиков медицинских систем к необходимости применения нового подхода, главным принципом которого является ориентация на рядового пациента, вовлеченного в информационную среду здравоохранения.

Для настоящего времени характерно разделение вопросов стандартизации медицинских информационных систем масштаба крупных организаций с одной стороны, и биомедицинских программных систем регистрации и обработки данных - с другой. Данный факт в какой-то степени является логичным, поскольку подходы, применяемые в этих двух случаях, фактически противоположны. В первом случае, большое значение уделяется структурам данных, полноте представления и универсальности номенклатуры, а во втором – производительности, универсальности интерфейса и скорости взаимодействия между отдельными компонентами системы.

Создание огромного количества близких по своим функциям, но совершенно несовместимых друг с другом систем, стало серьезным препятствием на пути развития отрасли в целом. Наиболее сложной задачей, стоящей перед разработчиками любой программной биомедицинской системы, является обеспечение высокого уровня совместимости между биомедицинскими приложениями, разработанными различными производителями и работающими на различных платформах. Одной из наиболее важных целей быстро развивающихся процессов стандартизации биомедицинских систем является преодоление имеющихся препятствий на пути интеграции систем и достижения, так называемой *интероперабельности* (от англ. *interoperability*), под которой понимается способность систем эффективно взаимодействовать друг с другом. Данное понятие имеет широкое значение: от возможности различным системам иметь доступ к единым информационным ресурсам, до способности систем эффективно взаимодействовать друг с другом для решения прикладных задач в масштабе реального времени.

Под способностью разнообразных систем к взаимодействию по принципу **plug-and-play** понимают совокупность их свойств, благодаря которым подобные системы могут эффективно быть интегрированы в рамках единого аппаратно-программного комплекса на основании установленных общих принципов и разработанных протоколов. Данное свойство систем подразумевает взаимную совместимость на основании сертификации их соответствия установленным нормам и стандартам. Примерами подобных комплексов могут выступать системы интенсивной терапии, состоящие из набора разнообразных устройств, управляемых единой информационной системой [8]. Укрупненная схема с основными направлениями стандартизации в области медицинской информатики представлена на Рис. 1.3.

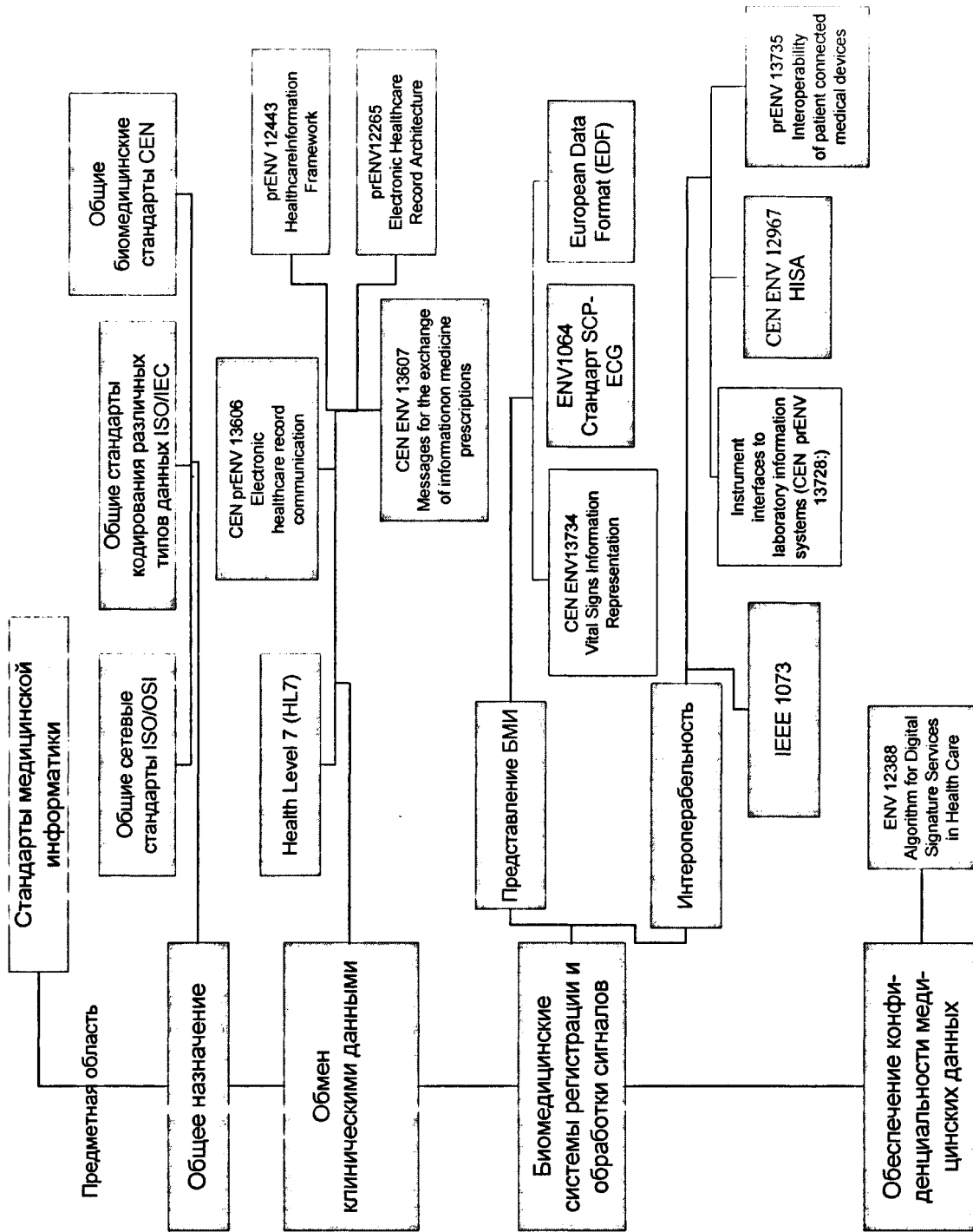


Рис. 1.3. Основные направления стандартизации современной медицинской информатики.

Различные по своей направленности стандарты разнесены по разным предметным областям: нормы общего назначения, стандарты обмена клиническими данными, обеспечения конфиденциальности, а также спецификации в области разработки биомедицинских систем регистрации и обработки БМИ.

Нормы **общего назначения** во многом не являются специфичными для области биомедицинской информатики и относятся к информационной науке в целом. Эту категорию формируют нормы общего использования, в частности, основные стандарты построения открытых сетей [9], широко используемые в разработке аппаратно-программных систем, общие стандарты кодирования простых типов данных, а также биомедицинские спецификации общего характера.

Стандарты **группы обмена клиническими данными** сосредотачивают свое внимание на информационных системах масштаба крупных учреждений. Среди подобных групп стандартов следует выделить хорошо зарекомендовавший себя стандарт HL7¹[10], официально получивший статус нормы ANSI². HL7 является спецификацией клинического интерфейса, соответствующего седьмому уровню в стеке протоколов OSI³, и решает задачи связанные с организацией электронного обмена разнообразной клинической, финансовой и административной информацией. Основой стандарта выступает так называемая реферативная информационная модель, которая служит основой для методологии построения клинических сообщений, служащих основой протокола обмена. Это позволяет придать процессу формирования сообщений и прочих элементов протокола необходимую логичность. Дополнительно в стандарте приведена модель предметной области с соответствующей номенклатурой и моделями использования. Аналогом HL7 выступает Европейский стандарт Electronic healthcare record communication [11], рассматривающий проблему взаимодействия информационных систем со сходных позиций. В дальнейшем возможно объединение этих двух стандартов в единую спецификацию: в настоящее время подобные работы ведутся в рамках совместной рабочей группы.

Таким образом, в настоящее время можно выделить несколько подходов к решению задач информационного взаимодействия систем:

- прямое использование стандартов представления и обмена медицинской информацией, таких как HL7 или EHCR;
- применение стандартов представления данных на уровне файлов общего доступа и обмена;

¹Health Level 7.

²American National Standards Institute

³Open System Interconnection-международная программа стандартизации

- комбинированное применение форматов представления данных на уровне файловой структуры и стандартов обмена медицинскими записями;
- разработка и реализация частных протоколов обмена.

Долгое время разработка специфичного для производителя внутреннего представления данных и собственных закрытых протоколов обмена являлась обычной практикой, поскольку разработка каких-либо интегрирующих стандартов в области биомедицинских систем является гораздо более сложной задачей в силу широты предметной области и разной специфики. Биомедицинские программные системы регистрации биологических сигналов имеют широкую функциональную направленность, и их стандартизация предусматривает построение абстрактных информационных моделей всех этих систем, жестко регламентирующих механизмы работы всех элементов. Комплекс подобных моделей составляет метамодель предметной области, в рамках которой определена система связей всех элементов, разработана соответствующая номенклатура и формы кодирования событий и данных. В рамках такой метамодели, любой формат представления данных является некоторым логическим следствием системных построений и не привязан к конкретному транспортному уровню.

Группа стандартов в области **биомедицинских систем** нацелена на решение подобных задач и концентрирует свое внимание на регламентации коммуникационного интерфейса для интеграции данных систем. Существующие интерфейсные стандарты в вычислительной технике (например, IEEE 802.x, USB и т.д.) не способны решить в полной мере задачи по обеспечению интерактивного взаимодействия по принципу plug-and-play разнообразных устройств, осуществляющих регистрацию и обработку физиологических сигналов в режиме реального времени. Универсальный коммуникационный стандарт должен учитывать особенности функционального назначения разнообразных подключаемых устройств – систем регистрации сигналов, систем мониторинга т.д.

Таким образом, можно выделить несколько подходов к разработке интегрируемых систем регистрации и обработки физиологических сигналов:

- 1) применение регламентированных форматов, решающих вопросы совместимости на уровне использования общих файлов хранения и обмена;
- 2) создание специализированного промежуточного ПО, реализующего необходимый открытый набор интерфейсов;
- 3) применение универсальных стандартов проектирования биомедицинских систем регистрации и обработки разнообразных

физиологических данных таких, как Medical Information Bus⁴ (MIB) [12];

- 4) комбинированной применение трех представленных выше методов.

Форматы записей медицинских сигналов содержат спецификации представления биомедицинских данных: физиологических сигналов, информации о пациенте, аннотаций и прочей информации в виде бинарной последовательности данных. Данное представление может быть использовано для сохранения полученной информации на различных носителях, централизованной БД или для организации обмена данными между различными программными биомедицинскими системами. В настоящий момент насчитывается большое количество разнообразных открытых форматов представления электрофизиологических данных, многие производители БМС имеют также свой закрытый формат данных.

1.1.3 Расширенные требования, предъявляемые к компьютерным биомедицинским системам

Современные программные биомедицинские системы являются сложными комплексами, состоящими из множества разнородных компонент. Основным подходом к их построению является создание монолитного приложения, обладающего заранее определенной функциональностью, которая обеспечивается жестко связанной группой компонент-модулей. В качестве примера можно привести типичную систему ЭКГ-диагностики, работающую под управлением операционной системы DOS. Она строится из законченных исполняемых модулей, взаимодействие между которыми осуществляется набором командных файлов: обмен данными обычно происходит через временные файлы. Обладая набором определенных характеристик, такая система способна обеспечить решение лишь тех задач, которые ставились перед разработчиками на этапе ее создания. С точки зрения пользователя система предоставляет законченный интерфейс, обладающий минимальными возможностями адаптации, а не редко и вовсе лишенный таковых. Одновременно с этим, такой интерфейс предполагает полную однозначность дерева реализуемых функций, прост в документировании и использовании. Однако по завершению работ над проектом развитие подобной системы застывает, или же она устаревает еще во время разработки. Причины этого кроются в проблемах, связанных с расширяемостью, надежностью и невозможностью повторного использования кода, свойственных такой архитектуре. Таким образом, подобная система не оправдывает как ресурсов, потраченных на ее

⁴ Также известен как семейство стандартов IEEE 1073.

создание, так и не оставляет конечному пользователю возможности модернизации существующего оборудования с развитием новых методов компьютерной диагностики.

На основе приведенных фактов можно сформулировать требования, предъявляемые к современным комплексам сбора и математической обработки ЭКГ-данных, которые можно разделить на две группы.

Архитектурные:

- модульная организация системы;
- гибкая модель взаимодействия и взаимозаменяемости модулей (динамическое связывание);
- открытая архитектура управления взаимодействием модулей (возможность создания новых модулей и моделей их взаимодействия сторонними разработчиками);
- надежность комплекса, которая не должна уступать надежности систем, построенных на базе жестких моделей взаимодействия.

Функциональные:

Не исключая требования, характерные для большинства систем этого класса, отметим наиболее важные для представленной реализации:

- обеспечение легкости в обучении пользовательскому интерфейсу;
- возможность адаптации комплекса как для различных применений (клинические, лабораторные, научно-исследовательские работы), так и для конкретного пользователя;
- документированный интерфейс взаимодействия с оборудованием, обеспечивающий возможность работы с приборами разных производителей;
- общее хранилище данных, доступное каждому подмодулю системы и инкапсулирующее методы взаимодействия с различными внешними источниками данных (СУБД, системы резервирования, сетевые ресурсы);
- интеграция комплекса в современные медицинские системы посредством реализации требований стандарта IEEE 1073.

Нетрудно заметить, что перечисленные требования обладают определенными противоречиями, которые в первую очередь продиктованы проблемой выбора между жестким и гибким архитектурными решениями.

Основная задача разработки подобных систем состоит в том, чтобы создать универсальную архитектуру построения программных БМС сбора и обработки БМИ, позволяющую наиболее полно разрешить дилемму между жестким и гибким архитектурными решениями.

1.2. *Проектирование компьютерных биомедицинских систем реального времени*

1.2.1 Основные понятия и общая терминология

Потребность в функциональном программном обеспечении за последние годы растет значительными темпами. Это связано с проникновением вычислительных систем в новые области применения, характеризующиеся постоянным повышением сложности программных систем, что приводит к увеличению времени реализации и падению надежности. Производительность же процесса разработки за последнее двадцатилетие выросла незначительно, в основном, за счет совершенствования вычислительных систем и средств разработки. Ранее, когда аппаратные ресурсы вычислительных систем были незначительны, основной задачей теории программирования являлось совершенствование алгоритмических методов решения конкретных задач. Данный подход получил название «программирования в малом». Разработка значительных проектов для целого спектра разнообразных задач привела к повышенному интересу к понятию **программная архитектура (ПА)**. В настоящий момент классическим является определение Г. Буча [13]: “ ПА – набор стратегических решений по организации программной системы, выбору ее структурных элементов (СЭ) и их интерфейсов, совместно с описанием поведения на уровне спецификаций взаимодействия СЭ, а также описание композиции СЭ в подсистемы, включающее архитектурный стиль и руководства по всем перечисленным пунктам”. Альтернативным и более математическим служит определение М. Шоу [14]: "**Программная архитектура** – структурная модель, представляющая программную систему множеством компонентов, соединением этих компонентов и (обычно) различными аспектами реализации". Разработка теории проектирования программных систем началась сравнительно недавно и в настоящий момент предметная область не является полностью и однозначно определенной. Можно выделить несколько категорий предметной области ПА, иными словами, терминологии.

К первой из таких категорий относится ПА потоков данных, которая основывается на рассмотрении модели программной системы, состоящей из двух разновидностей компонент: **каналов передачи и фильтров**. В рамках данной модели каналы обеспечивают доставку данных от одного компонента-фильтра к другому. Фильтр, обладающий входом и выходом, осуществляет некоторые операции с данными, поступившими на его вход, и выдающий результат на выход, при этом к одному выходу можно подключить несколько каналов. Данная архитектура является прямой реализацией модели потоков данных, известной достаточно давно.

Вторая категория – **ПА параметрического описания функций**, основанная на соответствующих классической парадигме программирования компонентах – подпрограмм, образующих иерархическую структуру. На взаимодействие компонент не налагаются какие-либо ограничения в зависимости от уровня их реализации. Механизм удаленного вызова подпрограмм, обеспечиваемый сервисами аналогичными сервисам удаленного вызова процедур, или ПО промежуточного слоя, является предметной областью данной категории.

Третьей категорией является **объектно-ориентированная ПА**, которая основывается на иерархической структуре объектов, инкапсулирующих данные и соответствующие функции работы с ними. Взаимодействие между компонентами сводится к механизмам межобъектного взаимодействия.

К четвертой категории можно отнести многослойные, или **многослойные, ПА**. Компоненты такой ПА являются слоями, реализующими дополнительную функциональность на основе функций, лежащих на нижнем уровне слоев. Реализация слоя не является существенной, он может являться, например, программной библиотекой, "надстроенной" над другой библиотекой, или же транслятором из одного языка в другой, для которого уже есть интерпретатор. Примерами подобных ПА могут служить виртуальные машины языков.

Пятой категорией являются **ПА независимых компонент**, представляющие собой исполняемые сущности, обменивающиеся сообщениями и реагирующие на сообщения на основе так называемой "модели событий". Последняя фактически представляет собой аналог уникальной для каждого компонента таблицы, ставящей в соответствие сообщению инициацию выполнения компонентом некоторого действия. Подобные таблицы получили название интерфейсов. Данная модель обеспечивает 100%-ное повторное использование кода, однако достаточно трудоемка при реализации.

Очевидно, что архитектура системы не может быть исчерпывающе описана на основании одной из пяти вышеописанных ПА. Структура любой системы может быть рассмотрена с различных позиций, например, ПА независимых компонент и объектно-ориентированной (ОО) ПА.

Основным средством поддержки разработки значительных по объему проектов в настоящее время является применение объектно-ориентированной методологии проектирования (ООП). ООП фактически единственный в настоящее время подход, позволяющий эффективно решать вопросы проектирования сложных программных систем. Любая зрелая методология должна включать в себя возможность качественного и количественного анализа архитектурных решений, выбранных при проектировании той или иной системы, оценивать ее количественные характеристики – сложность, структуру, ожидаемое количество дефектов,

длительность процесса отладки. Основными методами объектно-ориентированного анализа к середине 90-х годов стали Booch, названный по имени автора и одного из главных идеологов Г. Буча, OMT и Objectory. Объединение этих методов дало унифицированный объектно-ориентированный подход, получивший название UML – Unified modeling language. UML – это единый метод объектно-ориентированного анализа, включающий методы визуализации, описания и документирования ОО-проектов.

Широко используемым понятием в теории программирования является программная платформа, которая тесно связана с **интерфейсом прикладного программирования (ИПП)**. Под ИПП понимается набор базовых сервисных функций, предоставляемых операционной системой (ОС) или дополнительными компонентами ОС приложениям. Разнообразие используемых ИПП и выбранную ОС принято называть программной платформой (ПП). Оптимальный выбор ПП, инвестиции на приобретение качественных программных и аппаратных средств является одним из стратегических решений, определяющих успех разработки в целом [15].

1.2.2 Разработка биомедицинских программных систем на основе компонентной программной архитектуры

Стандартным подходом к проектированию программных систем является применение монолитной архитектуры, заключающей в себе все необходимые функции, заложенные на этапе разработки. Подобные системы обычно проектируются и реализуются с помощью того или иного языка высокого уровня, поддерживающего ООП. Такая архитектура не отличается гибкостью, так как невозможно произвести модернизацию системы без ее замены целиком. С другой стороны, такие монолитные приложения сравнительно надежны, так как на их отладку затрачивается достаточно много времени. Однако такой уровень надежности удается сохранить лишь до внесения каких-либо изменений и усовершенствований в систему, после чего фактически приходится тестировать все приложение целиком. Пример монолитной архитектуры приложения приведен на Рис. 1.4. В данной схеме функциональность системы, что свойственно методам объектно-ориентированного проектирования, реализована набором статических классов. Глобальную область видимости имеют общие переменные и основная функция *main*, которой после запуска приложения передается управление. Данная реализация предусматривает возможность применения только одного языка разработки всего приложения, так как только в этом случае удастся обеспечить непрерывный процесс компиляции и компоновки программы, осуществляемой статически, т.е. на этапе получения окончательного исполняемого модуля.

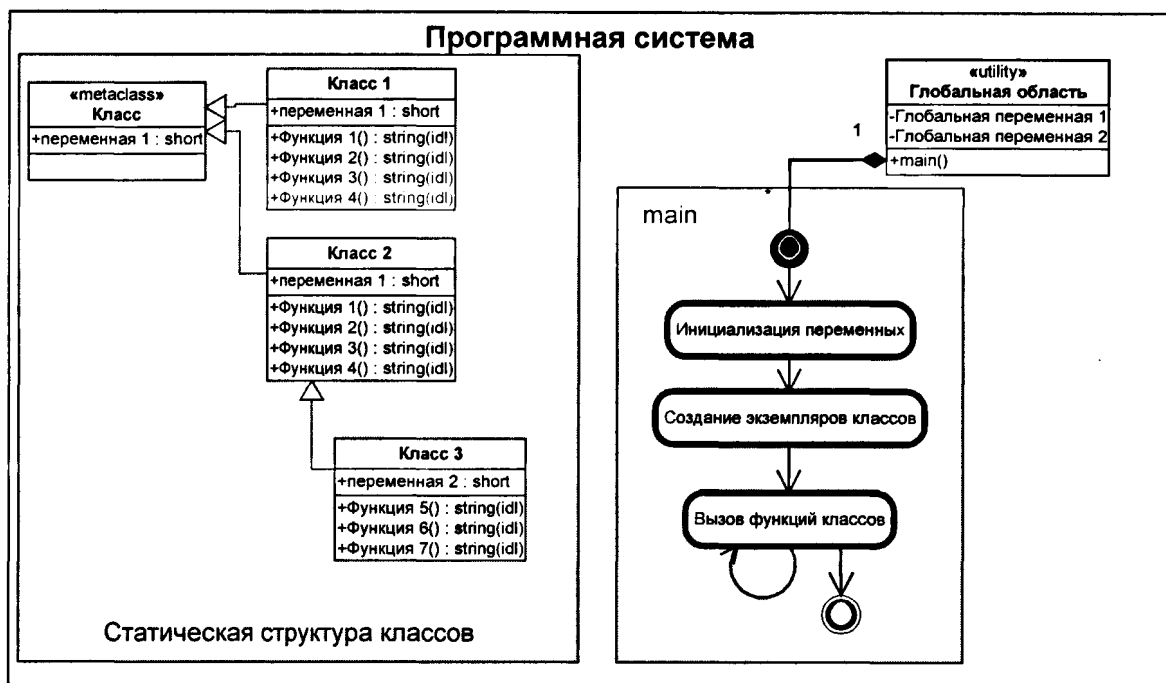


Рис. 1.4. Пример монолитной архитектуры приложения.

Полученное приложение является отдельной исполняемой операционной системой единицей, внесение изменений в которую возможно лишь путем повторной сборки и компоновки всего проекта.

В последнее время получила широкое распространение компонентная модель построения программного обеспечения, при которой вся программная система разбивается на ряд функционально законченных компонент, причем каждый такой компонент удовлетворяет определенным соглашениям (см. Рис. 1.5). Это дает возможность обеспечить априорную совместимость компонент, т.е. такую совместимость, при которой им не обязательно знать друг о друге что-то кроме того, что указано в соответствующем соглашении. Данное соглашение оформляется в виде интерфейсов, т.е. описаний функций использования. Это упрощает разделение задач между участниками разработки и в корне меняет процесс проектирования и реализации программного проекта.

Под компонентом, понимается некоторый законченный бинарный модуль, содержащий один или большее количество объектов. Отдельные компоненты являются независимыми исполняемыми единицами, разработанными с использованием различных инструментальных средств. Взаимодействие и управление независимыми компонентами обеспечивается на самом общем уровне средствами программной платформы, т.е. программным обеспечением ОС. При этом, сам компонент выступает отдельным элементом программной архитектуры системы, так как является в общем случае неделимым на более мелкие части, т.е. монолитным. Это позволяет рассматривать данные элементы в качестве

«черных ящиков», скрывающих особенности внутренней реализации, функциональность которых доступна извне посредством *интерфейсов*.

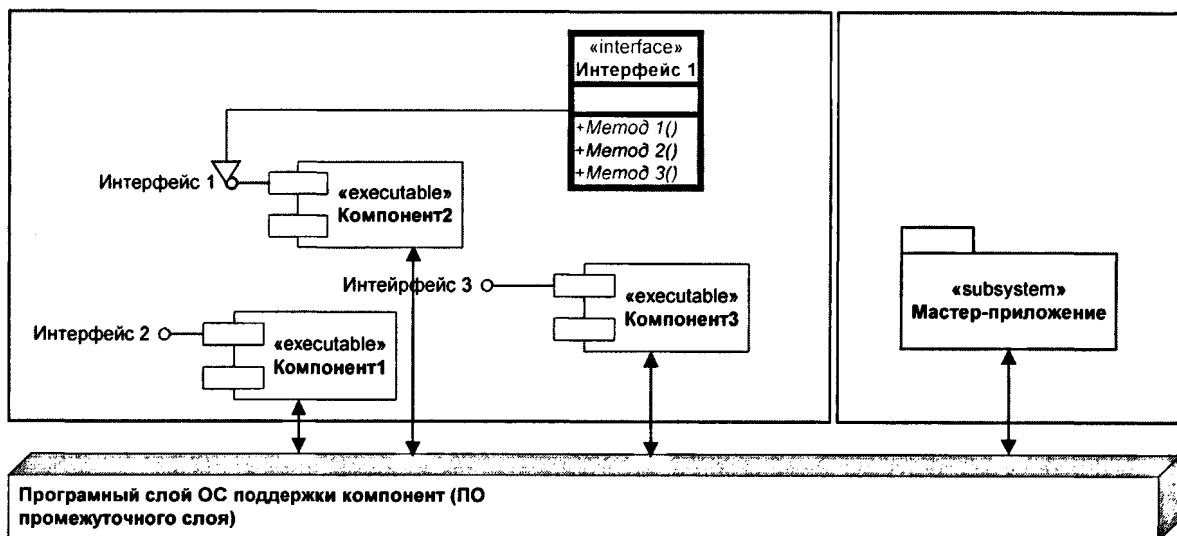


Рис. 1.5. Компонентная организация приложения.

Так называемое **промежуточное ПО**, или связующее программное обеспечение (от англ. *middleware*), решает задачи поддержки объектно-ориентированной компонентной системы, а также организации взаимодействия внутри нее. Под **интерфейсом** понимают строго типизированный, представленный на одном из языков описания, набор функций. **Объектом** является произвольная управляемая средствами **промежуточного программного обеспечения (ППО)** единица, соответствующая требованиям спецификации ППО. Промежуточное программное обеспечение выполняет посреднические функции для приложений, обеспечивая связующий интерфейс между приложением и сетью, между клиентской и серверной частью, а также возможность прозрачной работы приложений в неоднородной среде.

Таким образом, структура компонентной системы может быть рассмотрена с позиций нескольких программных архитектур: объектно-ориентированной ПА, так как любой компонент в конечном итоге представляется объектом, реализованным на любом языке высокого уровня, многозвенной, в случае если существуют условия разделения компонент по уровням абстракции, и, собственно, компонентной.

К настоящему времени сформировалось несколько устоявшихся компонентных архитектур, позволяющих решать определенные задачи в своих областях. Подобные архитектуры обеспечивают функционально законченный ИПП и набор базовых сервисов для функционирования и управления отдельными программными модулями. Области применения рассмотренных наиболее распространенных компонентных платформ представлены в [16].

Табл. 1.2. Сравнительная характеристика компонентных платформ.

Компонентная модель	Клиентские компоненты	Клиентские приложения	Серверные приложения	Серверные компоненты
COM+	+	+	+	+
JavaBean/RMI	+	+	+	+
CORBA	-	+	+	-

Получившая в настоящее время широкое распространение компонентная модель COM+ (Component object model), следующей фазой которой является реализация концепции языково-независимых ИПП в рамках проекта NET, была разработана компанией DEC в сотрудничестве с Microsoft corporation и в настоящее время активно развивается Microsoft [17,18,19]. COM+ является расширением хорошо известной архитектуры COM, разработанной в начале 90-х. Данная компонентная модель обеспечивает бинарную совместимость объектов, т.е. совместимость бинарного компилированного кода вне зависимости от языка программирования, который использовался для их написания. В той или иной форме подобная идея использовалась уже достаточно давно с появления C++ в Unix системах, а в Windows системах она вылилась в достаточно простую и прозрачную спецификацию модели COM (см. Приложение 1). Основной задачей, решаемой COM+, является обеспечение совместимости бинарных модулей, разработанных различными производителями в разное время и на разных языках программирования. Компонентная объектная модель определяет несколько основополагающих принципов, положенных в основу всей структуры COM:

- ◆ бинарный стандарт вызова функций между компонентами;
- ◆ обеспечение строго типизированного объединения функций в интерфейсы;
- ◆ обязательное присутствие базового интерфейса, обеспечивающего возможность доступа к другим доступным интерфейсам, а также подсчет ссылок на данный компонент, чтобы отслеживать его использование и динамически удалять, когда число ссылок становится равным нулю;
- ◆ механизм уникальной идентификации компонент и поддерживаемых ими интерфейсов.

Компонентная платформа JavaBean/RMI опирается, главным образом, на концепцию языка Java и существенно расширяет стандартную функциональность, обеспечивая возможность удаленной работы с интерфейсами компонент [20].

CORBA (Common Object Request Broker Architecture) является одной из наиболее распространенных платформ на целом множестве

разнообразных ОС и обеспечивает богатство разнообразных сервисов, характерных для сред промежуточного ПО [21]. CORBA, опираясь на так называемую технологию брокера объектных запросов, обеспечивающего взаимодействие удаленных объектов, сосредотачивается на задачах управления компонентами в распределенной среде. CORBA не обеспечивает поддержку собственных клиентских компонентов, опираясь в этом аспекте на технологию JavaBean. Таким образом, архитектура COM+ с точки зрения компонент конкурирует с технологией JavaBean, а с точки зрения распределенных решений – с технологией CORBA.

Все представленные выше компонентные платформы имеет хорошо развитую архитектуру, общую номенклатуру и терминологию, проработанный набор средств поддержки разработки и отладки компонент и приложений.

Наряду с объектно-ориентированными компонентными средствами ППО такими как CORBA, COM+, существует множество разнообразных систем посреднического ПО, основанных на промежуточных текстовых представлениях, в частности XML, которые не будут рассматриваться в дальнейшем в данной работе.

Обобщенное представление компонентной модели объектно-ориентированной ППО, состоящей из нескольких уровней, представлено на Рис. 1.6.

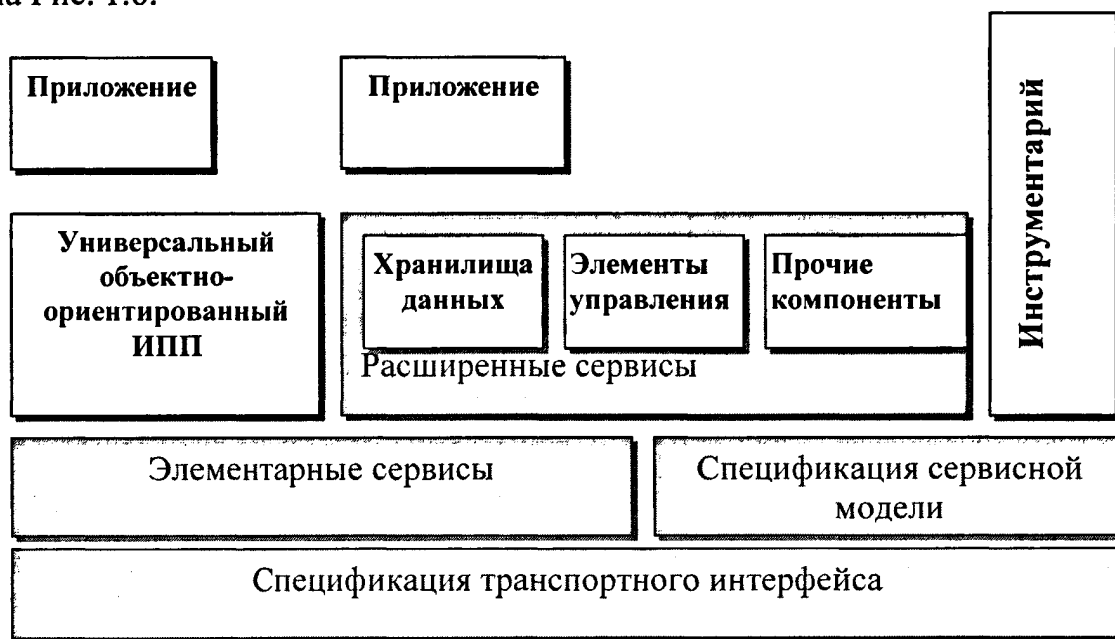


Рис. 1.6. Обобщенное представление объектно-ориентированной среды ППО.

Спецификация транспортного интерфейса и элементарные сервисы базового уровня составляют основу любой системы промежуточного ПО. Клиентские приложения, являющиеся сервисами высшего уровня, могут

быть разработаны на основе универсального ИПП или библиотек, предоставляемых средствами разработки.

Таким образом, могут быть сформулированы основные преимущества компонентно-ориентированной разработки на основе сред посреднического ПО:

1. Объектно-ориентированная методология в рамках UML получает мощную прикладную область применения.
2. Сокращается цикл разработки и проектирования. Сборка систем из отдельных компонент является более эффективным процессом, чем разработка монолитного приложения.
3. Компонентное программирование позволяет получать программные продукты более высокого качества.
4. Цикл разработки компонентных систем можно формализовать на основе универсального подхода.
5. В целом, затраты на разработку и сопровождение систем падают.
6. Разработка становится языково-независимой. Компоненты, разработанные в различных средах, могут быть использованы в рамках одного продукта.

Сравнительная характеристика основных компонентных платформ с точки зрения применимости в медицинской области приведена в [22].

1.2.3 Применение архитектур промежуточного ПО при проектировании программных биомедицинских систем

Развитие интеграционных стандартов в области биомедицинских программных систем сделало внедрение систем посреднического ПО неизбежным, благодаря их способности обеспечивать прозрачную интеграцию на принципах *plug-and-play* в гетерогенной среде разнообразных ОС. В настоящее время можно выделить несколько сфер применения архитектур ППО.

1. Наиболее распространенным подходом является разработка медицинских информационных систем на основе концепции сообщений стандартов обмена медицинскими записями, в частности Health Level 7, стандартов ASTM и EDIFACT при использовании объектной методологии промежуточного ПО [23]. Характерной реализацией является Европейский проект HELIOS[24], представляющий собой некоторую компонентную систему поддержки медицинской объектно-ориентированной БД, интеграция элементов которой осуществляется посредством обмена EDIFACT-сообщениями [25]. Данный подход предусматривает использование медиатора – сервиса-посредника, обеспечивающего передачу и хранение данных в неизменном формате. Это обеспечивает единство представления данных и позволяет интегрировать разнородные подсистемы, построенные на основе различных технологий

(DCOM/CORBA) и выступающие по отношению к медиатору в качестве агентов [26]. Сервисы-посредники могут использовать разнообразные представления данных, например, формат ASCII или XML [27]. В настоящее время доступно несколько реализаций объектно-ориентированного ИПП на основе стандарта сообщений HL7. Они основаны на получении описаний интерфейсов, исходя из правил структурно и семантически соответствующих реферативной информационной модели HL7. Подобным образом может быть получено универсальное объектное представление сообщений на уровне приложения, что позволяет работать с ними как с отдельными структурными единицами вне зависимости от используемого языка программирования и среды исполнения. Посредством сервиса-посредника объектные сообщения могут быть преобразованы в ASCII или XML-представления согласно стандарту HL7 и переданы через существующий транспортный интерфейс, которым обычно выступает протокол TCP/IP [28]. Сходным решением является открытая архитектура SynEx, поддерживаемая одноименным консорциумом, интегрирующая результаты нескольких Европейских проектов и стандартов, в частности, ENV-12967, который регламентирует архитектуру медицинских информационных систем [29,30]. SynEx состоит из набора компонент поддержки открытого стандарта консолидированных записей, являющегося развитием европейского стандарта ENV-12265 EHRC.

2. Достаточно распространенным приложением методологии ППО является реализация медицинских информационных систем полностью в рамках модели выбранной архитектуры ПО. Данный подход предусматривает концентрацию на задачах разработки гибкого программного интерфейса, а не интегрированного представления биомедицинских записей. В этом случае взаимодействие компонент системы в распределенной среде достигается за счет стандартизации интерфейсной части отдельных модулей, а основную нагрузку по поддержке информационного обмена несет среда промежуточного ПО. Задача разработки подобного специализированного промежуточного уровня программного обеспечения и форм ее решения сформулированы рядом авторов [31,32,33]. Наиболее известным на сегодняшний день проектом является программа CORBAMed, реализуемая консорциумом OMG, который занимается поддержкой стандарта CORBA в целом. Аналогичные решения доступны и на основе платформы DCOM [34,35,36,37]. Также доступны реализации систем на основе концепции стандарта ENV 12967-1 HISA, которая дает возможность последовательно конструировать медицинскую информационную систему, не меняя ее архитектуру по мере интеграции новых подсистем. Дополнительным развитием данной концепции является доработка стандарта ENV 12967-1 путем расширения функциональности Объектного Брокера Запросов (ОБЗ)

[38], не зависящего от выбранной распределенной платформы (DCOM\CORBA\HTTP).

3. В дополнение к уже указанным приложениям методов ППО не менее актуальной является реализация разнообразных медицинских приложений: интеллектуальных систем мониторинга, систем интенсивной терапии, а также комплексов регистрации и математической обработки физиологических сигналов. Задача разработки гибкого решения, позволяющего эффективно интегрировать разнообразные устройства регистрации сигналов, организовать транспортные потоки данных к АРМ врача, стоит достаточно жестко.

Решения данной задачи, которые, в основном, ориентированы на разрешение проблемы поиска эффективного механизма обмена данными в рамках расширенной локальной сети, опираются на систему *программных агентов*. Под **программным агентом** понимается некоторое приложение, соответствующее единым для данной конкретной системы интерфейсным нормам, которое реализует необходимые функции, например, функции источника данных. В этом случае, область *plug-and-play* взаимодействия разнообразных компонент ограничивается конкретной системой, для которой они разработаны.

Еще одним аспектом поставленной выше задачи является разработка эффективных систем регистрации биомедицинских сигналов, обеспечивающих возможность интеграции новых источников сигналов и эффективного доступа к унифицированным БД медицинского учреждения. Такие системы, в свою очередь, должны обеспечивать интеграцию в рамках медицинской информационной системы медицинского учреждения.

Важной стороной поставленной задачи является учет требований реального времени, соблюдение которых для систем регистрации и обработки сигналов необходимо.

В настоящий момент наметилась тенденция отказа от частных решений в пользу концентрации усилий на реализациях, опирающихся на общепринятые стандарты. Это, прежде всего, объясняется тем, что стандартизация в области подобных систем подошла к уровню, достаточному для начала работ по коммерческой реализации.

Для организации взаимодействия между различными устройствами регистрации медицинских сигналов и клиническими информационными системами необходимо использовать единый коммуникационный стандарт, удовлетворяющий ряду требований, наиболее важными из которых являются следующие:

- *уникальность имени*. Устройство должно иметь идентификатор, позволяющий однозначно выделить его в сети.
- *надежность*. Стандарт должен обеспечивать надежную работу в случае отказа одного или нескольких устройств.

- *совместимость с производимым оборудованием.* Технология реализации стандарта должна позволять использовать имеющуюся аппаратно-программную реализацию настолько, насколько это возможно.

- *поддержка множества физических интерфейсов.* Обеспечение возможности устанавливать соединение, используя множество доступных протоколов.

- *поддержка множества сетевых топологий.* Поддержка прозрачности доступа к компонентам системы (устройствам, системе контроля) из внешних сетей.

- *синхронизация по времени.* Обеспечение синхронизации передаваемых данных по времени.

- *совместимость с клиническими информационными системами.* Должна быть обеспечена возможность передачи данных в информационную среду Health Level-7, ENCR или прочего стандарта обмена медицинскими записями.

- *защищенность.* Интерфейс должен обеспечивать защищенность и конфиденциальность передаваемых данных.

- *масштабируемость.* Архитектура должна позволять изменять количество элементов системы.

Усилия по координации работ групп IEEE по стандарту MIB и комитета TC251 Европейского института по стандартизации говорит в пользу единого семейства стандартов, которые уже в ближайшем будущем, возможно, будут приняты ISO в качестве полноценных норм.

Глава 2. РАСПРЕДЕЛЕННАЯ КОМПОНЕНТНО-ОРИЕНТИРОВАННАЯ ПРОГРАММНАЯ АРХИТЕКТУРА ДЛЯ КОМПЬЮТЕРНЫХ БИМЕДИЦИНСКИХ СИСТЕМ РЕГИСТРАЦИИ ЭЛЕКТРОФИЗИОЛОГИЧЕСКОЙ ИНФОРМАЦИИ

2.1. Основные этапы исследования

Проектирование любой программной системы включает в себя целый комплекс вопросов, связанных с выбором методологии проектирования, разработки программной архитектуры и выборе программной платформы, применения методов оценки качества программных приложений, гибкости реализации и тестирования функционального содержания, формирования требований рекомендательного характера, ориентированных на повышение качества продукта. Важным этапом проектирования является исследование структуры (структуризации) подобного класса систем с целью выявления общих элементов структуры, выработки единых требований к реализации. Это дает возможность подойти к разработке с позиций универсального подхода, дающего возможность использовать единые методы для построения целого класса систем. Поэтапная структура исследования, включая первый этап, состоящий в изучении проблемы, формулирования целей и задач исследования, рассмотренный в предыдущих главах, представлена на Рис 2.1.

Основным методом структуризации в данном исследовании явился системный метод, являющийся основой теории системного анализа.

В системно-структурном методе акцент преимущественно делается на исследование внутренней структуры системы, внутренних связей, тогда как системный подход, кроме этого, подразумевает исследование поведения системы и ее элементов в зависимости от связей с внешней средой, от происхождения ситуации, в которую попадает система. В данном случае, в понятие структура (от латинского «structure», означающего строение, расположение, порядок) вкладывается смысл отображения определенных взаимосвязей, взаиморасположение составных частей системы, ее устройства (строения). Важной чертой системно-структурного метода является то, что он наряду с рассмотрением структурных связей рассматривает и структуру функций системы (структурно-функциональный аспект метода).

Методы системного анализа являются основой так называемого Рационального унифицированного процесса (РУП) разработки технических систем, в частности программного обеспечения [39] (этап 2). РУП регламентирует процессы проектирования ПО, от этапа создания и сопровождения моделей системы до фазы тестирования и внедрения.

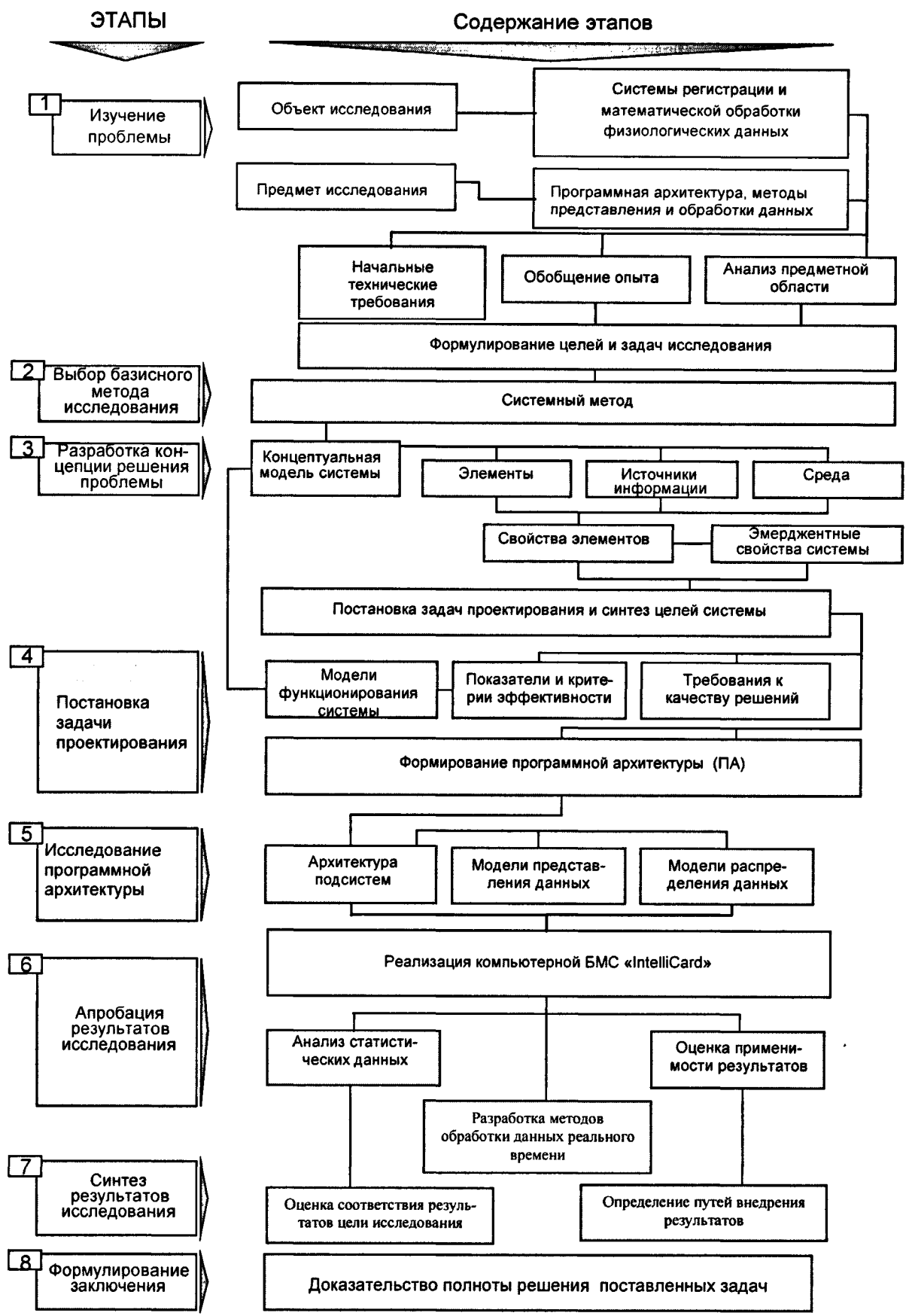


Рис. 2.1. Структурная схема проведения исследований.

В качестве метода формализованного представления систем (МФПС) в рамках данной методологии используются лингвистические, графические представления, в частности, язык моделирования UML[40].

Предварительный, поверхностный анализ систем требует их представления в виде схем, в которых фиксируются наиболее существенные параметры и связи между ними. При этом элементы данных схем не подвергаются какой-либо детализации до уровня конкретных используемых технологий. Такие схемы получили название концептуальных моделей, или структурных моделей [41], и являются отправной точкой исследования (этап 3).

Основными параметрами концептуальных моделей считаются элементы системы, источники информации, внешняя среда, каждый из которых характеризуется своими свойствами, а система – свойствами целостности (эмерджентными свойствами). Эти свойства формируют совокупность факторов, влияющих на поведение системы. Важнейшим этапом исследования является переход от разработанных структурных представлений о системе, к программной архитектуре (этап 4) и ее дальнейшей реализации. Архитектурные решения должны приниматься только тогда, когда стала ясна структура системы в целом, т.е. полностью сформулированы требования, предъявляемые к системе [42].

Для решения поставленных задач необходимо обладать определенным набором приемов решения, среди которых наиболее результативны модели функционирования системы, среди которых стоит выделить модели процессов, формализующих механизмы параллелизма и синхронизации, представления и накопления данных, а также модель развертывания, отражающую топологию используемых аппаратных средств. Модель реализации служит заключительным представлением, реализующим покомпонентную конфигурацию системы (этап 4).

На основании полученных результатов выносится общее суждение о функционировании исследуемой программной системы, выявляются закономерности, присущие системе.

Путем сопоставления полученных результатов при моделировании и фактических данных обобщенно оценивается степень применимости модельных характеристик, а затем определяется их соответствие цели исследования и выдвинутым требованиям, т.е. решается задача синтеза исследования (этап 7), позволяющая обосновать направления и способы реализации результатов.

Завершающим этапом методики исследования (этап 8) является доказательство полноты достижения цели исследования, которое приводит к формулированию заключения по теме.

2.2. Разработка концепции решения задач

2.2.1 Структуризация биомедицинских систем регистрации электрофизиологической информации

Особенностью цифровых компьютерных комплексов регистрации электрофизиологической информации, осуществляющих запись и обработку данных непосредственно во время записи сигнала, является принадлежность к системам реального времени, так как функциональная часть этих систем должна непрерывно обрабатывать и анализировать поступающую информацию и оперативно реагировать на изменившееся состояние внешних параметров. Так, например, для полиграфических систем регистрации электрофизиологической информации такими сигналами могут являться ЭКГ, ЭЭГ, ЭМГ и ЭОГ, а также целый спектр физиологических сигналов, приведенный в Табл. 1.1. Исследовать структуру подобных аппаратно-программных систем можно с использованием двух подходов: информационного и функционального. Информационный подход подразумевает анализ на основе рассмотрения транспортных потоков данных, циркулирующих между элементами системы, в то время как функциональный основывается на рассмотрении разнообразных связей внутри комплексной системы, выделяя в отдельные единицы функционально законченные компоненты. С точки зрения формализации наиболее проработанным является информационный подход, поскольку он предоставляет возможность количественно и качественно проанализировать и представить структуру связей на основе математических методов. Функциональная схема отражает структуру в виде множества законченных элементов, реализующих определенные функции в рамках требований, предъявляемых к комплексам ЭКГ-диагностики, а также предоставляющих различные дополнительные диагностические или сервисные возможности. Детализация функциональной схемы подразумевает учет конкретной программной платформы (ПП), на базе которой строится реализация, накладывающей свой отпечаток на структуру связей и набор функциональных компонент, присутствующих в структурной схеме. Более того, цели и задачи самой биомедицинской системы также играют важную роль. Например, функциональная схема компьютерных систем ЭКГ-диагностики может существенно различаться по структуре (рутинная ЭКГ, ЭКГ физических нагрузок, системы прикроватного мониторинга) и иметь четко выраженную близость с используемым аппаратным обеспечением. Все дальнейшее рассмотрение будет касаться систем, ориентированных на получение и обработку БМИ в режиме реального времени.

На Рис. 2.2 представлена структуризация полиграфической биомедицинской системы регистрации и обработки электрофизиологической информации, включающая основные элементы,

отвечающие общим требованиям, предъявляемым к системам такого рода. Данная схема дает только общие представления о структуре систем и не отражает функциональную логику связей между отдельными его элементами в рамках каждой подсистемы. Выделение независимых подсистем заключается в выделении отдельных элементов с сильными функциональными взаимными связями. Для анализа функциональных связей отдельных подсистем необходим детальный анализ структуры подобных систем.

Структурно, произвольная компьютерная БМС может быть разделена на 4 уровня: уровень сопряжения с устройствами регистрации, уровень обработки БМИ, представления и коммуникационных интерфейсов.

Для обеспечения доступа к устройствам в большинстве современных ОС, в том числе и семейства Windows, необходимо наличие драйвера, чтобы обеспечить полный доступ к аппаратному обеспечению на уровне ядра ОС. Множество поддерживаемых аппаратных интерфейсов для подключения внешних устройств в настоящее время является достаточно широким, поэтому использование шин ISA/PCI для обеспечения высокоскоростного обмена не является необходимым. Среди поддерживаемых производителями портативных ПЭВМ интерфейсов, необходимо выделить USB, IrDA, Wi-Fi (стандарт IEEE 802.11b), BlueTooth, как наиболее перспективные для сопряжения с устройствами регистрации БМИ, поддерживающие скорости передачи от нескольких Мбит/с, до десятков Мбит/с.

Таким образом, программный слой работы с оборудованием составляют компоненты-контроллеры, обеспечивающие прозрачный интерфейс доступа к соответствующим драйверам устройств, которые могут быть стандартными в случае использования поддерживаемых ОС интерфейсов, или специально разработанными для данного устройства при использовании стандартов ISA/PCI/PCMCIA. В предметную область стандарта IEEE 1073 контроллеры устройств внесены под наименованием виртуального медицинского устройства (ВМУ), так как с точки зрения БМС данный компонент является абстрактным представлением реального аппаратного устройства. Для обеспечения расширенной функциональности в подсистему управления устройствами регистрации может быть добавлен менеджер устройств, который позволяет пользовательскому процессу динамически подключаться к различным устройствам, сохраняя их параметры и обеспечивая стандартный интерфейс доступа. В качестве подобных устройств, например, в комплексе полнофункциональной диагностики ЭКГ могут выступать, собственно, регистратор ЭКГ, а также пульсовой оксиметр и монитор артериального давления (АД).

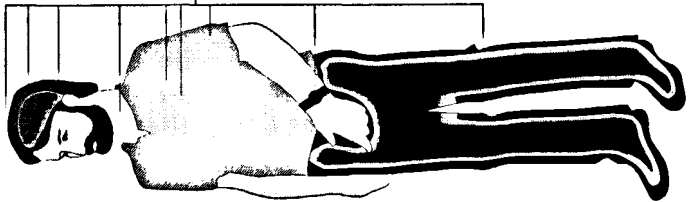
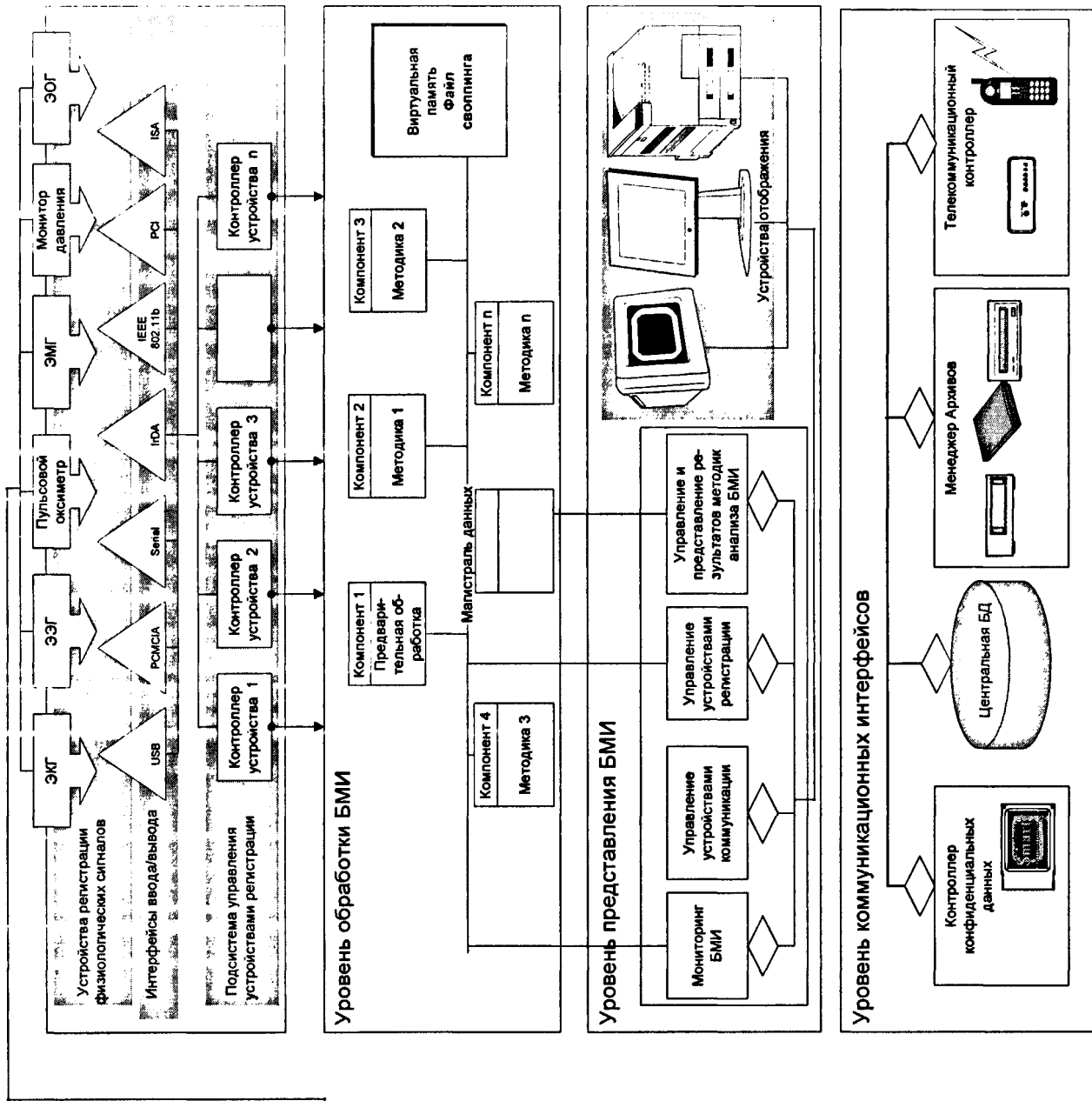


Рис. 2.2. Концептуальная модель полиграфической системы регистрации и анализа БМИ.

В целом, архитектура системы должна обеспечивать возможность подключения различных устройств в зависимости от назначения и требований диагностических методов, закладываемых в систему на этапе разработки или в процессе ее совершенствования.

Наиболее важным уровнем является уровень обработки данных, который составляют компоненты, выполняющие обработку данных, как в режиме реального времени, так и в автономном режиме. Здесь наибольшую важность приобретает возможность динамического изменения конфигурации обработчиков данных, добавление и удаление компонент во время работы. Это позволяет эффективно изменять заложенную на этапе проектирования системы функциональность в соответствии с новыми требованиями.

Уровень представления БМИ представлен компонентами, осуществляющими визуализацию БМИ и результатов анализа, а также пользовательский интерфейс управления различными подсистемами.

Выделение коммуникационных модулей в отдельный уровень обязано своим существованием быстрому развитию средств удаленной передачи и общим интеграционным процессам в здравоохранении. Коммуникационная подсистема может обеспечивать интеграцию БМС с централизованной БД медицинского учреждения (например, на основе стандарта HL7), сохранение и ведение архивов медицинских данных, поддержку телекоммуникационных функций, что особенно важно для портативных систем. При этом возможно широкое применение разнообразных доступных средств передачи данных [43]. Поддержка смарт-карт позволяет обеспечивать конфиденциальность доступа к БД и защищенное управление данными пациентов [44,45].

2.2.2 Показатели и критерии эффективности

Для сложных аппаратно-программных комплексов является весьма затруднительным выработка какого-либо единого совокупного критерия из-за проблем компенсации негативных показателей позитивными, поэтому оценка эффективности может быть проведена на основании исследования нескольких групп частных показателей, вес которых может быть различным в каждом конкретном случае.

В качестве таких групп были выбраны следующие множества частных показателей:

Клинические показатели характеризуют эффективность и степень соответствия функциональных элементов (медицинских методик обработки и анализа, качество аналого-цифрового преобразования медицинских сигналов и т.д.) установленным требованиям и могут являться в каждом конкретном случае различными значениями. Примером таких величин

является, в частности, чувствительность алгоритма, отношение сигнал/шум и т.д.

Технологические показатели характеризуют качество программных и аппаратных решений с общих позиций современной технологической базы. Для систем реального времени такими показателями могут быть максимальная пропускная способность, коэффициенты использования ЦП, оперативность отображения информации, надежность.

Структурно-топологические показатели являются количественными оценками иерархической топологии компонентной программной системы и позволяют характеризовать связность, структурную избыточность, степень централизации, а также ранг каждого из элементов архитектуры.

Экономические показатели характеризуют затраты, связанные с обеспечением требуемых функциональных характеристик, которые складываются из стоимости разработки ПО и минимально необходимой конфигурации аппаратных средств.

2.3. Разработка основных моделей компонентной архитектуры БМС

Основной задачей проектирования любой системы обработки данных реального времени, к которым относятся и компьютерные БМС сбора и обработки электрофизиологических сигналов реального времени, является разработка и реализация методов обработки информации. Эта задача может быть разделена на несколько этапов:

- а) определение задач обработки данных, выделение независимых процессов, решение задачи планирования параллельно выполняемых задач;
- б) проектирование магистрали данных, под которым понимается определение дисциплины доступа компонент к данным, физические модели представления и механизмы передачи данных;
- в) разработка и тестирование реализаций различных методов обработки данных.

2.3.1 Модель организации вычислительных процессов

Важнейшей составной частью любой системы обработки данных реального времени является подсистема распределения данных, обеспечивающая передачу необходимых данных всем компонентам системы.

Проблема планирования вычислительных задач возникает, когда есть необходимость разделения ограниченных ресурсов (ресурсов ЦП, постоянной памяти, устройств ввода/вывода) между несколькими задачами. Цель решения проблемы планирования – выделение в системе процессов, использующих в режиме разделения времени ресурсы системы, и разработка алгоритма обеспечения доступа параллельно выполняемых

процессов к этим ресурсам. Выделение параллельных задач не является самоцелью, последовательное выполнение задач при прочих равных условиях является более производительной организацией вычислений. Анализ концептуальной модели системы на предмет параллелизма процессов имеет смысл только в случаях недостатка вычислительных ресурсов [46]. Примером может служить организация мультипроцессорных или распределенных вычислений. Данная задача изучена к настоящему времени достаточно хорошо и известно, что в общем случае она имеет NP-сложность [47].

Поставленную задачу можно решить на основании анализа моделей процессов системы, формализующих ее функционирование в периоды максимальной вычислительной нагрузки, т.е. во время регистрации или обработки данных в реальном времени.

Разработка подобной модели требует определения параметров моделирования, в том числе характеристик планировщика задач, на основе анализа его реализации в каждой конкретной ОС, а также критериев эффективности, по которым можно было бы судить о целесообразности той или иной реализации.

Операционные системы семейства Windows NT имеют реализацию вытесняющего планировщика задач, обеспечивающего привилегированное исполнение задач с более высоким приоритетом. Каждое исполняемое приложение, являющееся с точки зрения ОС отдельным процессом с определенным набором параметров (объемом виртуальной памяти, приоритетом выполнения, административными правами), имеет единственный поток выполнения, создаваемый по умолчанию средствами ОС (от англ. *thread*). В терминах ОС каждый такой поток выполнения является отдельно выполняемой задачей, которой соответствует собственный стек вызовов функций, копия регистров ЦП, локальная область хранения переменных. Граф переходов состояний задач в Windows NT/2000/XP представлен на Рис. 2.3, подобная реализация свойственна и многим Unix-системам.

Таким образом, задача i может находиться в одном из следующих состояний:

WS_i —ожидает запуска;

AC_i —активное состояние, ЦП осуществляет выполнение задачи;

AW_i —ожидание в активном состоянии завершения обработки операции в ядре ОС или задачи более высокого приоритета;

SU_i —задача приостановлена на неограниченное время;

WA_i —ожидание определенного события, в том числе, завершения задачи;

SL_i —нахождение в «спящем», неактивном состоянии заданный интервал времени;

ST_i —удаление задачи из списка планировщика задач ОС.

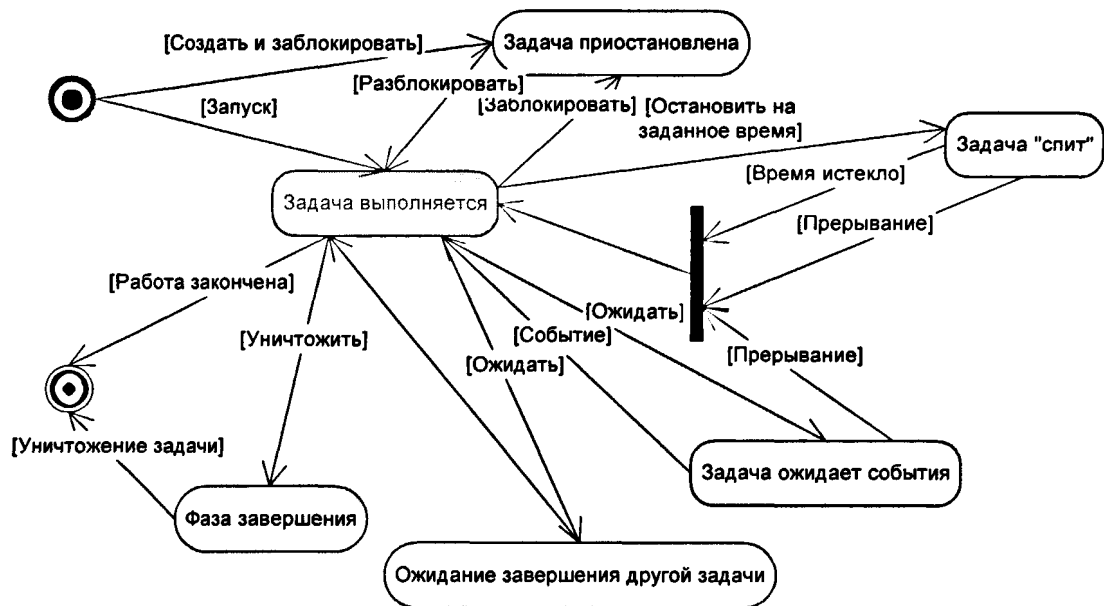


Рис. 2.3. Граф переходов состояний задачи в Windows NT.

Важнейшим элементом модели является алгоритм работы планировщика задач ОС, управляющего разделением ресурсов ЦП. Однако построить точную модель планировщика достаточно сложно, поскольку алгоритм его работы тесно связан с процессами, протекающими в ядре ОС, включая операции ввода-вывода с разнообразными устройствами. Поэтому необходимо пользоваться общими принципами алгоритма планирования процессов, которые заключаются в следующем:

- 1) В Windows NT/2000 каждый вычислительный процесс имеет свой приоритет $P_{\text{проц}}^i$ (до 4 градаций), определяющий базовый приоритет каждой вычислительной задачи P_{task}^i , который в свою очередь имеет до 7 градаций.
- 2) Максимально возможный квант времени ΔT , выделяемый одной задаче до переключения ЦП на выполнение других задач с таким же приоритетом, ограничен фиксированным значением, составляющим в зависимости от версии ядра ОС Windows NT/Windows 2000/XP и аппаратной платформы до нескольких мс.
- 3) В случае выхода из состояния ожидания задач с более высоким приоритетом, задачи с более низким приоритетом переходят в состояние AW.
- 4) Планировщик поддерживает инверсию приоритета, позволяя менее приоритетной задаче завершить выполнение в случае, если она использует разделяемые ресурсы системы.

Важнейшими величинами, характеризующими выполняемые задачи, являются C_{task} -время ЦП, необходимое для одного запуска задачи, T_{task} -период выполнения (частота запуска для периодических задач). Коэффициент использования ЦП задачей i составляет в данном случае

$$U_i = C_{task}^i / T_{task}^i$$

Периодические задачи, завершающиеся до истечения периода выполнения, называются планируемыми [46].

Наиболее предпочтительным вариантом перехода от последовательной архитектуры выполнения задач к параллельной — является рассмотрение модели информационных потоков данных, развитие которой началось вместе с первыми

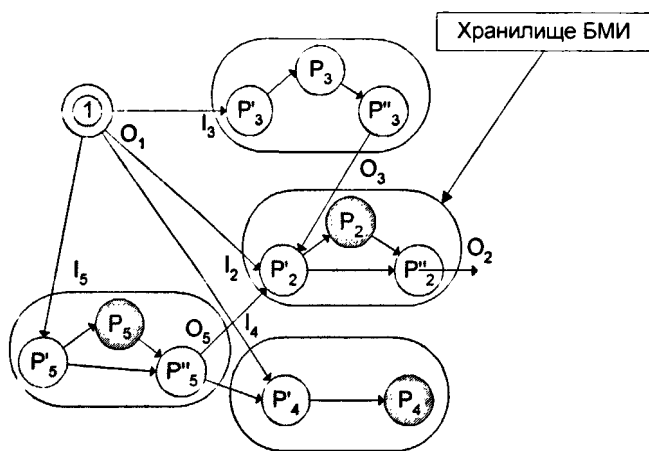


Рис. 2.4. Информационный граф потоков данных.

вычислительными архитектурами [48]. Частным случаем данного подхода является модель синхронных информационных потоков (англ. *Synchronous Data Flow*). Модель синхронных информационных потоков для формализации систем цифровой обработки сигналов известна достаточно давно [49]. Особенностью данной модели является то, что характеристики информационных потоков (количество каналов, частота дискретизации, и т.д.) в системах ЦОС могут быть определены заранее. Это позволяет свести задачу планирования к статической, т.е. решаемой на этапе проектирования потоков передачи данных [50]. Данная модель является наиболее естественным представлением аппаратно-программных систем регистрации биомедицинских сигналов с источниками БМИ с заданной частотой дискретизации. Информационная модель состоит из двух типов элементов — узлов, имеющих входные и выходные порты для передачи данных заранее заданного типа, а также связей коммутации узлов друг с другом. Модель синхронных потоков данных может быть эффективно использована для выделения в системе параллелизма [51]. В схеме, представленной на Рис. 2.4, существует единственный источник данных, а процессы обработки данных условно разнесены по функциональным модулям, в каждом из которых в отдельные процессы вынесены процедуры получения данных через коммуникационный интерфейс и преобразования их во внутренне представление P'_n и обратно P''_n , а также основной процесс обработки данных P_n . Параметры синхронной сети — количество обрабатываемых каждым процессом P_n отсчетов (входящих I_n и исходящих O_n) — считаются заданными на этапе проектирования.

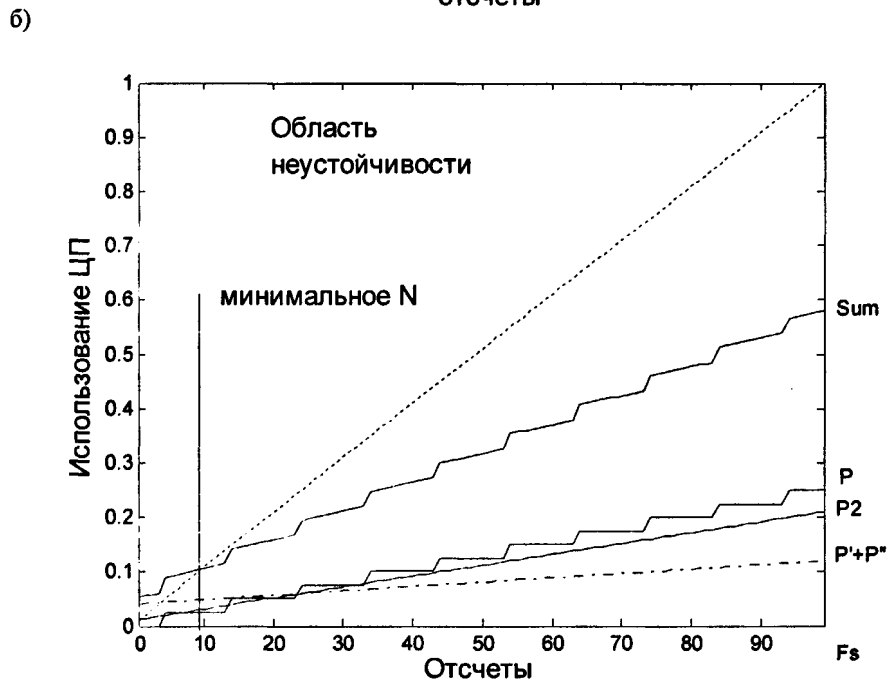
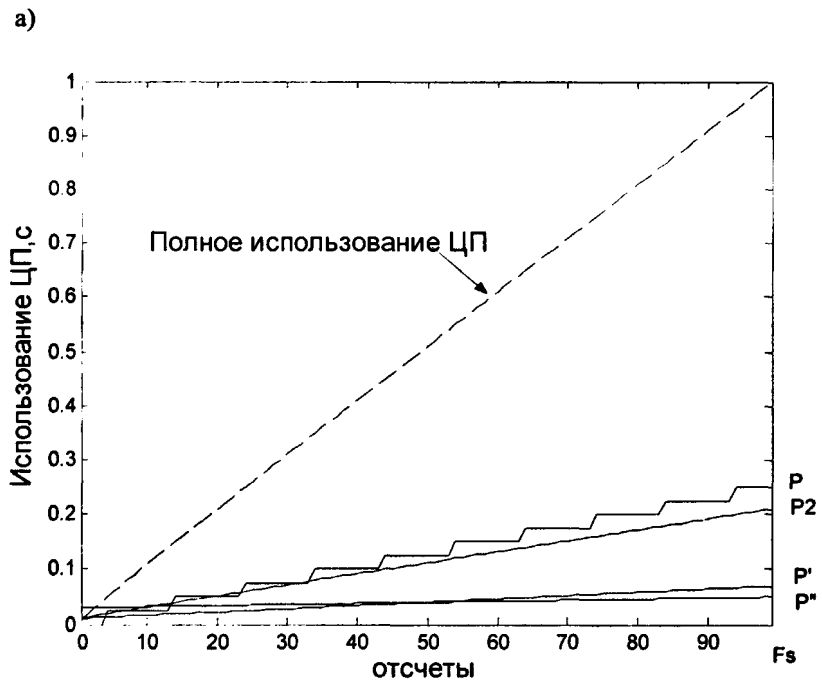


Рис. 2.5. Динамика времени ЦП, выделяемого процессам обработки данных.

Результатом решения задачи выделения параллельных процессов является граф, раскрашенный согласно принадлежности процессов обработки данных различным задачам ОС. В простейшей ситуации все процессы принадлежат одной задаче ОС, что соответствует приложению с одним потоком выполнения.

Затраты времени ЦП, выделенного на каждый процесс обработки данных с момента времени $t=0$, представлены на Рис. 2.5а. По оси абсцисс

отложено количество отсчетов, переданных источником данных 1. Суммарные затраты времени ЦП по сгруппированным процессам представлены на Рис. 2.5б. Анализ представленных зависимостей позволяет в каждом конкретном случае в зависимости от выбранной стратегии определить группы процессов, которые целесообразно сделать параллельными, выделив им различный приоритет выполнения. Наиболее оправданной является стратегия группировки в одну подгруппу задач с более коротким периодом, которая в нашем примере сводится к выделению процессов обслуживания распределения данных в сети компонент, каждый из которых может иметь независимый поток исполнения.

Статическое планирование процессов может быть осуществлено методом монотонных частот, предусматривающим присвоение более высокого приоритета задачам с меньшим периодом. И, соответственно, периодические задачи, полученные методом разбиения информационного графа, будут удовлетворять временным ограничениям при любой комбинации момента запуска тогда и только тогда [52]:

$$\forall i, 1 \leq i \leq n, \min \sum_{j=1}^i C_{task}^j \frac{1}{pT_{task}^k} \left[\frac{pT_{task}^k}{T_{task}^j} \right] \leq 1$$

$$(k, p) \in R_i, R_i = \left\{ (k, p) : 1 \leq k \leq i, p = 1, \dots, \left[\frac{T_{task}^i}{T_{task}^k} \right] \right\}$$
(2.1)

Соответствующую верхнюю грань коэффициента использования ЦП можно оценить следующим образом:

$$\sup U_i = \sum_{j \in H_j} \frac{C_{task}^j}{T_{task}^j} + \frac{1}{T_{task}^i} \left(C_{task}^i + B_{task}^i + \sum_{k \in H_i} C_{task}^k \right)$$
(2.2)

В данной формуле H_j - множество задач, вытесняющих задачу i , имеющую более низкий приоритет согласно алгоритму монотонных частот, H_j^* - множество более приоритетных задач, имеющих периоды длиннее, чем у i , B_{task}^i - время блокировки задачами с более низким приоритетом в случае инверсии приоритетов. Формула 2.2 с хорошей точностью позволяет учесть особенности вытесняющего планировщика задач Windows NT/2000/XP.

Итак, схема статических информационных потоков дает возможность эффективно осуществить разделение процесса обработки БМИ в программных биомедицинских системах реального времени на ряд параллельно выполняемых с разными периодами задач и обеспечить их статическое планирование. Однако к недостаткам такого подхода следует отнести невозможность обеспечить динамический контроль качества обслуживания (QoS) потребителей БМИ, ввиду статического характера планирования. Данная проблема может возникнуть, например, в случае

непредвиденного вытеснения задач биомедицинской системы посторонними процессами более высокого приоритета, что приведет к достижению ЦП 100%-загрузки. К значительным недостаткам следует отнести также невозможность обеспечения планирования случайных событий, например, вмешательства врача-специалиста и поддержки распределения БМИ, передаваемой аperiodическими устройствами, например, монитором артериального давления.

Расширение сетевой схемы движения данных может проводиться различными способами, в частности, наиболее проработанной является модель потоков и функций, заданных на них [53,54]. Подобный подход является наиболее естественным для систем ЦОС реального времени.

Решение поставленной задачи возможно также путем создания некоей гибридной модели сети информационных потоков, объединяющей свойства синхронных процессов и дискретных событий. Данная модель подразумевает рассмотрение общей информационной модели с двух позиций: микромоделей синхронных потоков и макромоделей событий [55]. Моделирование функционирования системы методом дискретных событий является чрезвычайно трудоемкой проблемой и решение проблемы планирования процессов, вообще говоря, является NP-полной задачей. Поэтому необходимо сохранить стабильность модели синхронных потоков данных для решения задач разбиения на параллельно выполняемые процессы и обеспечить системе динамические свойства.

В исходной концептуальной модели компьютерных систем обработки электрофизиологических данных можно выделить процессы, не имеющие входных данных и являющиеся лишь источниками сигналов. Таковыми являются процессы, реализующие ВМУ, обслуживающие аппаратные устройства регистрации биомедицинских сигналов и обеспечивающие непрерывность и защищенность потока данных. Подобные процессы обладают, безусловно, высоким приоритетом и конкурируют лишь между собой. Планирование таких, обычно периодических, процессов можно осуществить статически методом монотонных частот, при условии, что время выполнения таких задач значительно меньше любых других процессов системы $C \ll \min C^i$ и, следовательно, временем их выполнения можно пренебречь. Это соответствует действительности в случае, если подобные процессы не несут большой вычислительной нагрузки и заняты обслуживанием коммуникационного интерфейса. В данном контексте будем рассматривать оператора программной системы, как еще один источник недетерминированных событий и дальнейшее рассмотрение человеко-машинного интерфейса оставим за рамками данной работы [56].

При рассмотрении сетевой схемы процессов движения данных, подразумевается, что блоки, представляющие процессы, осуществляют вызовы передающей функции данных следующих в цепи элементов. При

этом задача планирования вырождается в задачу определения расписания вызовов, по возможности, статически, т.е. на этапе разработки. Модель синхронных информационных потоков позволяет найти решение этой задачи при любых условиях, для более обобщенных моделей это удастся сделать не всегда.

Таким образом, применение данного метода позволяет использовать статические методы определения параллелизма процессов в биомедицинских системах и разнообразные алгоритмы динамического планирования, в частности, метод иерархического планирования, позволяющий преодолеть трудности статического подхода – отсутствие динамического контроля качества обслуживания планировщиком ОС вычислительных задач. Так как для полиграфических биомедицинских систем реального времени является характерным наличие ограниченного числа источников событий, прежде всего, контроллеров устройств регистрации, то иерархия процессов, наделенных функцией планирования, может быть сведена к двум уровням: супервизора и задач, осуществляющих распределение данных от каждого устройства регистрации.

Будем рассматривать произвольное событие, как элемент множества $E_N = V^N \times T$, где N -количество реальных сигналов (потоков отсчетов), V^N -множество значений (v_1, v_2, \dots, v_n) , соответствующих сигналов, а множество временных интервалов $T = [0, \infty)$. Таким образом, в рамках данного подхода множество сигналов $S = \wp(E_1)$, т.е. является множеством всех подмножеств событий. S^N -множество наборов из N -сигналов $S^N = \{s_1, s_2, \dots, s_N\}$, $S^N = \wp(E_N)$. Подобным образом возможно связать набор из N сигналов с множеством событий. Каждое событие помечено некоторой временной меткой $t \in T$, $T(S^N) \subseteq T$ -множество временных меток набора исходных сигналов S^N . Таким образом, некоторая последовательность $S_d^N = \{(t, v_1, \dots, v_n) : t \in T, t_2 > t_1\}$ представляет собой последовательность событий с временной меткой t . Важным фактором является синхронизация отсчетов в каждом из сигналов с одной временной меткой t , что не является существенным ограничением для многоканальных систем регистрации биомедицинских сигналов.

Функционально произвольный процесс Φ в потоковой схеме передачи данных осуществляет преобразование входных данных $F: S^M \rightarrow S^N$, где M -количество входных сигналов, а N -выходных. В дискретно-событийной модели эквивалентный дискретный процесс Φ_d может быть описан в терминах преобразования F_d , которое в общем виде принадлежит множеству функций Γ :

$$\Gamma = \{ F : S^m \times T \rightarrow S^n \times T \} \quad (2.3)$$

Поток событий, описываемый последовательностью меток $s_n = \{t_1, t_2, t_3, \dots, t_n\}$, преобразовывается процессом Φ_d в аналогичный $s'_n = \{t_1, t_2, t_3, \dots,$

t_n }. В случае эквидистантных отсчетов для привязки события ко времени достаточно одной метки времени только для первой серии отсчетов. Поэтому событию E_t^n , помеченному временем t_n , соответствует $N = (t_{n+1} - t_n)F_s$ серий отсчетов. Следовательно, при $t_{n+1} - t_n = \Delta = \text{const}$ исходный поток превращается в стационарный и соответствует сети синхронных потоков данных. В простейшей ситуации период передачи Δ может быть задан фиксировано в каждом источнике данных в соответствии с интенсивностью обработки данных. В более сложных ситуациях необходимо решение задачи интервального планирования, т.е. определения следующего момента вызова функции передачи. Данная задача может быть решена методами планирования без вытеснения процессов.

2.3.2 Структурная модель данных

Внутреннее представление данных и объектов хранения играет определяющую роль во всех системах обработки данных в целом, а в системах и устройствах цифровой обработки сигналов (ЦОС) имеет принципиальное значение. Это объясняется тем, что системы ЦОС предъявляют дополнительные требования к форме представления данных многоканальных и многомерных сигналов [57]. Реализация методов ЦОС включает в себя несколько этапов, среди которых следует выделить несколько: разработка алгоритма ЦОС, программная реализация и отладка, выполнение алгоритма процессором. При этом каждому этапу соответствуют разные модели обработки данных: первому свойственна математическая, второму - программная, а третьему - физическая модель, рассматривающая представление данных на уровне регистров и ячеек памяти процессора. При этом большое значение имеет тот факт, что представление данных на уровне программной модели однозначно определяет представление на физическом уровне, обеспечиваемом набором аппаратно-программных средств разработки для той или иной программной платформы. Однозначное соответствие обеспечивается, прежде всего, механизмами типизацией данных, наиболее важными типами являются целый, вещественный, комплексный, логический и строковый. С точки зрения ЦОС различными видами данных, возникающих в процессе обработки, являются:

- 1) *вещественный тип* - последовательности отсчетов, значения импульсных и частотных характеристик;
- 2) *целый тип* - номера и количества отсчетов;
- 3) *логический тип* - результаты битовых операций.

Для систем регистрации и обработки медицинских данных согласно структурной модели (см. 2.2.1) исходными данными являются электрофизиологические сигналы, содержащие дискретизованные по

амплитуде отсчеты. Качество исходных данных определяется аналоговой частью системы и, прежде всего, АЦП. Для реализации эффективных процедур ЦОС необходим оптимальный выбор внутреннего представления именно вещественного типа данных. Причем выбранный тип данных должен подходить и для представления разнообразных промежуточных данных, возникающих в процессе реализации методов обработки. С точки зрения представления вещественных данных существуют 2 устойчивые альтернативы: тип вещественных данных с плавающей точкой (ПТ) и фиксированной точкой (ФТ). Для определения преимуществ той или иной формы представления данных для конкретной задачи необходимо проведение дополнительных исследований заключающихся в сравнении ряда характеристических величин. Важным является тот факт, что входными данными любой подсистемы обработки сигналов являются данные с ФТ, поскольку на этом принципе основана работа любого аналого-цифрового преобразователя. Результаты сравнения характеристик двух представлений вещественных чисел приведены в Табл. 2.1.

Табл. 2.1. Сравнительная характеристика представлений с ПТ и ФТ.

Параметры	ПТ (32 бита)	ФТ (16 бит)
Динамический диапазон, Дб	$2,9 \cdot 10^{16}$	2^{15}
Точность	25 бит	16 бит
Наличие специальных обозначений (бесконечность, не числа, нули)	да	нет
Представление отрицательных чисел	старшим знаковым разрядом	в дополнительном коде

Таким образом, анализ особенностей различного представления данных позволяет сформулировать следующие преимущества использования вычислений с ПТ:

- 1) При использовании 32-разрядного представления с ПТ существенно повышается точность внутреннего представления данных.
- 2) Представление чисел с ПТ позволяет расширить динамический диапазон представляемых величин, что представляет ценность при работе с биомедицинскими сигналами.
- 3) Использование вычислений с ПТ снимает проблемы масштабирования данных, что необходимо, чтобы избежать переполнений при выполнении различных арифметических операций.

- 4) Вычисления с ПТ являются более простыми с точки зрения реализации на языках высокого уровня, в частности на языке C++, что, в свою очередь, позволяет получать более эффективный код и минимизировать процедуры разработки и тестирования реализаций алгоритмов.

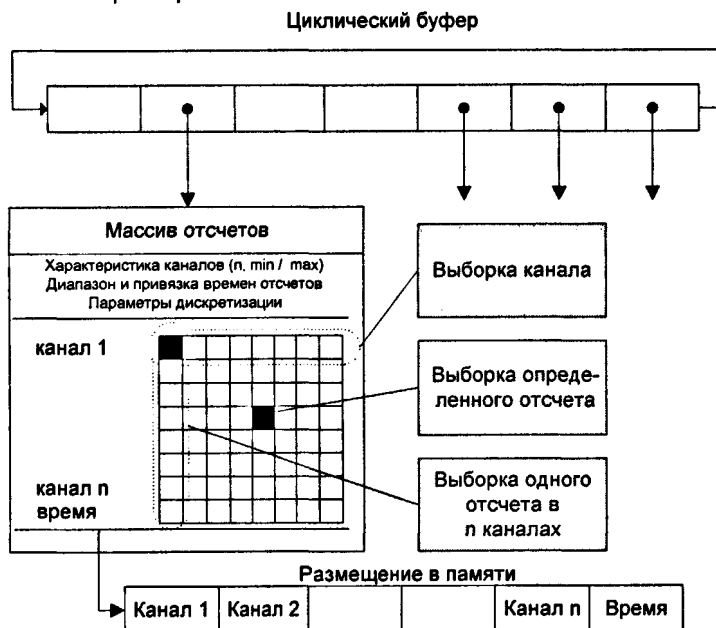
Среди недостатков представлений данных с ПТ стоит выделить:

- 1) несколько больший требуемый объем оперативной и дисковой памяти, в случае сигналов с низкой степенью квантования (менее 16 бит);
- 2) вычисления с ФТ зачастую выполняются устройствами вычисления, такими как цифровой процессор обработки сигналов (ЦПОС), быстрее.

Таким образом, исследование представлений данных, результаты которого приведены в Табл. 2.1, свидетельствует о том, что наиболее целесообразным является применение методов представления данных с ПТ, если решена проблема недостатка производительности операций с плавающей точкой, выполняемых ЦП. Эффективное представление БМИ в магистрали данных (см. Рис 2.2) управляемой дискретными событиями, может быть обеспечено массивами отсчетов, представляющими собой последовательные кортежи значений в представлении с плавающей точкой одинарной точности согласно стандарту IEEE 754[58] (см. Рис. 2.6).

Это позволяет обеспечить динамический диапазон представления амплитуд сигналов, определяемый как отношение максимальной величины к минимальной, равным:

$$D = \frac{(2 - 2^{-23})2^{127}}{2^{-126}} \approx 2,9 \cdot 10^{76} \text{ а точность представления, определяемая длиной}$$



мантиссы:

$$P = \log_2 \left| \frac{V_{\max}}{\Delta V} \right| = \log_2 \left| \frac{2 - 2^{-23}}{2^{-24}} \right| \approx 256 \text{ бит}$$

Значительный динамический диапазон и наличие специальных значений в представлении числа позволяет полностью избежать дополнительных затрат, связанных с контролем переполнения и переносов.

Таким образом, исходный набор из n сигналов может

Рис. 2.6. Физическая модель данных.

быть представлен множеством массивов $S_N = \{A_t^N: 0 < t_1 < t_k < T_{\max}\}$. Произвольный дискретный процесс Φ_d может быть описан в терминах дискретного преобразования $F_d: A_t^N \rightarrow A_t^N$.

В рамках данной модели реализация управления передачей данных процессом Φ_d может быть сведена к разработке логики работы функции передачи F_d , при этом организация очередей событий в межпроцессном обмене осуществляется циклическими буферами FIFO, использование которых представляется наиболее оптимальным в данном случае с точки зрения сложности реализации и производительности [59]. Пример реализации процесса вычисления скользящего ДПФ представлен на Рис. 2.7.

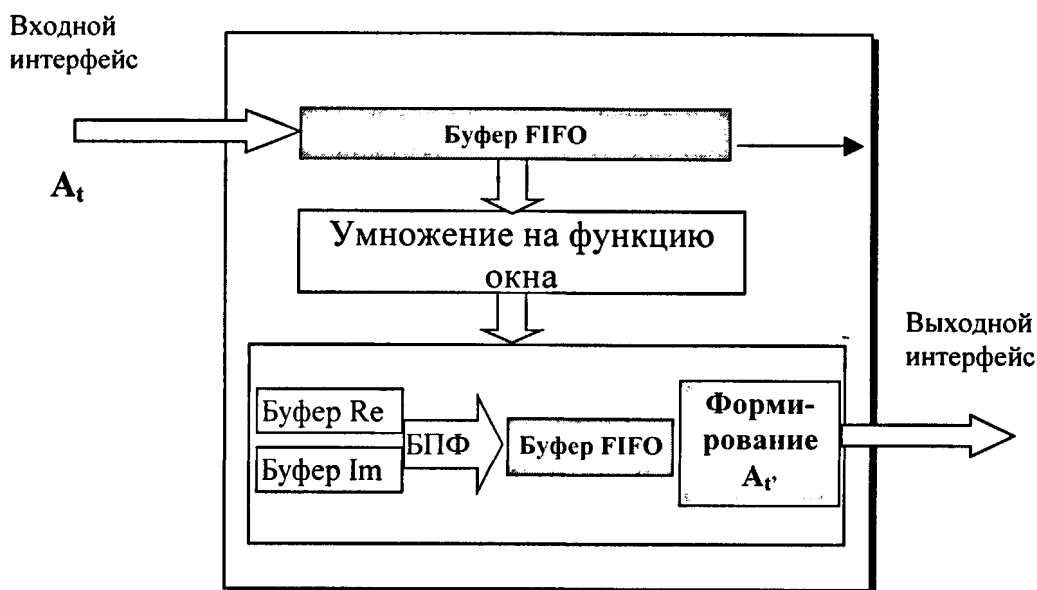


Рис. 2.7. Пример архитектуры процесса вычисления ДПФ.

2.3.3 Имитационная модель подсистемы распределения и обработки данных

Разработка методов и алгоритмов обработки данных требует решения задачи моделирования и тестирования программных реализаций отдельных элементов системы или отдельных подсистем в целом. Рассмотренная в предыдущем параграфе сетевая модель потоков данных, управляемая дискретными событиями, является достаточно хорошим представлением биомедицинских систем регистрации и математической обработки данных. Эксплуатация данной модели в имитационной среде позволяет не только провести моделирование поведения системы в условиях реальной эксплуатации, но и произвести тестирование отдельных процессов. В настоящее время наиболее эффективной и функционально развитой имитационной средой, основанной на концепции сетевой модели

информационных потоков данных, является среда SIMULINK, входящая в пакет MATLAB, компании MathWorks [60].

Согласно структурной схеме (Рис. 2.2) основными источниками данных служат устройства регистрации биомедицинских сигналов (в дальнейшем - УРС), которые представлены в программной модели контроллерами медицинских устройств, или виртуальными медицинскими устройствами (ВМУ). УРС выполняют ряд существенных операций: регистрация и предварительная обработка аналогового биологического сигнала, дискретизация сигнала, упаковка данных. Согласно современным представлениям о системах регистрации биомедицинской информации можно выделить несколько основных клинических сценариев использования подобных систем:

- 1) регистрация и передача информации в режиме реального времени (Рис. 2.8);
- 2) просмотр ранее записанной биомедицинской информации;
- 3) регистрации и обработка биомедицинской информации с последующей некритичной по времени передачей информации.



Рис. 2.8. Сценарий регистрации и передачи информации в реальном масштабе времени.



Рис. 2.9. Сценарий доступа к ранее записанной и обработанной информации.

Представленные выше сценарии характерны для целого ряда лабораторных и клинических систем диагностики и мониторинга (ЭКГ, ЭЭГ, ЭОГ). На практике возможно несколько способов построения моделей систем распределения данных, включающих все упомянутые выше клинические сценарии. Наиболее целесообразно рассматривать поведение системы в период максимальной загрузки, т.е. в моменты регистрации и обработки данных от одного и более устройств регистрации в реальном масштабе времени. Периоды максимальной нагрузки являются наиболее показательными при оценке эффективности и устойчивости любой системы обработки информации. Сценарий регистрации информации, представленный на Рис. 2.8, позволяет рассмотреть работу подобных систем с позиций теорий систем массового обслуживания. В рамках такого подхода, любой элемент реальной концептуальной модели аппаратно-программной системы может рассматриваться в качестве обслуживающего устройства, а блоки передаваемых данных от источников регистрации в виде дискретных событий – заявками на обслуживание. Такая система обслуживания имеет ряд особенностей:

- 1) последовательность обработки заявки компонентами системы четко фиксирована, нет возможность выбора между свободными сервисами;
- 2) обслуживающие устройства в реальных системах работают квазипараллельно в режиме разделения времени центрального процессора ЦП;
- 3) интенсивность поступления заявок λ даже при четко заданном алгоритме работы УРС может являться случайной величиной, распределенной по сложному закону.

Поскольку поток заявок в такой системе не подчиняется закону Пуассона, достаточно сложно получить аналитические результаты для подобной модели. В случае, когда λ -интенсивность поступления заявок в систему с одним сервисом и время обслуживания t имеет произвольное распределение с математическим ожиданием $M(t)$ и дисперсией $D(t)$, то среднее число находящихся в системе заявок L_s может быть получено по формуле Поллачека-Хинчина [61]:

$$L_s = \lambda M(t) + \frac{\lambda^2 (M^2(t) + D(t))}{2(1 - \lambda M(t))}, \lambda M(t) < 1 \quad (2.4)$$

Остальные параметры модели массового обслуживания вычисляются из L_s . Условие $\lambda M(t) < 1$ является естественным условием устойчивости системы реального времени. В противном случае, она будет не в состоянии обслуживать заявку за время поступления следующего задания и, следовательно, будет неустойчивой.

Таким образом, единственным средством получения достоверных результатов об эффективности функционирования компьютерных БМС и элементов их составляющих является имитационное моделирование. Целями подобного моделирования являются:

а) определение технических характеристик и описания режимов функционирования элементов программной архитектуры по выполнению задач передачи и обработки данных;

б) определение характеристик качества обслуживания пользователей, прежде всего показателей эффективности, характеризующих способность реализации алгоритма справляться с задачами обработки при условии налагаемых ограничений;

в) тестирование реализаций алгоритмов и методик обработки БМИ в имитационной среде.

В данном случае, наиболее целесообразным является формализация работы системы агрегатами, имитирующими функционирование каждого элемента системы. Организацию квазипараллельной работы такой агрегативной модели можно осуществлять дискретными событиями, или, что эквивалентно, транзактами. Формально под транзактом понимается некоторый динамический объект модели, представляющий собой заявку на обслуживание. Каждому транзакту принадлежит определенный набор параметров, интерпретируемых и изменяемых агрегатами, имитирующими работу элементов программной системы. Преимуществом подобного подхода является то, что транзакты являются хорошей моделью реальных элементов программных систем обработки и физической структуры представления данных.

Для задания структуры подобной имитационной модели можно ограничиться несколькими типами описаний элементов модели: источников, узлов и связей. Последовательность входных данных от источников, в данном случае таким источниками могут являться агрегаты, формализующие работу любого УРС, например, регистратора ЭКГ, представляется в виде потока транзактов.

Согласно общим характеристикам функциональной модели для задач имитационного моделирования целесообразно использовать несколько типов узлов:

В–узлы, позволяющие размножить транзакт на заданное число копий;

S–узлы, дающие возможность переключения транзактов в соответствии с указанными типами транзактов;

M–узлы, допускающие модификацию параметров транзактов;

D–узлы, в которых заканчивается время жизни транзактов.

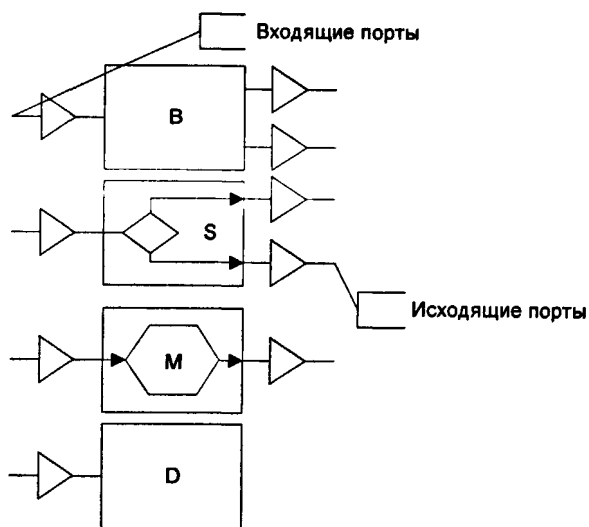


Рис. 2.10. Основные элементы имитационных моделей подсистемы передачи и обработки данных.

соответствует выходному каналу устройств регистрации. Внешняя нагрузка обладает рядом характеристик F , которые можно считать неизменными в ходе моделирования. Основными варьируемыми характеристиками модели является интенсивность поступления заданий (транзакций) из множества G (транзактов в секунду) λ -интенсивность, которая связана с периодом передачи транзактов ΔT следующим соотношением: $\lambda = \frac{1}{M(\Delta T)}$, где M - математическое ожидание периода передачи событий ΔT .

Для алгоритмизации работы источника транзактов решающее значение имеет распределение $F(\Delta T)$, соответствующее реальной системе. Формально $F(\Delta T)$ при реализации подсистемы распределения данных возможно сделать произвольным, согласно выбранному алгоритму планирования, рассмотренному в предыдущих параграфах. Однако на практике $F(\Delta T)$ зависит от качества реализации самого планировщика задач, точности доступного для вычислительных задач ОС пользовательского уровня, таймера. В ОС семейства Windows NT/2000 максимально возможное разрешение таймера, доступное пользовательским процессам составляет 1мс и, следовательно, использование более коротких интервалов ΔT невозможно [62]. Однако даже такое разрешение является недоступным из-за того, что квант времени, выделяемый планировщиком текущей задаче, составляет несколько мс в зависимости от версии Windows NT. Для того, чтобы оценить отклонение модельного времени от реального,

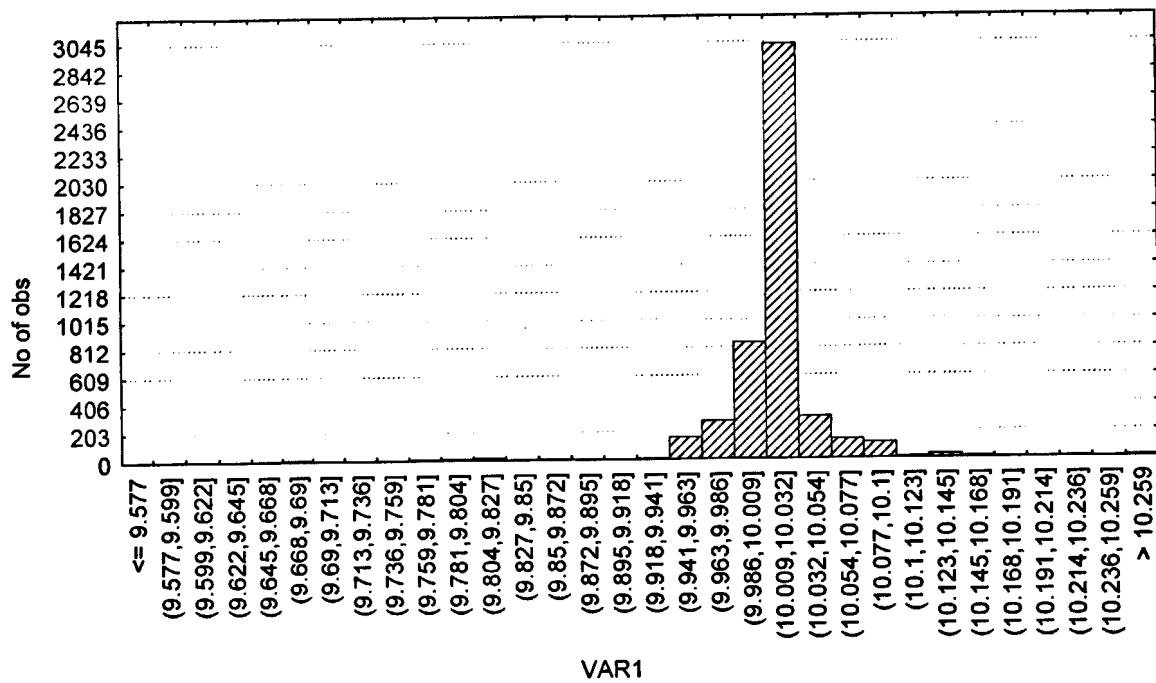
Типичными примерами узлов типа B , являются контроллеры биомедицинских устройств регистрации, S и M -различные компоненты обработки БМИ, D -подсистема накопления данных.

На первом этапе необходимо решить проблему алгоритмизации работы каждого агрегата, участвующего в схеме обработки. Формализация подсистемы распределения данных предусматривает наличие постоянного потока заданий G^* , который в реальном случае

необходимо исследовать точность системного таймера, доступного программным приложениям. Оценка точности этого таймера возможна путем использования более точной функций измерения промежутков, предоставляемой ядром ОС, которая позволяет получать значения счетчика, связанного прямым соотношением с частотами работы системной шины. Разности значений, выдаваемых данной функцией, могут быть переведены в микросекунды путем деления полученного значения на вес младшего разряда, который также можно получить через функции ИПП ОС. Таким образом, есть возможность измерения промежутков времени в рамках пользовательского процесса с точностью до 1 мкс. Экспериментальные результаты по тестированию точности системного таймера на выборке $N=5000$ при заданном $\Delta T = 10$ мс, приведены на Рис 2.11. Математическое ожидание полученного ряда составляет $M(\Delta T) = 10.01$, дисперсия $D(\Delta T) = 0.0395$. Смещение $M(\Delta T)$ свидетельствует о дополнительных затратах, которые сложно учесть, в частности, переключение контекстов задач планировщиком. Очевидно, что представленное распределение имеет четко выраженный характер зависимости от периодических процессов, происходящих в ядре (переключение задач, обработка прерываний, работа со страницами памяти), о чем свидетельствуют периодические выбросы (Рис. 2.11б). Следует отметить способность системы к стабилизации периода вызова таймера, с целью сохранения заданного периода вызова в среднем. Это означает, что в случае задержки вызова таймера по тем или иным причинам система пытается осуществить такой вызов в следующий раз раньше срока. Windows NT, как и многие другие многозадачные ОС, использует на несколько порядков более точную процедуру измерения промежутков времени по сравнению с доступным таймером ядра, которая гарантирует разрешение в зависимости от реализации до нескольких микросекунд. Достаточно малая дисперсия (порядка 40 мкс) и предсказуемый характер поведения позволяют при проектировании подсистемы распределения и обработки данных и, следовательно, ее моделировании, пренебречь погрешностями системного таймирования, по крайней мере, при частотах регистрации сигнала до нескольких килогерц. ОС семейства Windows NT относятся к так называемым мягким системам реального времени, которые дают возможность гарантировать начало или завершение процесса в заранее заданный промежуток времени, однако погрешности таймирования при этом достаточно велики.

Таким образом, имитационная модель без учета данных факторов будет являться хорошей аппроксимацией.

a)



b)

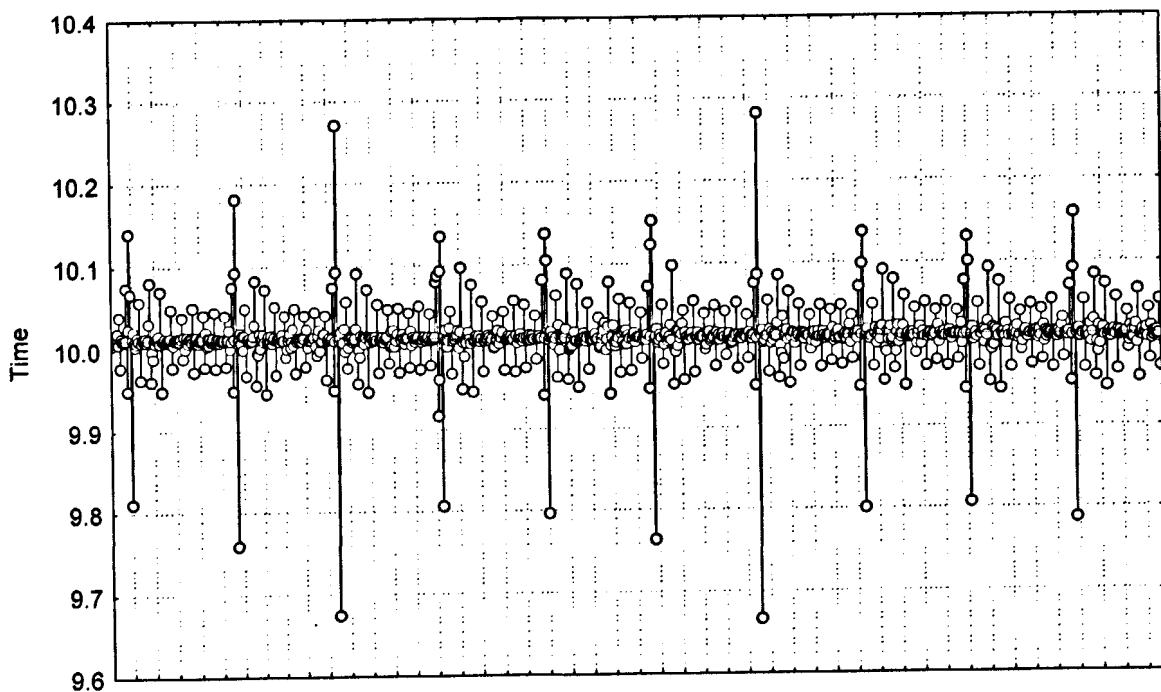


Рис. 2.11. Характеристики таймера ядра Windows NT.

а) Гистограмма частот;

б) Последовательность ΔT .

Пример имитационной модели, представленной на Рис. 2.12, включает в себя несколько элементов: источник сигнала, имитирующий

работу 8-канального ВМУ, детектор QRS-комплексов, модуль предварительной обработки, накопления данных, а также визуализатор БМИ. Сигнал ЭКГ может иметь произвольную частоту дискретизации, в данной модели она соответствует 500 Гц. Кванты модельного времени соответствуют периоду генерации дискретных событий передачи данных. В случае, когда источников сигнала более одного, необходимо выбрать минимальный период. Упаковка данных в массивы отсчетов (МО) осуществляется ВМУ и в простейшем случае размер кванта времени МО может быть выбран фиксировано. Элементы модели, в свою очередь, являются сложными элементами, имеющими внутреннюю архитектуру, которая, в значительной мере, может быть формализована аналогичным образом. Реализация, не допускающая формализацию на уровне потоков данных, представляется набором т.н. S-функций, которые позволяют подключить внешнюю реализацию алгоритмов обработки потока данных. S-функции позволяют, по сути, провести тестирование в имитационной среде реальной реализации того или иного метода, использование которого возможно в составе реальной БМС (см. Приложение 3). Использование стандартных элементов визуализации среды моделирования позволяет следить за развитием процессов во времени. Имитационная модель позволяет провести полное моделирование магистрали передачи данных, функций каждого из элементов, принимающих участие в создании, обработке и передаче БМИ, тестирование алгоритмов обработки данных и оценить пригодность и эффективность всей системы в целом.

На основании представленных результатов основные технические требования к магистрали данных и подсистеме распределения и обработки БМИ могут быть сформулированы следующим образом:

- поддержка возможности регистрации информации с различных устройств одновременно в режиме реального времени;
- поддержка контроллером УРС режима автономной работы;
- динамическое изменение структуры сети обработчиков данных и параметров узлов обработки;
- обеспечение быстрого доступа к данным для эффективной работы в режиме реального времени;
- интегрированное представление данных и унифицированный интерфейс управления ресурсами УРС;
- возможность интеграции дополнительных программных компонент без внесения изменений в уже существующие в подсистеме;
- ориентация на существующие стандарты биомедицинской информатики.

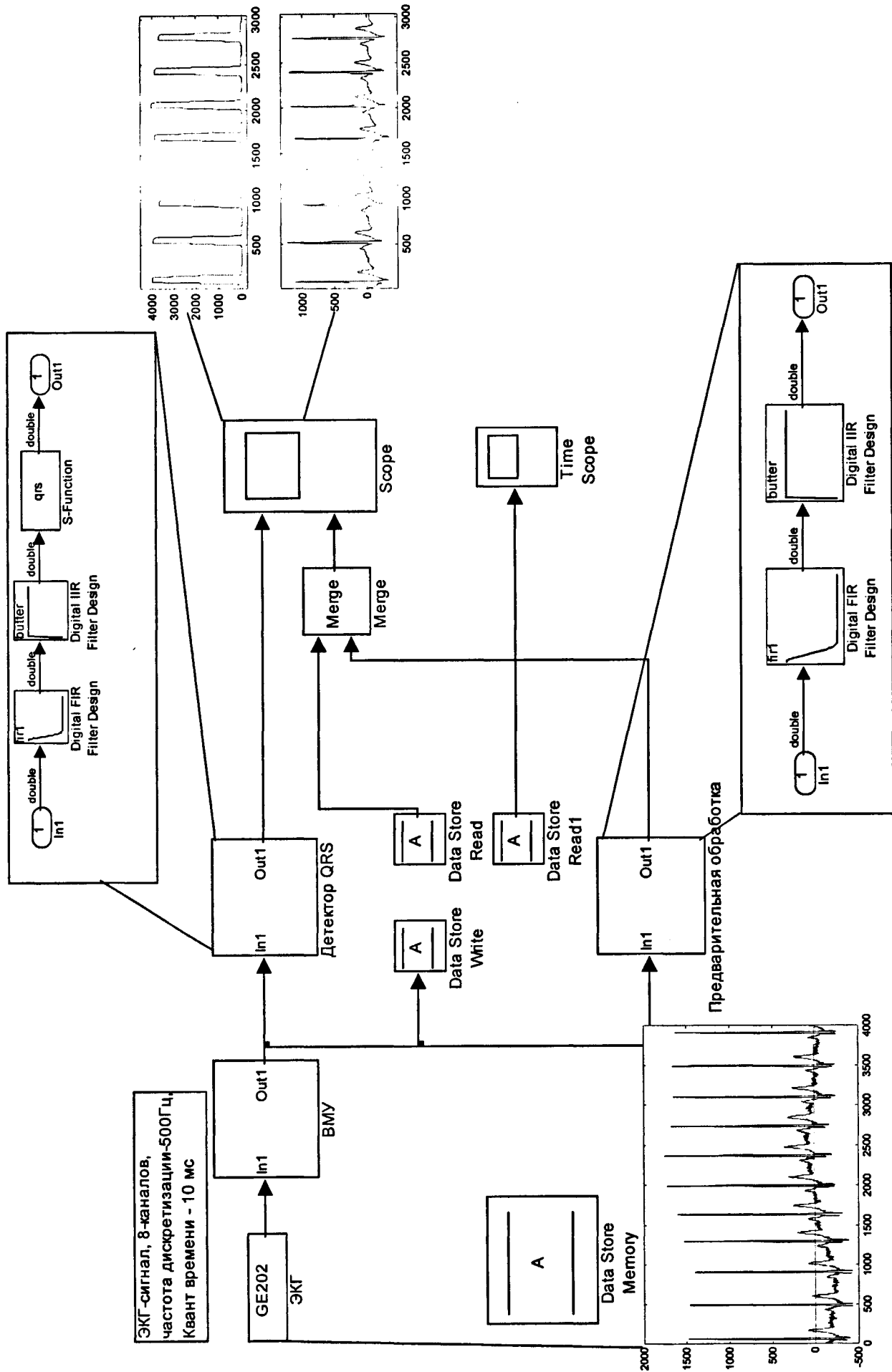


Рис. 2.12. Пример имитационной модели подсистемы обработки БМИ "Intellicard".

Глава 3. Экспериментальная интегрированная компьютерная система регистрации и математической обработки электрофизиологических сигналов «IntelliCard»

3.1. Основные функциональные характеристики аппаратно-программного комплекса

3.1.1 Краткая характеристика и назначение компьютерной БМС «IntelliCard»

На основе теоретических результатов, представленных во второй главе, разработана интегрированная компонентно-ориентированная программная система регистрации и математической обработки электрофизиологических сигналов реального времени «IntelliCard». Ядром программной системы выступает архитектура динамических модулей, дающая возможность реализовать программный комплекс полностью на компонентной основе. «IntelliCard», построенный на базе персональной ЭВМ под управлением ОС Windows 98/Windows 2000, может быть интегрирован практически с любой аппаратной реализацией устройств регистрации электрофизиологических сигналов, как посредством модулей экспорта/импорта в стандартные форматы хранения БМИ, так и с помощью динамически подключаемых модулей прямого ввода ЭКГ, разработанных для конкретного устройства регистрации.

Среди технических особенностей «IntelliCard» необходимо выделить следующие [63]:

- 1) динамическое подключение устройств регистрации электрофизиологических данных;
- 2) поддержка возможностей регистрации одновременно с нескольких устройств;
- 3) наличие коммуникационных интерфейсов для передачи биомедицинских записей через Internet и средства мобильной связи;
- 4) доступность программного обеспечения в двух конфигурациях: для портативной ПЭВМ и встраиваемом исполнении для построения БМС под управлением Windows-совместимых ОС.

Основными задачами, решаемыми аппаратно-программным комплексом «IntelliCard», являются:

1. Организация базы данных с возможностью накопления как электрокардиографических, так и прочих клинических данных, при этом ввод данных может осуществляться как с устройств регистрации, а также из внешних файлов, созданных другими приложениями.

2. Обработка цифровых сигналов различными математическими методами и сохранение результатов обработки в интегрированной БД:

а) предварительная обработка сигнала (линейная, адаптивная КИХ-, БИХ-фильтрация помех);

б) реализация ряда современных кардиографических методов (выделение и анализ элементов ЭКГ);

в) обеспечение возможности быстрой интеграции модулей, реализующих новые методы представления или обработки данных на основе разработанного программного инструментария в рамках компонентной архитектуры системы [64];

г) подготовка и представление данных в требуемом виде, в том числе для печати или публикации;

д) доступ к удаленным медицинским информационным ресурсам [65];

3. Поддержка функций обмена (в том числе в реальном масштабе времени) электрокардиографическими и клиническими данными с другими подобными системами на основании установленных стандартов.

“IntelliCard”, являясь ЭКГ-системой реального времени, представляет собой переносной аппаратно-программный комплекс, обеспечивающий основной набор функций комплекса функциональной диагностики, поддерживает, с точки зрения целевого применения, следующие операции:

- 1) регистрация ЭКГ и расчет по одной из следующих схем отведений: 12 стандартных, по Небу, по Франку, биполярные ортогональные, Мак Фи[66];
- 2) сопряжение с устройствами, регистрирующими вспомогательную биомедицинскую информацию (монитор артериального давления, пульсовой оксиметр и т.д.);
- 3) отображение на экране и печать от 1 до 12 отведений сигналов с частотами дискретизации до 1 кГц и разрядностью 22 бита при скоростях развертки 1.25, 2.5, 5, 10 см/сек с различными масштабами по амплитуде;
- 4) поддержка настраиваемых функций предварительной обработки ЭКГ реального времени для подавления внешних и инструментальных помех ЭКГ и артефактов;
- 5) возможность динамического подключения новых компонент, реализующих дополнительные функции обработки, анализа и визуализации данных;

- 6) обеспечение в дополнение к стандартному режиму записи ЭКГ режима исследований, который позволяет просматривать последовательность амплитуд и длительностей определенных QRS-комплексов;
- 7) гибкое управление процессом печати; кардиограмма, выводимая на любой матричный, струйный или лазерный принтер (до 1200 точек\дюйм включительно) соответствует принятым в практике масштабам (от 1.25 до 10 см/сек и от 0.5 до 4 мВ/см);
- 8) поддержка режима записи, обработки и визуализации длительных ЭКГ, объем данных ограничивается свободным пространством, выделенным на жестком диске;
- 9) повышенная надежность системы в режиме записи ЭКГ за счет реализации механизма непосредственного сохранения записанной информации на жесткий диск в реальном времени;
- 10) широкая область применения (клиническое, лабораторное, экспериментальное).

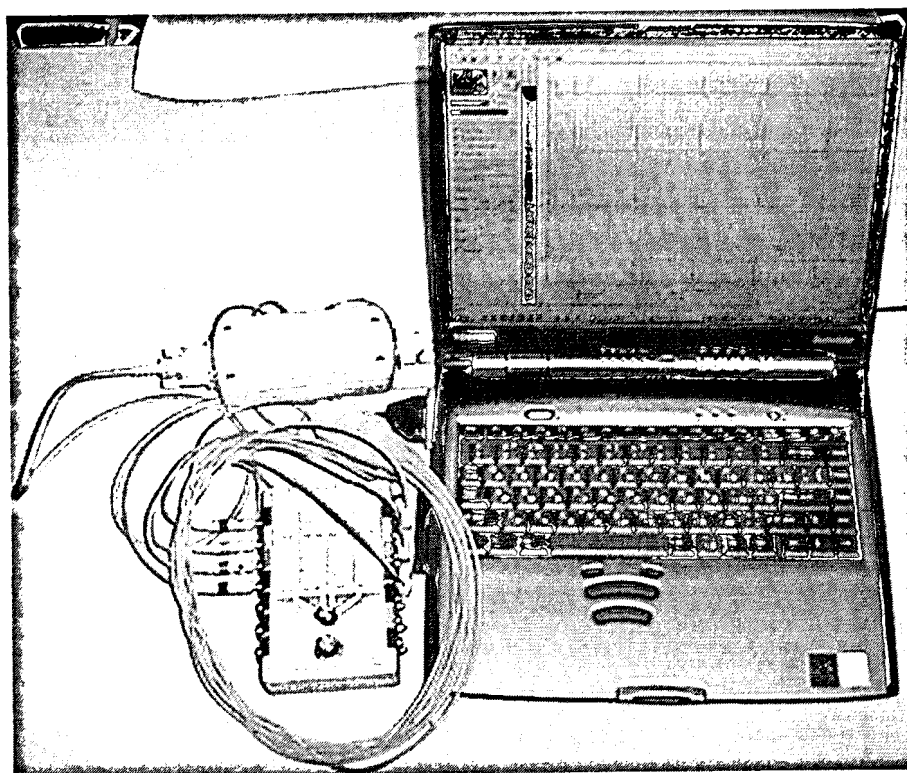


Рис. 3.1. Компьютерная БМС "Intellicard" в исследовательском исполнении.

В состав системы могут входить: устройство регистрации ЭКГ, монитор АД, пульсовой оксиметр, портативная ЭВМ, генератор тестовых ЭКГ-сигналов, а также другие устройства регистрации электрофизиологических сигналов.

В Табл. 3.1 представлены основные технические параметры системы «IntelliCard». Технические характеристики электрокардиографа представлены в Табл. 3.2.

Табл. 3.1. Технические характеристики "Intellcard".

1	Поддерживаемые операционные системы	Windows 98\2000, Windows NT
2	Разрешение монитора	800x600 и выше
3	Рекомендуемый размер памяти	Не менее 128Мб
4	Необходимый минимальный размер памяти на диске	10 Мб
5	Метод подключения устройств регистрации электрофизиологических сигналов	Без ограничений методом модулей расширения
6	Устройства регистрации сигналов	12-ти канальный сигма-дельта регистратор ЭКГ, пульсоксиметр, монитор артериального давления и т.д.
7	Поддержка устройств печати	Лазерные, светодиодные, струйные (при наличии драйвера) / 150 x 150 dpi, 300 x 300 dpi, 600 x 600 dpi, 1200 dpi. Поддержка режима непрерывной бумаги.
8	Процессор	Intel Pentium II , 350 MHz и выше

Табл. 3.2. Технические характеристики электрокардиографа.

1	Частота дискретизации	512000 Гц синхронно по всем каналам до цифровой децимации и фильтров прореживания
2	Динамический диапазон сигнала	600 мВ(22 разряда АЦП)
3	Полоса пропускания	0 – 150 Гц
4	Входное сопротивление	90 Мом
5	Разрешение	2 мкВ
6	Коэффициент ослабления синфазных сигналов	110 дБ

«IntelliCard» прошел сертификацию и допущен к продажам в Германии в качестве медицинского продукта на основании соответствия

директивам 93\42\EWG, Приложения II требований министерства здравоохранения ФРГ.

Визуальное представление ПО "IntelliCard" во время регистрации ЭКГ-данных содержится на Рис. 3.2.

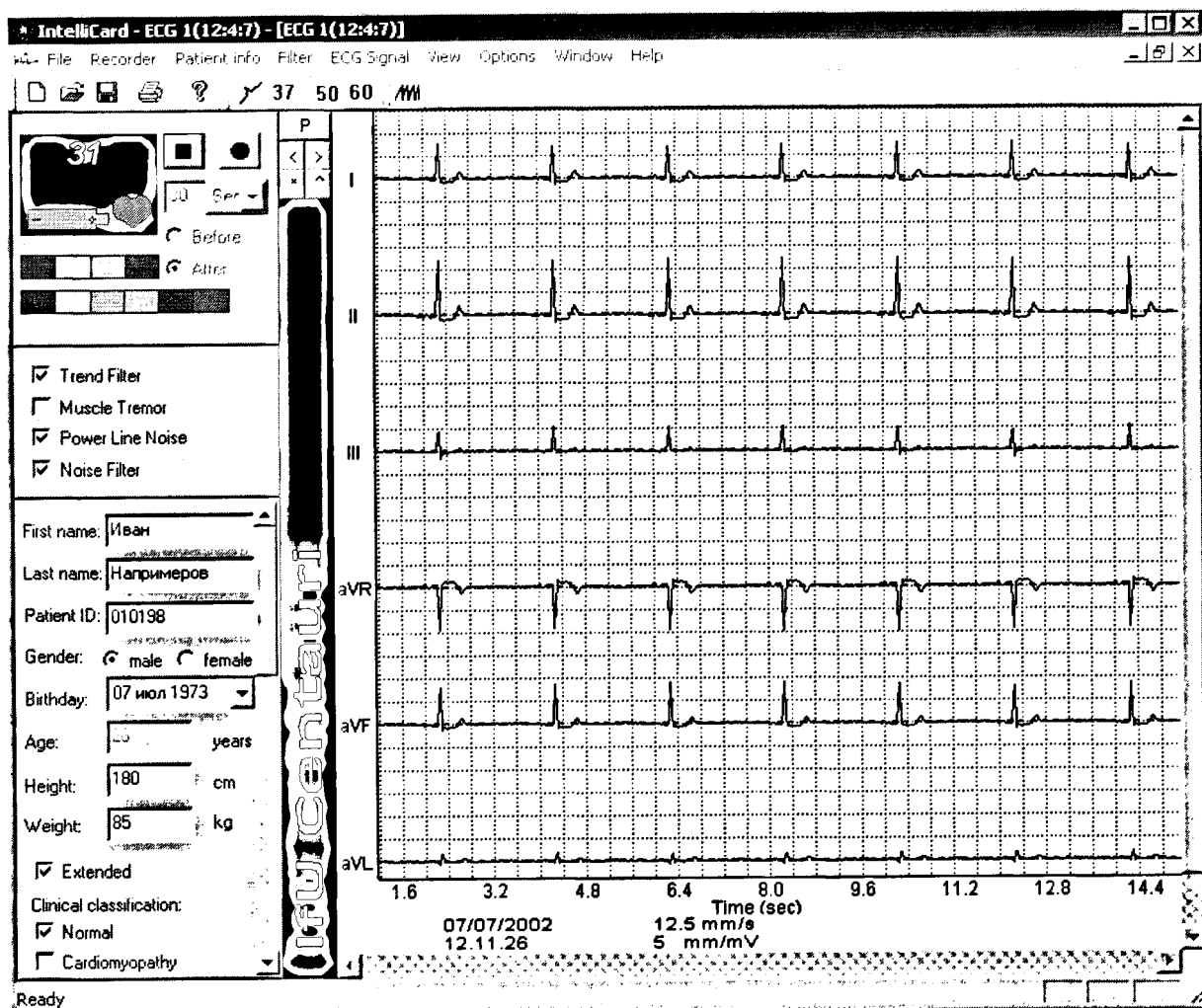


Рис. 3.2. Основной вид ПО "IntelliCard" во время записи 12-канальной ЭКГ.

Конфигурация подключаемых модулей зависит от конкретных начальных установок. В данном случае, в контексте регистрации ЭКГ активизированы компоненты отображения ЭКГ, предварительной обработки, интегрированной БД, а также модуль визуального управления устройством регистрации ЭКГ.

Конфигурация «IntelliCard» во встраиваемом исполнении построена на базе промышленной одноплатной ЭВМ, представляющей в качестве интерфейса управления 7 функциональных клавиш расположенных по периметру ЖК-экрана и манипулятор. Программной платформой данной ЭВМ может быть произвольная Windows-совместимая ОС, в том числе является возможной адаптация и под платформу Microsoft Pocket PC 2002, которая является развитием встраиваемых ОС линии Windows CE.

Функциональные назначения клавиш управления отображаются на экране в конкретный момент работы программы. В конфигурации, представленной на Рис. 3.3, активизированы компоненты визуального управления электрокардиографом, пульсовым оксиметром и монитором артериального давления, визуализации сигнала, а также предварительной обработки ЭКГ. Панели управляющих клавиш также реализованы в виде подключаемых модулей для унификации процедуры управления экранными видами.

Структура программного обеспечения «IntelliCard» во встраиваемом исполнении полностью соответствует полноценной версии. Встраиваемая версия «IntelliCard» функционирует в двух базовых режимах: в режиме съема и в режиме анализа ЭКГ. В режиме съема производится отображение в реальном времени кардиограммы получаемой от пациента. Последние 15 сек. отображаемой кардиограммы при переходе в режим анализа используется для анализа ЭКГ. Модуль анализа ЭКГ производит измерения характеристик определенных QRS-комплексов и сравнивает значения с нормативами, соответствующими, полу, возрасту, весу и росту пациента.

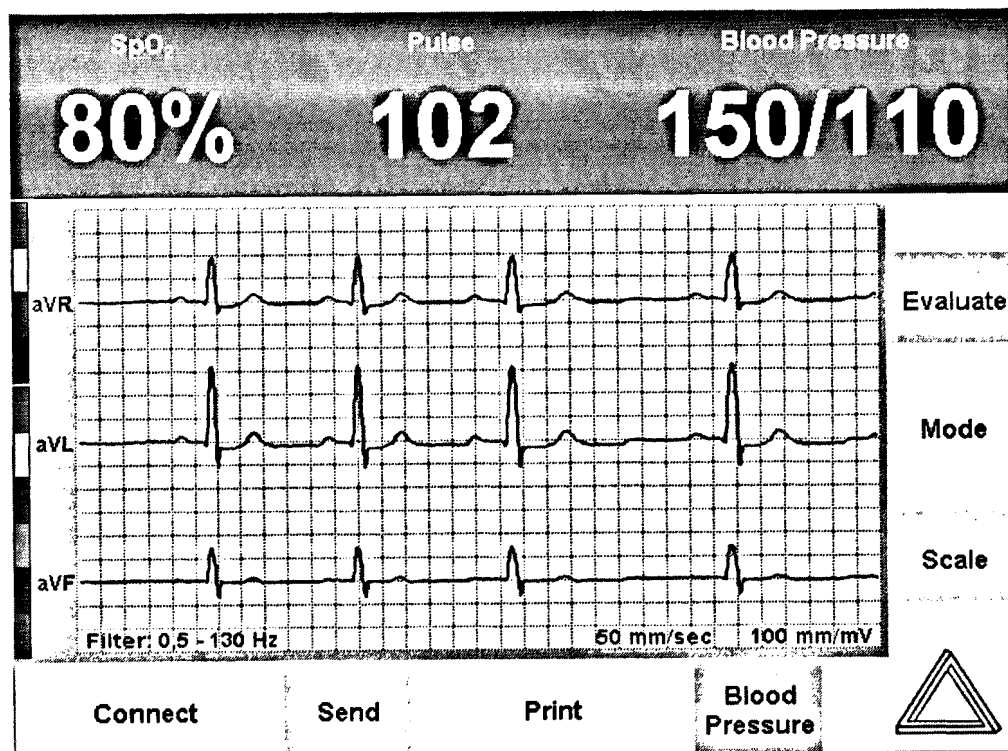


Рис. 3.3. Визуальное представление «IntelliCard» во встраиваемом исполнении.

Снятая кардиограмма и результаты анализа сохраняются на локальном диске и подключаемой карте памяти. Эти данные также могут быть перенаправлены на следующие устройства передачи данных: интерфейс IrDa, модем (в том числе для спутниковой и сотовой телефонии), стандартный сетевой интерфейс. «IntelliCard» в исследовательском и встраиваемом исполнении имеют единый менеджер печати, поддерживающий режим непрерывной бумаги, который

обеспечивает требуемые масштабы по амплитуде и времени представления электрокардиографического и сопутствующих сигналов (Рис. 3.4).

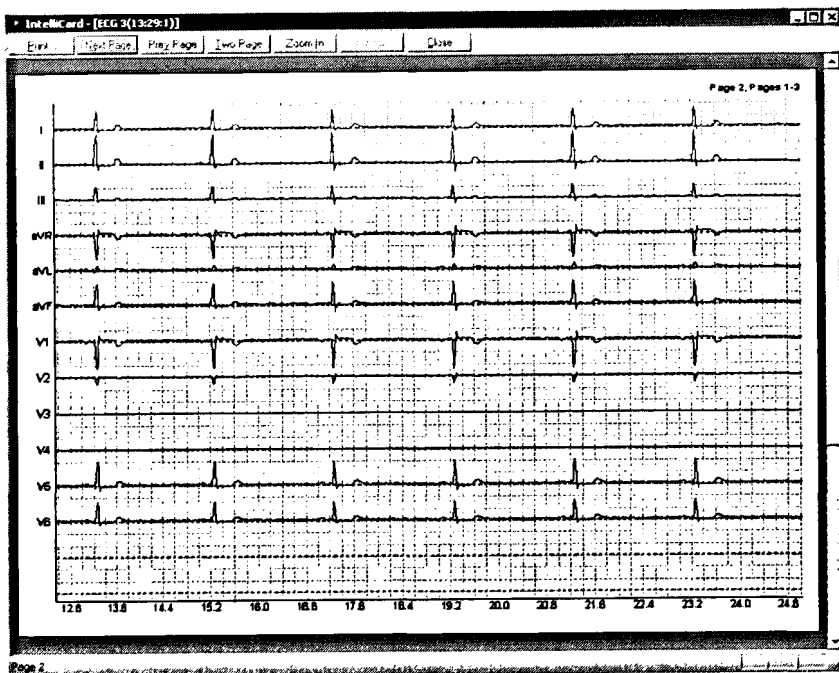


Рис. 3.4. Предварительный просмотр электрокардиограммы и печать.

3.1.2 Состав и структура программного комплекса

Поскольку в качестве базовой платформы промежуточного ПО была выбрана архитектура COM+, ее концептуальные особенности оказали влияние на формирование программной модели системы «IntelliCard». На этапе проектирования системы был проведен сравнительный анализ наиболее распространенных компонентных платформ, среди которых необходимо выделить COM+, JavaBeans/Remote Method Invocation(RMI)[67], а также Common Object Request Broker Architecture (CORBA). Исследования выявили некоторое преимущество первой из них для приложений реального времени. RMI, по-видимому, имеет пока ограниченное применение, вследствие привязанности к платформе Java, CORBA же накладывает достаточно широкие ограничения на системную архитектуру, в частности, требует наличия промежуточного звена-брокера объектных запросов. Решающим фактором в выборе платформы ПО промежуточного уровня является наличие поддержки собственной компонентной модели для подключаемых модулей, что обеспечивается только COM+ и Java.

Основные принципы выбранной платформы позволяют сформировать ряд основных свойств архитектуры программной системы, построенной на основе использования этой технологии:

- 1) соответствие общим требованиям COM+;
- 2) наличие уровневой, многозвенной архитектуры;

- 3) полная унификация механизма работы с данными;
- 4) поддержка внешних открытых интерфейсов для прозрачной интеграции дополнительных компонент.

Соответствие требованиям COM+ подразумевает использование операционных систем линии WindowsNT, в частности, Windows 2000. Основой многозвенной архитектуры (*от англ. multi-tier architecture*) может выступать хорошо зарекомендовавшая в последние годы модель «документ-представление». Ее принципом является заключение рабочей функциональности в компонентах, условно называемых документами, и визуального представления – в так называемых видах. Таким образом, рассмотренная общая концепция системной архитектуры вполне укладывается в трехуровневую модель, известную как Windows DNA – Distributed Internet Application Architecture. Следовательно, базовая архитектура системы «IntelliCard» может быть представлена в виде следующей трехуровневой структуры с выделенным слоем уровня операционной системы [68].

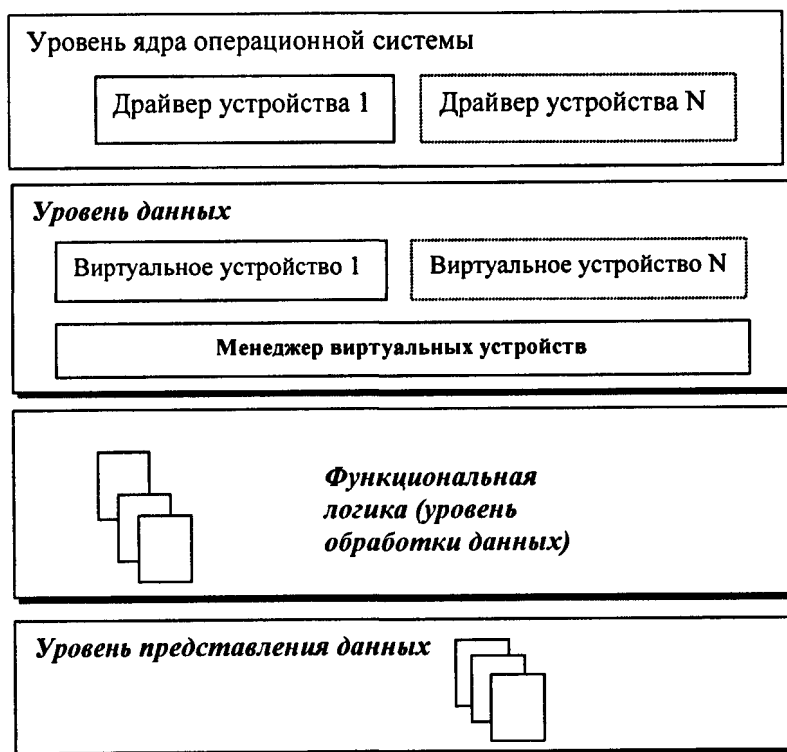


Рис. 3.5. 3-х уровневая архитектура «IntelliCard».

В рамках представленной многозвенной архитектуры системные компоненты условно по своему функциональному назначению разделяются на три уровня: уровень данных, функциональной логики и представления. Каждый из этих компонент реализует необходимые для данного уровня интерфейсы доступа к заключенной функциональности и пользуется предоставляемыми системными сервисами в рамках, доступных для этого уровня. Детализация программной архитектуры представлена на Рис. 3.6.

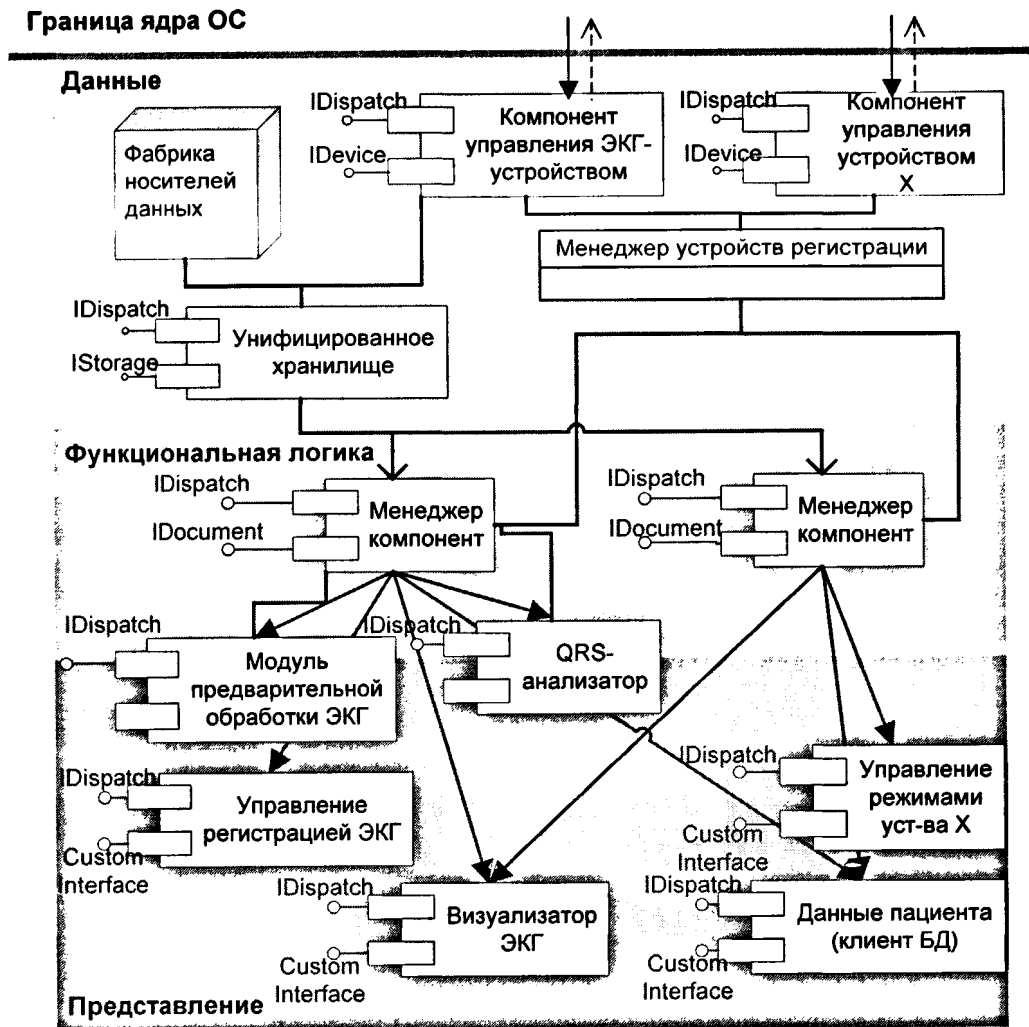


Рис. 3.6. Детализация структуры «IntelliCard».

Уровень данных составляют объекты, ответственные за операции ввода/вывода с внешними источниками данных (например, устройство регистрации ЭКГ, монитор артериального давления), а также универсальный компонент хранения гетерогенных, т.е. разнородных, данных, и, так называемая, фабрика контейнеров данных – объектов, используемых для передачи различного рода информации через интерфейсы компонент.

Функциональная логика реализуется набором объектов – менеджеров компонент, часто условно называемых документами, имеющих методы обработки данных и средства управления компонентами визуализации. В свою очередь, объекты визуализации могут иметь методы обработки данных, хотя их непосредственной задачей является отображение информации и обеспечение интерфейса с пользователем. Следует заметить, что представленная системная структура не накладывает ограничений на локализацию отдельных модулей, а подразумевает возможность

дистрибуции их в рамках локальной среды, поскольку такая возможность обеспечивается непосредственно функциями COM+. Благодаря принципу открытой архитектуры и наличию четкой спецификации, возможно расширение и интеграция в комплекс новых функциональных модулей, реализующих новые методики анализа или предоставляющих возможность осуществлять съем данных на новых устройствах ввода ЭКГ.

Таким образом, структурно программная система «IntelliCard» может рассматриваться как иерархическая архитектура отдельных элементов, среди которых основными элементами являются разнообразные компоненты, выполняющие заранее заданные функции в рамках следующих подсистем:

- 1) регистрации биомедицинских данных;
- 2) накопления и хранения БМИ;
- 3) клинических методов обработки данных, визуализации и печати БМИ;
- 4) внешних коммуникационных интерфейсов.

Полный состав компьютерной БМС реального времени «IntelliCard» в виде диаграммы поглощения представлен на Рис. 3.7.

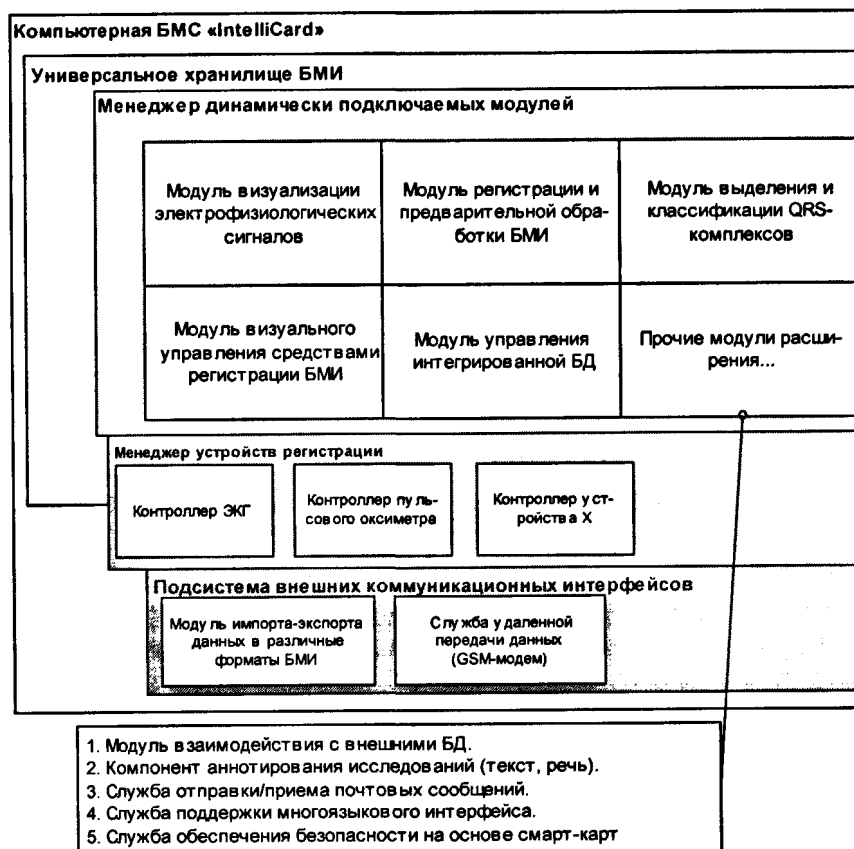


Рис. 3.7. Покомпонентный состав БМС «IntelliCard».

Полная покомпонентная структура разработанного программного обеспечения представлена в Приложении 5.

Компонентно-ориентированная архитектура приложения, построенная на нескольких уровнях реализации, требует наличия более сложной процедуры первоначальной активации программной системы. В отличие от монолитных приложений, данная процедура ввиду наличия множества независимых исполняемых единиц может стать достаточно сложной для реализации. Для решения данной задачи в рамках «IntelliCard» разработана многоуровневая процедура инициализации и динамического конфигурирования отдельных подсистем. Диаграмма состояний процедуры инициализации представлена на Рис. 3.8.

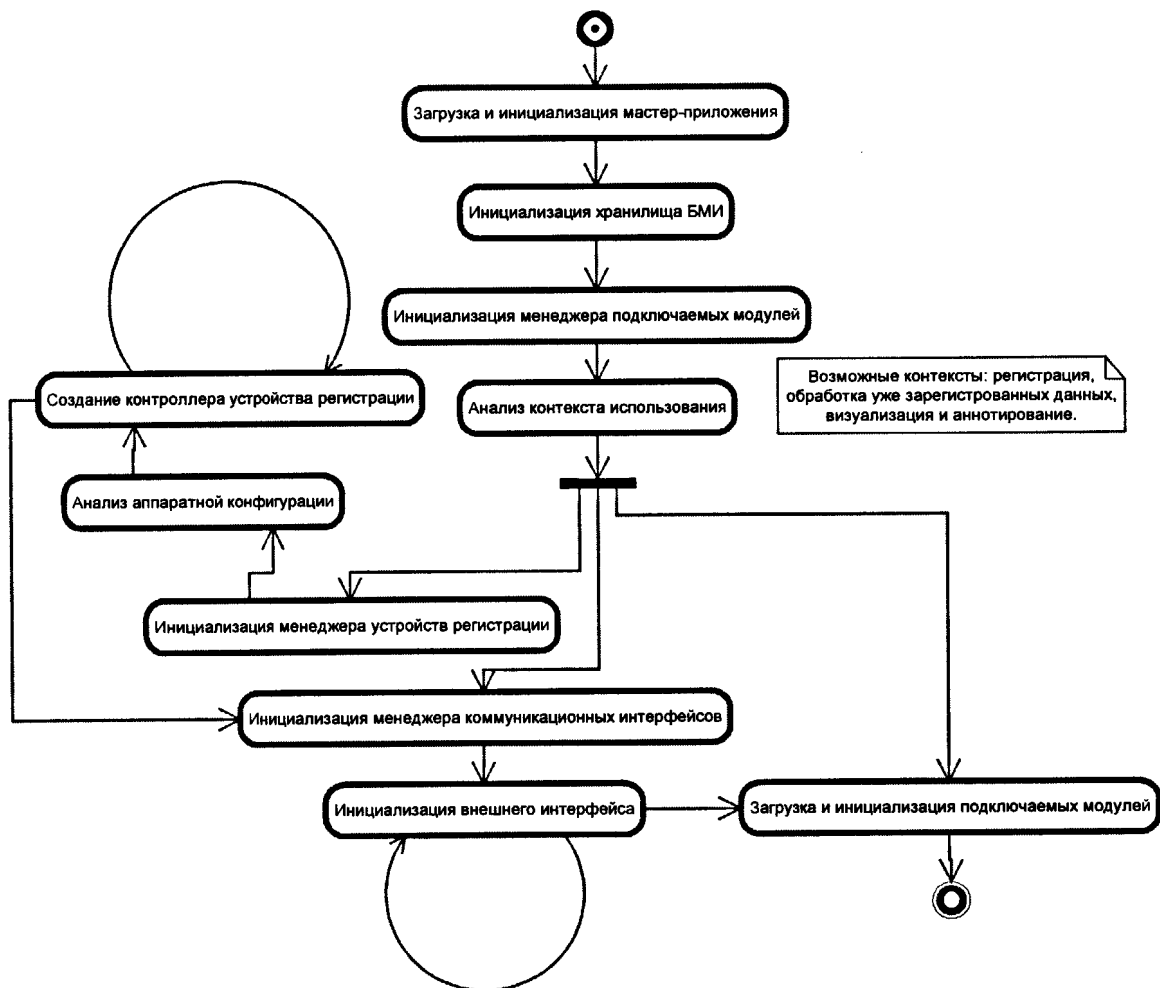


Рис. 3.8. Инициализация подсистем "IntelliCard".

Начальной фазой процедуры активации является загрузка и инициализация основных элементов ПО «IntelliCard», в частности, основного приложения, менеджера подключаемых модулей, без которых нормальное функционирование системы невозможно. Определение контекста использования позволяет определить дальнейшую процедуру инициализации приложения и список используемых в данном контексте подсистем. Так, например, в случае удаленной активации из приложения Microsoft Word для редактирования внедренного элемента OLE,

подсистема регистрации и телекоммуникационных интерфейсов не будет инициализирована. Завершающим этапом инициализации системы является загрузка всех подключаемых модулей, которые автоматически подключаются к системным сервисам, предоставляемым основным приложением и уже активизированными подсистемами.

Особенности реализации основных подсистем рассматриваются ниже в данной главе, некоторые математические методы обработки данных "IntelliCard" рассматриваются в Главе 4.

3.2. Основные программные компоненты системы "IntelliCard"

В состав ПО компьютерной электрокардиографической системы "IntelliCard" входит множество программных компонент (см. Рис. 3.7), сгруппированных по своему функциональному назначению в отдельные подсистемы, а по особенностям реализации – в отдельные уровни 3-х уровневой иерархической структуры (см. Рис. 3.5). Каждая из представленных подсистем имеет свои особенности проектирования и реализации.

3.2.1 Подсистема записи данных с устройств регистрации электрофизиологических сигналов и распределения данных

Современные подходы к проектированию межкомпонентного взаимодействия в программных системах предусматривают несколько возможных схем, в том числе сетевую модель и модель клиент\сервер. Согласно современным концепциям о механизме взаимодействия медицинских систем, осуществляющих регистрацию, обработку и сохранение биомедицинской информации [69], модель такого взаимодействия, предусматривающего возможность динамического установления связи (принцип plug&play), строится по классической схеме клиент/сервер. Данная схема является также основой единого коммуникационного стандарта MIB IEEE 1073 взаимодействия медицинских устройств. Развитием данной парадигмы взаимодействия объектов является модель так называемых программных агентов, которые мы можем определить, следуя Вудриджу [70] как аппаратную или программную систему, обладающую рядом свойств:

- а) автономность – агенты обладают способностью самостоятельно выполнять поставленные задачи или изменять свое внутренне состояние без прямого вмешательства извне;
- б) социальность – способность к взаимодействию с другими подобными системами;
- в) реактивность – свойство агента воспринимать свое окружение и отвечать в заданный промежуток времени на его изменение;
- г) проактивность – возможность принимать инициативу по выполнению того или иного действия на себя.

Таким образом, парадигма программных агентов, которая наиболее полно согласуется с основными принципами организации подсистемы обслуживания и поддержки устройств регистрации различной электрофизиологической информации, может быть эффективно использована для реализации подсистемы регистрации данных [71].

Схема взаимодействующих подсистем, следуя стандарту MIB(IEEE 1073), может быть представлена в виде следующей модели (Рис. 3.9).

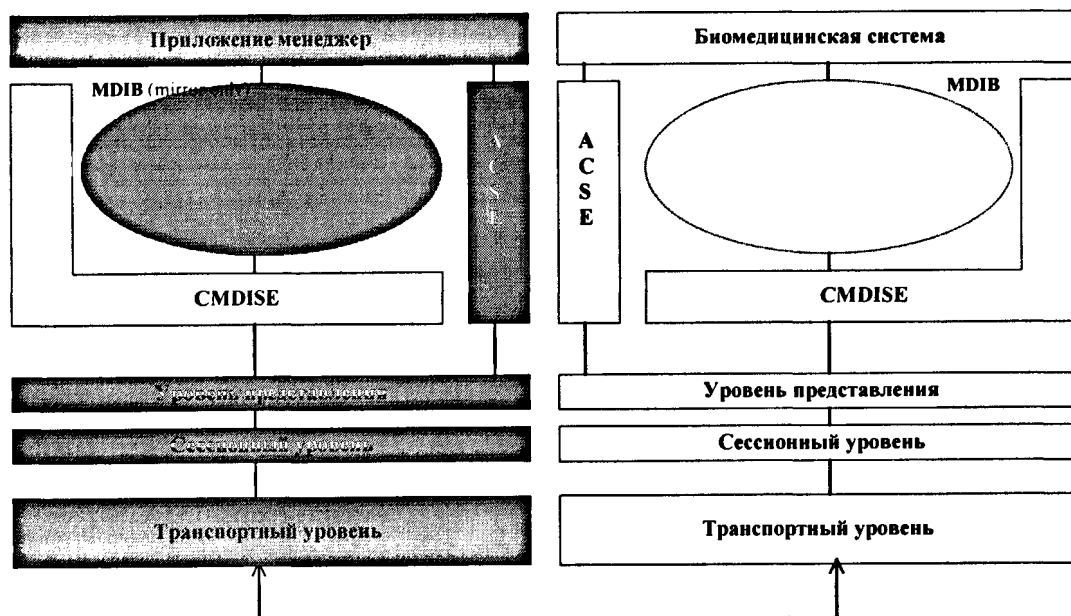


Рис. 3.9. Модель взаимодействующих биомедицинских систем.

Биомедицинская система, выступающая по отношению к приложению менеджеру в качестве программного агента, согласно модели предметной области IEEE 1073 является формальным обозначением для произвольной системы, обслуживающей одно или несколько биомедицинских устройств регистрации БМИ. Программой частью, обеспечивающей управление и доступ к ресурсам БУ, является контроллер биомедицинского устройства регистрации сигналов. Коммуникационный протокол строится на основе стандартных сетевых уровней ISO/OSI и поддерживается слоем Association Control Service Element (ACSE)[2], обеспечивающим логическую связь между взаимодействующими сторонами и уровнем Common Medical Device Information Service Element (CMDISE)[3], дающим доступ к внутренней объектной инфраструктуре приложения-агента, регламентируемой информационной базой медицинских данных (ИБМД) (англ. MDIB) стандарта IEEE 1073. MDIB является формальным описанием всех существующих в рамках логических моделей биомедицинских систем объектов. Данные объекты являются элементарными моделируемыми элементами из «реального мира» такими, как медицинская система (МС), виртуальное медицинское устройство (ВМУ), массив отсчетов реального времени (МОРВ), массив отсчетов

(МО), событие. Следовательно, в рамках данного подхода, детализация программной системы «IntelliCard» с выделением дополнительных компонент, не отраженных напрямую в MDIB, может быть представлена в следующей форме (см. Рис. 3.10).

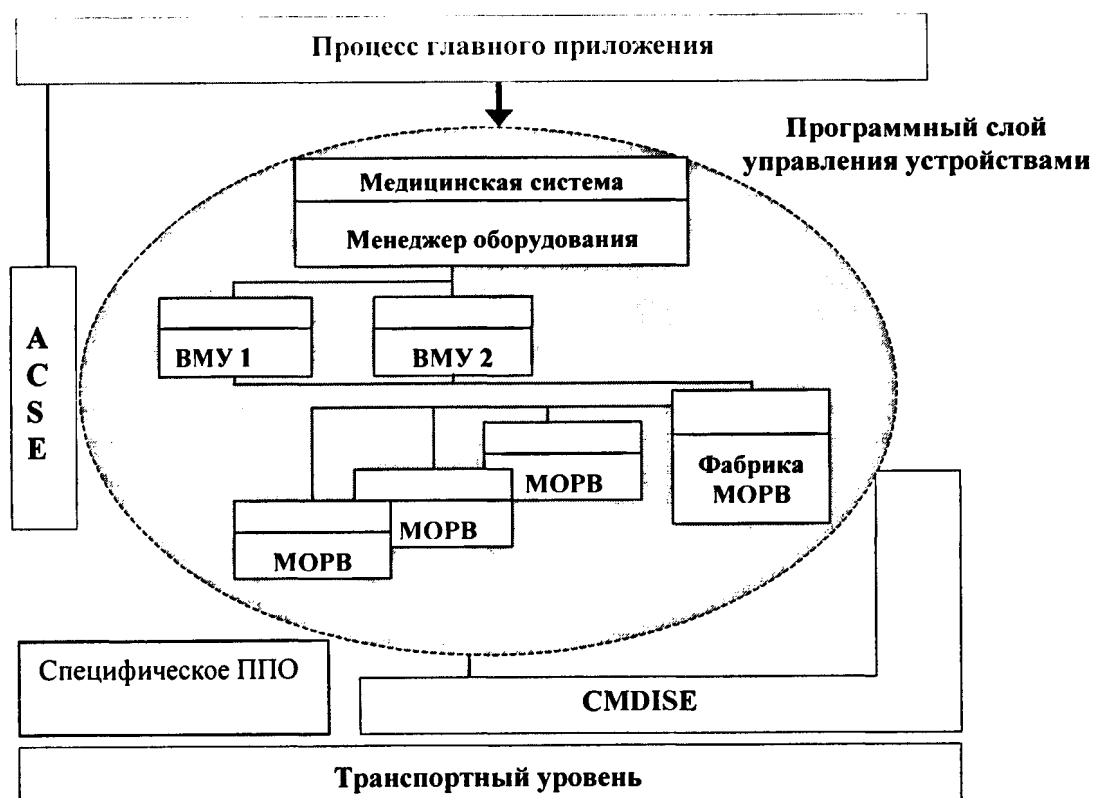


Рис. 3.10. Детализация подсистемы управления устройствами регистрации «Intellicard».

В целом, задача реализации подсистемы регистрации и распределения медицинских данных с устройств съема состоит во взаимно однозначном отображении представлений ИБМД в описания интерфейсов COM+.

Наиболее важным компонентом представленной схемы является ВМУ, который служит абстрактным представлением компонента, обслуживающего устройство регистрации БМИ, и компонент абстрактного доступа к оборудованию, обеспечивающего интеграцию с ВМУ. Принципы взаимодействия ВМУ построены на основе парадигмы программных агентов, как концепции позволяющей наиболее полно решить задачи децентрализованного управления устройствами. В рамках архитектуры «IntelliCard» ВМУ обладает следующими свойствами:

- 1) внутреннее состояние ВМУ однозначно соответствует состоянию физического устройства, изменения этих состояний строго детерминированы;

- 2) ВМУ обеспечивает функции управления и контроля состояний управляемого аппаратного устройства;
- 3) ВМУ управляет созданием, распределением массивов отсчетов реального времени и передачей событий.

Формально переходы состояния ВМУ могут быть заданы в виде конечного автомата (см. Рис. 3.11).



Рис. 3.11. Переход состояний виртуального медицинского устройства «Intellicard».

В медицинской системе может присутствовать несколько ВМУ, более того, не все присутствующие ВМУ могут иметь соединение с медицинской системой (МС). Ассоциирование ВМУ с МС означает подключение соответствующего ВМУ в схему компонента абстрактного управления оборудованием, после чего ВМУ становится управляемым БМС.

Следующей фазой является этап конфигурирования устройства, на котором решается проблема инициализация аппаратного обеспечения, настройка начальных параметров интерфейсов. В случае неудачного завершения данной фазы следует удаление ВМУ из контекста МС. После этапа конфигурирования устройство может быть переведено в состояние

работы в нормальном режиме, которое предусматривает получение информации от устройства на базе механизма событий.

Реализация ВМУ в виде программных агентов на основе архитектуры промежуточного ПО COM+ обеспечивает возможность дистрибуции источников данных. В частности, в этом возникает потребность при необходимости осуществлять запись БМИ с нескольких устройств, пространственно разделенных и сопряженных с различными рабочими станциями. В этом случае ВМУ может быть установлен на удаленной рабочей станции, при этом для всех остальных компонент системы ничего в структуре не изменится, поскольку взаимодействие и в этом случае будет осуществляться через подсистему регистрации БМИ, предоставляющей абстрактный интерфейс взаимодействия с оборудованием (см. Рис. 3.12). При этом запуск ВМУ может быть осуществлен в суррогатном процессе, предоставленном средствами поддержки COM+, на удаленной рабочей станции.

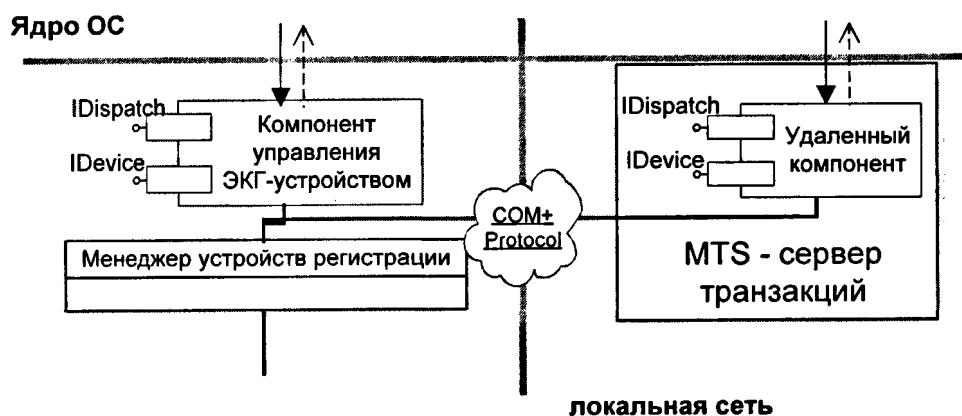


Рис. 3.12. Распределение элементов программной архитектуры.

Массивы отсчетов реального времени являются объектами, инкапсулирующими данные, передаваемые от узла обработки следующему в цепи обработки узлу. В гибридной модели дискретных событий, рассмотренной в главе 2, подобные объекты служат единственным носителем данных внутри системы и согласно имитационной модели имеют следующий набор свойств:

$Q_{MO}^* = (T_{MO}, \lambda_k, t)$, где T_{MO} - период времени, который связан с количеством отсчетов в случае эквидистантного сигнала, как $N = \frac{T_{MO}}{\Delta}$, Δ - период дискретизации, где λ_k - количество каналов, t - временная привязка первого отсчета в МО. В случае неэквидистантных отсчетов, например, при передаче сигнала от монитора артериального давления, отсчеты времени передаются в качестве дополнительного канала.

В рамках архитектуры «IntelliCard», МОРВ реализованы в качестве COM+ объектов, поэтому за их создание, поддержание в рабочем

состоянии и удаление должна отвечать операционная среда поддержки COM+. Однако, с точки зрения производительности, это весьма неэффективно, поскольку COM-библиотека приносит дополнительные временные затраты, связанные с поиском компонента в системном реестре, его настройкой. В «IntelliCard» данная проблема решена путем введения дополнительных вспомогательных компонент – фабрик МО, которые выполняют большинство операций COM+-библиотеки, но гораздо более эффективно, так как они работают с контейнерами данных как классами языка C++. Это позволяет пользоваться стандартными возможностями C++, операцией **new** и **delete** для создания и удаления объекта, получать прямой доступ к внутренним данным компонента. Таким образом, фабрики служат для создания, управления контейнерами данных и их удаления. Схематически реализация массива отсчетов может быть представлена следующим образом (Рис. 3.13).

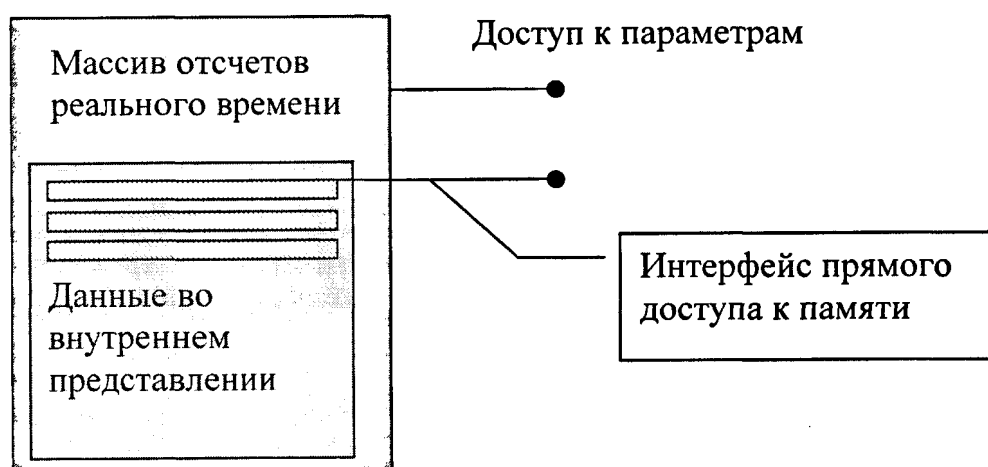


Рис. 3.13. Структура массивов отсчетов во внутреннем представлении «IntelliCard».

Единственным требованием архитектуры является поддержка массивом отсчетов интерфейса доступа к параметрам Q^* и поддержка дополнительного интерфейса для массивов реального времени, предоставляющего прямой доступ к памяти. Внутренняя организация подобного контейнера данных, каким является массив, зависит от требований конкретного приложения и типа хранимых данных, регистрируемого сигнала, результатов измерений сопутствующих величин. Исследования, результаты которых представлены в главе 2, позволяют говорить об эффективности матричного представления МОРВ. В случае, когда внутреннее представление данных отдельных сигналов МО осуществляется числовыми значениями с плавающей точкой одинарной точности, занимающими в памяти 4 байта на один отсчет, размер занимаемой одним массивом памяти описывается следующей формулой в зависимости от временного периода T :

$M(\tau) = M_{\text{com}} + M_{\text{конф}} + 4\lambda_k \tau F_{\text{диск}}$, где M_{com} – размер области для хранения внутренних переменных, требуемых COM+-библиотекой. Такая организация хранилища имеет ряд преимуществ, например, в этом случае возможна операция быстрого блочного копирования данных, доступ к одному отведению осуществляется, как к массиву.

Агрегатная имитационная модель, представленная во второй главе, содержит несколько типов разнообразных системных компонент «Intellcard»: источники данных (устройства регистрации, представляемые ВМУ), обработчики БМИ и потребители.

Формализованные в качестве агрегат, элементы программной архитектуры обеспечивают универсальный механизм установления взаимодействия, обмена данными, поддержания интерфейса и завершения соединения, схема которого представлена на Рис. 3.14.

Установление соединения и передача данных в рамках представленного механизма осуществляется в несколько этапов. На первом этапе после запроса потребителя информации, источник осуществляет проверку корректности интерфейса и в случае необходимости возможно изменение конфигурации соединения. Далее, интерфейс переходит в режим ожидания, в пределах которого возможны вызовы функций передачи данных, каждая из которых завершается проверкой корректности выполнения. В зависимости от результатов подобной проверки возникшая ошибка может быть обработана. Вызов функции завершения соединения в фазе ожидания информирует клиента о том, что сеанс передачи данных окончен, по крайней мере, до следующей установки соединения.

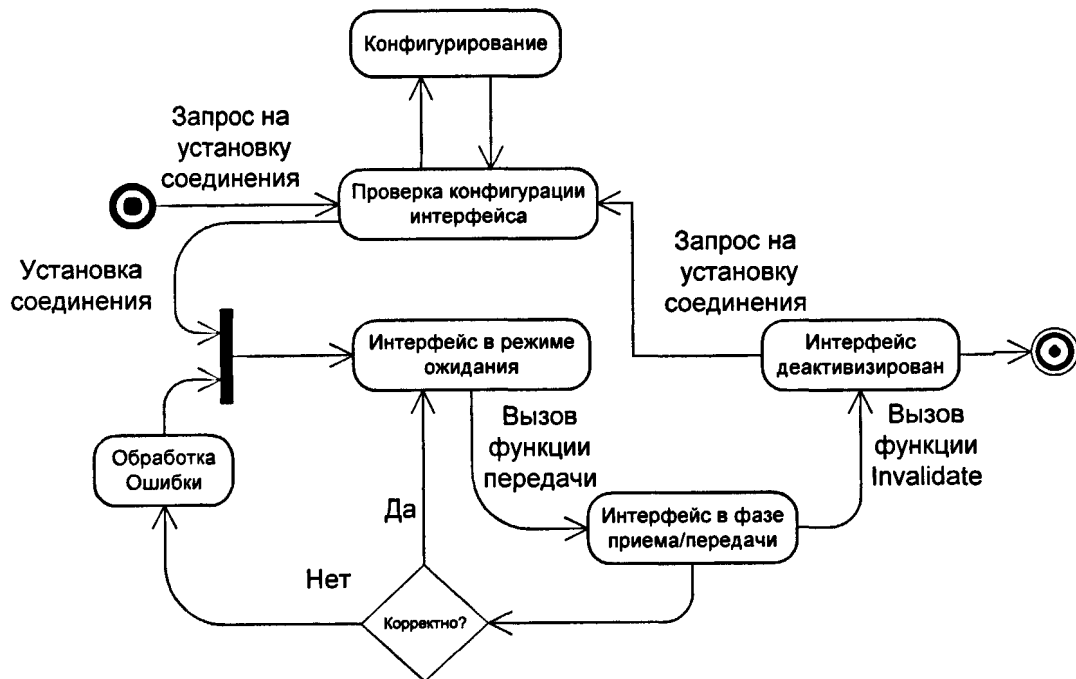


Рис. 3.14. Алгоритм работы интерфейса передачи данных между элементами системы обработки данных «IntelliCard».

В случае возникновения критической ошибки интерфейс соединения может вызвать переход в неактивное состояние.

3.2.2 Подсистема визуализации и печати биомедицинских данных

Основное назначение подсистемы визуализации любой системы регистрации электрофизиологической информации реального времени состоит в выполнении целого ряда функций, наиболее важными из которых являются:

1. отображение, передаваемой с различных источников регистрации или полученной в результате применения медицинских методик информации в наиболее удобном для восприятия врачом-специалистом виде;
2. своевременная реакция на изменившиеся параметры входных данных посредством графической или звуковой сигнализации.

В данном случае, время реакции зависит от ряда факторов [72]:

1. от времени доставки данных, зависящего от операционной системы (ОС), интерфейсных протоколов, обеспечивающих передачу информации от устройства регистрации до подсистемы визуализации;
2. от времени реакции медицинского работника, эксплуатирующего систему.

Таким образом, являющаяся основой любого интерфейса пользователя, подсистема визуализации выступает основным интерфейсом между системой регистрации и врачом-специалистом, интерпретирующим полученные данные с целью выработки диагностического заключения. Поэтому качественное представление информации является очень важным фактором качества диагностики.

Системы регистрации и обработки сигналов реального времени имеют два ярко выраженных режима визуализации: режим реального времени и режим повторного просмотра. В этом контексте печать рассматривается как частный случай режима просмотра с особым масштабом.

Современные методы диагностики выдвигают ряд серьезных требований к подсистеме визуализации, из которых наиболее важными и трудоемкими для реализации являются следующие:

- 1) поддержка синхронного отображения нескольких потоков данных, полученных от разных источников, имеющих различные частоты дискретизации и спектральный состав;
- 2) поддержка нескольких масштабов отображения данных по оси амплитуды сигнала и времени;
- 3) отображение вспомогательной и производной информации, например, отметки меток, событий и т.д.
- 4) поддержка нескольких форм отображения данных.

В режиме просмотра БМИ дополнительным требованием является поддержка измерения амплитуд и временных отрезков по двум и более произвольно установленным точкам. Таким образом, модуль визуализации является компонентом, обслуживающим несколько многоканальных потоков, причем объемы передаваемой информации через транспортное соединение могут достигать значительных размеров. Так передача стандартной 12-канальной ЭКГ с частотой дискретизации 1 кГц с учетом 4-байтового представления отсчетов с плавающей точкой составляет порядка 375 кБод, причем количество отсчетов, обрабатываемых в одну секунду, составляет 12000. Так как разрешающая способность современных стандартных средств визуализации составляет всего порядка 100-200 точек/дюйм, то для эффективного решения задач визуализации важной проблемой является сокращение количества отсчетов, которые используются для воспроизведения непрерывных сигналов при частотах дискретизации значительно превосходящих разрешающую способность с учетом выбранного масштаба. Согласно требованиям американской ассоциации кардиологов[73] рекомендуемыми минимальными частотами дискретизации являются 500 Гц, а для методик ЭКГ ВР – более 1 кГц, поэтому при отображении для большинства масштабов имеется значительная степень избыточности входных данных. Избыточность заключается в том, что различным отсчетам в исходном сигнале соответствует одна временная позиция на устройстве визуализации. Количество отображаемых отсчетов связано с заданным масштабом по оси X S_x следующим выражением:

$$N_x = f \frac{L_x \Delta_c}{S_x}, \text{ где } \Delta_c - \text{линейный размер зерна устройства отображения,}$$

f - частота дискретизации

Таким образом, избыточность при частотах дискретизации 500 Гц, масштабе 12,5 мм/с и линейном размере зерна электронно-лучевой трубки (ЭЛТ) 0,25 мм приводит к избыточности $N_{изб} = \frac{f \Delta_c}{S_x}$ равной 10 отсчетам на

одну отображаемую точку. Простейшим способом отображения подобных сигналов является визуализация кусочно-линейной функцией по всем отсчетам сигнала, однако это может привести к значительному падению производительности, поэтому необходима эффективная процедура фильтрации отсчетов.

Существует несколько возможных методов реализации процедуры прореживания, простейшим из которых является применение элементарного фильтра прореживания, заключающегося в последовательном взятии отсчетов с индексами k :

$$k_i = 1 + \frac{i}{S_x} f \Delta_c$$

Подобный метод является наиболее оптимальным с точки зрения производительности, однако обладает значительным недостатком – возможны потери острых пиков, в частности, R-зубцов, так как данный пик может достигаться произвольной точке на интервале $[k_i, k_{i+1}]$ исходного сигнала. Исходя из предположения, что ЭКГ обладает на интервале $[k_i, k_{i+1}]$, который достаточно мал, выпуклостью, можно сформулировать другой метод поиска отсчета на данном интервале:

$$C[k_i^{opt}] = \max_{k_i < j \leq k_{i+1}} |S[j] - S[k_{i-1}^{opt}]|$$

В случае, если масштаб таков, что $N_{изб} < 1$, то для формирования сигнала, может потребоваться интерполяция дискретного сигнала. Возможны любые методы интерполяции, в частности кусочно-линейной, биквадратной (сплайн-функциями) и т.д. Выбор того или иного метода диктуется рядом факторов: требуемой производительностью, максимальной абсолютной ошибкой.

Изложенный метод отображения данных дает значительные преимущества при малых масштабах, когда производительность определяется, во многом, количеством отображаемых пикселей. Экспериментальные данные по производительности подсистемы визуализации "Intellicard" при работе в режиме реального времени на различных масштабах представлены на Рис. 3.15. Параметры тестовой рабочей станции, участвовавшей в исследовании, следующие: ЦП Pentium III, 750, видео карта с аппаратной реализацией прокрутки без копирования растровых изображений (т.н. функции BitBlt).

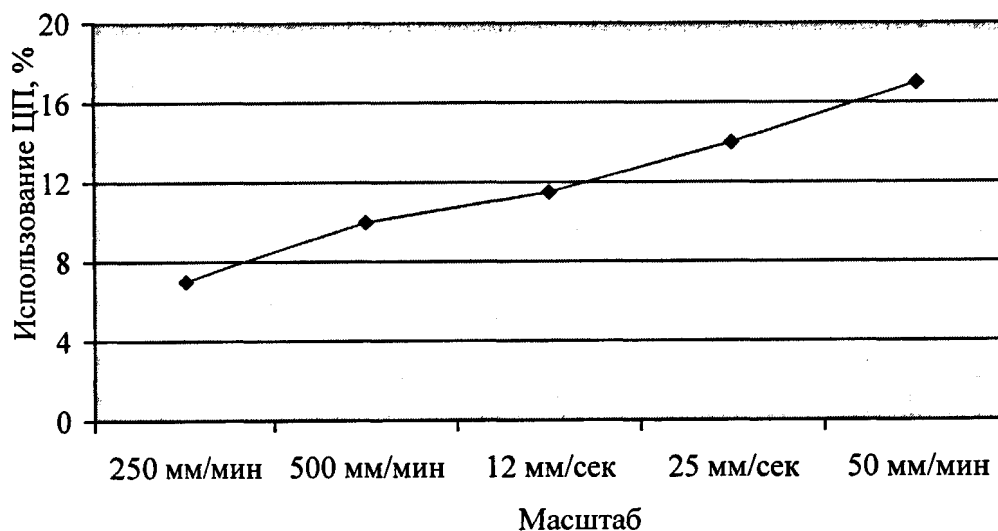


Рис. 3.15. Производительность подсистемы визуализации.

Использование видео адаптеров с достаточным объемом внутренней памяти и аппаратной поддержкой операции растрового копирования, позволяет сократить затраты ЦП до 2-5%. Однако количество таких

адаптеров до сих пор достаточно невелико и ограничивается профессиональным оборудованием.

3.2.3 Подсистема хранения и накопления биомедицинских данных

Важнейшей задачей подсистемы хранения и накопления данных является обеспечение надежного механизма краткосрочного и долгосрочного сохранения биомедицинских данных различного рода, а также поддержка эффективного доступа к ним. Под краткосрочным сохранением данных понимается временной период, не превышающий одной сессии работы программного приложения.

ЭКГ записи даже небольших длительностей занимают достаточно большие объемы. Простейший расчет показывает, что даже при регистрации ЭКГ со стандартной системой отведений, запись которой составляет порядка 1 минуты при частоте дискретизации 1000 Гц и квантования АЦП 16 бит, объем данных достигает:

$$W = 12 \cdot 1000 \cdot 16 = 192000 \text{ бит/сек или порядка } 1,4 \text{ МБ/мин.}$$

Поэтому при возрастании длительностей съема ЭКГ до минуты и более (например, в электрокардиографии физической нагрузки и холтеровском мониторинге) возникает острая необходимость в сжатии ЭКГ-сигнала. Такое сжатие может стать необходимым и при меньших объемах ЭКГ в случаях, когда необходимо сохранение большего количества записанных ЭКГ в интегрированной БД или существует необходимость в передаче ЭКГ посредством телефонной или мобильной связи, а также средствами сети Интернет.

С другой стороны, при записях ЭКГ большой длительности существует вероятность аппаратного или программного сбоя ЭВМ и потери уже обработанных данных и, следовательно, необходимо краткосрочное сохранение БМИ, позволяющее восстановить потерянные данные при необходимости, а также разгрузить оперативную память ЭВМ от излишней нагрузки (Рис. 3.16). Поэтому важность подсистемы накопления данных для ЭКГ-систем, построенных на платформе ЭВМ, обеспечивающей возможности сжатия биомедицинских данных и краткосрочного хранения (буферизации) достаточно высока.

В БМС «IntelliCard» хранилище обеспечивает настраиваемую на величины до нескольких минут буферизацию БМИ в рамках внутрисистемного буфера FIFO. При этом хранение данных осуществляется во внутреннем представлении системы. Краткосрочное хранение данных требует эффективной организации самого хранилища для обеспечения быстрого доступа к данным в режиме просмотра. Снижение объемов участвующей в обмене между дисковым хранилищем и буфером FIFO информации возможно двумя способами:

- а) уменьшения избыточности данных за счет сокращения количества бит для представления одного отсчета;

б) потокового сжатия данных.

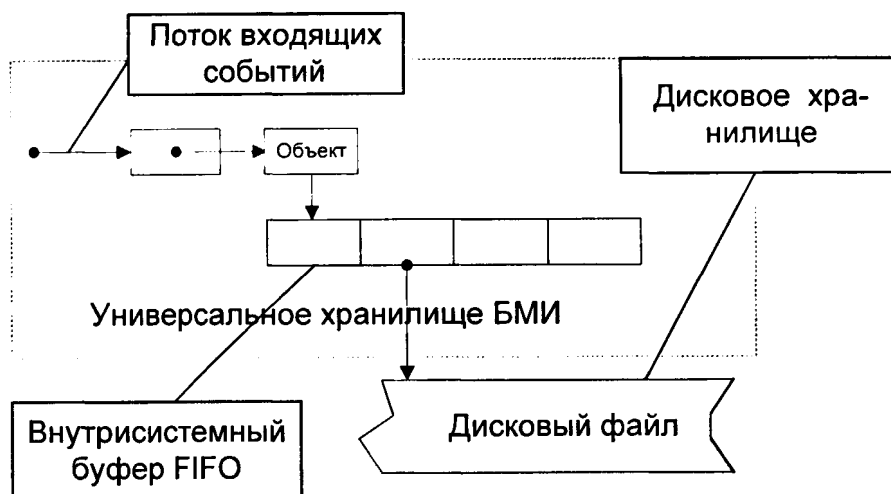


Рис. 3.16. Архитектура универсального хранилища БМИ «IntelliCard».

Уменьшение битности отчетов с 32-х до 16 бит в «IntelliCard» позволяет без значительных вычислительных затрат сократить интенсивность обмена с дисковым хранилищем. Сжатие данных на этом этапе имеет смысл лишь в том случае, когда сокращение объемов передаваемой информации дает выигрыш в производительности. Однако недостатки современных методов сжатия данных не позволяют использовать их в режиме реального времени и отводимая им роль сокращается до применений в реализациях функций импорта/экспорта в различные файловые форматы представления БМИ.

Цифровые записи электрофизиологических сигналов, в том числе и ЭКГ, представляют собой последовательности отчетов, поэтому обычные методики компрессии данных применимы и к ним (см. Прил. 2). Применение того или иного метода сжатия БМИ определяется требуемыми характеристиками метода (производительность, коэффициент сжатия), а также его свойствами, в частности, возможности работать в потоке, требуемыми ресурсами. Жестким требованиям реального времени способны удовлетворить достаточно узкий класс методов сжатия, способных к работе в потоке данных [74].

3.2.4 Подсистема внешних коммуникационных интерфейсов

Одним из наиболее важных элементов интеграции разнородных систем регистрации биомедицинских сигналов является выработка общих стандартов хранения и представления биомедицинских данных. Это является важным шагом на пути удешевления, универсализации и повышения качества биомедицинского программного обеспечения на базе

ЭВМ. Однако в решении проблем совместимости компьютерных электрокардиографов на уровне обмена данными за последние 30 лет серьезных изменений не произошло. Крупные фирмы-производители ЭКГ-приборов (Siemens, Hewlett Packard, Fukuda, NEC, Oxford, Del Mar и т.д.) разработали свои корпоративные стандарты, которые практически все являются закрытыми, т.е. недоступными для применения в системах других производителей.

Несомненным успехом процесса стандартизации явилась разработка принятого медицинским сообществом стандарта DICOM/MEDICOM[75], служащего, в основном, для представления изображений в радиологии. Стандартизация систем регистрации и обработки физиологических сигналов, начавшаяся сравнительно недавно и не подкрепленная мотивацией крупных производителей медицинских систем, не смогла выработать единый формат. Количество разнообразных спецификаций достаточно велико – каждый крупный производитель медицинского оборудования имеет в своем распоряжении свой собственный стандарт. Вместе с тем, в настоящее время существует ряд спецификаций и рекомендаций, принятых различными учреждениями по стандартизации, некоторые из которых являются национальными стандартами, поэтому достаточно сложной задачей является оптимальный выбор между существующими спецификациями.

Анализ круга задач, решаемых современными системами регистрации и обработки сигналов, позволяет сформулировать ряд требований, предъявляемых к форматам представления биомедицинских данных, наиболее важными из которых являются следующие положения:

- 1) универсальный формат представления данных, поддерживающий возможность представления сигналов разного рода: кардиологические, нейрофизиологические, респираторные и одновременного хранения разнородных сигналов, записанных с разной частотой дискретизации и разрядности отсчетов;
- 2) поддержка разнообразных форм аннотирования измерений: текстовая, голосовая, визуальная;
- 3) обеспечение механизма хранения второстепенных не привязанных к отметкам времени данных, например, результаты анализа записей, синдромальные заключения врача-специалиста;
- 4) многоуровневая система совместимости, при которой не требуется реализация полной функциональности, декларированной в стандарте, для достижения минимального уровня интеграции;
- 5) поддержка физических манипуляций с данными, с возможностью их осуществления в масштабе реального времени.

Дополнительными критериями, используемыми при анализе множества существующих стандартов, являются:

- 1) популярность стандарта в медицинской промышленности, которая оценивается процентным соотношением систем, поддерживающих данный формат;
- 2) простота реализации, которая обычно находится в обратном соотношении к объему спецификации;
- 3) компактность представления данных, наличие функций сжатия данных.

Существующие системы спецификаций можно разделить в целом на два класса. Первый класс составляют специальные стандарты, разработанные для узкой предметной области, в которых регламентации подвергается сама структура данных. Ко второй группе следует отнести, так называемые, метастандарты, в которых построена информационная модель предметной области, в рамках которой детализированы модели каждого предметного объекта, определена система связей компонент предметных объектов, разработана соответствующая номенклатура и система кодирования [76]. Под номенклатурой понимается некоторый список разнообразных типов данных, в представлении информационной модели – объектов, возникающих в процессе медицинского документооборота. В рамках таких метастандартов, любой формат представления данных является некоторым логическим построением разработанной метамоделю.

При реализации «IntelliCard» были рассмотрены ряд стандартов представления биомедицинских данных, оценены их качества на основе оценки соответствия представленным выше требованиям с целью определения необходимости их поддержки.

Самым ранним из стандартов, относящихся к первому классу, является спецификация EDF (European Data Format), появление которой относится к 1991 году. Первая спецификация, разработанная рядом инженеров, работающих в отрасли разработки биомедицинских систем, несмотря на отсутствие поддержки со стороны какого-либо института стандартизации, привлекла значительный интерес со стороны разработчиков систем [77]. EDF быстро завоевал относительную популярность, в основном, в области нейрофизиологии, благодаря, прежде всего, простоте реализации. Другим важным примером эффективного использования данного формата является возможность его применения в реферативных базах данных биомедицинских сигналов [78]. Структура формата представлена на Рис. 3.17. Файл данных имеет некоторый ASCII-заголовок, в котором содержатся основные данные о пациенте, условиях съема и длительностях сигналов. Дополнительно, для каждого из сигналов создается соответствующий заголовок, в котором содержится информация о типе сигнала и его физических характеристиках.

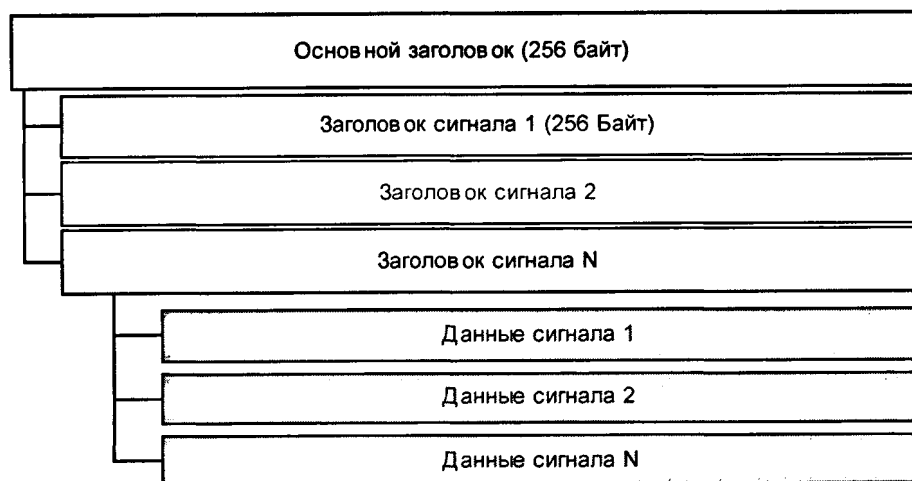


Рис. 3.17. Структура формата EDF.

Единственным стандартом представления нейрофизиологических сигналов, официально поддерживаемым организацией по стандартизации, является спецификация ASTM⁵ 1467[79], разработанная в соответствии с методологией создания сообщений стандарта обмена медицинскими записями Health Level-7. В соответствии с основной спецификацией обеспечивается поддержка представления основных электрофизиологических сигналов, а также ЭКГ. В дополнении к формату многоканальных записей, стандарт определяет множество вспомогательных элементов: идентификаторов каналов, параметров настройки фильтров, калибровочных данных, характеристик электродов, различных форм аннотирования и прочих параметров. Для представления этих параметров ASTM-1467 определяет множество типов данных, представляемых в виде ASCII-символов.

Стандартом, ориентированным, в основном, на задачи электрокардиографии, стал формат данных ENV1064[80], выработанный Европейским Институтом Стандартизации (CEN⁶), широко известный среди разработчиков электрокардиографов под названием SCP-ECG (Standard Communication Protocol for ECG)[81]. Являясь официальным стандартом и получив поддержку органов стандартизации, SCP-ECG быстро завоевал популярность среди производителей кардиологической техники. ENV1064 устанавливает единый протокол передачи ЭКГ-данных как между цифровым электрокардиографом и компьютеризированной системой управления, так и между компьютерными системами различных производителей. Стандарт SCP-ECG не накладывает ограничений на

⁵ American Society for Testing and Materials

⁶ European Committee for Standardization (CEN),

физический уровень протокола, а лишь определяет минимально необходимые требования для его поддержки. Он же регламентирует некоторые соглашения по передаче других данных: сведений о пациенте, результатов анализа ЭКГ, условиях проведения измерений и т.д.

Документ ENV1064 разбивает логическую последовательность ЭКГ данных (запись) на секции и описывает содержание и формат представления каждой секции. Начало каждой секции находится по четному адресу (смещению от 0). Т.о. все секции содержат четное количество байт - при необходимости секция дополняется нулевым байтом. Все секции имеют идентификационные номера (ID). Номера от 0 до 11 описаны в SCP-ECG протоколе (Табл. 3.2), номера от 12 до 127 и номера более 1024 включительно зарезервированы для будущего использования. Номера от 128 до 1023 предназначены для специфических секций производителей ЭКГ-устройств.

Табл. 3.3. Структура формата записи SCP-ECG.

Наличие	Содержание
Обязательно	2 байта Контрольная сумма CRC16 всех секций и размера всей записи
Обязательно	4 байта Размер всей записи в байтах
Обязательно	Секция 0 Список указателей на начало каждой последующей секции
Обязательно	Секция 1 Данные о пациенте (имя, идентификатор, пол, дата рождения и т.д.); данные об обследовании (дата, время, условия и т.д.)
Необязательно	Секция 2 Все таблицы Хафмена(при кодировании по Хафмену) для распаковки исходного сигнала (Секция 6) или репрезентативного комплекса (Секция 5) и разностного сигнала (Секция 6) в режиме "высокой компрессии"
Необязательно	Секция 3 Перечисление отведений, переданных в текущей записи, а также общей информации об их кодировании
Необязательно	Секция 4 Расположение QRS-комплексов (только в режиме "высокой" компрессии)
Необязательно	Секция 5 Для каждого отведения репрезентативный комплекс
Необязательно	Секция 6 Для каждого отведения исходный сигнал или, при использовании медианного кодирования, разностный сигнал, полученный путем вычитания репрезентативного комплекса из исходного сигнала

Необязательно	Секция 7 Общие для всех отведений измерения каждого комплекса в записи (длительности, углы поворота электрических осей и т.д.) и список артефактов от искусственного водителя ритма.
Необязательно	Секция 8 Текстовый диагноз от "интерпретирующего" устройства
Необязательно	Секция 9 Диагностические данные, специфичные для производителя
Необязательно	Секция 10 Измерения, произведенные для каждого отведения отдельно (амплитуды, длительности и т.д.)
Необязательно	Секция 11 Унифицированное закодированное заключение

Под репрезентативным комплексом понимается усредненный или наиболее типичный из всех найденных комплексов во всей записи ЭКГ. Вместе с желудочковым QRS комплексом в репрезентативном комплексе присутствует зубец Р (кроме случая фибрилляций предсердий) сегмент S-T и зубец Т.

В секциях 5 и 6 при кодировании цифровых сигналов допускается представление их в виде оригинальных отсчетов, первого или второго дифференциала.

В секции 11 используется система стандартных кодировок для получения унифицированного заключения. В отличие от этого в секции 8 необходимо сохранять заключение в текстовом виде. Секция 9 предназначена для записи уникальных диагностических данных, не специфицированных в описываемом стандарте. Если любая из секций 8, 9 или 11 присутствует, то не предполагается, что все три секции присутствуют.

Секции с 2 по 11 опциональны. Любая SCP-ECG запись должна в обязательном порядке содержать только технологическую секцию 0 (список указателей на начало каждой последующей секции) и секцию 1 (данные о пациенте; данные об обследовании). Предполагается наличие в записи и других секций, но проверка их присутствия не производится.

Таким образом, стандарт SCP-ECG является жестко ориентированным на задачи компьютерной электрокардиографии и его использование для представления произвольных электрофизиологических сигналов является затруднительным.

Спецификация формата CEN-TC251/FEF [82] для биомедицинских сигналов опирается на объектно-ориентированную модель и расширенную номенклатуру стандарта ENV 13734[83]. Номенклатура является перечнем разнообразных элементов, представляющих собой определенные типы медицинских данных, каждый из которых однозначно определяется

некоторым тегом, уникальным в своем контексте. Каждый такой элемент получен на основании моделирования предметной области, в данном случае биомедицинских систем регистрации и обработки данных. Элементы, обладая набором атрибутов и соответствующими бинарными представлениями, составляют информационную базу медицинских данных (ИБМД)⁷. Файл в формате FEF представляет собой последовательность секций, каждая из которых определяется своим тегом и содержит объектные элементы, кодирование которых осуществляется в соответствии со спецификацией ENV 13734. Укрупненная структура файла представлена на Рис. 3.18.

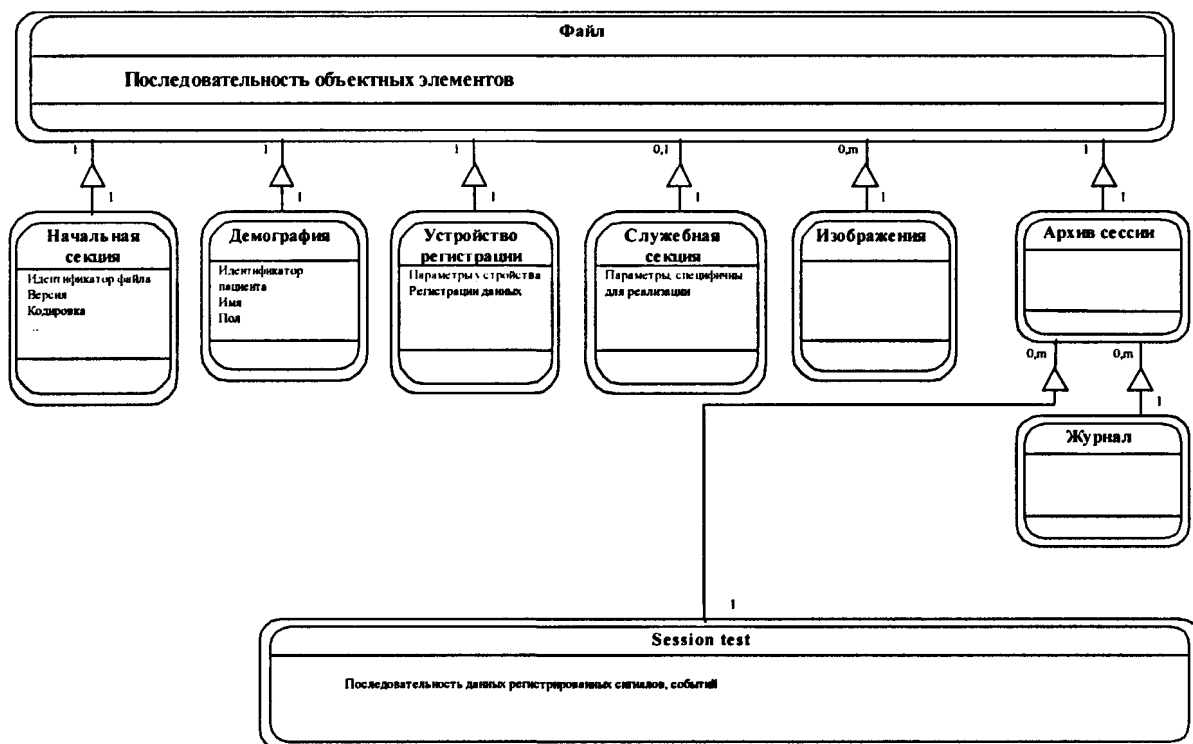


Рис. 3.18. Структура формата CEN-TC251/FEF.

Основные сравнительные характеристики представленных форматов сведены в Табл. 3.4. Кроме рассмотренных форматов в настоящий момент доступно ряд других спецификаций, однако они не имеют широкого распространения и поддерживаются узкими группами специалистов[84,85].

Таким образом, анализ разнородных стандартов показал, что имеет смысл поддержка стандарта SCP-ECG, как наиболее подготовленного для применения в электрокардиографии и EDF, ввиду его простоты и наличия большого количества данных (например, записей медицинских баз данных), представленных в данном формате. Стандарт CEN-TC251/FEF считается весьма перспективным, так как включает в себя возможности представления любых данных, представленных в ИБМД, покрывающей

⁷ Входит в семейство стандартов IEEE 1073.

фактические потребности ЭКГ, ЭЭГ, МЭГ и сопутствующих электрофизиологических исследований. Однако сдерживающим фактором применения стандарта CEN-TC251/FEF является сложность реализации, а также малая распространенность. Предполагается, что данная проблема будет разрешена, как только появятся реализации стандарта в виде встраиваемых библиотек. Таким образом, объем поддерживаемых форматов для «IntelliCard» был определен, как EDF, SCP-ECG, а также собственного формата представления данных.

Табл. 3.4. Сводная таблица характеристик основных стандартов представления электрофизиологических сигналов.

Характеристики	Стандарты			
	EDF	ASTM-1467	SCP-ECG	CEN-TC251/FEF
Поддержка сигналов разной природы	Любые эквидистантные сигналы	Нейрофизиологические измерения	Специфичен для ЭКГ	Согласно номенклатуре ИБМД
Формат кодирования данных/возможность сжатия данных	ASCII кодировка для параметрических данных и бинарная для сигналов/нет	ASCII кодировка/нет	Бинарная TLV-кодировка/да	Бинарная TLV-кодировка/да
Формы аннотирования	Текстовая	Текстовая	Текстовая	Текстовая Голосовая Визуальная
Система совместимости	Одноуровневая	Многоуровневая	Многоуровневая	Многоуровневая
Количество реализаций	Достаточно велико	Коммерчески доступные реализации не известны	Достаточно велико	Коммерчески доступные реализации редки
Сложность реализации	Низкая	Относительно высокая	Средняя	Высокая

Рассмотренный механизм интеграции систем на основе единых форматов файлов представления биомедицинских сигналов позволяет во многих случаях решить проблему несовместимости систем. Однако данный подход не может в полной мере решить проблему plug-and-play-взаимодействия систем, поскольку не обеспечивает поддержки динамического интерфейса и тем более режима реального времени. Поставленную задачу способен решить только единый стандарт взаимодействия устройств и информационных систем.

3.3. Анализ эффективности реализации элементов системы

Экспериментальная система «IntelliCard», благодаря своей открытой и гибкой архитектуре, является удобным средством отработки новых алгоритмов и аппаратных решений. Любой новый модуль может быть достаточно просто интегрирован в систему без изменений в структуре остальных компонент системы. Данное свойство комплекса позволяет использовать его как для тестирования и отработки новых алгоритмов путем интеграции их в рамках системы, так и для исследования эффективности реализации отдельных компонент системы. Это позволяет изолировать исследуемый компонент или сформировать именно ту рабочую конфигурацию, которая необходима для проведения анализа.

3.3.1 Планирование внутренних задач системы

Важнейшее значение при разработке сложных аппаратно-программных систем имеет анализ эффективности организации взаимодействия внутренних процессов. Данная проблема может быть решена путем построения временных диаграмм функционирования отдельных потоков выполнения. Наиболее целесообразным является исследование поведения системы в моменты максимальной загрузки, каковыми для систем реального времени являются периоды регистрации и обработки полученной информации. На временной диаграмме, представленной на Рис. 3.19, присутствуют четыре задачи: супервизор, осуществляющий планирование подчиненных процессов системы, задача распределения данных от произвольного ВМУ (в данном случае электрокардиографа), визуализации БМИ, а также обслуживания графического интерфейса пользователя. Согласно п.2.3.1, рассматриваемыми состояниями задач Windows NT являются: ожидание запуска (WS), активное состояние (AC), а также ожидание освобождения ресурсов ЦП (AW). На основании анализа динамики загрузки ЦП (см. п. 2.3.1), приоритеты выполнения представленных задач варьируются от максимального (задача супервизора) до минимального (обслуживание GUI). В простейшем случае интервального планирования задача распределения данных может иметь фиксированный период запуска, рассчитанный на этапе разработки системы. Решающим фактором для определения периода является учет того факта, что обработка данных (AC₁-AC₃) должна завершаться до следующего момента вызова функции передачи. Если по тем или иным причинам функция передачи не смогла обработать данные за определенный период, то супервизор может принять решение об увеличении периода обработки. Задача визуализации БМИ также управляется супервизором путем произвольно изменяемого периода вызова, однако при этом необходимо использовать точные процедуры измерения промежутков времени для повышения качества плавности отображения.

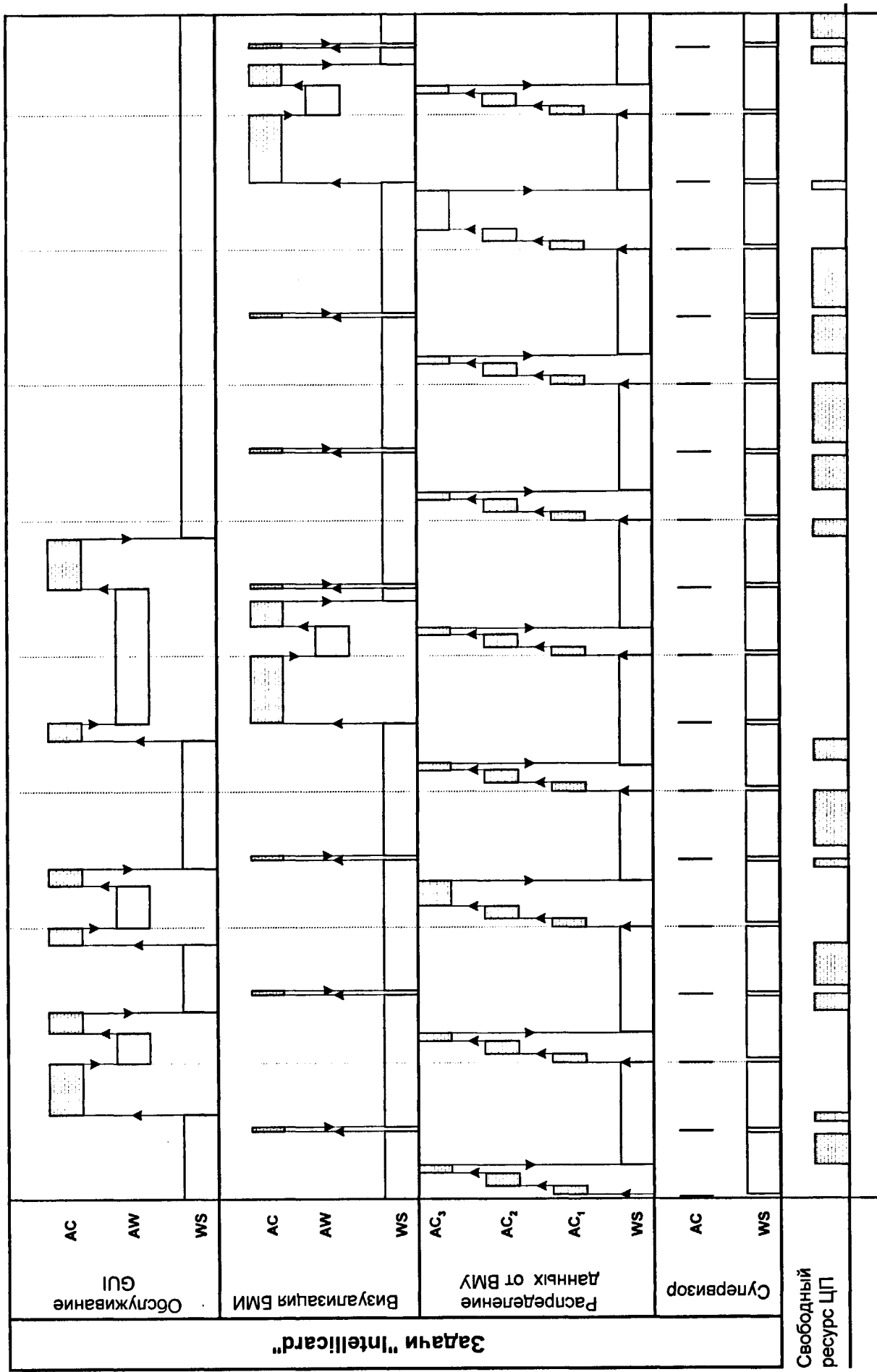


Рис. 3.19. Временная диаграмма взаимодействия задач "Intellcard".

Обслуживание пользовательского интерфейса является асинхронной, непланируемой задачей, поэтому ей присвоен минимальный приоритет. Данная задача может быть вытеснена любым другим потоком исполнения. Супервизор имеет максимальный приоритет, доступный пользовательским процессам в Windows NT, и способен вытеснить как выполняющиеся задачи системы, так и ряд процессов ядра ОС. При этом он имеет возможность в критической ситуации остановить работу системы, когда производительность по тем или иным причинам является недостаточной. Большое значение на рассматриваемой диаграмме имеет анализ динамики свободного ресурса ЦП, который позволяет оценить мгновенную нагрузку ОС. Недостатком представленного анализа на основе временной диаграммы является отсутствие учета влияния других процессов ОС на функционирование программной системы, так как такое исследование провести достаточно тяжело.

3.3.2 Исследование производительности программной реализации на основе COM+

Важным показателем эффективности любой объектно-ориентированной архитектуры может служить отношение времени, затраченного на поддержку специфических механизмов обработки данных и дополнительных сервисных функций компонентной модели, к общему времени, выделенному приложению ОС. Подобным способом можно оценить дополнительную вычислительную нагрузку, приходящуюся на реализацию компонентно-ориентированной программной архитектуры, по сравнению с монолитной [86]. Это позволяет провести оценку дополнительной нагрузки, вносимой реализацией компонентной архитектуры. Для проведения статистических исследований необходимо подготовить инструментальную сборку ПО системы, что позволяет провести вычисления затраченного времени на выполнение каждой функции, которое является суммой периодов выполнения за время сбора статистики. Дополнительным параметром, участвующим в оценке, является количество вызовов той или иной функции в единицу времени. Каждая функция может быть отнесена к реализующему ее модулю и характеризуется, как основная, т.е. реализующая заданные операции, а также дополнительная, присутствие которой диктуется требованиями компонентной архитектуры.

Количественные значения коэффициента использования ЦП, полученные в результате нескольких испытаний системы "IntelliCard" в инструментальном режиме приведены в Табл. 3.5.

Полученные данные позволяют утверждать, что в случае применения COM+ подобное соотношение не превышает 4-6% от общего времени ЦП, выделенного приложению операционной системой, что является достаточно хорошим показателем для систем такого рода.

Табл. 3.5. Распределение коэффициента использования ЦП между компонентами системы.

Системный компонент	Коэффициент использования ЦП		
	U _i (в процентах)		
	U _i	U _i ^{СОМ+} процедуры поддержки СОМ+ (%)	U _i ^{СОМ+} / U _i (%)
Контроллер электрокардиографа	5.0%	0.05%	1%
Фабрика контейнеров	0.1%	0.1%	100%
Унифицированное хранилище БМИ	0.2%	0.05%	25%
Менеджер компонент	1.5 %	0.8%	53%
Модуль предварительной обработки БМИ	3.0%	0.3%	10%
Визуализатор БМИ	14%	0.08%	<1%
Клиент БД пациентов	0.2%	0.0	0
Модуль управления устройством регистрации	0.2%	0.08%	40%
Прочие компоненты	1%	0	0
Система	<26%	<1.5%	<6%

3.3.3 Оценка затрат на реализацию дополнительных требований программной архитектуры

Использование любой унифицированной архитектуры, несмотря на те преимущества, которые она предоставляет, подразумевает наличие дополнительных затрат, связанных с реализацией требований той или иной программной модели. Оценка дополнительных трудозатрат, связанных с реализацией специфических функций СОМ+, произвести достаточно сложно, так как само использование компонентной модели подразумевает принципиально иную схему программного обеспечения. В таблице 3.5 приведена оценка количества исполняемых строк исходного кода ряда компонент с выделенным числом строк, реализующих требования компонентной модели. Исходный код, динамически генерируемый средствами разработки, не участвует в подсчете.

Табл. 3.5. Распределение затрат на реализацию функций программной модели.

Системный компонент	Общее количество строк	Количество строк кода поддержки СОМ+	Отношение, %
Контроллер электрокардиографа	9843	200	2%
Унифицированное хранилище БМИ	1896	400	21%
Менеджер подключаемых компонент	1501	150	10%
Визуализатор БМИ	12060	250	2%
Модуль предварительной обработки БМИ	7443	200	3%
Модуль управления устройством регистрации	4304	250	6%

Таким способом можно рассчитать объем ожидаемых трудозатрат, соотнеся его с общим размером реализации, и оценить ожидаемую экономическую целесообразность.

Глава 4. Прикладное математическое обеспечение компьютерной биомедицинской системы «IntelliCard»

Компьютерные системы регистрации и обработки электрофизиологических сигналов, осуществляющие запись и обработку БМИ непосредственно во время записи сигнала, относятся к системам реального времени, поскольку функциональная часть этих систем должна непрерывно обрабатывать и анализировать поступающую информацию и оперативно, по возможности с меньшей задержкой, реагировать на изменившееся состояние внешних параметров. Так, для систем компьютерной электрокардиографии поступающей информацией является ЭКГ-сигнал, а внешними параметрами – характеристики записываемого сигнала: частота сердечных сокращений (ЧСС), фактор присутствия фибрилляций и т.д. В силу этих специфических особенностей ЭКГ-системы реального времени предъявляют повышенные требования к алгоритмам, реализующим обработку поступающей информации. Среди наиболее важных требований к подобным алгоритмам следует выделить:

- 1) соответствие производительности алгоритма скорости входящего потока данных, т.е. период дискретизации входной выборки T не должен превышать время τ_s обработки системой порции данных;
- 2) гарантированность получения результата или выработки управляющего воздействия в течение заранее заданного промежутка времени;
- 3) возможность эффективной работы при ограниченной доступности входных данных, т.е. при доступности данных для времен $t < t_{\text{тек}}$, где $t_{\text{тек}}$ - текущее время системы.

Последнее требование является существенным для систем реального времени, поскольку его выполнение требует определенных качеств самого алгоритма и не может быть обеспечено вспомогательными аппаратно-программными средствами.

В данной главе рассмотрены методы математической обработки электрокардиограмм и особенности их реализации в системе «IntelliCard». Рассмотрение методов обработки других электрофизиологических сигналов находится за рамками данной работы.

В подсистеме обработки данных «IntelliCard» можно выделить несколько функциональных модулей:

- 1) расчет отведений и предварительная обработка электрофизиологических сигналов;
- 2) буферизация, сжатие и накопление биомедицинских данных электрофизиологических исследований;
- 3) подготовка биомедицинских данных к визуализации и печати;

- 4) выделение QRS-комплексов и измерение параметров элементов ЭКГ;
- 5) реализация разнообразных дополнительных медицинских методик компьютерной кардиографии в том числе, анализа variability сердечного ритма;

Поскольку присутствие подобных модулей с требуемым набором реализованных свойств в составе комплекса во многом диктуется требованиями, предъявляемыми к системам ЭКГ в целом, большое значение при реализации имеет разумный компромисс между эффективностью того или иного метода и его и производительностью.

Компонентно-ориентированные системы, построенные на основе ПО промежуточного уровня, предъявляют некоторые дополнительные требования, которые не являются специфическими для медицинских систем. Среди подобных требований по степени важности можно выделить следующие:

- 1) универсализм отдельных компонент системы;
- 2) единство интерфейса манипуляций с данными;
- 3) минимизация затрат на реализацию дополнительных подсистем, необходимых согласно требованиям выбранной модели компонентной архитектуры.

Разработка систем реального времени в компонентно-распределенной среде, имеет свои особенности, которые и рассматриваются в данной главе на примере реализации подсистемы обработки данных «IntelliCard».

4.1. Предварительная обработка ЭКГ реального времени

Предварительная обработка сигнала в компьютерных ЭКГ-комплексах заключается, в основном, в цифровой фильтрации ЭКГ. Цель фильтрации - подавление неблагоприятных факторов, таких как шумы, выделение желаемых характеристик сигнала и его коррекция.

4.1.1 Характеристики и классификация шумов, присутствующих в электрокардиограмме

Шумы, присутствующие в электрокардиограмме, можно разделить на две основные категории:

- 1) шумы, обусловленные присутствием внешних физических факторов;
- 2) инструментальные шумы, связанные с особенностями реализации устройства регистрации и программного обеспечения обработки ЭКГ.

Внешние факторы обусловлены физиологическими особенностями человека, условиями проведения эксперимента, а инструментальные –

связаны с особенностями измерительной техники, применяемой для регистрации сигнала.

Среди помех, обусловленных присутствием внешних факторов, наиболее значимыми являются [87]:

- 1) помехи силовых электрических линий;
- 2) шум контактирующих электродов;
- 3) электрическая активность мышц;
- 4) скачки или смещение изолинии;
- 5) дрейф изолинии вследствие дыхательной активности;
- 6) шум периферийного оборудования.

Особенностью вышеперечисленных помех является то, что отсутствует реальная возможность значительно уменьшить или полностью нейтрализовать воздействия источников этих шумов, поэтому единственным решением проблемы является аппаратное или программное их подавление.

Помехи внешних силовых электрических линий представляют собой наводку, являющуюся суперпозицией гармоник разных фаз, с частотами кратными частоте колебаний напряжения в силовой линии. Первая гармоника, имеющая наибольшую мощность, равна частоте колебания электрического тока в сети, что составляет 50 Гц в Европейских странах и 60 Гц в США и Канаде.

Контакт электродов и поверхности кожи имеет явно выраженный емкостной характер, поэтому кратковременное нарушение контакта может приводить к появлению на ЭКГ артефактов, имеющих вид скачков большой амплитуды. Подобная потеря или нарушение контакта электродов носит кратковременный характер (до нескольких миллисекунд) и часто не может быть идентифицирована как отрыв электрода.

Электрическая активность мышечной массы человека приводит к появлению на ЭКГ некоторой помехи, значительная спектральная мощность которой сосредоточена в диапазоне 30-40 Гц. Амплитуда шума зависит от интенсивности процессов сокращения, происходящих в мышцах в период регистрации ЭКГ.

Смещение электродов при записи ЭКГ приводит к изменению поверхностного омического сопротивления в месте касания электрода, что в свою очередь может приводить к смещениям уровня сигнала ЭКГ - скачкам изолинии с амплитудой достигающей нескольких милливольт.

Дыхательная деятельность человека приводит к появлению еще одной помехи - дрейфа изолинии, имеющего низкочастотный спектральный состав (основная мощность сосредоточена в области менее 1 Гц).

Шум внешних периферических устройств заключается в разнообразном широкополосном электромагнитном излучении, генерируемом разнообразными электромагнитными приборами: устройствами визуализации, обработки и передачи данных. В случае компьютерной ЭКГ

шумами внешних устройств являются помехи, генерируемые аппаратными блоками ЭВМ, такими как процессорная плата, импульсный источник питания, системные шины передачи данных, шумы которых характеризуются достаточно высокой частотой и при обычных частотах регистрации до 2 кГц могут смешиваться с частотами полезного сигнала ЭКГ. Такого смешения часто возможно избежать лишь путем аппаратной ФНЧ-фильтрации аналогового сигнала до фазы аналого-цифрового преобразования в регистраторе ЭКГ, т.н. антиэлайсинговым ФНЧ. Интерфейс ввода ЭКГ в ЭВМ должен обеспечивать надежное экранирование цифрового сигнала от внешних помех.

Таким образом, в спектре ЭКГ можно выделить несколько полос, в которых обычно сосредоточена основная мощность шума, в то время как мощность полезного сигнала в данных областях минимальна. В частности, такими полосами являются:

1. частота менее 0.5 Гц – основная мощность шума дрейфа изолинии;
2. 50/60 Гц – шумы электрических силовых линий;
3. 35-40 Гц – шумы электрической активности мышц;
4. более 100 Гц – высокочастотные шумы разной природы.

Следовательно, модель ЭКГ с присутствующими шумами внешней природы схематически может быть сформулирована следующим образом (Рис. 4.1). В данном случае, в качестве исходного сигнала выступает идеальный ЭКГ-сигнал, полученный на основе общих представлений о ЭКГ из теории. Подобный сигнал может быть получен при помощи тестовых генераторов ЭКГ, использующих различные методы получения эталонного сигнала. Модель представляет дрейф изолинии синусоидой заданной частоты, шумы электрических линий синусоидами 50 и 60 Гц, соответственно. Высокочастотный шум и шум электрической активности мышц учитывается синусоидальным сигналом переменной частоты, находящейся в заданном диапазоне. Дополнительно в схему введен генератор случайного шума, а также переключаемое смещение, позволяющее представить смещение или отрыв электродов.

Представленная модель, разработанная в визуальной среде SIMULINK пакета Matlab, может быть произвольно расширена путем введения дополнительных источников шума. Это позволяет ее использовать для оценки эффективности процедур подавления помех, в частности, фильтрации. Эффективность может быть оценена количественно на основе вычисления отношения мощности шума в выделенных областях до и после процесса фильтрации.

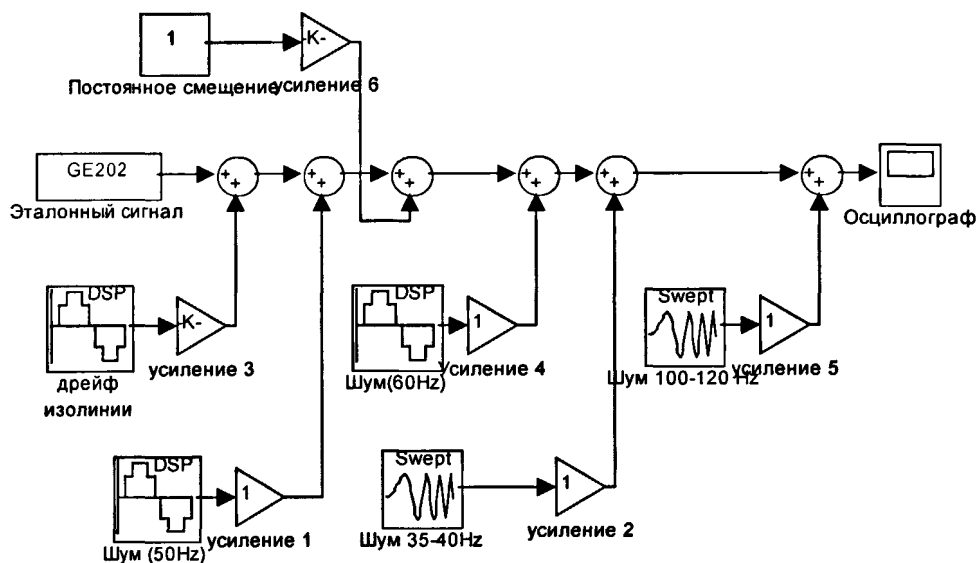


Рис. 4.1. Модель шумов ЭКГ.

Ко второй категории шумов относятся помехи, обусловленные особенностями схемотехники регистратора ЭКГ, реализацией канала передачи данных к управляющей ЭВМ или выбранными алгоритмами обработки. Среди подобных помех по степени важности следует выделить следующие:

- 1) инструментальный шум квантования АЦП регистратора ЭКГ;
- 2) температурно-зависимое смещение электрического потенциала на выходе АЦП;
- 3) шумы квантования цифровых фильтров, если представление сигналов осуществляется числами с фиксированной точкой;
- 4) ошибки в канале передачи данных и ряд других помех.

Шумы этой категории допускают возможность нейтрализации или уменьшения воздействия источников этих помех на ЭКГ. Шумы аналого-цифровых схем электрокардиографа определяются использованием конкретной элементной базы и обычно характеризуются номинальными параметрами использованных схем. Необходимо, чтобы ЭКГ-регистраторы обладали схемотехническими решениями, позволяющими минимизировать шум аналоговых схем, подавлять внешние наводки. С этой точки зрения очень интересным представляется схемотехническая реализация кардиографа, который был использован в данной работе, на основе сигма-дельта АЦП[88]. Основные шумовые и динамические характеристики такого аппаратного решения представлены в следующем параграфе.

В сигма-дельта АЦП частота дискретизации во много раз больше удвоенной верхней граничной частоты аналогового сигнала (частоты

Найквиста)[89], что позволяет увеличить число значащих разрядов АЦП и улучшить шумовые характеристики преобразования. С использованием в схемотехнике электрокардиографов сигма-дельта АЦП [90] в первую очередь решается проблема входного диапазона. Постоянная составляющая на входе ЭКГ компенсируется за счет 5-6 дополнительных бит сигма-дельта АЦП. Последний обеспечивает 16-17 значащих разрядов в требуемой полосе частот, то есть перекрывает динамический диапазон входного сигнала как минимум в 105 дБ. Схемотехника сигма-дельта АЦП такова, что постоянное смещение, составляющее до 6 старших значащих бит, зависит и от температуры. Это не приводит к потере точности при дискретизации, однако обработка и визуализация сигнала до его коррекции невозможна. Учет влияния температуры на величину смещения на выходе АЦП может быть учтен соответствующей калибровкой.

Шум квантования цифровых фильтров состоит в том, что округление коэффициентов линейных цифровых фильтров может часто приводить к значительному отклонению АЧХ от расчетной или к неустойчивости фильтра в целом [91]. Округление коэффициентов может потребоваться при реализации целочисленной фильтрации или при уменьшении количества значащих знаков в действительном представлении чисел с плавающей точкой. Спектральный состав этого шума может произвольно варьироваться и обычно определяется на этапе расчета цифрового фильтра. В этот же момент, основываясь на результатах анализа устойчивости, выбирается наиболее эффективная реализация фильтра.

Ошибки канала передачи данных в большой степени определяются выбранным интерфейсом связи с ЭВМ, обрабатывающей БМИ. Использование доступного широкого набора интерфейсов и выбор соответствующей аппаратной реализации позволяет переложить контроль над целостностью передаваемых данных на аппаратную реализацию. Наиболее перспективными в настоящее время являются интерфейсы USB [92], SCSI, и Bluetooth, выбор которых диктуется, прежде всего, спецификой целей, которые ставятся перед компьютерной БМС.

4.1.2 Методы подавления ряда инструментальных шумов

Недостатки классических схем построения регистраторов ЭКГ приводят к необходимости использовать новые схемотехнические решения. В течение последнего времени появилось достаточно много сообщений, об использовании схемотехники сигма-дельта АЦП в электрокардиографии [93,94,95,96]. Анализ инструментальных шумов ЭКГ-усилителей проведем на примере модели SIMULINK пакета Matlab устройства на основе сигма-дельта АЦП, которое также используется в системе «IntelliCard».

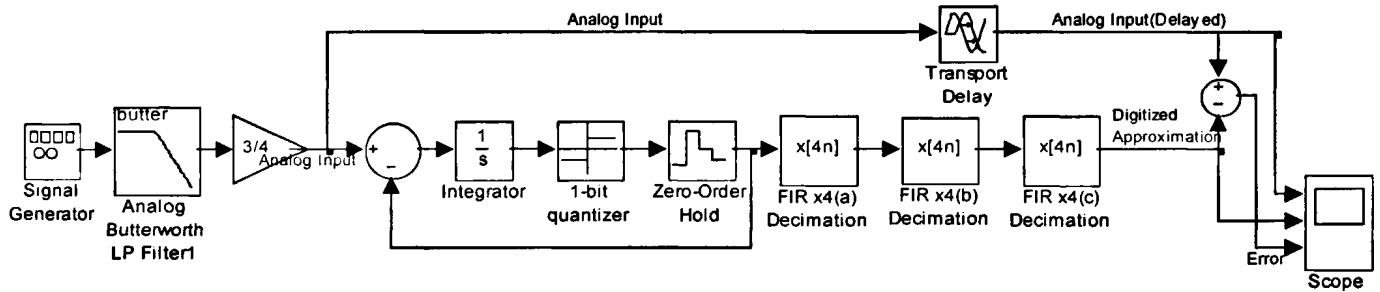


Рис. 4.2. Модель сигма-дельта аналого-цифрового преобразования.

В данной модели выходной аналоговый сигнал генератора предварительно обработан антиэлайсинговым ФНЧ Батерворда. Замкнутая цепь обратной связи состоит из вычитающего устройства, интегратора, компаратора (1-бит АЦП) и 1-бит ЦАП. Этот ЦАП принимает последовательный поток данных, преобразовывает его в аналоговый вид и данный сигнал с его выхода вычитается из входного сигнала. Интегратор может быть представлен как фильтр, амплитуда отклика которого пропорциональна $1/f$, где f – частота входного воздействия. Разностный сигнал «дельта» накапливается интегратором и составляет т.н. значение «сигма». Компаратор синхронизируется тактовыми импульсами, следующими с частотой, в данном примере, kF_s , (где k – коэффициент передискретизации в данной модели равный 64) преобразуя медленный входной сигнал в сигнал переменного тока высокой частоты, меняющейся в зависимости от среднего значения напряжения на входе. Таким образом, эффективное значение шума квантования на низких частотах пренебрежимо мало, а интегратор выступает в роли фильтра высоких частот для шума квантования 1-битного АЦП, смещая шум квантования в область высоких частот (k частотам $kF_s/2$, т.е. за границу желаемой частоты преобразования F_s) (Рис. 4.3.)

Для различных входных величин в одном интервале квантования данные от 1-бит АЦП значат мало. Только когда накоплено большое число отсчетов, мы получим результирующее значение. Если входной сигнал близок к положительному краю полной шкалы, то в битовом потоке на выходе больше единиц, чем нулей, и наоборот, если сигнал ближе к отрицательному краю, то больше нулей. Для восстановления цифрового сигнала в данной модели применяется каскад КИХ-фильтров прореживания, который сокращает исходную частоту дискретизации и подавляет шум квантования, внесенный петлей обратной связи, а также прочие высокочастотные шумы, не устраненные входным аналоговым фильтром (Рис. 4.3). Таким образом, мощность шума квантования сигма-

дельта АЦП незначительна и может быть подавлена путем НЧ-фильтрации с частотами 100-130 Гц.

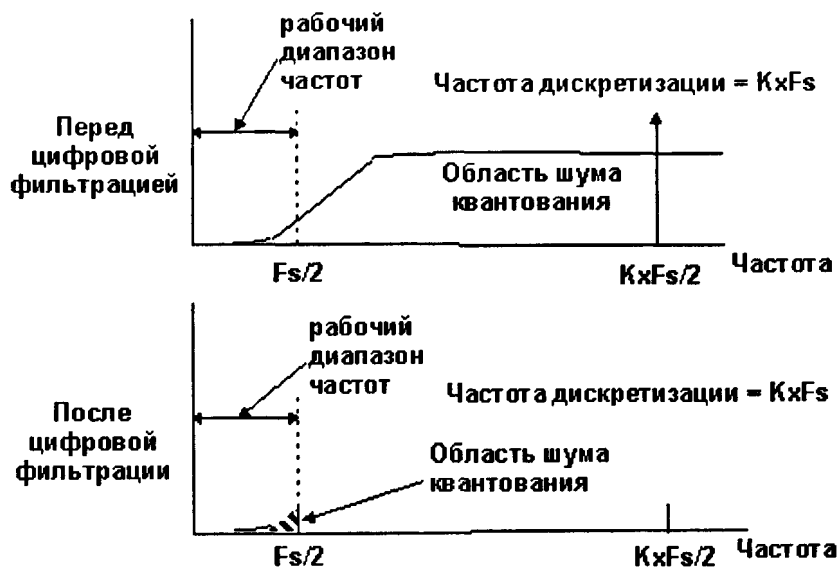


Рис. 4.3. Распределение шума квантования до и после децимации.

Еще одним классом шумов, подавление которых необходимо на самом раннем этапе обработки, является постоянное смещение выходного сигнала АЦП, достигающего величин, превышающих в несколько раз амплитуду полезного сигнала, а также обрывы контактов электродов с кожным покровом, приводящих к появлению на ЭКГ разнообразных артефактов. Постоянное смещение объясняется особенностями метода записи электрокардиограммы, в частности, в случае сигма-дельта АЦП такой причиной является отсутствие аппаратных ВЧ-фильтров ввиду широкого динамического диапазона АЦП. Так как амплитуда подобного смещения может быть достаточно велика, применение ВЧ-фильтрации на основе КИХ- или БИХ-фильтров может вносить определенные искажения в спектральный состав полезного сигнала. Подобные проблемы могут быть решены следующим образом, как это реализовано в программном модуле виртуального медицинского устройства (ВМУ) системы «IntelliCard».

Коррекция смещения необходима отведениям I, II, V1, V2, V3, V4, V5 и V6, поскольку остальные могут быть вычислены из уже рассчитанных потоков. Параллельно для каждого сигнала отведения $S_{lead}[n]$, вычисляется абсолютное значение первой разности в каждой точке $\Delta S_{lead}[n] = |S_{lead}[n] - S_{lead}[n-1]|$. Критерием отсутствия артефактов отрыва электрода является выполнение следующего соотношения $\Delta S_{lead}[n] < S_{крит}$, где $5\text{ мВ} < S_{крит} < 7\text{ мВ}$. Для определения оценки смещения возможно произвести усреднение

первых n отсчетов $D = \frac{1}{n} \sum_{i=1}^n S[i]; S_{lead}^{cor}[n] = S_{lead}[n] - D$. N целесообразно

выбрать от четверти до половины секунды и отображать во время вычисления среднего калибровочный прямоугольник. Вычисление смещения необходимо проводить каждый раз после обнаружения отрыва электрода (Рис. 4.4).

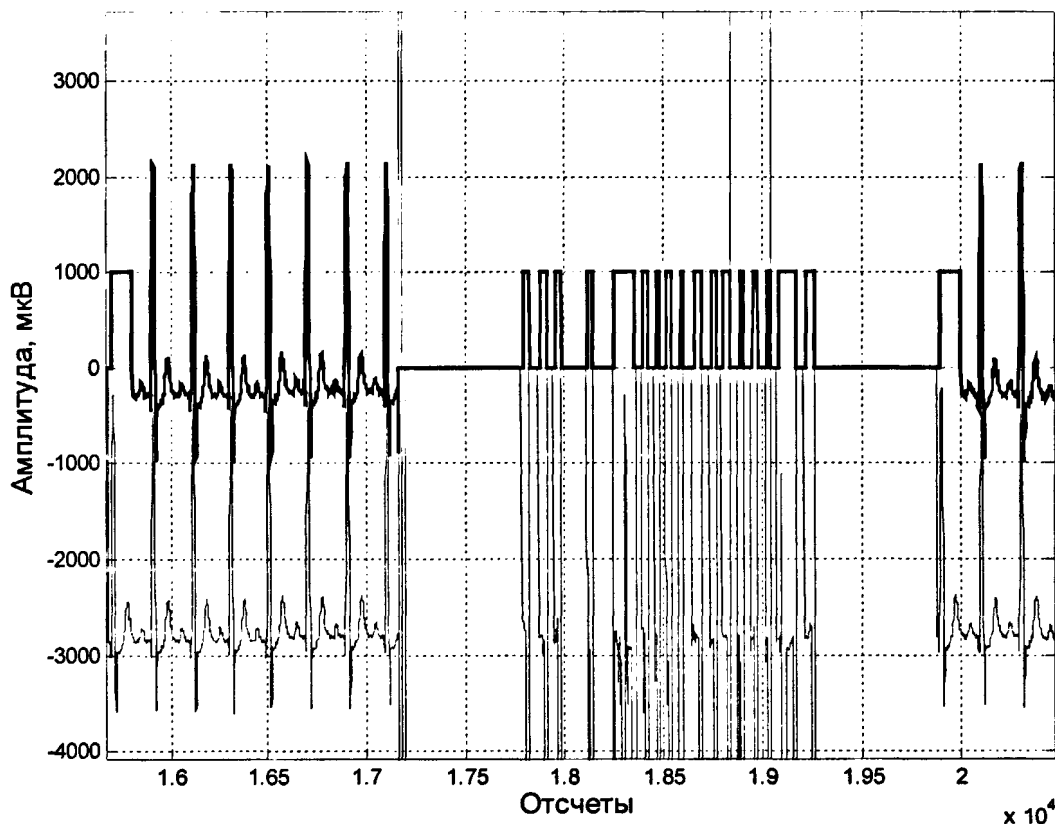


Рис. 4.4. Подавление постоянного смещения и эффектов обрыва электродов.

4.1.3 Реализация подсистемы подавления шумов ЭКГ методами цифровой фильтрации

Линейные цифровые фильтры, применяемые в системах ЦОС компьютерной электрокардиографии, делятся на 3 основные группы – фильтры с конечной импульсной характеристикой, или КИХ-фильтры, фильтры с бесконечной импульсной характеристикой (БИХ), или БИХ-фильтры, а также адаптивные фильтры. Применение того или иного фильтра имеет ряд особенностей и определяется задачами, которые необходимо решать на этапе предварительной фильтрации ЭКГ. Реализация фильтра должна удовлетворять заранее заданным условиям: устойчивости, эффективности подавления шумов, минимальности искажений ЭКГ. Каждый класс из вышеуказанных фильтров наиболее эффективен при подавлении помех определенного рода, поэтому возникает необходимость

детального анализа характеристик каждого из фильтров. Важнейшими параметрами линейных фильтров, на основе которых необходимо проведение анализа, являются: АЧХ, ФЧХ, групповая задержка, устойчивость[97].

Характеристики присутствующих в ЭКГ помех (см. 4.1.1), ясно демонстрируют потребность в избирательном подавлении шумовой составляющей в ряде частотных диапазонов. Линейные цифровые фильтры способны эффективно подавлять сигнал в заданных диапазонах частот, искажая тем самым полезный сигнал ЭКГ. Более того, в зависимости от вида ФЧХ и групповой задержки эти искажения могут затрагивать весь спектр сигнала. Все эти особенности необходимо учитывать при синтезе подсистемы цифровой обработки сигналов.

В основу структуры подсистема цифровой обработки «IntelliCard» положен синтез рекурсивных, нерекурсивных и адаптивных фильтров, работающих в определенной полосе частот с возможностью их включения и отключения в произвольный момент времени.

Фильтры с конечным импульсным откликом описываются следующим уравнением:

$$y(n) = \sum_{m=0, M-1} b_m x(n-m) \quad (4.1)$$

Соответственно, передаточная функция, которая определяется как отношение Фурье-образа результирующего сигнала $y[n]$ к входному сигналу $x[n]$, выглядит следующим образом:

$$H(z) = \sum_{m=0, M-1} b_m z^{-m} \quad (4.2)$$

что следует из z -преобразования правой и левой частей выражения 4.1.

Групповая задержка фильтра, функция, характеризующая запаздывание каждой частотной составляющей, определяется выражением:

$$\tau_H(f) = -\frac{1}{2\pi} \frac{d}{df} \arg\{H(f)\} \quad (4.3)$$

Следовательно, фильтры, имеющие нулевую или постоянную задержку, или, что является эквивалентным требованием - линейную ФЧХ, не будут вносить в сигнал дополнительных искажений, кроме линейной задержки отклика.

Фильтры с бесконечным импульсным откликом описываются следующим выражением в стандартной форме:

$$y(n) = - \sum_{k=1, N-1} a_k y(n-k) + \sum_{m=0, M-1} b_m x(n-m) \quad (4.4)$$

Передаточная функция имеет вид:

$$H(z) = \frac{\sum_{m=0, M-1} b_m z^{-m}}{1 + \sum_{k=1, N-1} a_k z^{-k}} \quad (4.5)$$

Таким образом, групповая задержка τ_{11} для БИХ-фильтров определяется также и коэффициентами обратной связи a_k . При соответствующем проектировании групповая задержка фильтра на широком диапазоне частот

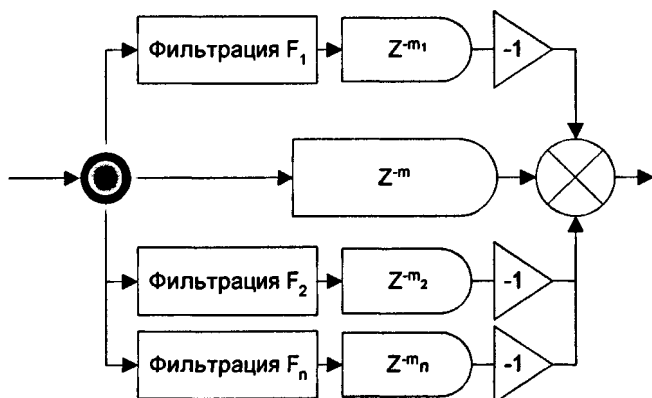


Рис. 4.5. Система потоковых фильтров «Intellicard».

пропускания определяется только количеством коэффициентов обратной связи a_k . Обычно в полосе подавления τ_H имеет сложный вид и может достигать значительных величин и, следовательно, вносить значительные искажения в обрабатываемый сигнал в случае плавности АЧХ.

Задача линейной многополосной фильтрации ЭКГ, состоящая в подавлении нежелательных

помех до последующего анализа сигнала, решается в «Intellicard» путем синтеза КИХ- и БИХ- фильтров, работающих в некотором диапазоне частот [98].

При этом согласование осуществляется на основе учета групповой задержки τ_H каждого из фильтров в системе. Максимальная групповая задержка, определяется как

$$\tau_{\max} = \max_{\text{по всем } n} \theta(n) \tau_n, \text{ где } \theta(n) = \begin{cases} 1, \text{ включен} \\ 0, \text{ выключен} \end{cases} \quad (4.6)$$

Таким образом, τ_{\max} представляет собой задержку подсистемы обработки сигналов. Интеграция адаптивных фильтров, имеющих структуру, представленную на Рис 4.6 имеет некоторые особенности и может быть произведена следующим образом. Пусть исходный сигнал $d(t)$ представляет собой сумму «чистого» не зашумленного сигнала $s(t)$ и шума $n(t)$, $u(t)$ – сигнал, так или иначе связанный с входным шумом. Результирующий сигнал, вырабатываемый адаптивным фильтром, вычитается из исходного сигнала.

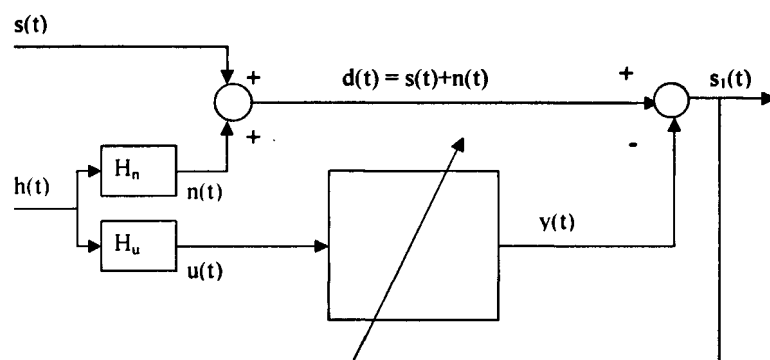



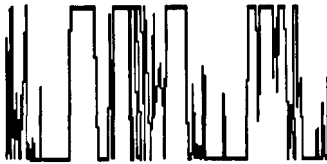
Рис. 4.6. Схема адаптивного фильтра.

Следовательно, адаптивный фильтр может быть интегрирован в систему фильтров с задержкой $\tau_H=0$ и выходом $u(t)$, непосредственно подаваемым на сумматор. Большим преимуществом адаптивных фильтров является большая гибкость по сравнению с КИХ- и БИХ-фильтрами, а также большая эффективность.

Мощность шумов, сосредоточенная в различных частотных диапазонах ЭКГ, зависит от множества факторов (оборудования, условий измерения, особенностей физиологии пациента). На Табл. 4.1 представлены причины повышенного влияния внешних шумов и методы борьбы с ними.

Табл. 4.1. Причины повышения влияния шумов и методы борьбы с ними.

Вид шума	Причины возникновения	Методы устранения
Шум силовых линий 50/60 Гц. 	1) Плохой контакт электродов; 2) Незаземленное оборудование регистрации ЭКГ или в непосредственной близости у пациента.	Заново установить электроды, проверить цепи заземления оборудования.
Шумы электрической активности мышц 	1) пациент находится в состоянии возбуждения; 2) плохой контакт электродов; 3) тремор, обильное потоотделение.	Пациент должен быть приведен в спокойное состояние. Необходимо проверить, как установлены грудные электроды.
Нерегулярности изолинии	1) плохой электрический контакт электродов;	Необходимо изменить положение электродов, а в случае

	2) сильное влияние респираторной активности; 3) Плохие электроды.	необходимости заменить их. Можно активизировать ВЧ-фильтрацию ЭКГ до 1 Гц.
Дрейф изолинии 	1) движение пациента; 2) сильное влияние респираторной активности; 3) некорректная установка электродов	Пациент должен находиться в спокойном состоянии. Электроды должны быть установлены в местах наименьшего влияния дыхательной активности.
Появление калибровочных прямоугольников (на «IntelliCard»), скачки амплитуды. 	1) потеря контакта электродов с кожным покровом; 2) дефект кабеля (обрыв, нарушения экранирования).	Заново установить электроды, проверить качество кабеля электродов.

Влияние представленных в таблице шумов не однозначно: при различных условиях воздействие, оказываемое на ЭКГ-сигнал, может оказаться различным. Поэтому необходима гибкая процедура конфигурирования, позволяющая управлять работой фильтров из каждой группы. Целесообразно, разделить цифровые фильтры на следующие основные категории согласно их функциональному назначению:

- 1) коррекции изолинии;
- 2) фильтрации помех электрических линий;
- 3) подавления влияния мышечных электрических процессов;
- 4) вспомогательные, включая фильтрацию эффектов квантования, шумов периферийного оборудования и т.д.

Как уже отмечалось выше, преимуществом линейной фильтрации перед другими методами борьбы с помехами, является не только простота реализации и удобство расчета, но также и возможность синтеза многочисленных, параллельно работающих цифровых фильтров.

Исходный поток отсчетов разбивается на несколько параллельных потоков, соответствующих каждому типу фильтров $F_1 \dots F_n$. Каждый поток F_i обрабатывается одним режекторным, адаптивным или парой ФНЧ- и ФВЧ-фильтров, формируя поток шума E_i , который, проходя линию

задержки длительности $m_i = \tau_{\max} - \tau_n^i$ (см. Рис. 4.5), вычитается из исходного потока. Важно определить максимальную задержку, которая вносится системой параллельных фильтров относительно исходного сигнала.

Подавление основной мощности мышечного шума может быть осуществлено парой ФНЧ и ФВЧ, с линейной ФЧХ. В «IntelliCard» для подобных целей использовался фильтр Чебышева 170-го порядка, обеспечивающий подавление до 50 дБ в области закрытия и порядка 30 дБ в интересующем нас диапазоне 35-40 Гц, вносит задержку в 87 отсчетов (Рис. 4.7а). Для подавления шумов высоких частот, можно использовать аналогичный ФНЧ Чебышева такого же порядка с частотами среза 100-130 Гц (Рис. 4.7б).

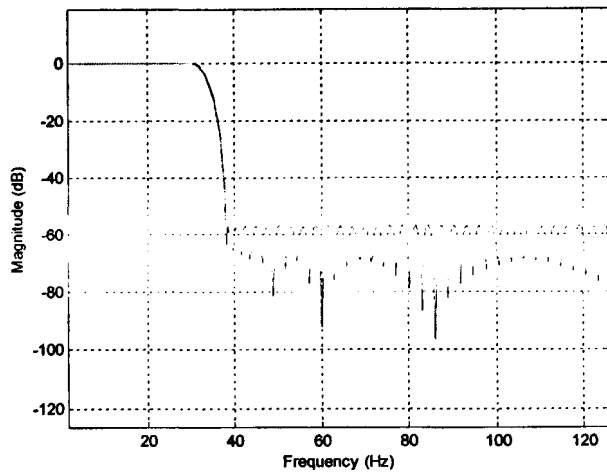
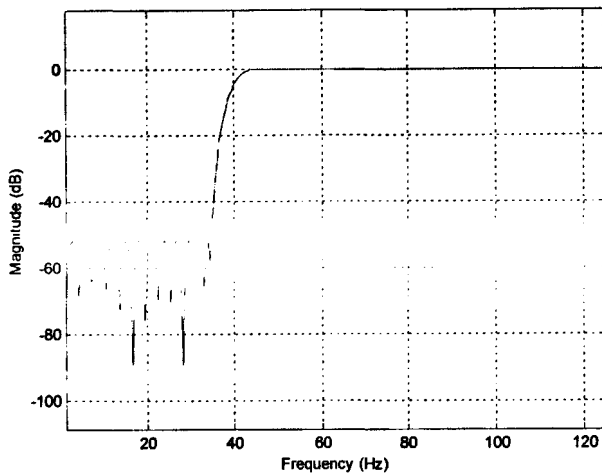
Подавление шума электрических силовых линий требует гораздо большей эффективности подавления из-за обычно значительной амплитуды данной помехи. Возможно применение пар режекторных БИХ-фильтров Батерворда для основной и старших гармоник (Рис. 4.7в). Для сравнимой эффективности подавления соответствующий КИХ-фильтр потребует порядка 500 отводов. Недостатком такого метода является то, что нелинейные искажения, вносимые режекторными фильтрами в полосе пропускания, приведут к тому, что сигнал ошибки E_i будет искажен, что в результате приведет к неполному подавлению помех и появления шумов более сложной природы. Логичной заменой эффективных, с точки зрения подавления, но шумящих БИХ-фильтров, могут служить адаптивные фильтры. В IntelliCard применялся адаптивный КИХ-фильтр с алгоритмом адаптации, основанном на методе наименьших квадратов (МНК) (Рис. 4.8). МНК-адаптивный фильтр производит следующую последовательность операций для получения потока шума $E_i(n)$ (99):

$$\begin{aligned} E_i(n) &= \hat{w}^H(n-1)u(n) \\ e(n) &= d(n) - E_i(n) \\ \hat{w}^H(n) &= \hat{w}^H(n-1) + \frac{u(n)}{\alpha + u^H(n)u(n)} \mu e^*(n) \end{aligned} \quad (4.7)$$

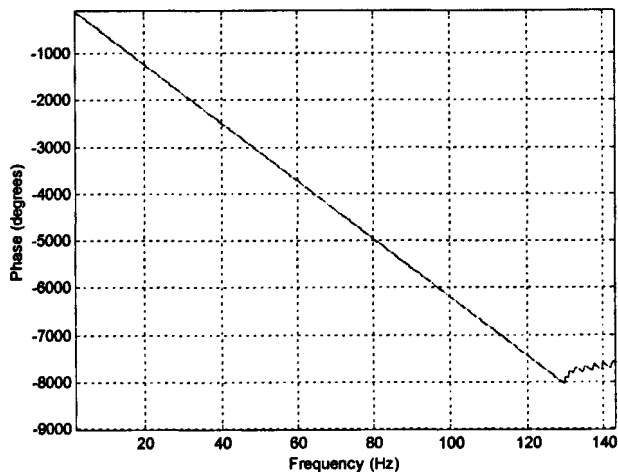
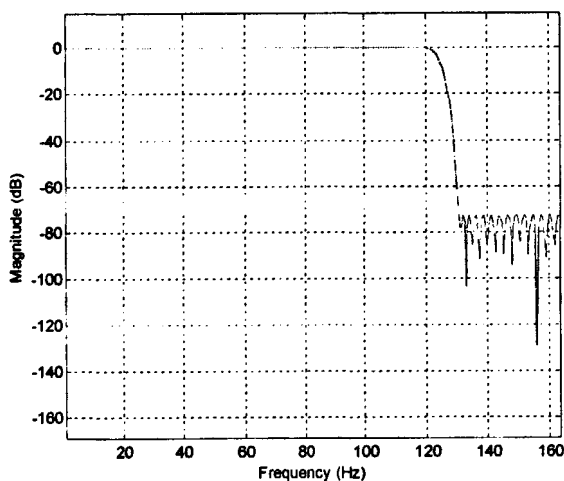
где n -текущая итерация, $u(n)$ – сигнал эталона шума, $w(n)$ -вектор текущей оценки коэффициентов КИХ-фильтра, $e(n)$ -оценка ошибки на шаге n , $d(n)$ -ожидаемый ответ, μ -шаг адаптации.

Например, в случае шума силовых линий (50 Гц), в качестве $u(n)$ выступает дискретизованная синусоида (см. Рис. 4.8).

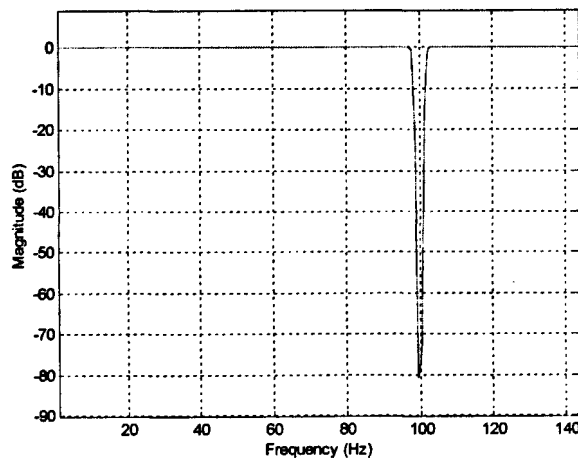
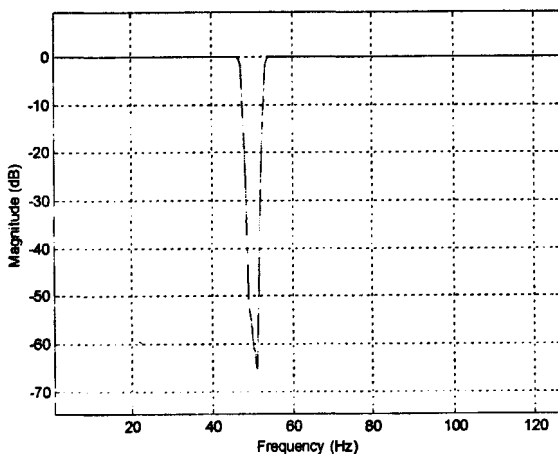
Коррекция дрейфа изолинии имеет большое значение для точных измерений амплитуд и длительностей элементов ЭКГ. Среди множества возможных подходов к данной проблеме наиболее разумными методами является цифровая ВЧ-фильтрация, а также полиномиальная аппроксимация, рассмотренная ниже. ВЧ-фильтрация в системе «IntelliCard» реализована тремя переключаемыми фильтрами с $F_s = 0.05, 0.1,$



а) АЧХ КИХ ВЧ и НЧ-фильтров Чебышева 170 порядка для подавления 35-40Гц помехи. Групповая задержка составляет 87 отсчетов.



б) АЧХ и ФЧХ ФНЧ КИХ-фильтра Чебышева 170 порядка для подавления высокочастотных шумов.



в) Режекторные БИХ-фильтры 8-порядка подавления помех электрических силовых линий. Групповая задержка равна порядку фильтра.

Рис. 4.7. Характеристики цифровых фильтров подсистемы ЦОС «IntelliCard».

1 Гц соответственно. Стоит отметить, что последний из упомянутых фильтров при определенных условиях может вызывать помехи в частотной области QRS-комплекса, искажая тем самым амплитуду R-зубца. Фильтры с частотами среза 0.05 и 0.1 Гц основаны на вычислении адаптивного среднего:

$$E(i) = \frac{1}{i} \sum_{k=1}^i x(k), i < 11 \quad (4.8)$$

$$E(i) = E(i-1) - c(E(i-1) - x(i)), i \geq 11$$

Для $F_s=0.05$ Гц константа адаптации c принята равной 0.0006, что обеспечивает подавление шума в -52 дБ на частоте 0 Гц. Для $F_s=0.1$ Гц $c = 0,00125$, что соответствует подавлению постоянного смещения шума в -45 дБ.

КИХ-фильтр $F_s=1$ Гц с 500 отводами построен с использованием оконной функции Кайзера и обеспечивает подавление постоянной составляющей на -22 дБ (Рис. 4.9). К сожалению, при таком соотношении F_s и частоты дискретизации 500Гц и более чрезвычайно тяжело получить КИХ-фильтр, который бы обеспечивал хорошее подавление постоянной составляющей с одной стороны и имел бы гладкую АЧХ без осцилляций. Приемлемую АЧХ удастся получить только при очень высоких порядках, в частности в нашем примере, это потребует порядка 500 отводов, что соответствует задержке $\tau_H=250$ отсчетов.

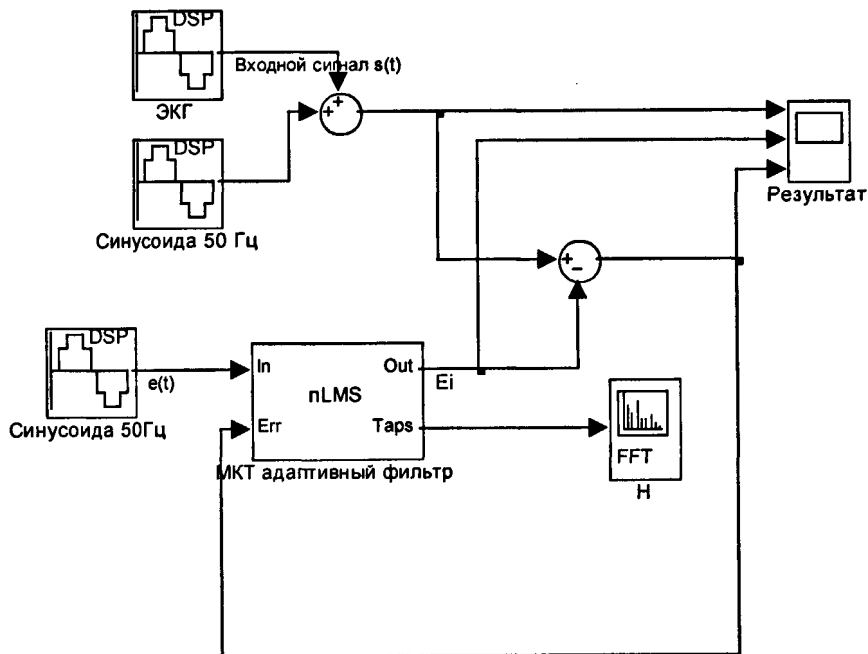


Рис. 4.8. Модель подавления шумов электрических линий МНК-адаптивным фильтром.

Таким образом, процедура предварительной обработки является постоянным компромиссом между сложностью реализации, производительностью и необходимой эффективностью подавления помех.

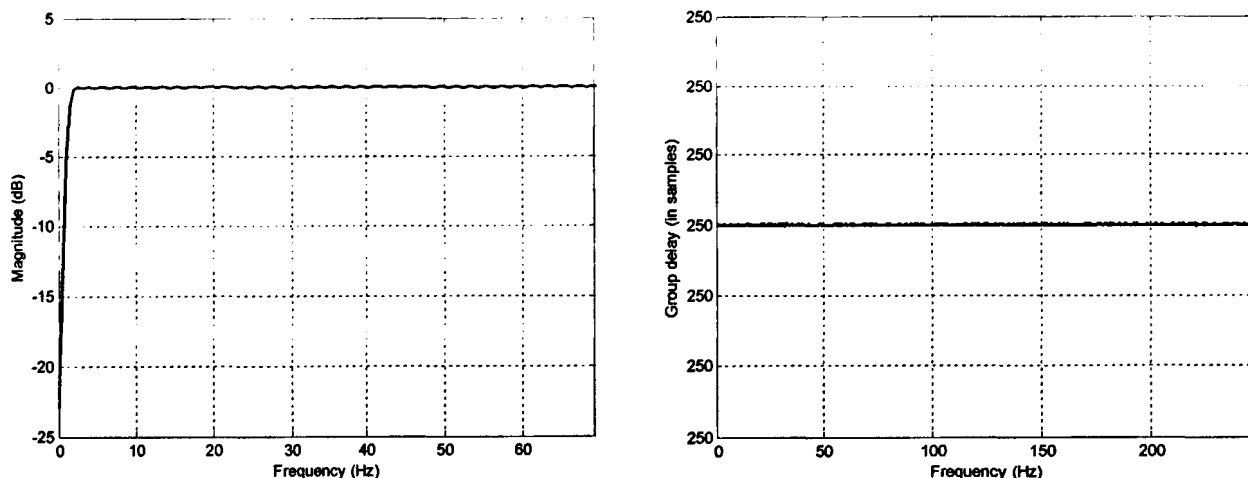


Рис. 4.9. АЧХ и групповая задержка фильтра коррекции изолинии.

4.2. Выделение и анализ QRS-комплексов в ЭКГ-системах реального времени

Отправной точкой ряда современных методик компьютерной электрокардиографии является локализация QRS-комплексов, положение которых определяется позицией максимума – R-зубца, а временные границы – параметрами Q- и S- волн. Полученная последовательность R-зубцов используется для сегментации кардиоцикла, измерения амплитуд и длительностей отдельных его элементов, определения изолинии ЭКГ методом полиномиальной аппроксимации, а также в методиках анализа variability сердечного ритма (BCP). Задача выбора оптимального метода выделения, анализа и сегментации QRS-комплексов на отдельные элементы в потоковом режиме для применений в ЭКГ-системах реального времени стоит в настоящее время достаточно актуально.

Предлагаемая методика, ориентированная на применение в реальном времени и реализованная в системе «IntelliCard», состоит из нескольких шагов[100]:

- 1) выделение R-зубца QRS-детектором;
- 2) анализ Q-волны, определение амплитуды и первого отсчета;
- 3) полиномиальная коррекция изолинии;
- 4) анализ S-волны, определение последнего отсчета QRS-комплекса;
- 5) коррекция определенного последнего отсчета QRS-комплекса;
- 6) верификация Q- и S-зубцов.

4.2.1 Классификация методик выделения QRS-комплексов

Электрокардиограмма представляет собой запись электрических потенциалов активности сердца, снятую с одного или нескольких отведений, и состоит из периодической последовательности сердечных циклов. На ЭКГ сердечный цикл обычно отображается в виде трех легко различимых элементов – комплексов. P-комплекс соответствует деполяризации предсердий, QRS-деполяризации желудочков, T – их реполяризации, реполяризация предсердий на ЭКГ не проявляется. Каждый комплекс состоит из нескольких разнонаправленных пиков, или зубцов (Рис. 4.10). Число зубцов в каждом комплексе различно в разных отведениях и у разных пациентов. Так P- и T-комплексы обычно содержат один или два зубца, а QRS-комплекс – от одного до семи.

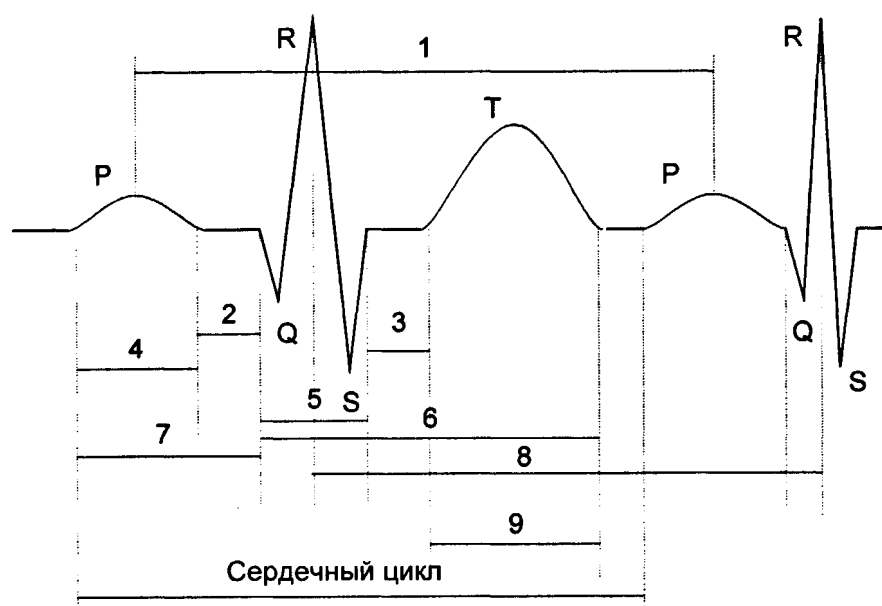


Рис. 4.10. Структура сердечного цикла.

1 - PP-интервал; 2 - PR-сегмент; 3 - ST-сегмент;
4 - P-комплекс; 5 - QRS-комплекс; 6 - QT-интервал;
7 - PR-интервал; 8 - RR-интервал; 9 - T-комплекс.

Продолжительность QRS-комплекса выражает время, необходимое для деполяризации мускулатуры желудочков и измеряется от начала Q-волны до конца S-волны. За начало Q-волны принимается точка, за которой уровень ЭКГ-сигнала становится меньше уровня изолинии, а конец S-волны – точка выхода сигнала на уровень изолинии, что является во многом условным. Нормальная продолжительность QRS-комплекса составляет 0.07-0.10 с. Если продолжительность превышает 0.11 с, то это может свидетельствовать о наличии нарушений проводящих путей. Максимальная амплитуда QRS-комплекса определяется амплитудой R-волны и составляет

обычно порядка 0.5-1.0 мВ. Амплитуда QRS-комплекса менее 0.4 мВ может свидетельствовать об определенных отклонениях.

В настоящее время существует множество методик выделения QRS-комплексов, которые по используемому подходу можно разделить на 5 основных групп:

- а) алгоритмы анализа ЭКГ во временной области;
- б) алгоритмы, основанные на частотно-временных, в том числе и нелинейных преобразованиях сигнала ЭКГ;
- в) алгоритмы на основе применения нейросетевых моделей;
- г) синтаксические методы;
- д) комбинированные алгоритмы.

Количественными характеристиками эффективности QRS-алгоритмов принято считать несколько величин: вероятность распознавания произвольного QRS-комплекса, выраженную в процентах (чувствительность), вероятность того, что произвольный выделенный QRS-комплекс является истинным (предсказуемость, или специфичность) и вероятность выдачи детектором правильного результата (эффективность). Чувствительность определяется количеством ложно отсеянных QRS-отсчетов, предсказуемость зависит от количества ложно определенных комплексов, а эффективность является интегральным показателем качества метода. Таким образом, указанные величины соответственно равны $S = \frac{N_{ip}}{N_{ip} + N_{fn}}$, $P_+ = \frac{N_{ip}}{N_{ip} + N_{fp}}$, $E = \frac{N_{ip}}{N_{ip} + N_{fn} + N_{fp}}$, где N_{ip} – количество правильно определенных комплексов, N_{fn} – количество ложно отвергнутых, а N_{fp} – ложно определенных.

Алгоритмы группы а), многие из которых используют принципы, заложенные в работе [101], основаны на применении к входному сигналу, кроме процедур линейной фильтрации (обычно ФВЧ и ФНЧ последовательно), некоторого нелинейного преобразования, включающего в себя обычно процедуру интегрирования сигнала в скользящем окне. Положение R-зубца определяются при помощи порогового детектора уровня сигнала, значения которого могут быть как фиксированными, так и вычисляться адаптивно на каждом шаге работы алгоритма. Дополнительно, чтобы повысить чувствительность, могут использоваться другие процедуры повышения точности: дополнительная предварительная обработка ЭКГ, введение вспомогательных оценочных процедур, позволяющих выявлять ложно определенные QRS-комплексы. Альтернативным подходом является применение согласованной фильтрации [102], представляющей собой вычисление тех или иных корреляционных соотношений между отрезком входного сигнала соответствующей длительности и базовыми шаблонами QRS-комплекса. Данный подход предусматривает различные

вспомогательные процедуры, позволяющие производить коррекцию морфологии исходных шаблонов. Эффективность методов этой группы составляет 96-98% корректно определяемых QRS-комплексов из общего числа присутствующих в ЭКГ.

Алгоритмы группы б) основаны на применении после ряда процедур предварительной обработки ЭКГ различных частотно-временных преобразований, таких как локальное преобразование Фурье, преобразование Карунена-Лоева, дискретное вейвлет-преобразование [103]. При использовании вейвлет-преобразования применяется несколько основных базовых вейвлет-функций, причем ряд методов предусматривает обратную связь для коррекции их параметров. При этом локализация положения QRS-комплекса осуществляется в области вейвлет-спектра, в простейшем случае, путем простого детектирования уровня. Методики этой группы характеризуются относительно невысокой производительностью, достаточно низкой чувствительностью к шумам и эффективностью, превышающей 99%.

Алгоритмы группы в) используют нейросетевые методы обработки данных и обычно применяются для анализа морфологии и классификации элементов ЭКГ. Нейросетевые модели позволяют значительно более эффективно адаптироваться к нестационарному характеру ЭКГ, поэтому в задачах выделения QRS-комплексов используются при адаптивной согласованной фильтрации [104,105]. Чувствительность методик варьируется в широком диапазоне, но в целом достигает 99%.

Группу г) составляют синтаксические алгоритмы, также известные как лингвистические или грамматические. Исходный анализируемый сигнал представляется в виде определенной последовательности примитивов, определяются специальные правила (грамматики), порождающие тот или иной элемент ЭКГ из множества примитивов. Выделение элементов ЭКГ сводится к определению порождающей грамматики [106].

Группу д) составляют различные комбинации методов, обычно являющиеся синтезом алгоритмов группы в) и б) или в) и а). Особенно выигрышным оказалось первое сочетание, поскольку подобный подход позволяет достичь максимальной на сегодняшний день чувствительности – 99,9% на тестовых подборках ЭКГ. К недостаткам методов этой группы стоит отнести требовательность к вычислительным ресурсам.

Проведенный анализ показал, что методики групп б), в), г) и д) требуют в единицу времени на порядок больше элементарных математических операций, чем методы группы а), при этом чувствительность и эффективность QRS-детектора обычно повышается на 1-3%. Более того, необходим достаточно большой интервал времени для адаптации параметров детектора, что делает подобные алгоритмы трудно применимыми для систем реального времени, когда необходимо получение

ЧСС с первых секунд после начала съема ЭКГ. Вейвлет-анализаторы обычно применяются при автоматизированном анализе уже записанной длительной ЭКГ и отличаются повышенной устойчивостью к шумам, нейросетевые алгоритмы активно применяются для классификации элементов ЭКГ.

4.2.2 Алгоритм детектирования QRS-комплексов

Предлагаемый алгоритм детектирования QRS-комплексов для программных ЭКГ-систем реального времени основывается на общих принципах построения подобных алгоритмов [107] и ряде хорошо себя зарекомендовавших методов [87,101], относящихся к группе а). QRS-детектор ориентирован на автономную работу и не связан с процедурами предварительной обработки ЭКГ. Методика разработана с учетом требований систем ЭКГ реального времени, специально адаптирована для реализации с использованием высоко оптимизированной библиотеки цифровой обработки сигналов NSP компании «Интел» [108] и позволяет надежно в режиме реального времени определять положение QRS-комплексов в каждом из 12 стандартных отведений.

Рассматриваемый алгоритм [109,110] делится на несколько шагов:

- 1) Предварительная обработка ЭКГ, которая заключается в ВЧ- и НЧ-фильтрации входящего потока ЭКГ;
- 2) Вычисление 2-х входных потоков данных – сигнала, прошедшего предварительную обработку и сигнала, дополнительно подвергнутого ряду нелинейных преобразований [111];
- 3) Вычисление и сравнение адаптивных пороговых значений соответственно с двумя входными потоками;
- 4) Определение интервала, в котором предположительно содержится R-зубец;
- 5) Фильтрация некорректно определенных R-зубцов;
- 6) Определение положения R-зубцов в каждом из 12 стандартных отведений.

Предварительная обработка ЭКГ заключается в фильтрации помех, в основном, являющихся сетевым шумом с частотой 50/60 Гц и шумом электрической активности мышц, что можно сделать достаточно эффективно одним ФНЧ. Был применен КИХ-фильтр 48-го порядка, рассчитанный с окном Кайзера, имеющий линейную ФЧХ в полосе пропускания. Данный фильтр обеспечивает подавление 50 Гц на 65 дБ, 60 Гц – на 60 дБ, мышечного тремора – 10-20 дБ (Рис. 4.11).

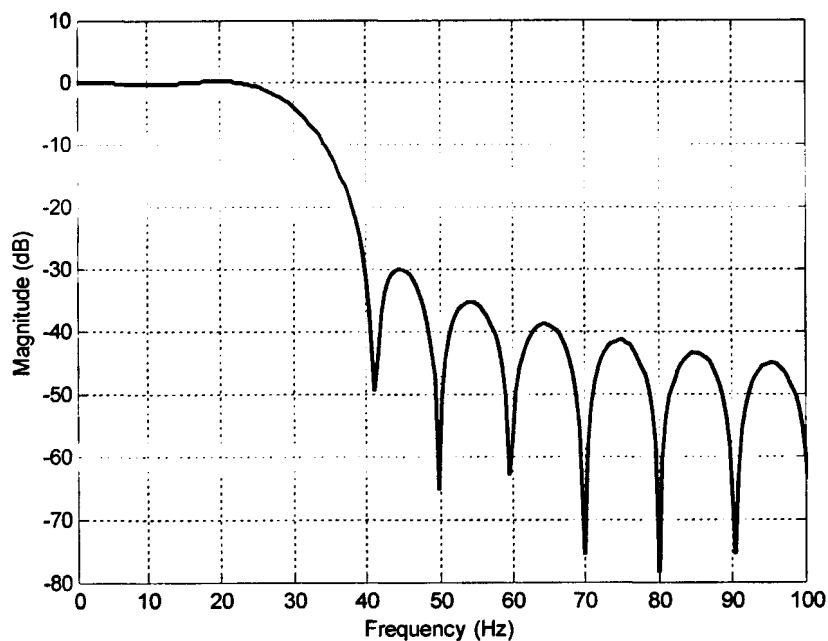


Рис. 4.11. АЧХ и ФЧХ ФНЧ-фильтра Кайзера 48-го порядка.

ФЧХ данного фильтра имеет линейный вид в полосе пропускания, что минимизирует искажения ЭКГ в интересующей нас области. Возможно применение и других КИХ-фильтров, имеющих линейную ФЧХ, например, фильтров, синтезированных на основе метода Паркса-Маклеллана [97], который представляет собой оптимальный алгоритм расчета КИХ-фильтров с требуемой АЧХ на основе алгоритма полиномиальной аппроксимации Ремеза. Выбор того или иного фильтра диктуется, прежде всего, вычислительной нагрузкой и необходимой эффективностью подавления шума. Современные вычислительные возможности позволяют производить обработку КИХ-фильтрами достаточно высокого порядка, что дает возможность отказаться от использования заведомо более эффективных, но приводящих к значительным искажениям, БИХ-фильтров. В качестве альтернативного решения может быть предложена адаптивная фильтрация [99].

Поскольку исходная ЭКГ содержит некоторую шумовую составляющую, имеющую низкую частоту (менее 1 Гц), – так называемый дрейф изолинии – необходима коррекция ЭКГ. Для этого возможно применение различных ФВЧ, например, БИХ-фильтра Баттерворда низкого порядка. Большой порядок фильтра в данном случае неважен, поскольку даже фильтры 2-3 порядка обеспечивают достаточное для нормальной работы алгоритма подавление постоянной составляющей. При необходимости, подобный БИХ-фильтр может быть эффективно заменен адаптивным КИХ-фильтром. Интересным представляется также

использование морфологических фильтров[112], что, однако, приводит к увеличению вычислительной нагрузки.

Следующим этапом работы алгоритма является вычисление адаптивных пороговых значений. Для повышения эффективности метода вычисляются два адаптивных пороговых значения, которые применяются соответственно к двум потокам данных: первый поток данных содержит сигнал, прошедший предварительную обработку - фильтрацию, второй поток – сигнал, подвергнутый некоторым нелинейным преобразованиям. Первое пороговое значение T_f вычисляется следующим образом:

$$\begin{aligned}
 T_f[n] &= N_f[n] + q(S_f[n] - N_f[n]) \\
 N_f[n+1] &= \begin{cases} pX_f[n] + (1-p)N_f[n], & \text{если } X_f[n] < T_f[n] \\ N_f[n], & \text{если } X_f[n] \geq T_f[n] \end{cases} \\
 S_f[n+1] &= \begin{cases} S_f[n], & \text{если } X_f[n] < T_f[n] \\ pX_f[n] + (1-p)S_f[n], & \text{если } X_f[n] \geq T_f[n] \end{cases} \\
 S_f[0] &= 0, N_f[0] = 0,
 \end{aligned} \tag{4.8}$$

где X_f - предварительно обработанная ЭКГ, т.е. прошедшая НЧ- и ВЧ-фильтрацию, $p = 0.125$, $q=0.25$ -весовые коэффициенты, S_f -скользящая оценка уровня сигнала, N_f -скользящая оценка уровня шума, T_f - пороговое значение, сравниваемое с X_f на каждом шаге.

Для вычисления второго потока за основу берется исходный сигнал X_f , который дифференцируется и каждый отсчет последовательно возводится в квадрат:

$$y[n] = (\text{diff}(X_f[n]))^2 \tag{4.9}$$

Аппроксимацию дифференцирующего оператора достаточно осуществлять с точностью 2-го порядка:

$$\text{diff}(X_f[i]) = \frac{x[i+1] - x[i-1]}{2\Delta}, \Delta = \frac{1}{f_s} \tag{4.10}$$

где f_s - частота дискретизации.

В данном выражении множитель Δ приводит к дополнительному усилению результирующего сигнала, поэтому при реализации его следует опустить. При наличии достаточной производительности в качестве дифференцирующего звена возможно применение дифференцирующего фильтра, особенность которого состоит в том, что он, в идеальном случае, обладает линейной АЧХ в определенной полосе частот, сохраняя линейность ФЧХ. Это позволяет эффективнее производить выделение определенной полосы частот, которая соответствует QRS-комплексу (Рис. 4.12).

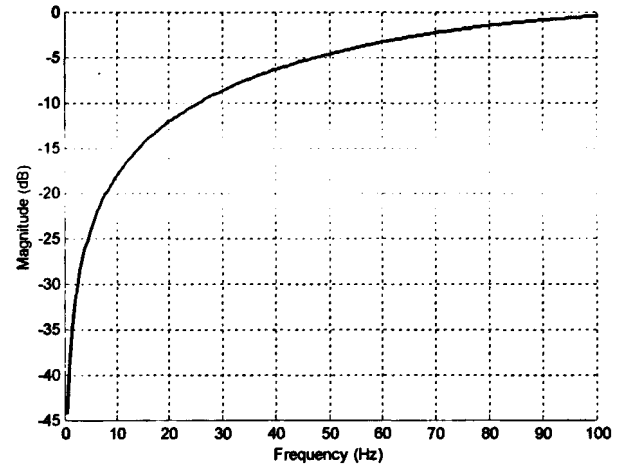
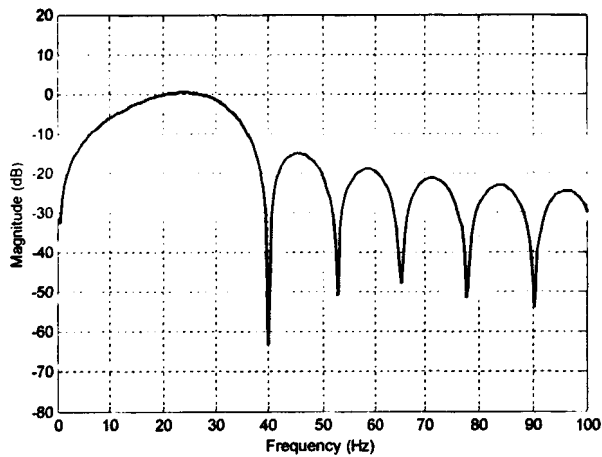


Рис. 4.12. АЧХ дифференцирующего фильтра и аппроксимации производной 2-го порядка.

После процедуры дифференцирования осуществляется интегрирование этого сигнала скользящим окном, которое выбирается равным 140-160 мс, полностью содержащим QRS-комплекс:

$$Y_i[k] = \frac{1}{W_{qrs}} \sum_{n=k-n_d+1}^{k+n_d} y[n] \quad (4.11)$$

при $k \geq n_d = \left\lceil \frac{w_{qrs}}{2} \right\rceil$, где w_{qrs} - величина окна интегрирования.

Адаптивное пороговое значение T_i вычисляется следующим образом:

$$\begin{aligned} T_i[n] &= N_i[n] + q(S_i[n] - N_i[n]) \\ N_i[n+1] &= \begin{cases} pY_i[n] + (1-p)N_i[n], & \text{если } Y_i[n] < T_i[n] \\ N_i[n], & \text{если } Y_i[n] \geq T_i[n] \end{cases} \\ S_i[n+1] &= \begin{cases} S_i[n], & \text{если } Y_i[n] < T_i[n] \\ pY_i[n] + (1-p)S_i[n], & \text{если } Y_i[n] \geq T_i[n] \end{cases} \\ S_i[0] &= 0, N_i[0] = 0, \end{aligned} \quad (4.12)$$

где S_i - скользящая оценка уровня сигнала Y_i , $p=0.125$, $q=0.25$ - весовые коэффициенты, N_i - скользящая оценка уровня шума, T_i - пороговое значение, сравниваемое с Y_i на каждом шаге.

Адаптивные пороговые значения T_f и T_i вычисляются на каждом шаге заново и сравниваются соответственно с отсчетами потоков 1 и 2. Интервалы, в которых происходит превышение обоих порогов одновременно, помечаются как содержащие R-зубец (Рис. 4.13). Функция, описывающая возможное положение R-зубца, имеет следующий вид:

$$\Theta(n) = \begin{cases} 1, & \text{при } T_i[n] < Y_i[n], T_f[n] < X_f[n] \\ 0, & \text{в остальных случаях} \end{cases} \quad (4.13)$$

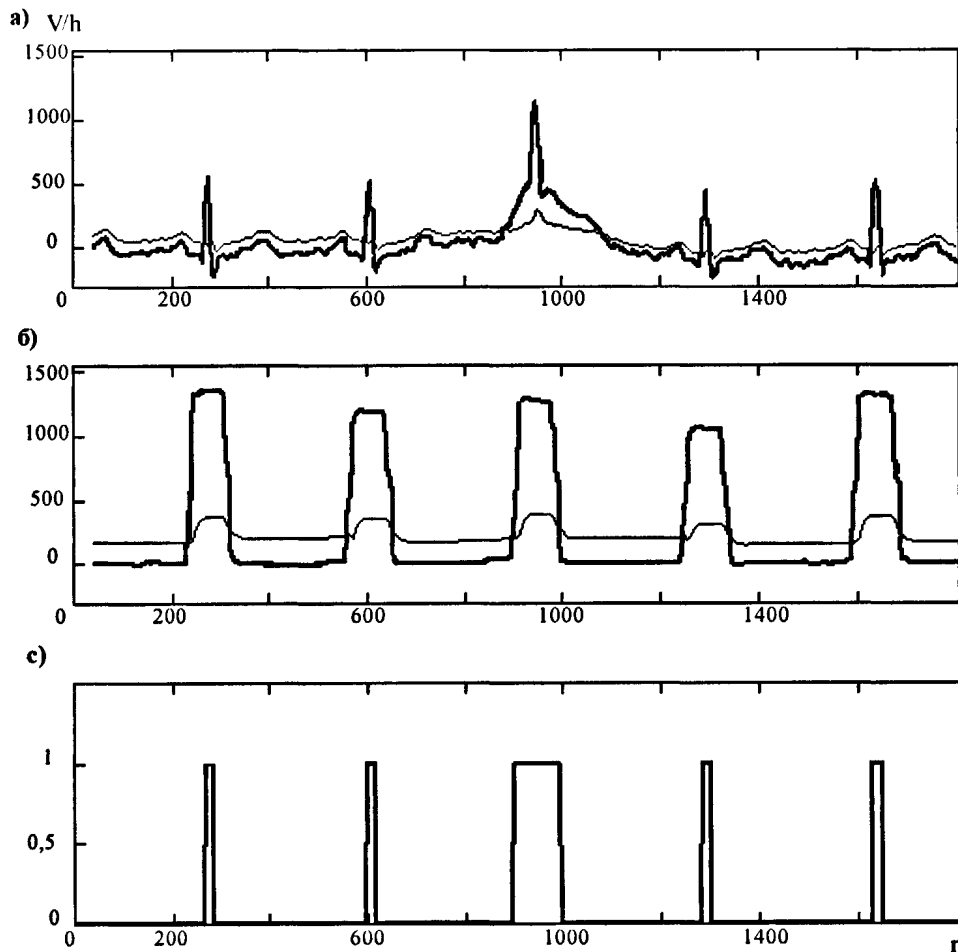


Рис. 4.13. Работа адаптивного порогового детектора.

а) Сигнал X_f и соответствующие пороговые значения T_f ;

б) Сигнал Y_i и соответствующие пороговые значения T_i ;

в) Функция превышения порогов $\Theta(n)$;

В качестве безразмерной величины по оси ординат отложено отношение амплитуды сигнала в мкВ к величине единицы младшего разряда АЦП.

Вероятность того, что R-зубец в действительности содержится на интервалах, где $\theta(n)=1$, достаточно высока, но для повышения точности алгоритма необходима процедура фильтрации некорректно определенных позиций R-зубцов. Это можно сделать за счет применения эвристических правил анализа характеристик исследуемого R-зубца. Для этого, после определения предполагаемого положения R-зубца, создается массив определенных свойств: длительности RR-интервала между исследуемым зубцом и предыдущим и модуля производной сигнала RA, вычисляемой в

предполагаемых точках нахождения R-зубца. Эти параметры сохраняются в циклическом буфере, способном сохранять не менее 3-х последних значений R-зубцов, дополнительно к этому необходимо вычисление среднего значения ЧСС за последние 10 циклов (см. Прил. 3).

Если RR-интервал между двумя последующими зубцами менее 300 мс (что соответствует пульсу 200 мин-1), то решение о том, какой из зубцов является истинным, принимается исходя из максимума RA, поскольку известно, что R-зубец обладает максимальным градиентом. Таким способом можно отсеять узкие и высокие P- и T-волны, часто определяемые пороговым детектором как R-зубцы. Существует возможность дополнительно повысить эффективность алгоритма, усложнив эвристические правила анализа, если вести статистику не только корректно определенных R-зубцов, но и отсеянных. Это позволяет, в случае необходимости, в течение последующих нескольких шагов скорректировать работу алгоритма, не потеряв ложно отсеянного QRS-комплекса.

В случае, когда доступно более одного отведения, можно несколько повысить точность алгоритма за счет использования большего количества отведений в качестве базовых. На практике оказывается полезным использовать отведения I и II стандартной системы 12 отведений. Применение суперпозиции 2-х отведений позволяет несколько увеличить отношение сигнал/шум, а так же подавить нежелательное воздействие на пороговый детектор артефактов, связанных с отрывом одного из электродов. Отведения I и II складываются с определенными весами, формируя исходный сигнал для детектора X[n]:

$$X[n] = a[n]X_1[n] + b[n]X_2[n],$$

где X_1, X_2 – сигналы I и II отведения, прошедшие предварительную обработку (НЧ- и ВЧ-фильтрацию), $a[n], b[n]$ – адаптивные весовые коэффициенты, которые на каждом шаге вычисляются следующим образом:

$$a[n] = \frac{1}{2}, 0 < n < 3f_s$$

$$a[n] = \frac{1}{2} + \frac{n - 3f_s}{2f_s} \left(\frac{\sum_{k=n-3f_s+n_d}^{n-n_d} \sqrt{T_i^I[k]}}{\sum_{k=n-3f_s+n_d}^{n-n_d} \sqrt{T_i^I[k]} + \sum_{k=n-3f_s+n_d}^{n-n_d} \sqrt{T_i^{II}[k]}} - \frac{1}{2} \right), 3f_s < n < 5f_s \quad (4.15)$$

$$a[n] = \frac{\sum_{k=n-3f_s+n_d}^{n-n_d} \sqrt{T_i^I[k]}}{\sum_{k=n-3f_s+n_d}^{n-n_d} \sqrt{T_i^I[k]} + \sum_{k=n-3f_s+n_d}^{n-n_d} \sqrt{T_i^{II}[k]}} , n > 5f_s$$

$$b[n] = 1 - a[n], \forall n$$

где T^I и T^{II} – интегральные пороги отведений I и II соответственно, n_d – задержка интегрированного сигнала.

Для одного и того же кардиоцикла позиции R-зубцов несколько сдвинуты относительно друг друга в различных отведениях. Определение позиции в каждом из оставшихся отведений осуществляется путем поиска локального экстремума в интервале $R_{pos} \pm 0,1$ сек., где R_{pos} – уже определенная позиция базового отведения (Рис. 4.14).

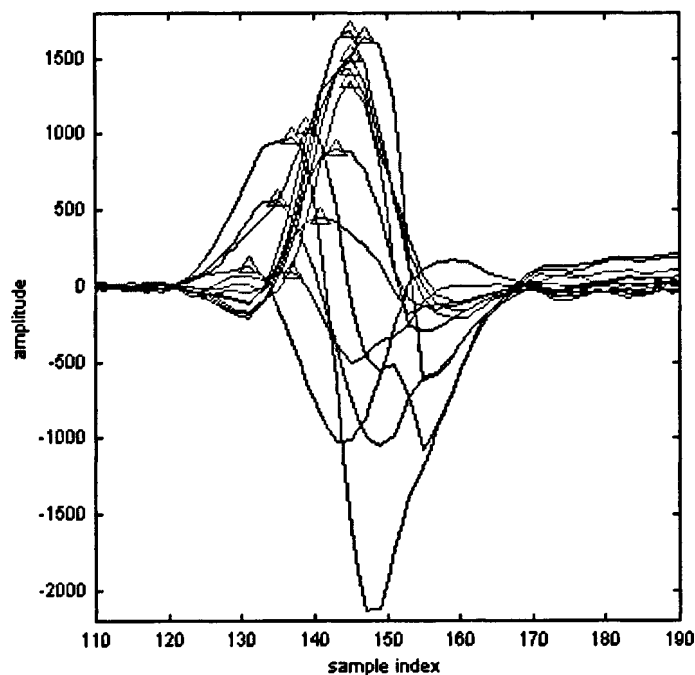


Рис. 4.14. Определение положения R-зубца в каждом из отведений.

Так как частоты дискретизации невелики (несколько десятков отсчетов в диапазоне поиска), то в качестве алгоритма поиска максимума достаточно использовать простой метод последовательного перебора.

Таким образом, выходными данными детектора, используемыми при дальнейшем анализе комплексов, является вектор R-зубцов по каждому определенному комплексу, а также набор соответствующих амплитуд.

4.2.3 Определение параметров Q-волны

Для выделения всего QRS-комплекса из ЭКГ необходимо применение дополнительных методов обработки данных, проводимых независимо от QRS-детектора, представленного в предыдущем параграфе. После этого необходима обработка каждого выделенного комплекса с целью выделения Q- и S-зубцов, а также точных границ комплекса. Процесс выделения QRS-комплекса может быть начат только тогда, когда для детектированного R-зубца существует верифицированный предшествующий зубец, а также 2-х следующих за ним. После это определяется грубый интервал нахождения

QRS-комплекса, который принимается равным $\left[R_{pos}^{lead} - \frac{1}{3} RR_{-}, R_{pos}^{lead} + \frac{2}{3} RR_{+} \right]$,

где RR_{-} и RR_{+} - интервал между предыдущим R-зубцом и последующим соответственно. Поиск точных границ начала и окончания QRS-комплекса будет проводиться в выделенном интервале, составляющем

$$L_{QRS} = \left[\frac{1}{3} RR_{-} + \frac{2}{3} RR_{+} \right] \text{ отсчетов.}$$

Следующие операции необходимо выполнить для всех независимых отведений I, II, V₁, V₂, V₃, V₄, V₅, V₆:

а) вычисление адаптивного среднего:

$$S_{15}(i) = \frac{x(i)}{2}, \text{ при } i = 1$$

$$S_{15}(i) = S_{15}(i-1) - c(S_{15}(i-1) - x(i)), i > 1 \quad (4.16)$$

константа адаптации $c = 0.15$.

б) аналогичным способом при $c=0.01$ получается поток S_1 , который в отличие от предыдущего вычитается из исходного сигнала:

$$S_1(i) = x(i) - S_1(i)'$$

в) третий поток S_{56} может быть получен НЧ-фильтрацией фильтром Кайзера 56 порядка с $F_s = 30$ Гц.

Для определения предпочтительных каналов для поиска начала QRS-интервала производится анализ потока S_1 , и выбираются только те отведения, которые попадают в определенный временно-амплитудный диапазон (Рис. 4.15). Пригодные для дальнейшего анализа отведения помечены на рисунке стрелками.

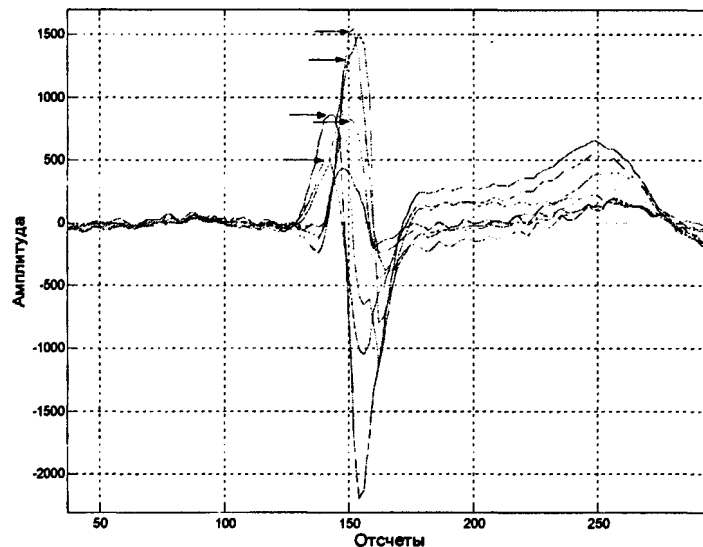


Рис. 4.15. Выбор предпочтительных отведений.

Данный диапазон по времени ограничен отсчетом R_{pos} , которому соответствует максимальное значение амплитуды сигнала в данной точке, а по амплитуде границей выступает значение минимальной рассматриваемой амплитуды равной 400 мкВ. В случае, если ни одно из значение не достигает нижней границы в 400 мкВ, то для анализа выбирается то отведение, амплитуда R-зубца которого максимальна. Анализ выбранного отведения необходимо производить на интервале $\left[R_{pos}^{lead} - \frac{1}{3} RR_-, R_{pos}^{lead} \right]$ сигнала S_1^{lead} .

Далее сигнал S_1^{lead} подвергается нормированию:

$$\bar{S}_1^{lead}(k) = \frac{S_1^{lead}(k)}{\max |S_1^{lead}(k)|} \frac{1}{3} RR_- \quad (4.17)$$

После этого определяется отсчет k_Q^{max} , в котором достигает минимума разностная величина:

$$\min_{0 < k < \lfloor \frac{1}{3} RR \rfloor} \left(\bar{S}_1^{lead}(k) - k \right) \quad (4.18)$$

Минимум этой функции может соответствовать минимуму Q-волны или соответствовать началу QRS-комплекса. Принять решение о принадлежности данной точки позволяет анализ потоков S_{15} и ЭКГ-сигнала S_{56} , прошедшего обработку ФНЧ. S_{15} усредняет сигнал и в точках скачков амплитуды может значительно расходиться с исходным сигналом, обрабатывая изменения с некоторой задержкой. Это позволяет построить следующий критерий для определения Q-зубца и начала QRS-комплекса. От начала Q-зубца разность $S_{15}(i) - S_{56}(i)$ возрастает и достигает своего минимума на вершине Q-волны. Таким образом, первый отсчет Q-волны может быть найден путем поиска в обратном направлении от k_Q^{max} , пока выполняется следующий критерий:

$$\sum_{k_Q^{max}}^{k_Q^{max} - i + 1} (S_{15}(k) - S_{56}(k)) > (1 + \alpha) \sum_{k_Q^{max}}^{k_Q^{max} - i} S_{15}(k) - S_{56}(k) \quad (4.19)$$

где α - некоторое значение, характеризующее минимальный прирост разности амплитуд этих двух потоков, связанное с доверительной вероятностью определения начала QRS-комплекса. Для $F_s = 500$ Гц это значение было выбрано равным 0.05.

Таким образом, начало Q-волны определяется в отсчете $k_Q^{min} = k_Q^{max} - i$.

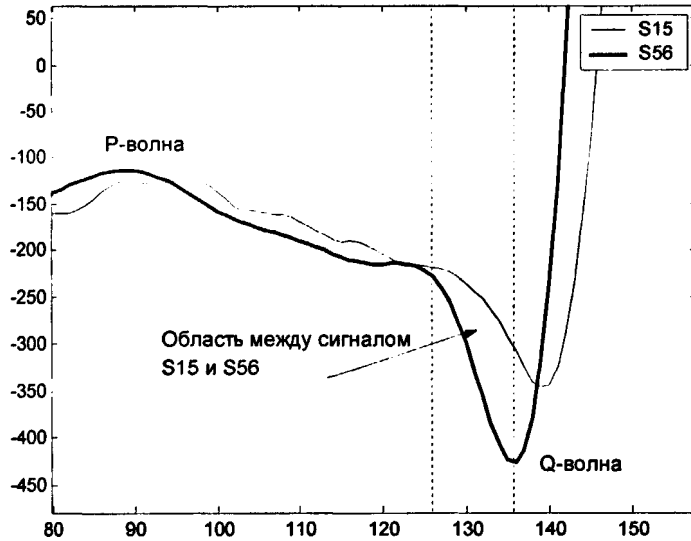


Рис. 4.16. Определение первого отсчета Q-волны.

Для определения амплитуды Q-волны необходимо определить точку нулевого потенциала, относительно которой можно сделать вычисления. В качестве подобной изолинии может выступать отрезок, соединяющий первый отсчет сигнала S_{56} и k_Q^{\min} (см. Рис. 4.17). Следовательно, амплитуда

$$Q\text{-волны составляет: } A_Q = S_{56}(1) + \frac{k_Q^{\max}}{k_Q^{\min}} (S_{56}(k_Q^{\min}) - S_{56}(1)) \quad (4.20)$$

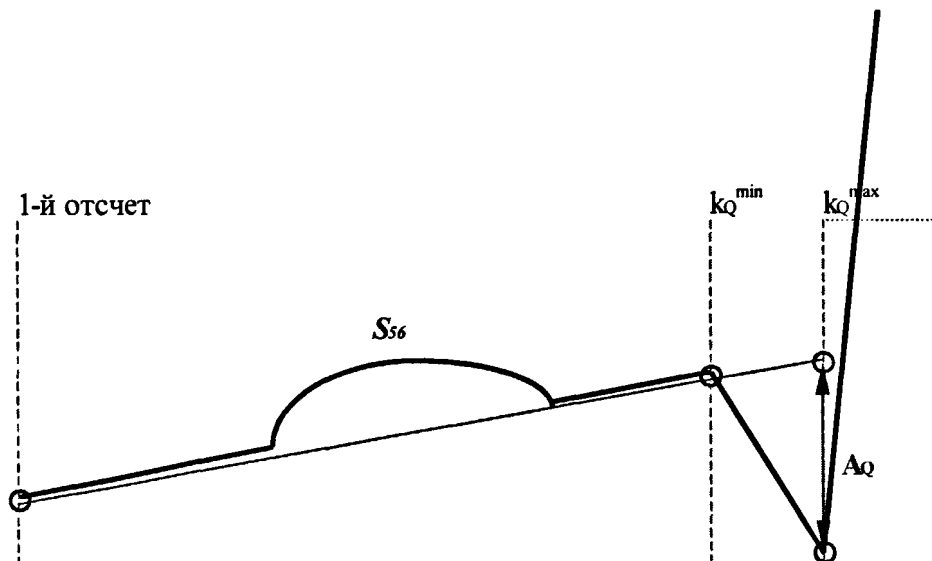


Рис. 4.17. Определение амплитуды Q-волны.

Критерием истинности Q-волны является попадание менее 2% отсчетов в диапазон ограниченный $0.7A_q$. В противном случае, k_Q^{\min} принимается равным k_Q^{\max} , Q-волна помечается как неопределенная. После это необходимо принять окончательное решение о подлинности определенной Q-волны, которая может являться шумовым всплеском. Для этого делается оценка шума в каждом из отведений путем вычисления разностного сигнала $S_{res} = S_{56-x}$ в диапазоне от 1 до k_Q^{\min} . Окончательное значение параметров Q-волны (k_Q^{\min} и k_Q^{\max}) усредняется по всем отведениям.

4.2.4 Коррекция дрейфа изолинии

Коррекция изолинии производится по единому алгоритму в каждом из присутствующих отведений. Возможно несколько методов интерполяции изолинии: кусочно-линейная, полиномиальная и т.д. Смешанная интерполяция, подразумевающая аппроксимацию на определенных отрезках изолинии линейной функцией и полиномами, показала свою эффективность. Ниже рассматривается вариант реализации интерполяции, при которой полагается, что отрезок изолинии на интервале нахождения QRS-комплекса, т.е. с отсчета $\left[R_{pos}^{lead} - \frac{1}{3}RR_- \right]$, линеен, а на других интервалах – аппроксимируется полиномами 3-го порядка.

Таким образом, опять переходя от абсолютных значений отсчетов к величинам отсчетов, привязанных к диапазону нахождения QRS-комплекса, получаем выражение для аппроксимирующей изолинию прямой:

$$f(k) = S_{56}(1) + \frac{S_{15}(k_Q^{\min}) - S_{56}(1)}{k_Q^{\min}}(k-1), 1 \leq k \leq k_Q^{\min} - 1 \quad (4.21)$$

На оставшемся отрезке, изолиния интерполируется многочленом 3-го порядка: $f(k) = ak^3 + bk^2 + ck + d, k_Q^{\min} < k \leq L_{QRS}$ (4.22)

при следующих условиях:

$$\begin{aligned} f(k_Q^{\min}) &= S_{15}(k_Q^{\min}) \\ f'(k_Q^{\min}) &= \frac{S_{15}(k_Q^{\min}) - S_{56}(1)}{k_Q^{\min}} \\ f(L_{QRS}) &= S_{56}(L_{QRS}) \end{aligned} \quad (4.23)$$

$$f'(L_{QRS}) = \frac{\sum_{i=L_{QRS}-\Delta}^{L_{QRS}} (i - (L_{QRS} - \frac{\Delta}{2}))(S_{15}(i) - \bar{S}_{15})}{\sum_{i=L_{QRS}-\Delta}^{L_{QRS}} (i - (L_{QRS} - \frac{\Delta}{2}))^2},$$

где $\Delta = \left[\frac{1}{9} RR_+ \right]$ -интервал линейной регрессии.

Результат коррекции изолинии представлен на Рис. 4.18.

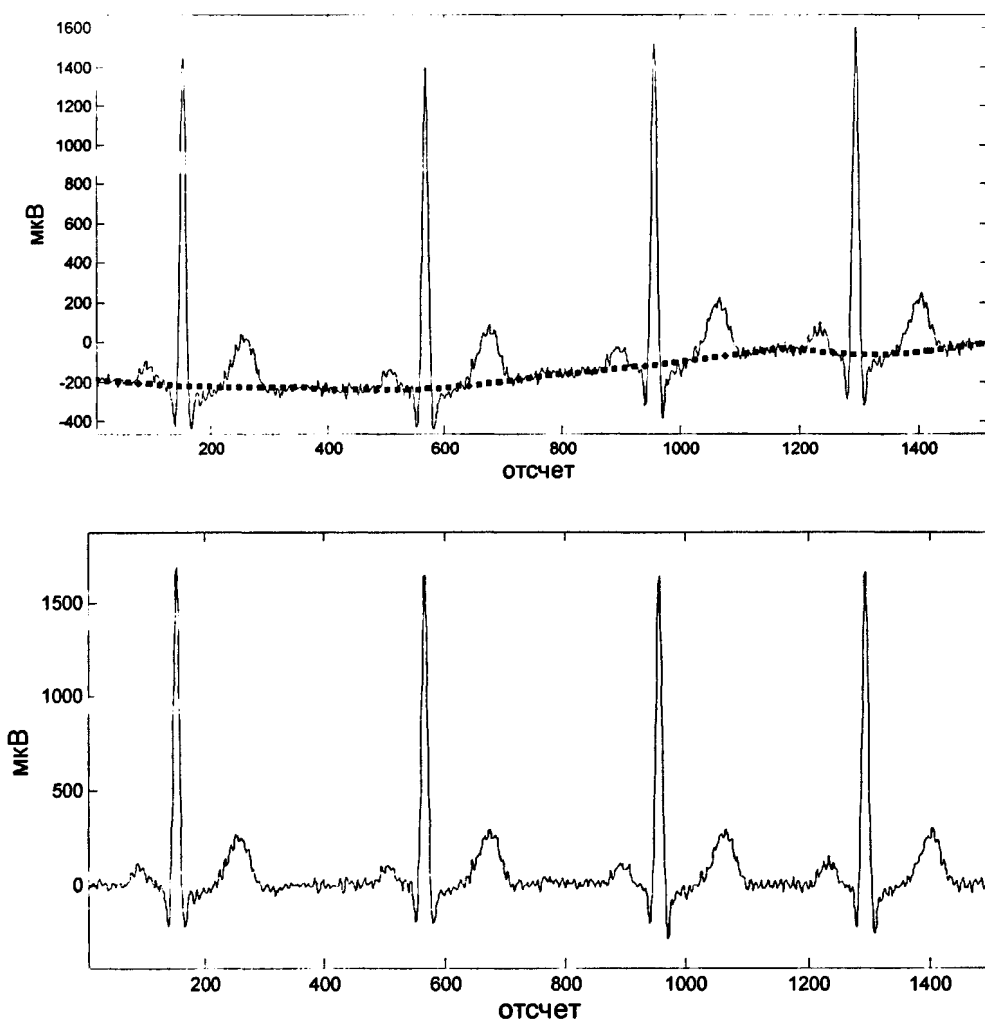


Рис. 4.18. Коррекция изолинии (отведение II).

Определение изолинии необходимо проводить для всех сигналов за исключением зависимых отведений, в частности, в случае системы 12 стандартных отведений – для III, aVR, aVL, aVF коррекция изолинии не нужна. Недостатком данного метода коррекции изолинии является то, что в случае если P-волна находится в точке ожидаемой границы комплекса, т.е. на расстоянии двух третей от R-зубца, будет наблюдаться смещение потенциала от корректной изолинии.

4.2.5 Определение параметров S-волны и окончания QRS-комплекса

Параметры S-волны и длительность всего QRS-комплекса имеют большое значение в медицинской диагностике[113]. Выделение S-зубца и определение окончания QRS-комплекса проводится в несколько этапов:

- 1) корректировка R_{pos} в каждом отведении;
- 2) определение наиболее подходящих отведений для анализа;
- 3) проверка наличия явно выраженного S-зубца;
- 4) определение вершины S-зубца и соответствующей амплитуды;
- 5) вычисление границы завершения QRS- комплекса;
- 6) верификация окончания QRS-комплекса.

Корректировка R_{pos} заключается в проверке условия $R_{pos}^{lead} > k_Q^{min}$, где k_Q^{min} усреднено по отведениям. Определение наиболее подходящих отведений производится путем поиска минимального значения по всем отведениям на интервале $[R_{pos}^{lead}, R_{pos}^{lead} + 2(R_{pos}^{lead} - k_Q^{min})]$. По результатам анализа выбирается до 4 отведений с минимальными экстремумами в каждом из них. Далее проводится анализ на интервале $[R_{pos}^{lead}, R_{pos}^{lead} + 0.08c]$ выраженного S-зубца путем построения регрессионного детектора. Работа регрессионного детектора заключается в разбиении исходного интервала на два сегмента на каждом шаге итерации и вычислении линейной регрессионной функции на каждом из них. Анализ функции разности углов между двумя регрессионными прямыми позволяет сделать вывод о наличии S-зубца. Результатом такого анализа является подтверждение факта присутствия S-зубца и его уточненный временной интервал. Положение S-зубца k_s^{max} может быть определено путем поиска минимума в данном диапазоне.

Поиск границы S-волны может быть осуществлен подобным образом, т.е. путем разбиения интервала от отсчета k_s^{max} до ожидаемого завершения S-волны на 2 сегмента и определения параметров линейной регрессии на каждом из них. Экстремум функции разности углов между регрессионными прямыми на двух интервалах позволяет сделать вывод об определении границы завершения QRS-комплекса.

Завершающим этапом анализа S-волны является коррекция границы QRS-комплекса k_s^{min} , которая заключается в группировке трех наиболее близких значений k_s^{min} каждого из четырех выбранных отведений и последующего их усреднения.

Таким образом, определяются параметры QRS-комплекса R_{pos} , k_s^{min} , k_s^{max} , k_Q^{min} , k_Q^{max} и набор амплитуд A_R, A_S, A_Q . Заключительным этапом анализа QRS-интервала является процедура уточнения значений найденных точек и амплитуд элементов в комплексе.

4.2.6 Особенности реализации QRS-детектора

Реализация данной методики выделения QRS-комплексов на основе прикладных библиотек цифровой обработки сигналов подразумевает формализацию шагов до уровня конкретной используемой прикладной библиотеки. Под этим понимается то, что шаги алгоритма следует, по возможности, осуществлять с помощью процедур, оптимизированных для

выбранного процессора, предоставляемых конкретной библиотекой. Блок-схема реализации на основе программного обеспечения для цифровой обработки сигналов NSP компании Intel представлена на рис. 5. Блок начальной обработки включает в себя функции фильтрации, полностью реализуемые на основе функций библиотеки NSP. Возможна эффективная реализация процедуры фильтрации как целочисленно, так и с плавающей точкой. Более того, эта прикладная библиотека содержит оптимизированные функции операций с векторами, что позволяет эффективно реализовать следующие операции: дифференцирование, интегрирование сигнала, сравнение адаптивных пороговых значений и вычисление функции $\theta(n)$, нахождение максимума. Все эти операции, в действительности, сводятся к простым манипуляциям с векторами, поэтому оказывается эффективной группировка входных данных в блоки с длительностью по времени до 1 сек. Эти блоки данных являются входящими для модуля начальной обработки и, следовательно, QRS-детектор не может выдавать значения R-зубцов с частотой, превышающей частоту поступления входных данных. Отдельным вопросом является реализация архитектуры линейных фильтров. Кроме использования одной из известных форм их реализации возможно сведение исходной процедуры фильтрации к свертке [114]. При этом потоковая фильтрация распадается на короткие секции фильтра, которые могут быть эффективно вычислены короткими БПФ, а исходный сигнал получен методами так называемого перекрытия с суммированием или накоплением. В настоящий момент существует огромное количество разновидностей БПФ, берущих свое начало с известной публикации Кули и Тьюки.[115]. Первый алгоритм был применен к сигналам длиной в 2^N отсчетов и выполнялся итерационно, разбивая на каждом шаге обрабатываемую секцию на 2 части, и получил наименование прореживания по времени. Позднее была получена схема прореживания по частоте, которая, наоборот, на каждой следующей итерации объединяла секции. Впоследствии стали применяться алгоритмы с основанием, отличным от 2 (4,8), а также с переменным основанием. Это дает возможность еще больше сократить количество операций, сокращая количество итераций и переходя к оптимизированным 4,8-размерным секциям. Если длина сигнала N допускает достаточно хорошее разложение на множители, то для таких длин тоже можно построить эффективную схему БПФ. Другим подходом воспользовался Виноград, построив эффективные алгоритмы БПФ для малых длин $N=2,3,4,5,7,8,11,13,16$. Предложенная Виноградом переиндексация позволила существенно сократить количество умножений за счет сложений. Однако, как показал опыт, применение БПФ на основе секций Винограда в случае операций с плавающей точкой не приносит желаемых преимуществ.

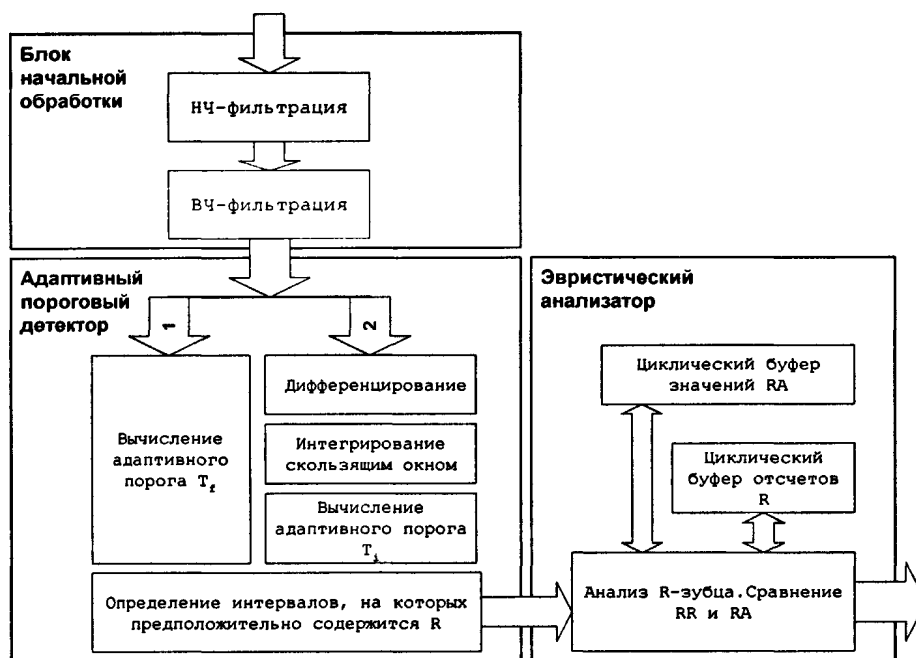


Рис. 4.19. Реализация QRS-детектора на основе библиотеки ЦОС Intel.

Существующие быстрые алгоритмы для вычисления вещественного ДПФ, преобразования Хартли, синусного или косинусного преобразования Фурье сводят их к уже существующим схемам БПФ.

Представленная реализация QRS-детектора интегрирована в рамках программного ЭКГ-комплекса «IntelliCard», осуществляющего полную сегментацию ЭКГ в режиме реального времени. Метод был апробирован как в пакетном режиме на сериях тестовых ЭКГ с расставленными метками R-зубцов, так и в режиме реального времени с использованием эмулятора ЭКГ. Тестирование алгоритма проводилось на ЭКГ-записях, полученных на 12-канальном цифровом компьютерном электрокардиографе KARDi [88].

Чувствительность алгоритма – количество корректно определенных QRS-комплексов из общего числа присутствующих в тестовых ЭКГ – оказалась на уровне 96-99,5%, в зависимости от соотношения присутствующих в тестовой подборке патологических ЭКГ и характера патологий. Выяснено, что основные ошибки алгоритма происходят за счет того, что используемые пороги могут оказаться достаточно высокими, и существует небольшая вероятность того, что настоящий R-зубец просто не будет определен пороговым детектором, особенно в первые секунды работы алгоритма, когда происходит первоначальная подстройка пороговых значений [116].

Итак, анализ различных методов выделения QRS-комплексов показал, что хотя существуют и более точные методы, до сих пор наибольшую целесообразность для систем реального времени имеет применение классического подхода при использовании различных методов,

позволяющих повысить чувствительность алгоритма. При реализации критичных ко времени выполнения процедур необходимо опираться на существующие оптимизированные библиотеки цифровой обработки сигналов.

4.2.7 Реализация алгоритма детектирования QRS- комплексов на базе микроконтроллера

Для целого ряда медицинских устройств, таких как, например, портативные кардиомониторы, интеллектуальные дефибрилляторы, необходима реализация алгоритма выделения QRS-комплексов для определения RR-интервалов, а применение дополнительного специализированного DSP-процессора часто невозможно из-за достаточно высокого потребления. К настоящему времени появилось несколько сообщений о реализации различных методов детектирования QRS-комплексов на базе неспециализированных микроконтроллеров[117]. Ряд современных микроконтроллеров, в частности микроконтроллеры серии C166 фирмы Infineon, обладают рядом специфических особенностей: акселерация операций целочисленного умножения и деления, и наличие 16 регистров общего назначения, что является характерным для RISC-архитектуры (см. Рис. 4.20). Каждый из этих регистров может использоваться как аккумулятор, указатель при косвенной адресации или для хранения индекса.

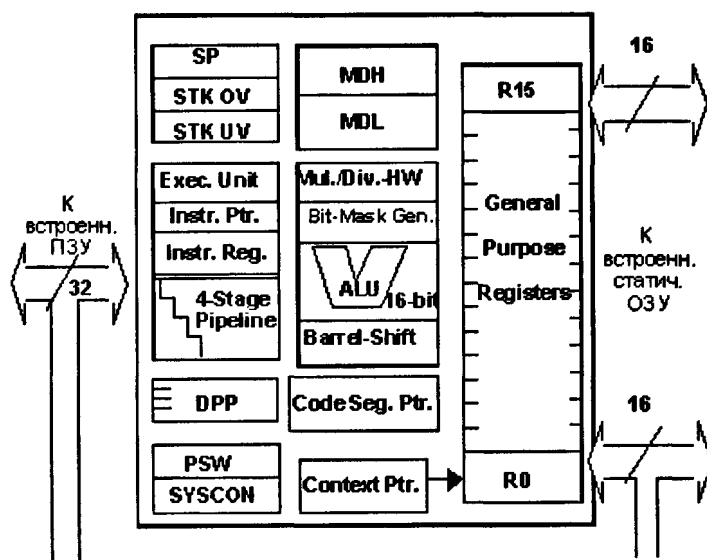


Рис. 4.20. Архитектура микроконтроллеров семейства C166.

Регистровый банк включает до 8 регистров с пословной адресацией и 8 регистров с возможностью адресации младшего и старшего байта, все регистры общего назначения адресуемы побитно (Рис. 4.21). При столь большом числе регистров становится возможным держать все или почти

все локальные и промежуточные переменные в регистрах, что сокращает частоту обращений к внешней памяти и существенно повышает скорость.

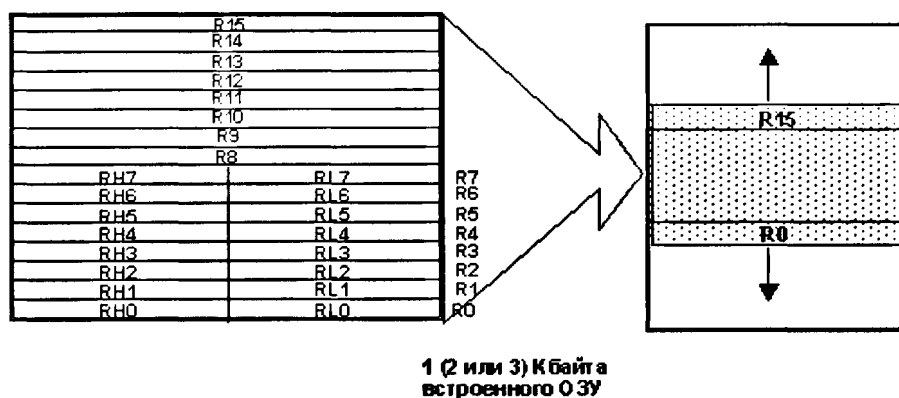


Рис. 4.21. Архитектура регистров C166.

Представленные факты позволяют утверждать, что микроконтроллеры этой серии наряду с функциями управления способны одновременно выполнять задачи цифровой обработки сигналов.

Алгоритм выделения QRS-комплексов, рассмотренный в этой главе, для эффективной реализации на базе данного микроконтроллера должен быть специально подготовлен для целочисленных операций процессора [118]. Особенности реализации состоят в том, что необходимо избегать применения КИХ-фильтров из-за того, что для эффективной фильтрации необходимо большее количество математических операций по сравнению с БИХ-фильтрами. БИХ-фильтры произвольного порядка могут быть построены на основе каскадирования секций БИХ-фильтра 2-го порядка. Если входные АЦП не осуществляют НЧ-фильтрацию на 37-40 Гц, тогда необходим дополнительный ФНЧ с $f_{cp}=37$ Гц, который одновременно осуществляет подавление помехи 50/60 Гц и шума электрической активности мышц.

Особенности реализации алгоритма состоят с том, что все математические операции должны производиться целочисленно. Реализуется это следующим образом. Предполагается, что все перемножаемые величины нормированы на 1, т.е. лежат в интервале $[-1,1]$. Для перевода дробной части числа в 16-битное представление необходимо умножить каждое значение на 32768(8000h) и преобразовать отрицательные числа в двоичный дополнительный код. Таким образом, после умножения результат операции помещается в регистры MDL и MDH, причем дробная часть числа заканчивается на 13 бите регистра MDH. При сложении чисел не требуется дополнительных преобразования формата числа, но необходимо следить за флагом переполнения.

Реализованный алгоритм был апробирован и в настоящее время применяется в дефибриляторе с микроконтроллером SIEMENS C164ci [119].

Заключение

1. Распределенная компонентно-ориентированная архитектура ПО на базе принципов динамического связывания и распределения вычислений обеспечивает системы сбора и обработки электрофизиологической информации возможностями постоянного расширения и наращивания их функциональных свойств, интеграции в единую систему разнородных компьютерных биомедицинских систем. Особенно следует отметить обеспечение полной *интероперабельности* в рамках программной архитектуры и широкие возможности повторного использования кода уже разработанных элементов системы.
2. На основе разработанной программной архитектуры, включающей спецификации интерфейсов взаимодействия, модели организации вычислительных процессов, физическую модель данных и модель распределения БМИ, построен экспериментальный переносной электрокардиографический компьютерный комплекс «Intellicard», обеспечивающий функции регистрации и обработки данных.
3. Разработанные программные компоненты, в том числе методики предварительной обработки сигналов, сжатия и визуализации БМИ, выделения и анализа QRS-комплексов, подсистема управления устройствами регистрации, могут быть применены в качестве базового набора модулей при проектировании произвольной БМС регистрации и обработки электрофизиологических сигналов.
4. Использование современных компонентно-ориентированных систем промежуточного ПО позволяет избежать дополнительных затрат и падения производительности при переходе от монолитной к распределенной программной архитектуре. Применение компонентной архитектуры промежуточного ПО COM+ не приводит к сколько-нибудь заметной потере производительности и не уступает в этом параметре монолитным архитектурам. Затраты выделяемого времени ЦП на выполнение специфических процедур поддержки распределенных компонент, в частности, создание, удаление и управление объектами, не превышает нескольких процентов от общего времени, выделяемого ОС всей системе в целом.
5. Решающее значение при проектировании программных БМС сбора и математической обработки физиологических сигналов имеет ориентация на общепризнанные стандарты в данной области, в частности, IEEE 1073.
6. Выделение вычислительных задач регистрации БМИ и математической обработки данных в отдельные процессы на основе технологий промежуточного ПО позволяет организовать

распределенную схему вычислений без дополнительных затрат на синхронизацию и взаимодействие удаленных процессов.

7. Внедрение принципов разработки распределенных компьютерных БМС реального времени на основе компонентных архитектур промежуточного ПО позволяет существенно сократить время проектирования, повысить их качество и эффективность, решить многие проблемы интеграции и взаимодействия БМС различных производителей.

Литература

1. Kurekov S. F., Prilutski D. A., Selishchev S. V. Sigma-Delta Analogue-to-Digital converters for Biomedical Data Acquisition Systems // Proceeding of 4th European conference on engineering and medicine.- Warsaw, May 25-28, 1997.-P.163-164.
2. Distributed Component Object Model Protocol. – Microsoft, version 1.0, January 1998.
3. Bronzino J. Biomedical Engineering Handbook, Biomedical Signals: Origin and Dynamic Characteristics; Frequency-Domain Analysis. - CRC Press. –1999. – pp. 805-827.
4. Е.В. Аксенов и др., Системы сбора и обработки электрофизиологической информации на основе сигма-дельта аналого-цифрового преобразования //Биомедицинские технологии и радиоэлектроника. – №12-2001. – стр. 56-65.
5. Водолазский Л. А. Основы техники клинической электрографии.– Москва: Медицина.–1966.–270 с.
6. Шакин В. В. Вычислительная электрокардиография.– М.: Наука.–1981.–166 с.
7. Дроздов Д. В. Персональный компьютер в качестве электрокардиографа — за и против. // М: Компьютерные технологии в медицине.–1996.–№1.–С. 18-23.
8. Metnitz P. G. H., Lenz K. Patient data management systems in intensive care - the situation in Europe. – Intensive Care Medicine, Springer-Verlag. –1995, 21(7). – pp. 703-710.
9. ISO/IEC 7498: Information Technology – Open System Interconnection. – Internation Organization for Standartization. – Geneva.
10. Health Level 7. – ANSI Standard. – version 2.4, 2000.
11. ENV 13606-1: Health informatics – Electronic healthcare record communication. – European committee for standardization.
12. IEEE 1073: Standard for Medical Device Communications. Overview and Framework.– The Institute of Electrical and Electronics Engineers, 1996.
13. Booch G. Object Solutions.–Addison-Wesley, 1996.- 322 p.
14. Shaw M., DeLine R., Klein D., Ross T., Young D., Zelesnik G. Abstraction for Software Architecture and Tools to Support Them. – IEEE Transactions on Software Engineering. – Vol. 21, No. 4. – 1995.
15. Селищев С.В. Автоматизированное проектирование биомедицинских электронных систем // Биомедицинские технологии и радиоэлектроника. – М.: №12, 2001. – с. 5-17.
16. Роджерсон Д. Основы COM. – М: Русская редакция ТОО “ Channel Trading Ltd.”. – 1997
17. The component object model specification. – Draft Version 0.9. Microsoft.–1995.
18. Box D. Essential COM.– Addison Wesley Longman, Inc, 1998.
19. Чаппел Д. Технологии ActiveX и COM. – Microsoft Press.– 1997.
20. Хабибуллин И.Ш. Создание распределенных приложений на Java 2 // М.- 2002 г. 704 стр.
21. CORBA/IIOP 2.2 Specification. – OMG. – version 2.2. – June, 1998.
22. CEN/TC251/N98-108: Short strategic study: enabling technologies – CORBA and COM/DCOM, Final Report. –TC251 Secretariat. –Brussels, Belgium. – 1998.
23. Recommendations for HL7 messaging over component technologies.–HL7 Committee.– ver. 1.0. – 1998.
24. Jean F.C., и др. The HELIOS medical connection services // Computer methods programs. biomed. – 1994; 45.– pp. 117-126.
25. Lavril M., и др. A Reuse oriented development database: The HELIOS object information system // Comput. methods programs biomed., 1994, pp. 35-45.

26. Wiederhold G, Mediators in the architecture of future information systems// IEEE Computer, 1992: 25 (3): 38-49.
27. Xu Y., Degoulet P. Using XML in a component based mediation architecture for the integration of applications // CD-ROM Proceedings of International Conference "XML Europe-2001".– Berlin.– May, 2001.
28. ActiveX for Healthcare messaging, Programmer's guide.– Microsoft Healthcare user group, ActiveX for Healthcare Committee, version 1.60, November 1998.
29. CEN ENV 12967-1: Healthcare Information System Architecture (HISA) – Part 1: Healthcare Middleware Layer.– European Prestandard. – CEN.– 1998.
30. Ferrara F.M. The standard "Healthcare Information Systems Architecture" and the DHE middleware // Int. J. med. Inf. 52 (1998), p. 39-51.
31. Нагин В.А., Потапов И.В. Применение технологий OLE и COM в разработке программного обеспечения медицинских информационных систем.// Тезисы докладов Всероссийской межвузовской конференции студентов и аспирантов «Микроэлектроника и информатика-98», – М., 1998, – ч. 1, с. 201.
32. B. Cesnik, A.T.McCray, J.-R. Scherrer (Eds.) Systems // MEDINFO 98 . – IOS Press, Amsterdam, 1998, p. 217-21.
33. Spahni S., Scherrer J.R., Sauquet D., Sottile P.A. Middleware for healthcare information systems// In: MEDINFO 98. – IOS Press.– Amsterdam, 1998, pp. 212-16.
34. Nagin.V.A, Potapov I,V, Selishchev S.V. A universal object-oriented medical database system // CD-ROM proceeding of 23 IEEE International conference on engineering in medicine and biology. – Istanbul, October 25-28, 2001.
35. Sakellaris George C.Fotiadis D.I. ;Xmedia Communication System: An XML driven medical image processing and archiving environment // // CD-ROM Proceedings of 23rd IEEE International conference on engineering in medicine and biology. – Istanbul, October 25-28, 2001.
36. Нагин В.А., Потапов И.В., Прилуцкий Д.А., Селищев С.В. Программная система для ЭКГ на основе компонентной архитектуры COM// Тезисы докладов международной конференции по биомедицинскому приборостроению «Биомедприбор-98», – М.,1998 – с. 215-217.
37. Нагин В.А., Потапов И.В. Программный комплекс для компьютерной электрокардиографии на основе компонентной архитектуры COM// Тезисы докладов Международного симпозиума "Компьютерная электрокардиография на рубеже столетий XX–XXI". – Москва, 27-30 апреля 1999.–с. 270-272.
38. Scherrer J.R., MD, Spahni S., Healthcare Information System Architecture (HISA) and its Middleware Models// In: MEDINFO 98.– IOS Press.– Amsterdam 1998. – pp. 212-16.
39. Крачтен Ф. Введение в Rational Unified Process, 2-е изд.: пер. с англ. – М.:Издательский дом «Вильямс», 2002.-240 стр.
40. Буч Г., Рамбо Д., Джекобсон А. UML: руководство пользователя: пер. с англ. – М.: ДМК, 2000.- 432 с.
41. Волкова В.Н., Денисов А.А. Основы теории систем и системного анализа. – изд. 2-е.- СПб.: СПбГТУ, 1999.-512 с.
42. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++.– СПб.: Бином.– 1998.– 560 с.
43. Потапов И.В., Нагин В.А. Использование сервиса USSD в сетях GSM для задач телемедицины //Тезисы докладов Международной научно-технической конференции «Электроника и информатика-2002», – М., 2002.
44. Потапов И.В., Нагин В.А., Селищев С.В. Применение микропроцессорных smart-карт для защиты данных медицинских приложений// Труды международной

- конференции по биомедицинскому приборостроению – «Биомедприбор-2000», – М.,2000 – ч. 2. с. 217-218
45. Нагин В.А. Система защищенного хранения медицинских данных // Тезисы докладов Всероссийской межвузовской конференции студентов и аспирантов «Микроэлектроника и информатика-2002», – М.,2002. – с. 85.
46. Гома Х. Проектирование систем реального времени, распределенных и параллельных приложений: Пер. с англ. – М.:ДМК Пресс,2002. – 704 с.
47. Michael R. Garey and David S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness.–W. H. Freeman and Company.– New York. – 1979.
48. Walid A. Najjar, Edward A. Lee,Guang R. Gao Advances in the data flow computational model, // Parallel Computing, 25 (1999). – Elsevier, pp. 1907±1929,
49. Lee E. A. and Messerschmitt D. G. Static scheduling of synchronous data flow Programs for Digital Signal Processing// IEEE Trans. on Computers. – January 1987. –с-36(2).
50. Halbwachs N., P. Caspi, P. Raymond, and Pilaud D. The synchronous data flow programming language LUSTRE.// Proc. of the IEEE, 79(9).– pp. 1305–1321. –Sept. 1991.
51. Нагин В.А. Планирование вычислительных процессов в компьютерных биомедицинских системах// Тезисы докладов Международной конференции «Электроника и Информатика - 2002», Москва, Зеленоград.
52. Sha L., Goodenough J.B. Real-time scheduling theory and Ada // IEEE Computer, 23, no. 4. – 1990.
53. Ida T., Tanaka J. Functional programming with streams. – Information Processing '83. – Elsevier Science pubs. – 1993.
54. Franco J., Friedman D. P., and Johnson S. D., Multi-way streams in scheme. –Comput. Lang., Vol. 15, No. 2. – pp. 109-125. – 1990.
55. Chang W.T., Ha S., Lee E. A., Heterogeneous simulation—mixing discrete-event models with dataflow// Journal of VLSI Signal Processing 15.–127–144 (1997).– Kluwer Academic Publishers.
56. Шибанов Г.П. Количественная оценка деятельности человека в системах «человек-техника» – М.: Машиностроение, 1983 – 263 с.
57. Lee E. A., Murthy P. K. Multidimensional synchronous dataflow// IEEE Transactions on Signal Processing.– July 2002.
58. IEEE 754-1985: Standard for Binary Floating-Point Arithmetic. – IEEE. – 1985.
59. Shuvra S. Bhattacharyya Edward A. Lee A Memory management for dataflow programming of multirate signal processing algorithms.– IEEE Transactions on Signal Processing, Vol. 42, No. 5. – May 1994.
60. Гульяев А. К. MATLAB 5.3. Имитационное моделирование в среде Windows. //М.: Корона Принт, – 2001 – 400 с.
61. Таха Х.А., Введение в исследование операций, 6-е издание: пер. с англ. – М: Издательский дом «Вильямс», 912 с.
62. Ramamritham K., and others, Using Windows NT for real-time applications: experimental observations and recommendations// Proceedings of the IEEE Real-time technology and applications symposium. -1998.-Denver, Colorado.
63. Нагин В.А., Потапов И.В., Селищев С.В., Шарф В. Программный комплекс сбора и математической обработки ЭКГ-данных на основе компонентной архитектуры COM. // М.: Медицинская техника.–2001.–№1. – С. 3-7.
64. Нагин В.А., Потапов И.В. Высокопроизводительная компонентная архитектура цифровой обработки сигналов для систем компьютерной электрокардиографии //

Тезисы докладов Всероссийской межвузовской научно-технической конференции «Микроэлектроника и информатика – 99», – Москва, 1999, с. 121.

65. Нагин В.А., Потапов И.В., С.В. Селищев. Система удаленного доступа к данным медицинских приложений на основе микропроцессорных smart-карт// Материалы 4 международной конференции «Физика и радиоэлектроника в медицине и экологии-2000», – Владимир, 2000. – ч. 2, с. 122-124.

66. Цимерман Ф. Клиническая электрокардиография. - М.: Восточная Книжная Компания, 1997.- 448с.

67. Java Remote Method Invocation (RMI), Developer documentation // <http://java.sun.com/products/jdk/rmi/>.

68. Nagin V.A., Potapov I.V., Selishchev S. V. A Distributed component-oriented architecture for real-time ECG data acquisition systems. CD-ROM Proceedings of 23rd IEEE International conference on engineering in medicine and biology.- Istanbul, October 25-28, 2001.

69. CEN ENV 13735: Interoperability of patient connected medical devices. – European Commission for Standardisation (CEN), Brüssel.

70. Wooldridge M., Jennings N. R. Agent Theories, Architectures and languages: a survey and in intelligent agents. – Berlin: Springer-Verlag. – 1995.

71. Nagin V.A., Potapov I.V., Selishchev S. V. Design of acquisition devices management subsystem for IEEE 1073 compliant software agents// Studies in Health Technology and Informatics, vol. 90, pp. 774-779, – IOS Press, Amsterdam, 2002.

72. Куцевич Н.А Средства визуализации в системах реального времени// CD-ROM Proceedings of ICSNET'99. –Moscow, 1999.

73. American national standard: ambulatory electrocardiographs. Arlington, VA: ANSI/AAMI; 1994. EC 38.

74. Нагин В.А. Сжатие ЭКГ в компьютерных ЭКГ-системах // Тезисы докладов Всероссийской межвузовской конференции студентов и аспирантов «Микроэлектроника и информатика-2001», – М., – 2001, с. 129.

75. Плотников А. В., Прилуцкий Д. А., Селищев С. В. Стандарт DICOM в компьютерных медицинских технологиях. // М: "Медицинская техника".-1997.-№2.- С.18-24.

76. ENV-1613 Messages for exchange of laboratory information// European Committee for Standardisation (CEN). – Brussel. – Belgium.

77. B.Kemp, A. Vaerri, A.C. Rosa, K.D. Nielsen, J.Gade, Data interchange for clinical neurophysiology. – Electroencephalogr. Clin. Neurophysiol., vol. 82., pp. 391-393, 1992.

78. A. Vaerri, B. Kemp, T. Penzel, A. Schloegl, Standards for biomedical signal databases. – IEEE Engineering in medicine and biology mag. – vol. 20, No. 3, 2001. – pp. 33-44.

79. ASTM E1467-94: Standard specification for transferring digital neuropsychological data between independent computer systems.– American Society for Testing and Materials ASTM, 1994.

80. CEN ENV 1064. Standard Communications Protocol for Computer-Assisted Electrocardiography// European Committee for Standardisation (CEN), Brussel, Belgium, 1996.

81. Нагин В.А. Применение стандарта SCP-ECG в компьютерной электрокардиографии// Тезисы докладов II Всероссийской конференции с международным участием «Электроника и информатика-97», – М., 1997 – ч. 2, с. 144.

82. CEN/TC251/PT-40: File exchange format for vital signs, Interim Report, Revision 2. – TC251 Secretariat. – Sweden. – 2000.

83. CEN ENV 13734: Vital Signs Information Representation. – CEN. –Brussels, Belgium, 2000.
84. Hellmann G., Kuhn M., Prosch M., Spreng M., Extensible biosignal (EBS) file format - simple method for EEG data exchange // EEG clinic. Neurophysiol., in press, 1996.
85. Cunha, M.B., Cunha, J.P. and Oliveira e Silva, T. SIGIF: A digital signal interchange format for biological signals// Proceedings of the 15th Annual International Conference of the IEEE Engineering Medicine and Biology Society, 1993, 2:644-5.
86. Нагин В.А., Потапов И.В., Селищев С.В. Разработка и тестирование в распределенной объектно-ориентированной среде COM+ программных электрокардиографических систем реального времени // М.:Биомедицинские технологии и радиоэлектроника. – 2001. – №12. – с. 67-72.
87. Friesen G.M., Jannett T.C., Jadallah M.A., Yates S.L., Quint S.R., Nagle H.T, A comparison of the noise sensitivity of nine QRS detection algorithms// IEEE Trans. Biomed. Eng., vol. BME-37, pp. 85-98, 1990.
88. Куриков С. Ф., Прилуцкий Д. А., Селищев С. В. Применение технологии многоуровневого сигма-дельта преобразования в цифровых многоканальных электрокардиографах. // М.: Медицинская техника.–1997.–№4. – С. 7-10.
89. Солоница А.И. и др. Алгоритмы и процессоры цифровой обработки сигналов // СПб.:БХВ-Петербург, 2001-464 с.
90. Curtin M. Sigma-Delta techniques reduce hardware count and power consumption in biomedical analog front end // Analog Dialogue Journal.-1994.-V. 28.-№2.-. 6-8.
91. Куприянов М.С., Матюшкин Б.Д. Цифровая обработка сигналов // СПб: Политехника, 2002, стр. 304-358.
92. Ветвицкий Е.В., Плотников А.В., Прилуцкий Д.А., Селищев С.В. Применение универсального последовательного интерфейса USB в компьютерных медицинских комплексах. // М: Медицинская техника. – 2000. – №4. – С.3-7.
93. Куриков С. Ф., Прилуцкий Д. А., Селищев С. В. Применение технологии многоуровневого сигма-дельта преобразования в цифровых многоканальных электрокардиографах. // М.: Медицинская техника.–1997.–№4. – С. 7-10.
94. Curtin M. Sigma-Delta techniques reduce hardware count and power consumption in biomedical analog front end // Analog Dialogue Journal. 1994. V. 28. №2. P. 6-8.
95. Kurekov S. F., Prilutski D. A., Selishchev S. V.Sigma-Delta Analogue-to-Digital converters for Biomedical Data Acquisition Systems // Proceedings of 4th European conference on engineering and medicine.– Warsaw, May 25–28, 1997. – P.163-164.
96. McKee J. J., Evans N. E. and Wallace D.Sigma-Delta analogue-to-digital converters for ECG signal acquisition. // in CD-ROM Proceedings of 18th Annual International Conference of the IEEE EMBS. – Amsterdam, 1996.
97. Oppenheim, A.V., and R.W. Schaffer, Discrete-Time Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1989,
98. Нагин В.А. Методика предварительной цифровой обработки ЭКГ для программных ЭКГ-систем // Тезисы докладов Всероссийской межвузовской конференции студентов и аспирантов «Микроэлектроника и информатика – 2000», – М., 2000, – с. 97.
99. Haykin S. Adaptive Filter Theory, 3rd ed Englewood Cliffs// NJ: Prentice Hall, 1996.
100. Нагин В.А., Селищев С.В. Методика анализа QRS-комплексов для ЭКГ// Материалы I Евразийского конгресса по медицинской физике и инженерии, часть XI, "Математические, компьютерные и информационные технологии в медицине"// Медицинская физика. – М.:2001. – с 78.
101. Pan J. , Tompkins, A real-time QRS detection algorithm// IEEE Trans. Biomed. Eng., vol. BME-32, pp. 230-236, 1985.

-
102. Kaplan D., Simultaneous QRS detection and feature extraction using simple matched filter basis functions// Computers in Cardiology.– IEEE Computer Society Press. – 1990, pp.503-506.
 103. Li C., Zheng C, Tai C. Detection of ECG characteristic points using the wavelet transform// IEEE Trans. Biomed. Eng., 1995;42: pp.21-28.
 104. Xue Q., Hu Y. H., and Tompkins W. J., Neural-network- based adaptive matched filtering for QRS detection// IEEE Trans. Biomed. Eng., BME-39(4): pp. 317–329, 1992.
 105. Mahalingam N., Kumar D., Neural networks for for signal processing applications: ECG classification// Australas. Phys. Eng. Sci. Med.,vol. 20, no. 3, pp 147-151,1997.
 106. Trahanias, P.; Skordalakis, E. Syntactic pattern recognition of the ECG// IEEE Transactions on Pattern Analysis and Machine Intelligence, July 1990, vol.12, (no.7): pp 648-657.
 107. Kohler B., Hennig C., Orglmeister R., The principles of software QRS detection// IEEE Engineering in Medicine and Biology, No.1, 2002,pp. 42-57.
 108. Intel Signal Processing Library, Library Manual. – Intel Corp. – 1995.
 109. Нагин В.А., Потапов И.В., Селищев С.В. Алгоритм выделения QRS-комплексов для ЭКГ-систем реального времени // Материалы 4 международной конференции «Физика и радиоэлектроника в медицине и экологии-2000»,- – Владимир, 2000. – ч. 1, с. 201-204.
 110. Нагин В.А., Селищев С.В. Выделение QRS-комплексов в ЭКГ-системах реального времени // М: Медицинская техника. – 2001. – № 6– с. 18-23.
 111. Нагин В.А., Потапов И.В., Селищев С.В. Выделение QRS-комплексов в компьютерных ЭКГ-системах // Труды международной конференции по биомедицинскому приборостроению – «Биомедприбор-2000», - М.,2000 – ч. 1. с. 120-121.
 112. Sedaaghi M. H., ECG wave detection using morphological filters.– Applied Signal Processing, Springer-Verlag, London. – 1998. – vol. 5. – pp. 182-194.
 113. Иванов Г.Г., Потапова Н.П., Буланова Н.А. Современные неинвазивные методы оценки и прогнозирования развития потенциально опасных и угрожающих жизни аритмий: состояние проблемы и перспективы развития. Кардиология. 1997.– N2.– с. 70-75.
 114. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов: Пер. с англ. – М.: Мир, 1989.- 448 с.
 115. Cooley J., Tukey J., An algorithm for the machine computation of complex Fourier Series.– Math. Comp. 19(1065):pp. 297-301.
 116. Нагин В.А. Алгоритм выделения QRS-комплексов для ЭКГ-систем реального времени// Тезисы докладов III Международной конференции «Электроника и информатика-2000», – М., – 2000, с. 85.
 117. Sun Y., Suprappola S., Wrublewski T.A., A microcontroller-based real-time QRS detection// Biomed. Instrum. Technol., vol 26, no. 6, pp. 447-484, 1992.
 118. Нагин В.А., Селищев С.В. Реализация алгоритма выделения QRS-комплексов на микроконтроллерах семейства C166 // Труды международной конференции «Радиоэлектроника-2001» – М.,2001, с. 72-74.
 119. Горбунов Б.Б. и др. Внешний дефибрилятор-монитор с программируемой формой электрического импульса. //М.: Биомедицинские технологии и радиоэлектроника. №12-2001. – стр. 56-65.

П. 1. ОСНОВНЫЕ СВОЙСТВА КОМПОНЕНТНОЙ АРХИТЕКТУРЫ COM+

Компонентная Объектная Модель (Component Object Model) – это компонентная программная архитектура, которая позволяет строить приложения и системы из компонент, созданных разными производителями и выступает в качестве базовой платформы для сервисов высокого уровня.

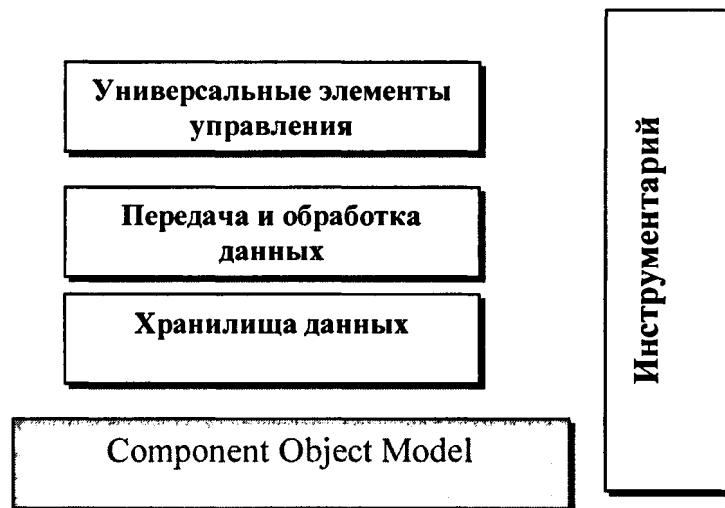


Рис. 1. Представление о СОМ.

Самый важный вопрос, решаемый этой компонентной моделью, состоит в следующем.

Как разработать систему таким образом, чтобы бинарные модули от разных производителей, разбросанных по всему миру, написанные в разное время и на разных языках программирования имели возможность активно взаимодействовать друг с другом?

Компонентная объектная модель определяет несколько основополагающих принципов, положенных в основу всей структуры СОМ:

- ◆ бинарный стандарт вызова функций между компонентами;
- ◆ обеспечение строго типизированного объединения функций в интерфейсы;
- ◆ обязательное присутствие базового интерфейса, обеспечивающего возможность доступа к другим доступным интерфейсам, а также подсчет ссылок на данный компонент, чтобы отслеживать его использование и динамически удалять, когда число ссылок становится равным нулю;
- ◆ механизм для уникальной идентификации компонент и поддерживаемых ими интерфейсов;
- ◆ загрузчик объектов, обеспечивающий их создание и настройку.

Бинарный стандарт

Для любой платформы (комбинации программного и аппаратного обеспечения) COM определяет стандартный способ размещения таблицы виртуальных функций (vtable) в памяти и стандартный способ вызова этих функций. Поэтому любой язык, поддерживающий вызовы функций через указатель (C, C++, Small Talk®, Ada, Delphi и даже Basic) может быть использован для написания универсальных компонентов, которые могут взаимодействовать с любыми другими компонентами, поддерживающим такой же бинарный стандарт.

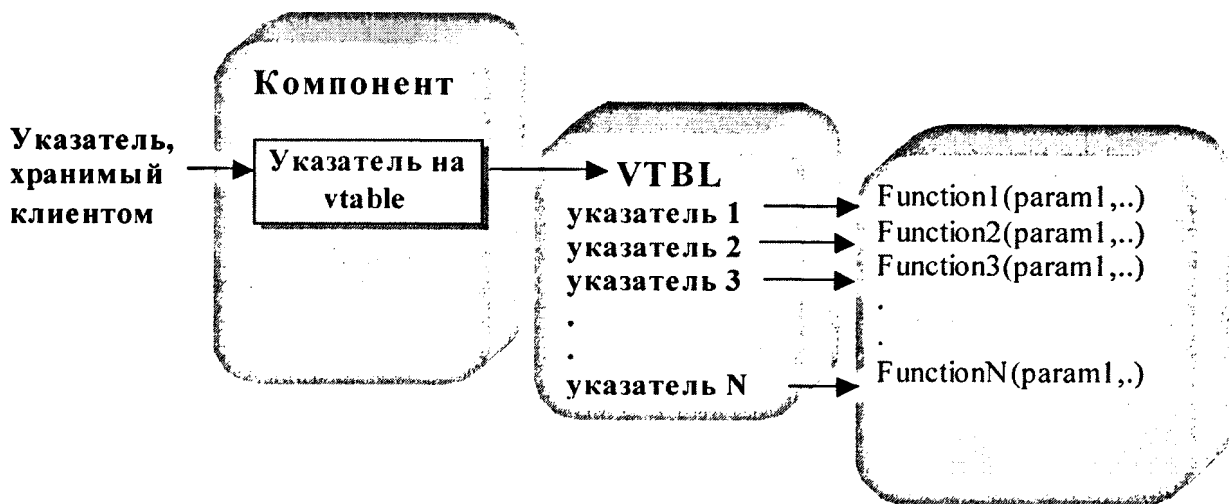


Рис. 2. Реализация механизма интерфейсов.

Использование двойной ссылки (клиент хранит указатель на указатель на vtable) дает возможность использовать один и тот же экземпляр компонента несколькими клиентами.

Объекты, в рамках этой компонентной модели, могут иметь свои внутренние данные, однако доступ к ним возможен только через интерфейс. Это является самым главным свойством архитектуры COM, фундаментальным ее требованием, дающим возможность обеспечить инкапсуляцию данных в соответствии с требованиями ООП.

Интерфейсы

В рамках COM все приложения взаимодействуют друг с другом исключительно через сгруппированные наборы функций, называемые интерфейсами. Интерфейс COM – это определенная договоренность между компонентами, заключенная в виде строго типизированного описания, несущего информацию о поведении интерфейса и ответственности компонента, предоставившего этот интерфейс.

По традиции COM, имя интерфейса начинается с “I”. Приведем пример объявления интерфейса в синтаксисе C++:

```

interface ILookup : public Iunknown
{ public:
    virtual HRESULT __stdcall LookupByName( LPTSTR lpName, TCHAR
    **lpIpNumber) = 0;
    virtual HRESULT __stdcall LookupByNumber(LPTSTR lpNumber,
    TCHAR **lpIpName) = 0;
};

```

В приведенном примере, интерфейс ILookup, имеет 2 метода, объявленных как чистые виртуальные функции. Компонент, поддерживающий интерфейс ILookup, таким образом предоставляет клиентам 2 функции: поиск по имени и поиск по значению.

Применение интерфейсов в архитектуре COM несет следующие преимущества:

1. Удобный путь модернизации уже имеющихся компонентов путем поддержки новых интерфейсов, расширяющих функциональность уже существующих. Это достигается через присутствие функции QueryInterface у каждого компонента, обрабатывающей запросы на получение указателя на любой интерфейс, поддерживаемый данным компонентом.
2. Быстрое и простое взаимодействие компонентов. Это достигается путем непосредственного вызова функции интерфейса через указатели в случае, если как компонент, так и клиент, его использующий, находятся в одном и том же процессе. В этом случае вызов функции интерфейса всего лишь на несколько инструкций процессора больше, чем вызов обычной функции C++.
3. Повторное использование интерфейсов. Как показывает практика, часто одни и те же операции приходится реализовывать по-разному в каждом конкретном случае. COM позволяет одному и тому же интерфейсу быть реализованным несколькими компонентами по-разному.
4. Прозрачность доступа к компоненту, вне зависимости от того, локально или удаленно он выполняется. Бинарный стандарт позволяет перехватывать вызовы функций, сделанных локально и преобразовывать их в удаленные вызовы.
5. Независимость от используемого языка программирования. Любой язык программирования, имеющий возможность обращаться с указателями и явно или неявно вызывать функции через указатель, может быть использован в качестве языка разработки COM.

Глобально уникальные идентификаторы

COM использует глобально уникальные идентификаторы – 128 разрядные целые числа, которые гарантируют свою уникальность в рамках нашего

пространства физических координат и времен и применяются для идентификации интерфейсов и COM компонентов. Читательские имена, присваиваемые компонентам и интерфейсам, не гарантируют уникальность и используются лишь для удобства разработки.

Точки соединения объектов архитектуры COM

Одним из основных механизмов архитектуры COM является механизм точек соединения, позволяющий объектам устанавливать взаимные связи друг с другом. Так называемая точка соединения представляет собой интерфейс, предоставляемый объектом-сервером объектам-клиентам для установления связи с реализацией этого интерфейса в объектах-клиентах, с которыми объект-сервер устанавливает взаимодействие.

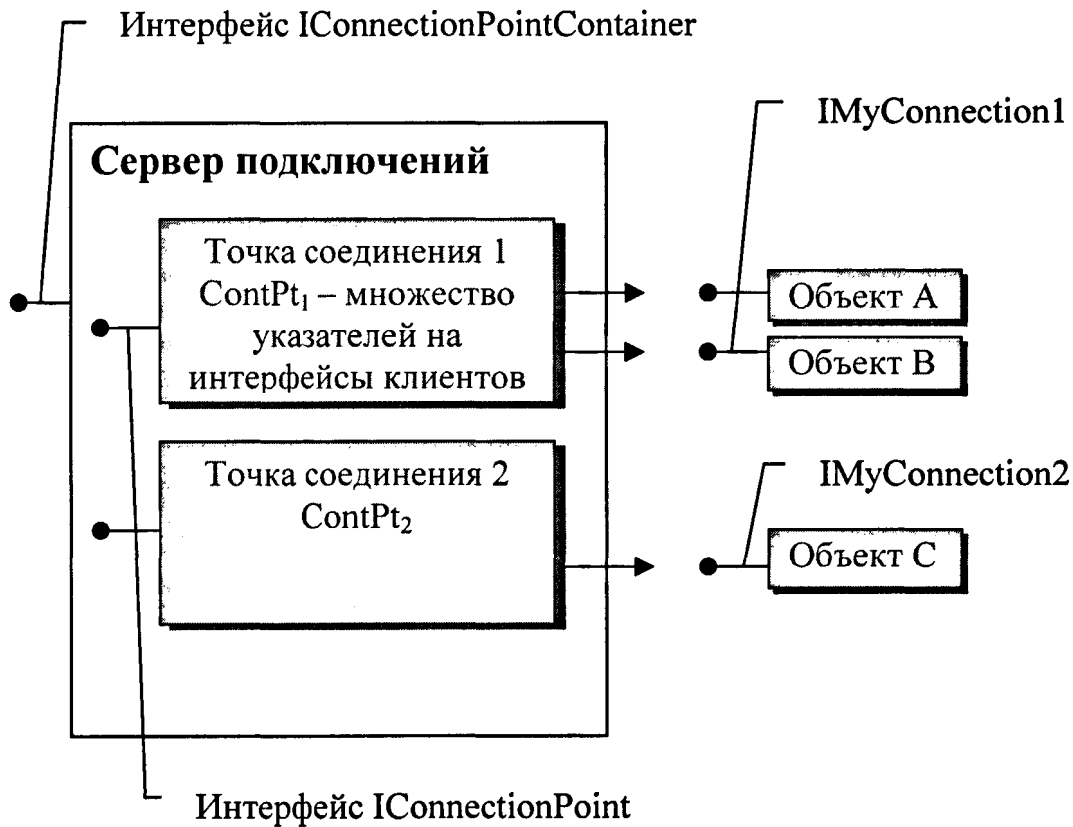


Рис. 3. Механизм точек соединения.

Сам интерфейс не реализуется сервером, а лишь предоставляет точку соединения, через которую объекты-клиенты могут подключить свои реализации этого интерфейса. В этом случае объекту-серверу соединения не нужно ничего знать о внутренней структуре клиентов, более того, ему даже не нужно ничего знать об их существовании, так как процедура подключения инициируется клиентом.

Интерфейс IUnknown

COM определяет один специальный интерфейс IUnknown, который должен поддерживаться любым COM объектом; любой COM интерфейс должен наследоваться от IUnknown.

В синтаксисе C++, IUnknown выглядит следующим образом:

```
interface IUnknown {  
    virtual HRESULT QueryInterface(IID& iid, void** ppvObj) = 0;  
    virtual ULONG AddRef() = 0;  
    virtual ULONG Release() = 0;  
}
```

Методы **AddRef** и **Release** являются простыми счетчиками ссылок. Метод **AddRef** вызывается, когда другой компонент получает указатель на интерфейс, **Release** – когда этот указатель освобождается. До тех пор, пока счетчик ссылок не обратился в нуль, объект должен оставаться в памяти. Как только счетчик достиг нуля, объект может безопасно выгружаться из памяти.

QueryInterface является механизмом, позволяющим компонентам динамически (на этапе выполнения) определять поддерживается тот или иной интерфейс этим компонентом или нет. Если этот интерфейс поддерживается, клиенты могут через этот метод запросить указатель на него.

Итак, компонентная архитектура COM берет на себя решение следующих проблем:

- ◆ обеспечение высокопроизводительного взаимодействия компонентов;
- ◆ отслеживание версий компонентов;
- ◆ независимость от языка программирования;
- ◆ прозрачность процедуры взаимодействия компонентов, принадлежащих разным процессам.

Самым важным и наиболее сложным является последний пункт, поскольку он требует весьма важного допущения – компоненты, разрабатываемые для использования в рамках адресного пространства, одного процесса без каких-либо изменений могут взаимодействовать с другими компонентами через границы процессов. Подобное взаимодействие гарантируется системной библиотекой поддержки COM, которая принимает на себя обязанности, связанные с поиском, загрузкой и управлением выполнением запущенных компонентных объектов. Фактически это означает, что любой компонент может свободно взаимодействовать с другими компонентами, исполняемыми в другом процессе, или даже удаленно на другой машине.

Современная реализация COM+ имеет множество дополнительных функций, которые остались за рамками данного обзора.

П. 2. ОБЗОР ХАРАКТЕРИСТИК НАИБОЛЕЕ ЗНАЧИМЫХ МЕТОДОВ СЖАТИЯ ЭКГ

Все основные методы сжатия данных могут быть разделены на 2 группы: сжатие с потерями данных и без потерь. Под сжатием с потерями подразумевается то, что исходный сигнал может быть восстановлен только с заведомо заданной степенью точности.

Методы сжатия без потерь используют стандартные методы кодирования избыточности, применяемые для сжатия произвольных данных, такие как например, DPCM (Differential Pulse Code Modulation), метод энтропийного кодирования вторичных разностей. Эти методики обычно используют особенности ЭКГ сигнала: периодичность, наличие участков практически постоянного потенциала. Чтобы извлечь из этого пользу, сигнал зачастую дифференцируется, генерируется усредненный шаблон периодического участка (обычно QRS комплекс), который вычитается из сигнала на каждом периоде.

Другим подходом является применение какого-либо нелинейного дискретного преобразования сигнала. В настоящее время в качестве дискретных преобразований широко применяются дискретное преобразование Фурье (ДПФ), дискретное косинусное преобразование, преобразование Карунена-Лоева, а также вейвлет-преобразование. Эти методы формируют группу методик сжатия ЭКГ с потерей информации и строятся обычно по стандартной схеме. После применения дискретного преобразования сигнала, полученные коэффициенты дополнительно квантуются и далее применяются стандартная процедура сжатия квантованных коэффициентов без потерь. Следует заметить, что единственный шаг кодирования, на котором осуществляется потеря данных, - это процедура квантования или отбрасывания части коэффициентов.

Эффективность компрессии определяется коэффициентом сжатия, который обычно тем больше, чем выше величина потерь данных, и чем больше время, затраченное на кодирование информации. Величину, характеризующую потерю данных, обычно выражают в терминах среднеквадратичного отклонения, выраженного в процентах:

$$CKO = \sqrt{\frac{\sum_{k=1}^n (y(k) - \hat{y}(k))^2}{\sum_{k=1}^n y(k)^2}} * 100$$

Все это делается для того, чтобы впоследствии как можно эффективнее применить операцию кодирования без потери информации. Основные

характеристики наиболее известных методов на основании результатов, представленных в различных источниках, содержатся в Табл. 1.

Табл. 1.
Обзор эффективности методов сжатия ЭКГ.

Метод	Коэффициент сжатия	Частота дискретизации	СКО
AZTEC	10	500	28.0
TP	2.0	200	5.3
CORTES	4.8	200	7.0
FAN/SAPA	3.0	250	4.0
Энтропийное кодирование 2-й производной	2.8	250	-
Слайн интерполяция и сжатие энтропии (Хаффмена)	10.0	500	14.0
DMSP	4.0	300	-
DPCM с линейным предсказыванием	2.5	250	-
DPCM, линейное предсказывание, интерполяция, энтропийное сжатие	7.8	500	3.5
Ортогональные преобразования (преобразование Фурье, Карунена-Лоева, Хартли)	3.0	250	-
Ортогональное вейвлет преобразование			
Двойное применение Преобразования Карунена-Лоева	12	250	-

П. 3. ИМИТАЦИОННАЯ МОДЕЛЬ QRS-ДЕТЕКТОРА

Реализация простейшего QRS-детектора в виде S-функции для моделирования в среде SIMULINK представлена ниже:

```
/*=====*/
* S-function methods *
*=====*/
/* Function: mdlInitializeSizes =====
* Abstract:
* The sizes information is used by Simulink to determine the S-function
* block's characteristics (number of inputs, outputs, states, etc.).
*/
static void mdlInitializeSizes(SimStruct *S)
{
    /* See sfuntmpl.doc for more details on the macros below */
    ssSetNumSFcnParams(S, 0); /* Number of expected parameters = 0 in this version*/
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        /* Return if number of expected != number of actual parameters */
        return;
    }
    ssSetNumContStates(S, 0);
    ssSetNumDiscStates(S, 0);
    if (!ssSetNumInputPorts(S, 1)) return;

// Customize the only input
    ssSetInputPortWidth( S, 0, DYNAMICALLY_SIZED );
    ssSetInputPortDataType( S, 0, DYNAMICALLY_TYPED );
    ssSetInputPortDirectFeedThrough( S, 0, 1 );

    /* outputs */
    if ( !ssSetNumOutputPorts( S, 1 ) ) return;

    /* output 0 is dynamically sized and typed. */
    ssSetOutputPortWidth( S, 0, DYNAMICALLY_SIZED );
    ssSetOutputPortDataType( S, 0, DYNAMICALLY_TYPED );

    /* sample times */
    ssSetNumSampleTimes(S, 1 );

    /* options */
    ssSetOptions( S, SS_OPTION_EXCEPTION_FREE_CODE );

    ssSetNumSampleTimes(S, 1);
    ssSetNumRWork(S, 0);
    ssSetNumIWork(S, 0);
    ssSetNumPWork(S, 1); // reserve element in the pointers vector
    ssSetNumModes(S, 0); // to store a C++ object
    ssSetNumNonsampledZCs(S, 0);

    //InitQRSData();
}
}
```



```

/* Function: mdlInitializeSampleTimes =====
* Abstract:
* This function is used to specify the sample time(s) for your
* S-function. You must register the same number of sample times as
* specified in ssSetNumSampleTimes.
*/

static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime( S, 0, INHERITED_SAMPLE_TIME );
    ssSetOffsetTime( S, 0, FIXED_IN_MINOR_STEP_OFFSET );
}

#define MDL_START /* Change to #undef to remove function */
#if defined(MDL_START)
/* Function: mdlStart =====
* Abstract:
* This function is called once at start of model execution. If you
* have states that should be initialized once, this is the place
* to do it.
*/
static void mdlStart(SimStruct *S)
{
    ssGetPWork(S)[0] = (void *) new counter; // store new C++ object in the
                                           // pointers vector
}

#endif /* MDL_START */
/* =====
/
/ QRS - detection algorithm
/
=====*/

#define INT long
#define ISOLINE_CORRECTION //Дополнительно включена коррекция изолинии
#define WINDOW_SIZE 80 //!!!Для 500 Гц. Рекомендуемый размер окна 150мс.
#define SIGNAL_DELAY 40 //Задержка сигнала половина окна интегрирования

//Глобальные величины
INT Counter; // Счетчик отсчетов
INT LastSample; // Последний отсчет
INT DelayLine[WINDOW_SIZE]; // Линия задержки
INT SignalDelLine[SIGNAL_DELAY];

INT PEAKI, // current peak of (I)ntegration window
    SPKI, // running estimate of signal peak (I)
    NPKI; // running estimate of noise peak (I)

INT Rpos, // Позиция интервала в потоке отсчетов
    RR, // Текущее значение RR-интервала
    RA,
    RRmean1; // Среднее значение за 10 последних отсчетов
INT IntValue; // Текущее значение интеграла
INT IntNormValue; // Нормированное значение интеграла
BOOL bQRSStarted; // Начало QRS

/* Function: InitQRSData =====
* Abstract:
*

```

```

*      Начальная инициализация данных детектора
*/
void InitQRSData()
{
    Counter = 0;
    LastSample = 0;      // Предыдущий отсчет
    IntValue =
    IntNormValue = 0;
    PEAKI = 0;          // current peak of (I)ntegration window
    SPKI = 0;           // running estimate of signal peak (I)
    NPKE = 0;           // running estimate of noise peak (I)
    Rpos = 0;           // Позиция интервала в потоке отсчетов
    RR = 0;             // Текущее значение RR-интервала
    RA = 0;
    RRmean1 = 0;        // Среднее значение за 10 последних отсчетов

    memset(DelayLine,0,sizeof(DelayLine));
    memset(SignalDelLine,0,sizeof(SignalDelLine));

    IntValue = 0;
    bQRSStarted = FALSE;
    QRSAmplitude = 0;   //Скользкая амплитуда QRS-комплекса
}

/* Function: mdlOutputs =====
* Abstract:
* In this function, you compute the outputs of your S-function
* block. Generally outputs are placed in the output vector, ssGetY(S).
*/
static void mdlOutputs(SimStruct *S, int_T tid)
{
    /*
    * Get "uPtrs" for input port 0 and 1.
    * uPtrs is essentially a vector of pointers because the input signal may
    * not be contiguous.
    */

    InputPtrsType u0Ptrs = ssGetInputPortSignalPtrs(S,0); // Получить указатель на буфер входных данных

    /*
    * Get data type Identifier for output port 0. Note, this matches the data
    * type ID for input port 0 and 1.
    */
    DTypeId y0DataType = ssGetOutputPortDataType(S, 0); // Получить входных данных. Будет double
    int_T y0Width = ssGetOutputPortWidth(S, 0); // Получить ширину (количество отсчетов) во входных
данных. Будет равно 1.
    switch (y0DataType)
    {
        case SS_DOUBLE:
        {
            //Данные поступают как double.
            //Это сделано для простоты отображения
            //Следовательно первым шагом следует преобразование к 32 битам
            //Преобразование к 16 битам может переводить к переполнениям и этот вопрос следует прорабатывать
            //отдельно
            real_T *pY0 = (double *)ssGetOutputPortSignal(S,0); // Получить указатель на поток выходных данных

            InputRealPtrsType pU0 = (InputRealPtrsType)u0Ptrs; // Получить указатель на поток входных данных

```

```

#ifdef ISOLINE_CORRECTION
    //Коррекция изолинии скользящим средним
#endif
for(long i = 0; i < y0Width; ++i)
    {
    //pY0[i] = -*pU0[i];
        INT NewSample = (*pU0)[i]; //Новый отсчет
        INT Diff = NewSample - LastSample; //Дифференцирование
        Diff = Diff*Diff; //Возведение в квадрат
        LastSample = NewSample; //Сохраняем значение отсчета
        SignalDelLine[Counter%SIGNAL_DELAY] = NewSample;
    if (Counter<WINDOW_SIZE)
        {
        IntValue+=Diff;
            IntNormValue = (Counter == WINDOW_SIZE-1)?IntValue/WINDOW_SIZE:0;
        }
        else
        {
        IntValue-=DelayLine[Counter % WINDOW_SIZE]; //Вычитаем выходящий из интеграла последний
отсчет
            IntValue+=Diff;
        IntNormValue=IntValue/WINDOW_SIZE;
        }

        DelayLine[Counter % WINDOW_SIZE] = Diff; //Сохраняем в линии задержки

        INT CurrentSample = 0;
        if (Counter<SIGNAL_DELAY)
        {
        Counter++; //Накапливаем задержку сигнала
        break;
        }
        else
            CurrentSample =
SignalDelLine[(Counter+1)%SIGNAL_DELAY]; //Сигнал с запаздыванием на половину окна интегрирования.
//Рассчитываем адаптивные пороги
        PEAKI = IntNormValue; // IntNormValue - после интеграции
        INT TII = NPKI + 1*(SPKI - NPKI)/4;
        if (PEAKI >= TII)
            SPKI = (1*PEAKI)/8 + (7*SPKI)/8;
        else
            NPKI = (1*PEAKI)/8 + (7*NPKI)/8;
        pY0[i] = PEAKI;

        Counter++;

        //Проверка на переполнение
        if (PEAKI > TII && !bQRSStarted)
            {
            QRSSstart = Counter-SIGNAL_DELAY; //QRS начинается со сдвигом
            bQRSStarted = TRUE;
            QRSAmpitude = SignalDelLine[(Counter+1)%SIGNAL_DELAY]
            }
        // Начало QRS
        .....
        //
        }
        break;
    } } // object

```

П. 4. IDL-СПЕЦИФИКАЦИИ ОСНОВНЫХ ИНТЕРФЕЙСОВ ПРОГРАММНОЙ АРХИТЕКТУРЫ

Наиболее важные интерфейсы объектно-ориентированных ИПП программной архитектуры приведены в следующих IDL-спецификациях.

Интерфейс доступа к контейнеру данных:

```
interface ISample : IDispatch
{
    [propget,readonly,id(1), helpstring("Number of Channels")] HRESULT Channels([out, retval] long *pVal);
    [propget,readonly,id(2), helpstring("Number of Slices")] HRESULT Slices([out, retval] long *pVal);
    [propget,id(3), helpstring("XValue")] HRESULT X([out, retval] float *pVal);
    [propput,id(3), helpstring("XValue")] HRESULT X([in] float pVal);
    [propget,id(4), helpstring("YValues")] HRESULT Y([in] long Channel,[out, retval] float *pVal);
    [propput,id(4), helpstring("YValues")] HRESULT Y([in] long Channel,[in] float pVal);
    [propget,id(5), helpstring("Position")] HRESULT Position([out,retval] long* Position);
    [propput,id(5), helpstring("Position")] HRESULT Position([in] long Position);
    [propget,readonly,id(6),helpstring("XBegin: bounds")] HRESULT XBegin([out, retval] float *pVal);
    [propget,readonly,id(7),helpstring("XEnd: bounds")] HRESULT XEnd([out, retval] float *pVal);
    [propget,readonly,id(8),helpstring("Data")] HRESULT Data([out, retval] IUnknown **pVal);
    [id(9),helpstring("Seek")] HRESULT Seek([in] float X);
};
```

```
interface ISampleFactory : IDispatch
{
    [id(1), helpstring("Returns GUID of creatable objects")] HRESULT _stdcall GetGUID([out,retval] VARIANT* pGUID);
    [id(2), helpstring("Create a new sample")] HRESULT _stdcall CreateInstance([in] VARIANT* varParam,ISample **pSample);
    [id(3), helpstring("Duplicate a sample")] HRESULT _stdcall DuplicateInstance([in] ISample *pInSample ,[out] ISample **pOutSample);
};
```

Общие интерфейсы управления подсистемой устройств регистрации электрофизиологических сигналов.

```
interface IDeviceStatus : IDispatch
{
    [propget,readonly,id(1), helpstring("Number of Channels")] HRESULT Channels([out, retval] long *pVal);
    [propget,readonly,id(2), helpstring("Number of Slices")] HRESULT Slices([out, retval] long *pVal);
    [propget,id(3), helpstring("XValue")] HRESULT X([out, retval] float *pVal);
    [propget,id(4), helpstring("Status")] HRESULT Status([in] long Channel,[out, retval] short int *pVal);
    [propget,id(5), helpstring("Position")] HRESULT Position([out,retval] long* Position);
    [propput,id(5), helpstring("Position")] HRESULT Position([in] long Position);
    [propget,readonly,id(8),helpstring("Data")] HRESULT Data([out, retval] IUnknown **pVal);
};

}
interface IEnumDevices : IUnknown {
    HRESULT Next ( [in] ULONG cDevices, [out] IDevice ** ppDevice,[out] ULONG * pcFetched);
```

```

HRESULT Skip( [in] ULONG cDevices);
HRESULT Reset(void);
HRESULT Clone( [out] IEnumDevices **ppEnum);
}

```

```

interface IDeviceManager : IDispatch

```

```

{
    [id(DISPID_NEWENUM)] HRESULT _NewEnum([out, retval] IEnumDevices* *pVal);
    [id(1),helpstring("Looks for devices of the type")]
    HRESULT GetDevicesOfType([in] DeviceType Type,[out, retval] IEnumDevices* *pVal);
    [id(2),helpstring("Looks for a device having provided name")]
    HRESULT GetDevice([in] BSTR Name,[out, retval] IDevice* *pVal);
};

```

```

interface IDeviceInfo : IDispatch

```

```

{
    [propget, id(0), helpstring("property Name")]           HRESULT Name([out, retval] BSTR *pVal);
    [propget, id(1), helpstring("property DeviceID")]      HRESULT ID([out, retval] BSTR *pVal);
    [propget, id(2), helpstring("property DeviceClass")]   HRESULT Class([out, retval] BSTR *pVal);
    [propget, id(3), helpstring("property Description")]    HRESULT Description([out, retval] BSTR *pVal);
    [propget, id(4), helpstring("property Driver")]        HRESULT Driver([out, retval] BSTR *pVal);
    [propget, id(5), helpstring("property DriverDescription")] HRESULT DriverDescription([out, retval] BSTR
    *pVal);
    [propget, id(6), helpstring("property Type")]          HRESULT Type([out, retval] DeviceType *pVal);
    [propget, id(7), helpstring("property Version")]       HRESULT Version([out, retval] BSTR *pVal);
};

```

```

interface IDevice: IUnknown

```

```

{
    [helpstring("Opens Device")] HRESULT Open([in,out] VARIANT* varParam);
    [helpstring("Closes Device")] HRESULT Close();
    [helpstring("Starts device")] HRESULT Start([in,out] VARIANT* varParam);
    [helpstring("Stops device")] HRESULT Stop();
    [helpstring("Brings Up device property box")] HRESULT Setup();
    [propget, helpstring("property State")] HRESULT State([out, retval] DeviceState *pVal);
};

```

```

interface IDeviceError: IUnknown

```

```

{
    [helpstring("method RaiseError")] HRESULT RaiseError([in] long nCode, [in] VARIANT data);
};

```

Интерфейсы управления точками соединения и интерфейсом передачи БМИ.

```

interface ITransport : IDispatch

```

```

{
    [id(1),helpstring("Writes measured values")] HRESULT _stdcall Write([in] ISample* pVal);
    [id(2),helpstring("Invalidates a signal")] HRESULT _stdcall Invalidate([in] VARIANT varParam);
    [id(3),helpstring("Validates a signal")] HRESULT _stdcall Validate([in] IUnknown* pSamplesFactory);
};

```

```

interface IStatusTransport : IDispatch

```

```

{
    [id(1),helpstring("Write status values")] HRESULT _stdcall StatusWrite([in] IDeviceStatus* pVal);
};

```

```

[id(2),helpstring("Invalidates a signal")] HRESULT STDMETHODCALLTYPE StatusInvalidate([in] VARIANT varParam);
[id(3),helpstring("Validates a signal")] HRESULT STDMETHODCALLTYPE StatusValidate([in] VARIANT varParam);
};

```

```

interface IPin : IDispatch

```

```

{
    [id(1), helpstring("Name of the connection")] HRESULT STDMETHODCALLTYPE Name([out, retval] BSTR *pVal);
    [id(2), helpstring("Number of available Channels")] HRESULT STDMETHODCALLTYPE Channels([out, retval] long *pVal);
    [id(3), helpstring("X Name")] HRESULT STDMETHODCALLTYPE XName([out, retval] BSTR *pVal);
    [id(4), helpstring("X Unit")] HRESULT STDMETHODCALLTYPE XUnit([out, retval] BSTR *pVal);
    [id(5), helpstring("Y Name")] HRESULT STDMETHODCALLTYPE YName([in] long Channel, [out, retval] BSTR *pVal);
    [id(6), helpstring("Y Unit")] HRESULT STDMETHODCALLTYPE YUnit([in] long Channel, [out, retval] BSTR *pVal);
    [id(7), helpstring("Sampling period")] HRESULT STDMETHODCALLTYPE XInterval([out, retval] float* pVal);
    [id(8), helpstring("Low Y limit")] HRESULT STDMETHODCALLTYPE YBegin([out, retval] float *pVal);
    [id(9), helpstring("High Y limit")] HRESULT STDMETHODCALLTYPE YEnd([out, retval] float* newVal);
    [id(10),helpstring("Is connection accepted?")] HRESULT STDMETHODCALLTYPE Accept([in] VARIANT
newVal,[out,retval] VARIANT_BOOL* bAccepted);
};

```

Интерфейсы, реализация которых необходима в каждом подключаемом модуле:

```

interface IActivePlugin : IDispatch

```

```

{
    [id(1), helpstring("method OnInitialDisplay")]
    HRESULT OnInitialDisplay(long *pHrResult);
    [id(2), helpstring("method OnReactivateDisplay")]
    HRESULT OnReactivateDisplay([in,out] long *pHrResult);
    [id(3), helpstring("method OnDeactivateDisplay")]
    HRESULT OnDeactivateDisplay([in,out] long *pHrResult);
    [id(4), helpstring("method OnClose")]
    HRESULT OnClose([in,out] long *pHrResult);
    [id(5), helpstring("method OkToClose")]
    HRESULT OkToClose([in,out] VARIANT_BOOL *pbCanClose,
[in,out] BSTR *pbMsg,
[in,out] long *pHrResult);
[id(6), helpstring("method CallTopDlgBar")]
    HRESULT CallTopDlgBar([in]short nWhichBtn,
[in]VARIANT vData,
[in,out] long *pHrResult);
[id(7), helpstring("method GetCtrlDefinedData")]
    HRESULT GetCtrlDefinedData([in]int nWhichData,
[in]int nIndex,
[in,out]int *pnOptionalDataSize,
[in,out]VARIANT* pvData,
[in,out] long *pHrResult);
[id(8), helpstring("method SetCtrlDefinedData")]
    HRESULT SetCtrlDefinedData([in]int nWhichData,
[in]int nIndex,
[in,out]int *pnOptionalDataSize,
[in]VARIANT vData,
[in,out] long *pHrResult);
[id(9), helpstring("method On Connection")]
    HRESULT OnConnection(
[in] IDispatch* pApp,
[in] IDispatch* pDocument,
[in] VARIANT_BOOL bFirstTime,
[in] long dwCookie,

```

```

        [out, retval] VARIANT_BOOL* OnConnection);
        [id(10), helpstring("method On Connection")]
HRESULT OnDisconnection([in] VARIANT_BOOL bLastTime);
        [propget, id(11), helpstring("property ControlID")]
        HRESULT ControlID([out, retval] short *pVal);
        [propput, id(11), helpstring("property ControlID")]
        HRESULT ControlID([in] short newVal);
        [id(12), helpstring("GetControlDefaults Settings")]
        HRESULT GetControlDefaults([out] long* hModule,[out] long* hResource,[out] VARIANT*
MoreData);
    };

```

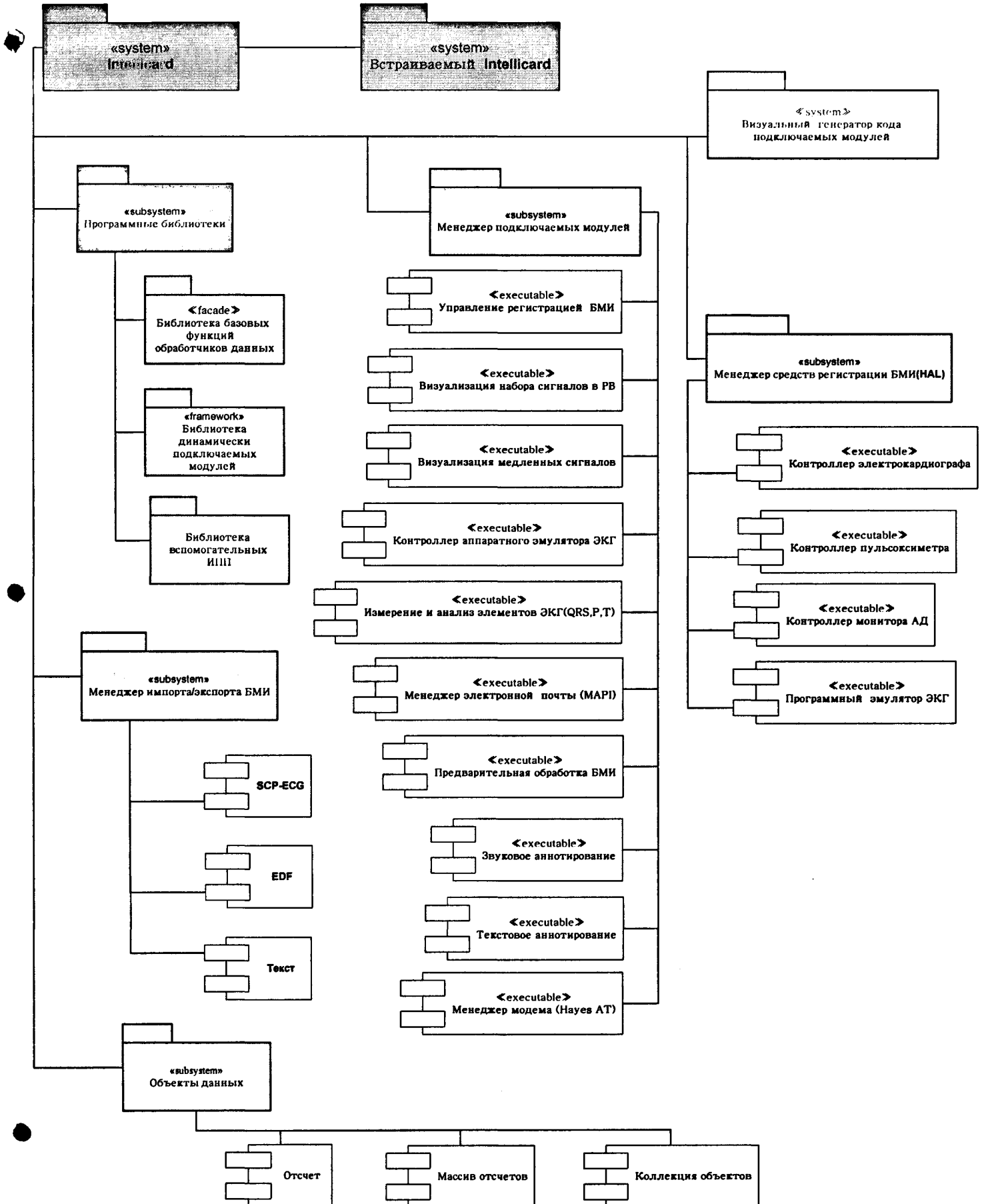
interface IPluginAbout : IUnknown

```

{
    [helpstring("Text for the snap-in description box")]
    HRESULT GetPluginDescription([ out] LPOLESTR* lpDescription);
    [helpstring("Provider name")]
    HRESULT GetProvider([out] LPOLESTR* lpName);
    [helpstring("Version number for the snap-in")]
    HRESULT GetPluginVersion([out] LPOLESTR* lpVersion);
    [helpstring("Main icon for about box")]
    HRESULT GetPluginImage([out] HICON* hAppIcon);
    [helpstring("Static folder images for scope and result panes")]
    HRESULT GetStaticFolderImage([out] HBITMAP* hSmallImage,
        [out] HBITMAP* hSmallImageOpen,
        [out] HBITMAP* hLargeImage,
        [out] COLORREF* cMask);
};

```

ПОКОМПОНЕНТНАЯ СТРУКТУРА РАЗРАБОТАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ





АКТ


Внедрения результатов диссертационной работы В.А. Нагина «Распределенная компьютерная система сбора и математической обработки электрофизиологических сигналов», представленной на соискание ученой степени кандидата технических наук по специальности 05.13.01 – Системный анализ, управление и обработка информации.

Результаты диссертационной работы были использованы при подготовке и реализации следующих проектов кафедры биомедицинских систем МИЭТ.

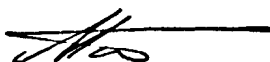
- **Научно-исследовательские работы по тематическому плану вуза:**
 - «Моделирование биомедицинских электронных систем на базе фундаментальных принципов биоинформатики, твердотельной электроники, микроэлектроники и нанозлектроники», 2002-2004 гг.
 - **Научно-технические программы Министерства образования Российской Федерации:**
 - «Внешний дефибриллятор-монитор нового поколения с программируемой формой электрического импульса», 2001-2002 гг.
 - « Системы сбора и обработки биомедицинских сигналов на основе сигма-дельта АЦП», 2001-2002 гг.
 - **Договорные научно-исследовательские работы:**
 - «Разработка и изготовление экспериментальных образцов блоков управления и индикатора для дефибриллятора», 2001-2002 гг.
 - «Разработка и изготовление экспериментального образца силового блока», 2001-2002 гг.
 - «Разработка наземного программного обеспечения (ПО) для обработки и анализа физиологических данных, получаемых в ходе проведения эксперимента ПУЛЬС», 2001 г.
 - «Разработка алгоритмов и ТЗ к программному обеспечению для автоматического анализа данных, получаемых с помощью комплекса СОНОКАРД», 2002 г.
 - Разработка технического предложения на создание «Системы оперативного медицинского контроля, сбора научной информации, прогноза и диагностики состояния человека-оператора в условиях моделирования динамических факторов авиационного полета на базе центрифуги», 2002 г.
 - **Международное сотрудничество:**
 - Разработка макетного образца носимого внешнего дефибриллятора-монитора», 2002-2003 гг.

Результаты диссертационной работы используются в учебном процессе кафедры биомедицинских систем МИЭТ, дисциплина «Системы медицинской диагностики».

Зав.каф. БМС

 д.ф.-м.н., проф. С.В.Селищев

Ученый секретарь каф.БМС

 к.ф.-м.н., доц. Ю.П. Маслбоев