

С.К. Ганиев, Т.А. Кучкаров

# ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ И СЕТИ



38.9  
719

МИНИСТЕРСТВО ВЫСШЕГО  
И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ  
РЕСПУБЛИКИ УЗБЕКИСТАН

ЦЕНТР СРЕДНЕГО СПЕЦИАЛЬНОГО  
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

С.К. Ганиев, Т.А. Кучкаров

# ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ И СЕТИ

Учебное пособие  
для профессионально-технических колледжей

Издательско-полиграфический творческий дом  
имени Гафура Гуляма  
Ташкент - 2008



32.973  
Г19

Рекомендовано к изданию Советом по координации  
деятельности учебно-методических объединений высшего  
и среднего специального профессионально-технического  
образования

Рецензенты:

**А. Каххаров** – доцент кафедры «Прикладная информатика»  
Ташкентского университета информационных технологий,

**Г.А. Собирова** – зам.директора по учебной работе Мирзо-  
Улугбекского профессионального колледжа информатики.

Ганиев С.К.

Электронные вычислительные машины и сети: Учеб. пособие  
для проф.-техн. колледжей/ С.К. Ганиев, Т.А. Кучкаров; МВ и ССО  
РУз, Центр сред. спец. проф. образования. – Т.: Издательско-  
полиграфический творческий дом им. Гафура Гуляма, 2008. – 140 с.

И. Кучкаров Т.А.

Настоящее учебное пособие посвящено вопросам организации современных  
электронных вычислительных машин – компьютеров, а также реализованных на их  
основе компьютерных сетей. При этом основное внимание уделено изложению  
вопросов архитектуры микропроцессоров, модулей памяти, организации ввода-вывода  
и прерываний, внешних устройств; общим принципам построения вычислительных  
сетей и аппаратно-программного обеспечения. В учебном пособии также рассмотрены  
вопросы эксплуатации компьютеров и компьютерных сетей.

Учебное пособие предназначено для учащихся профессиональных колледжей по  
направлению «Информатика и информационные технологии».

ББК 32.973я72

Г-К 2302020200 – 107  
M 352(04) – 2008 Тв. заказ – 2008

© С.К. Ганиев, Т.А. Кучкаров.  
Издательско-полиграфический  
творческий дом им.Г.Гуляма.  
2008 г.

ISBN 978-9943-03-118-0

## ПРЕДИСЛОВИЕ

Электронные вычислительные машины, или компьютеры являются преобразователями информации. В соответствии с используемой формой представления информации машины делятся на два класса: непрерывного действия — аналоговые и дискретного действия — цифровые.

В силу универсальности цифровой формы представления информации цифровые электронные вычислительные машины представляют собой наиболее универсальный тип устройства обработки информации. Именно эти машины, в дальнейшем называемые сокращенно ЭВМ, а также реализованные на их основе вычислительные сети, составляют предмет изучения в настоящей книге.

Замечательные свойства ЭВМ — автоматизация вычислительного процесса на основе программного управления, огромная скорость выполнения арифметических и логических операций, возможность хранения большого количества различных данных делают эти машины мощным средством научно-технического прогресса.

В развитии вычислительной техники можно указать на несколько поколений ЭВМ. Появление новых поколений ЭВМ вызывалось расширением областей и развития методов их применения, требовавших более производительных, более дешевых и более надежных машин.

**Первое поколение** образовали ламповые ЭВМ, в которых в качестве компонентов логических элементов использовались электронные лампы. Ламповые ЭВМ потребляли много энергии, имели большие габариты, малую емкость оперативной памяти и, что особенно важно, невысокую надежность, в первую очередь из-за частого выхода из строя электронных ламп.

В вычислительных машинах **второго поколения** полупроводниковые приборы — транзисторы заменили электронные лампы, что существенно повысило надежность, снизило потребляемую мощность, уменьшило габариты ЭВМ. Большим достижением явилось применение печатного монтажа. Это позволило создать ЭВМ, обладающие большими логическими возможностями и более высокой производительностью.

Наряду с техническим совершенствованием ЭВМ развивались методы и приемы программирования вычислений. С появлением машин второго поколения, особенно многопрограммных машин, значительно расширилась сфера применения электронной вычислительной техники.

**Третье поколение** ЭВМ характеризуется широким применением интегральных схем, заменивших большинство транзисторов и различных деталей. Благодаря интегральным схемам удалось существенно улучшить технические и эксплуатационные характеристики машин. Этому способствовало также применение многослойного печатного монтажа.

Программное обеспечение машин третьего поколения получило дальнейшее развитие, особенно это касается операционных систем. Развитие операционных систем многопрограммных машин, снабженных периферийными устройствами ввода-вывода с автономными пунктами абонентов, обеспечивали управление работой ЭВМ в различных режимах.

Конструктивно-технологической основой ЭВМ **четвертого поколения** являются интегральные микросхемы с большой (БИС) и сверхбольшой (СБИС) степенями интеграции, содержащие тысячи, десятки и сотни тысяч транзисторов на одном кристалле. К этому поколению ЭВМ относятся реализованные на СБИС такие новые средства вычислительной техники, как микропроцессоры и создаваемые на их основе микроЭВМ.

Микропроцессоры и микроЭВМ нашли широкое применение в устройствах и системах автоматизации измерений, обработки данных и управления технологическими процессами, при построении различных специализированных цифровых устройств и машин.

Вычислительные возможности микро-ЭВМ оказались достаточными для создания на их основе нового по ряду эксплуатационных характеристик и способу использования типа вычислительных устройств – *персональных ЭВМ (персональных компьютеров)*, получивших в настоящее время широкое распространение.

В рамках этого поколения создаются многопроцессорные вычислительные системы с быстродействием в несколько сотен миллионов и более операций в секунду и многопроцессорные управляющие комплексы повышенной надежности.

В последнее время определились контуры нового **пятого поколения** ЭВМ. ЭВМ и вычислительные системы пятого поколения, помимо более высокой производительности и надежности при более низкой стоимости, должны обладать следующими качественными новыми свойствами: возможностью взаимодействия с ЭВМ при помощи естественного языка, человеческой речи и графических

изображений; способностью системы обучаться, производить ассоциативную обработку информации, делать логические суждения, вести «разумную» беседу с человеком в форме вопросов и ответов; способностью системы «понимать» содержимое базы данных, которая при этом превращается в базу «знаний», и использовать эти «знания» при решении задач.

Шло время, потребности пользователей вычислительной техники росли, им стало недостаточно собственных компьютеров, им уже хотелось получить возможность обмена данными с территориально удаленными компьютерами. В ответ на эту потребность предприятия и организации стали соединять свои компьютеры вместе и разрабатывать программное обеспечение, необходимое для их взаимодействия. В результате появились *вычислительные сети*.

По территориальному признаку различают глобальные (WAN – Wide Area Network), локальные (LAN – Local Area Network) и городские (MAN – Metropolitan Area Network) вычислительные сети. Хронологически первыми появились глобальные вычислительные сети. Они объединяют компьютеры, рассредоточенные на расстоянии сотен и тысяч километров. Часто используются уже существующие не очень качественные линии связи. Более низкие, чем в локальных сетях, скорости передачи данных (десятки килобит в секунду) ограничивают набор предоставляемых услуг.

Локальные вычислительные сети сосредоточены на территории не более 1-2 км; построены с использованием дорогих высококачественных линий связи, которые позволяют достигать высоких скоростей обмена данными — порядка 100 Мбит/с.

Городские сети (или сети мегаполиса) занимают промежуточное положение между локальными и глобальными сетями. При достаточно больших расстояниях между узлами (десятки километров) они обладают качественными линиями связи и высокими скоростями обмена.

# ГЛАВА 1. ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ. ОБЩИЕ СВЕДЕНИЯ

## 1.1. Базовая структура ЭВМ. Классификация

В настоящее время примерно одинаковое распространение получили два способа построения вычислительных машин: *с непосредственными связями* и *на основе шины*.

Типичным представителем первого способа может служить классическая фон-неймановская вычислительная машина (ВМ) (рис. 1.1). В ней между взаимодействующими устройствами (процессор, память, устройство ввода/вывода) имеются непосредственные связи. Особенности связей (число линий в шинах, пропускная способность и т. п.) определяются видом информации, характером и интенсивностью обмена. Достоинством архитектуры с непосредственными связями можно считать возможность развязки «узких мест» путем улучшения структуры и характеристик только определенных связей, что экономически может быть наиболее выгодным решением. У фон-неймановских ВМ таким «узким местом» является канал пересылки данных между ЦП и памятью, и «развязать» его достаточно непросто. Кроме того, ВМ с непосредственными связями плохо поддаются реконфигурации.



Рис. 1.1. Структура фон-неймановской вычислительной машины.

В варианте с общей шиной все устройства вычислительной машины подключены к магистральной шине, служащей единственным трактом для потоков команд, данных и управления (рис. 1.2). Наличие общей шины существенно упрощает реализацию вычислительной машины, позволяет легко менять состав и конфигурацию машины. Благодаря этим свойствам шинная архитектура получила широкое распространение в мини- и микроЭВМ. Вместе с тем именно с шиной связан и основной недостаток архитектуры: в каждый момент передавать информацию по шине может только одно устройство. Основную нагрузку на шину создают обмены между процессором и памятью, связанные с извлечением из памяти команд и данных и записью в память результатов вычислений. На операции ввода/вывода остается лишь часть пропускной способности шины. Практика показывает, что даже при достаточно быстрой шине для 90% приложений этих остаточных ресурсов обычно не хватает, особенно в случае ввода или вывода больших массивов данных.

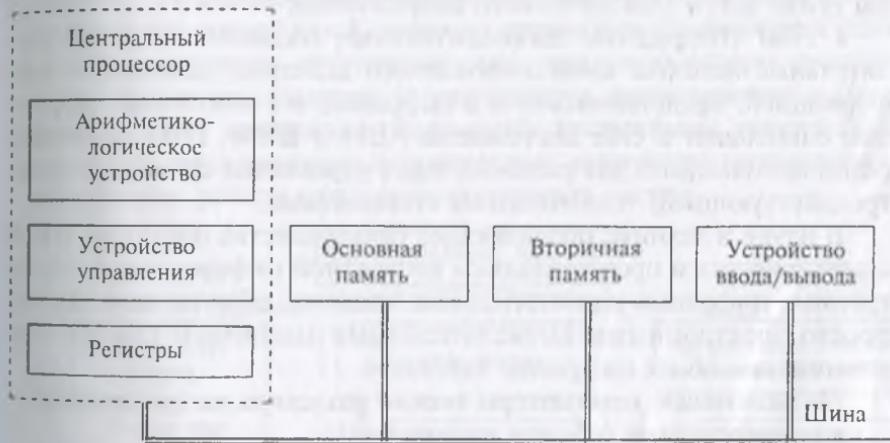


Рис. 1.2. Структура вычислительной машины на базе общей шины.

В целом следует признать, что при сохранении фон-неймановской концепции последовательного выполнения команд программы шинная архитектура в чистом ее виде оказывается недостаточно эффективной. Более распространена *архитектура с иерархией шин*, где помимо магистральной шины имеется еще несколько дополнительных шин. Они могут обеспечивать непосредственную связь между устройствами с наиболее интенсивным обменом, например процессором и кэш-памятью. Другой вариант использования дополнительных шин – объединение однотипных устройств ввода/вывода с последующим выходом с дополнительной шины на

магистральную. Все эти меры позволяют снизить нагрузку на общую шину и более эффективно расходовать ее пропускную способность.

Вычислительные машины могут быть классифицированы по различным признакам (рис. 1.3.).

*По принципу действия* вычислительные машины делятся на три больших класса: аналоговые, цифровые и гибридные.

Критерием деления вычислительных машин на эти три класса является форма представления информации, с которой они работают:

- *ЦВМ* (цифровые вычислительные машины), или вычислительные машины дискретного действия, работают с информацией, представленной в дискретной, а точнее в цифровой форме.

- *АВМ* (аналоговые вычислительные машины), или вычислительные машины непрерывного действия, работают с информацией, представленной в непрерывной (аналоговой) форме, то есть в виде непрерывного ряда значений какой-либо физической величины (чаще всего электрического напряжения).

- *ГВМ* (гибридные вычислительные машины), или вычислительные машины комбинированного действия, работают с информацией, представленной и в цифровой, и в аналоговой форме; они совмещают в себе достоинства АВМ и ЦВМ. ГВМ целесообразно использовать для решения задач управления сложными быстродействующими техническими комплексами.

В науке и технике подавляющее преимущество получили ЦВМ с электрическим представлением дискретной информации – электронные цифровые вычислительные машины, обычно называемые просто электронными вычислительными машинами (ЭВМ), без упоминания об их цифровом характере.

*По назначению* компьютеры можно разделить на три группы:

- универсальные (общего назначения);
- проблемно-ориентированные;
- специализированные.

*Универсальные компьютеры* предназначены для решения самых различных инженерно-технических, экономических, математических, информационных и им подобных задач, отличающихся сложностью алгоритмов и большим объемом обрабатываемых данных. Они широко применяются в вычислительных центрах коллективного пользования и в других мощных вычислительных комплексах.

Характерными чертами универсальных компьютеров являются:

- высокая производительность;
- разнообразие форм обрабатываемых данных: двоичные, десятичные, символьные, – при большом диапазоне их изменения и высокой точности их представления;

- обширная номенклатура выполняемых операций, как арифметических, логических, так и специальных;
- большая емкость оперативной памяти;
- развитая организация системы ввода-вывода информации, обеспечивающая подключение разнообразных видов внешних устройств.

*Проблемно-ориентированные компьютеры* предназначены для решения более узкого круга задач, связанных, как правило, с управлением технологическими объектами; регистрацией, накоплением и обработкой относительно небольших объемов данных; выполнением расчетов по относительно несложным алгоритмам; они обладают ограниченными, по сравнению с универсальными компьютерами, аппаратными и программными ресурсами.

*Специализированные компьютеры* предназначены для решения определенного узкого круга задач или реализации строго определенной группы функций. Такая узкая ориентация компьютеров позволяет четко специализировать их структуру, существенно снизить их сложность и стоимость при сохранении высокой производительности и надежности работы. К специализированным компьютерам можно отнести, например, программируемые микропроцессоры специального назначения; адаптеры и контроллеры, выполняющие логические функции управления отдельными несложными техническими устройствами, агрегатами и процессами; устройства согласования и сопряжения работы узлов вычислительных систем.

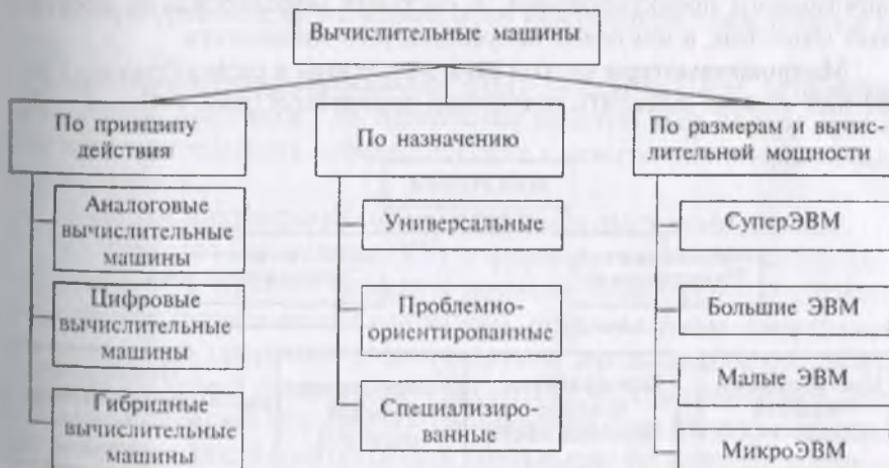


Рис. 1.3. Классификация вычислительных машин.

*По размерам и вычислительной мощности* компьютеры можно разделить на сверхбольшие (суперкомпьютеры, суперЭВМ); большие; малые; сверхмалые (микрокомпьютеры или микроЭВМ).

**Суперкомпьютерам** присуще: очень высокое быстродействие — порядка сотен миллионов операций в секунду (для чисел с плавающей запятой). Они предназначены для решения научно-технических задач большой сложности, работы с большими базами данных, они используются для управления ресурсами вычислительных сетей. Это, как правило, многопроцессорная вычислительная машины.

**Большие компьютеры** (за рубежом их часто называют **мэйнфреймами**) используются их в качестве серверов компьютерных сетей. Они имеют основную память емкостью от 512 до 10000 Мбайт, внешнюю память не менее 100 Гбайт, а также обеспечивают многопользовательский режим работы (одновременно обслуживаются от 16 до 1000 пользователей). Основные направления эффективного применения мэйнфреймов — решение научно-технических задач, работа в вычислительных системах с пакетной обработкой информации, при создании баз данных предприятий и организаций, управление вычислительными сетями и их ресурсами.

**Малые компьютеры (миниЭВМ)** — надежные, недорогие и удобные в эксплуатации компьютеры, обладающие несколько более низкими, по сравнению с мэйнфреймами и большими компьютерами, возможностями. В основном ориентированы на использование в качестве управляющих вычислительных комплексов. Наряду с использованием их для управления технологическими процессами миникомпьютеры успешно применяются в системах автоматизированного проектирования, в системах моделирования несложных объектов, в системах искусственного интеллекта.

**МикроКомпьютеры** весьма многочисленны и разнообразны. Среди них можно выделить несколько подклассов (рис. 1.4):



Рис. 1.4. Классификация микроЭВМ.

*Многопользовательские микрокомпьютеры* — это мощные микрокомпьютеры, оборудованные несколькими видеотерминалами и функционирующие в режиме разделения времени, что позволяет эффективно работать на них сразу нескольким пользователям.

*Персональные компьютеры* — однопользовательские микрокомпьютеры, удовлетворяющие требованиям общедоступности и универсальности применения.

*Рабочие станции* представляют собой однопользовательские микрокомпьютеры, часто специализированные для выполнения определенного вида работ (графических, инженерных, издательских и т. д.).

*Серверы* — многопользовательские мощные микрокомпьютеры в вычислительных сетях, выделенные для обработки запросов от всех рабочих станций сети.

*Сетевые компьютеры* — упрощенные микрокомпьютеры, обеспечивающие работу в сети и доступ к сетевым ресурсам, часто специализированные для выполнения определенного вида работ (защита сети от несанкционированного доступа, организация просмотра сетевых ресурсов, электронной почты и т. д.).

## 1.2. Устройства электронных вычислительных машин

Структурная схема персонального компьютера представлена на рис. 1.5.

**Процессор** (микропроцессор, МП) — центральное устройство ПК, предназначенное для управления работой всех блоков машины и для выполнения арифметических и логических операций над информацией.

В состав микропроцессора входят несколько компонентов:

- устройство управления (УУ) — формирует и подает во все блоки машины в нужные моменты времени определенные сигналы управления (управляющие импульсы), обусловленные спецификой выполняемой операции и результатами предыдущих операций; формирует адреса ячеек памяти, используемых выполняемой операцией, и передает эти адреса в соответствующие блоки компьютера; опорную последовательность импульсов устройство управления получает от генератора тактовых импульсов;

- арифметико-логическое устройство (АЛУ) — предназначено для выполнения всех арифметических и логических операций над числовой и символьной информацией (в некоторых моделях ПК для ускорения выполнения операций к АЛУ подключается дополнительный математический сопроцессор);

- микропроцессорная память (МПП) – предназначена для кратковременного хранения, записи и выдачи информации, непосредственно используемой в ближайшие такты работы машины; МПП строится на регистрах для обеспечения высокого быстродействия машины, ибо основная память (ОП) не всегда обеспечивает скорость записи, поиска и считывания информации, необходимую для эффективной работы быстродействующего микропроцессора. Регистры – быстродействующие ячейки памяти различной длины (в отличие от ячеек ОП, имеющих стандартную длину один байт и более низкое быстродействие);

- интерфейсная система микропроцессора – предназначена для сопряжения и связи с другими устройствами ПК; включает в себя внутренний интерфейс МП, буферные запоминающие регистры и схемы управления портами ввода-вывода (ПВВ) и системной шиной.

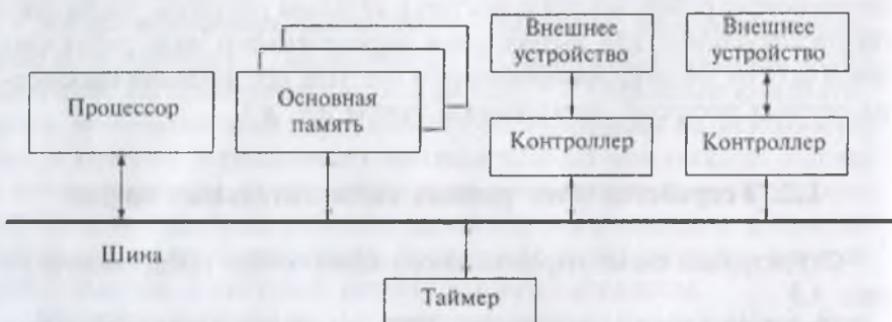


Рис. 1.5. Структурная схема персонального компьютера.

**Основная память (ОП)** – предназначена для хранения и оперативного обмена информацией с прочими блоками машины. ОП содержит два вида запоминающих устройств: постоянное запоминающее устройство (ПЗУ) и оперативное запоминающее устройство (ОЗУ).

**Внешние устройства (ВУ)** персонального компьютера – важнейшая составная часть любого вычислительного комплекса, достаточно сказать, что по стоимости ВУ составляют до 80-85 % стоимости всего ПК.

ВУ ПК обеспечивают взаимодействие машины с окружающей средой: пользователями, объектами управления и другими компьютерами.

К внешним устройствам относятся:

- внешние запоминающие устройства (ВЗУ) или внешняя память ПК;

- диалоговые средства пользователя;
- устройства ввода информации;
- устройства вывода информации;
- средства связи и телекоммуникаций.

**Системная шина** – основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой. Системная шина включает в себя:

- кодовую шину данных (КШД), содержащую провода и схемы сопряжения для параллельной передачи всех разрядов числового кода (машинного слова) операнда;
- кодовую шину адреса (КША), содержащую провода и схемы сопряжения для параллельной передачи всех разрядов кода адреса ячейки основной памяти или порта ввода-вывода внешнего устройства;
- кодовую шину инструкций (КШИ), содержащую провода и схемы сопряжения для передачи инструкций (управляющих сигналов, импульсов) во все блоки машины;
- шину питания, содержащую провода и схемы сопряжения для подключения блоков ПК к системе энергопитания.

Системная шина обеспечивает три направления передачи информации:

- между микропроцессором и основной памятью;
- между микропроцессором и портами ввода-вывода внешних устройств;
- между основной памятью и портами ввода-вывода внешних устройств (в режиме прямого доступа к памяти).

Все блоки, а точнее их порты ввода-вывода, через соответствующие унифицированные разъемы (стыки) подключаются к шине единообразно: непосредственно или через контроллеры (адAPTERы). Управление системной шиной осуществляется микропроцессором либо непосредственно, либо, что чаще, через дополнительную микросхему контроллера шины, формирующую основные сигналы управления. Обмен информацией между внешними устройствами и системной шиной выполняется с использованием ASCII-кодов.

**Таймер** – внутримашинные электронные часы реального времени, обеспечивающие при необходимости автоматический съем текущего момента времени (год, месяц, часы, минуты, секунды и доли секунд). Таймер подключается к автономному источнику питания – аккумулятору и при отключении машины от электросети продолжает работать.

### 1.3. Особенности представления информации в персональных компьютерах

Числовая информация внутри персональных компьютеров (ПК) кодируется в двоичной или в двоично-десятичной системах счисления; при вводе и выводе любой информации используются специальные коды представления информации — коды ASCII, эти же коды применяются для кодирования буквенной и символьной информации и внутри ПК.

Для удобства работы введены следующие термины для обозначения совокупностей двоичных разрядов (см. табл. 1.1). Эти термины обычно используются в качестве единиц измерения объемов информации, хранимой или обрабатываемой в компьютере.

Последовательность нескольких битов или байтов часто называют *полям данных*.

Биты в числе (в слове, в поле и т. п.) нумеруются справа налево, начиная с 0-го разряда. В ПК могут обрабатываться поля постоянной и переменной длины.

Поля постоянной длины:

- слово — 2 байта;
- двойное слово — 4 байта;
- полуслово — 1 байт;
- расширенное слово — 8 байтов.

Таблица 1.1.

Количество двоичных разрядов в группе	Наименование единицы измерения
1	Бит
8	Байт
16	Параграф
8•1024	Кбайт (килобайт)
8•1024 <sup>2</sup>	Мбайт (мегабайт)
8•1024 <sup>3</sup>	Гбайт (гигабайт)
8•1024 <sup>4</sup>	Тбайт (терабайт)
8•1024 <sup>5</sup>	Пбайт (пентабайт)

Числа с фиксированной запятой чаще всего имеют формат слова и полуслова; числа с плавающей запятой — формат двойного и расширенного слова (математические сопроцессоры IBM PC могут работать с 10-байтными словами).

Поля переменной длины могут иметь любой размер от 0 до 255 байтов, но обязательно равный целому числу байтов.

Например, структурно запись двоичного числа  $-11000001_{(2)}$ , равного десятичному  $-193_{(10)}$ , в разрядной сетке ПК, выглядит следующим образом (рис. 1.6, а,б).

Разряд	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Число	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
Знак числа	Абсолютная величина числа															

а ) число с фиксированной запятой в формате слово со знаком

Разряд	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10
Число	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	1	0
Знак числа	Порядок															Мантисса						

б) число с плавающей запятой в формате двойное слово

Рис. 1.6. Структурное представление двоичных чисел в персональном компьютере.

Двоично-кодированные десятичные числа могут быть представлены в ПК полями переменной длины в так называемых упакованном (рис. 1.7) и распакованном форматах. В упакованном формате для каждой десятичной цифры отводится по 4 двоичных разряда (полбайта), при этом знак числа кодируется в крайнем правом полубайте числа (1100 – знак «+» и 1101 – знак «-»).

Цф	Цф	Цф	Цф	...	Цф	Знак
----	----	----	----	-----	----	------

Рис. 1.7. Структура поля упакованного формата.

Здесь и далее: Цф – цифра, Знак – знак числа. Упакованный формат используется в ПК обычно при выполнении операций сложения и вычитания двоично-десятичных чисел.

В распакованном формате (рис. 1.8) для каждой десятичной цифры выделяется по целому байту, при этом старшие полубайты (зона) каждого байта (кроме самого младшего) в ПК заполняются

кодом ООН (в соответствии с ASCII-кодом), а в младших (левых) полубайтах обычным образом кодируются десятичные цифры. Старший полуbyte (зона) самого младшего (правого) байта используется для кодирования знака числа.

Зона	Цф	...	Цф	...	Зона	Цф	Знак	Цф
------	----	-----	----	-----	------	----	------	----

Рис. 1.8. Структура поля распакованного формата.

Распакованный формат используется в ПК при вводе-выводе информации, а также при выполнении операций умножения и деления двоично-десятичных чисел.

Например, число  $-193_{(10)} = -000110010011_{(2)}$  в ПК будет представлено:

- в упакованном формате:

0001	1001	0011	1101
------	------	------	------

- в распакованном формате:

0011	0001	0011	1001	1101	0011
------	------	------	------	------	------

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Оцените достоинства и недостатки структур вычислительных машин с непосредственными связями и общей шиной.
2. Дайте общую характеристику классов ЭВМ аналоговых, цифровых и гибридных машин.
3. Приведите классификацию ЭВМ по принципу их работы.
4. Перечислите основные типы микроЭВМ.
5. Приведите структурную схему персонального компьютера.
6. Расскажите об основных устройствах, входящих в состав персонального компьютера, и задачах, выполняемых ими.
7. Объясните формы представления информации с плавающей и фиксированной запятой.
8. Что это значит, когда говорят об упакованном и распакованном форматах представления двоично-кодированных десятичных чисел?

## ГЛАВА 2. АРХИТЕКТУРА МИКРОПРОЦЕССОРОВ

### 2.1. Структура 16-разрядного микропроцессора i8086

Структурная схема микропроцессора i8086 представлена на рис. 2.1. Микропроцессор включает в себя три основных устройства:

- УОД – устройство обработки данных;
- УСМ – устройство связи с магистралью;
- УУС – устройство управления и синхронизации.

УОД предназначено для выполнения команд и включает в себя 16-разрядное АЛУ, системные регистры и другие вспомогательные схемы; блок регистров (РОН, базовые и индексные) и блок микропрограммного управления.

УСМ обеспечивает формирование 20-разрядного физического адреса памяти и 16-разрядного адреса ВУ, выбор команд из памяти, обмен данными с ЗУ, ВУ, другими процессорами по магистрали. УСМ включает в себя сумматор адреса, блок регистров очереди команд и блок сегментных регистров.

УУС обеспечивает синхронизацию работы устройств МП, выработку управляющих сигналов и сигналов состояния для обмена с другими устройствами, анализ и соответствующую реакцию на сигналы других устройств МПС. Микропроцессор i8086 может работать в одном из двух режимов – *минимальном* и *максимальном*. Минимальный режим предназначен для реализации однопроцессорной конфигурации МПС с организацией, подобной МПС на базе i8080, но с увеличенным адресным пространством, более высоким быстродействием и значительно расширенной системой команд. Максимальный режим предполагает наличие в системе нескольких микропроцессоров, работающих на общую системную шину. МПС на базе i8086 с использованием максимального режима не получили широкого распространения. Более того, в последующих моделях своих микропроцессоров (80286, 80386, 80486) фирма Intel отказалась от поддержки мультипроцессорной архитектуры. Поэтому мы здесь не будем рассматривать особенности организации максимального режима.

На внешних выводах МП i8086 широко используется принцип *мультиплексирования сигналов* – передача разных сигналов по общим линиям с разделением во времени. Кроме того, одни и те же выводы могут использоваться для передачи разных сигналов в зависимости от режима (min – max).

В таблице приведено описание внешних выводов МП i8086. При описании выводов косой чертой (/) разделены сигналы, появляющиеся на выводе в разные моменты машинного цикла. В круглых скобках указаны сигналы, характерные только для максимального режима. Символ \* после имени сигнала – знак его инверсии.

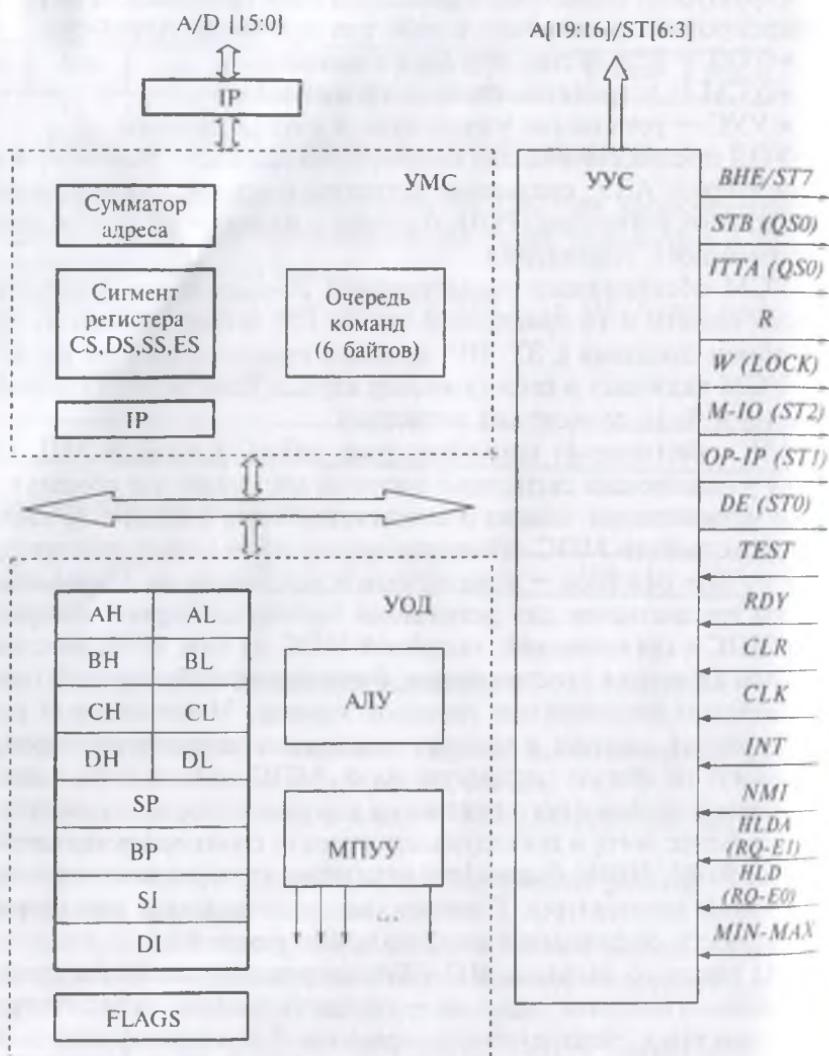


Рис.2.1. Структура микропроцессора i8086.

Таблица 2.1.

## Внешние выводы МП i3086

Внешний вывод	Описание
A/D[15:0]	Младшие 0 – 15 разряды адреса/данные
A[19:16]/ST[6:3]	Старшие 16 – 19 разряды адреса/сигналы состояния
BHE* /ST[7]	Разрешение передачи старшего байта данных/сигнал состояния
STB(QSO)	Строб адреса (состояние очереди команд)
R*	Чтение
W* /(LOCK*)	Запись (блокировка канала)
M-IO*(ST2*)	Память – внешнее устройство (состояние цикла)
OP-IP*(ST1*)	Выдача/прием (состояние цикла)
DE*(STO*)	Разрешение передачи данных (состояние цикла)
TEST*	Проверка
RDY	Готовность
CLR	Сброс
CLC	Тактовый сигнал
INT	Запрос внешнего прерывания
INTA*(QSI)	Подтверждение прерывания (состояние очереди команд)
NMI	Запрос немаскируемого прерывания
HLD(RQ*/EO)	Запрос ПДП (запрос/подтверждение доступа к магистрали)
NLDA(RQ*/EI)	Подтверждение ПДП (запрос/подтверждение доступа к магистрали)
MIN/MAX*	Потенциал задания режима (min = 1, max = 0)

## 2.2. Структура 32-разрядного микропроцессора x86 (серии Pentium)

Обобщенная структурная схема 32-разрядного микропроцессора x86 (серии Pentium) приведена на рис. 2.2.

Условно **микропроцессор** можно разделить на три части: исполнительный блок (Execution Unit – EU), устройство сопряжения с

системной магистралью (Bus Interface Unit — BIU) и блок управления микропроцессором (Control Unit — CU).

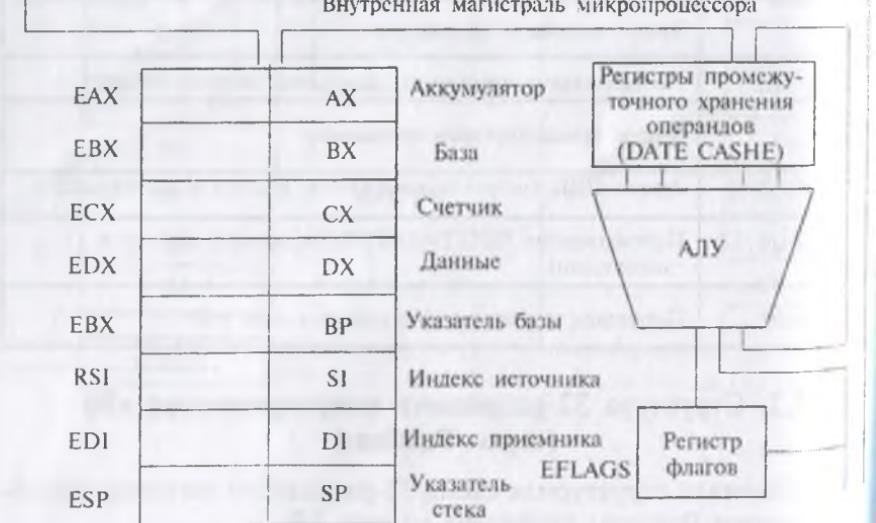
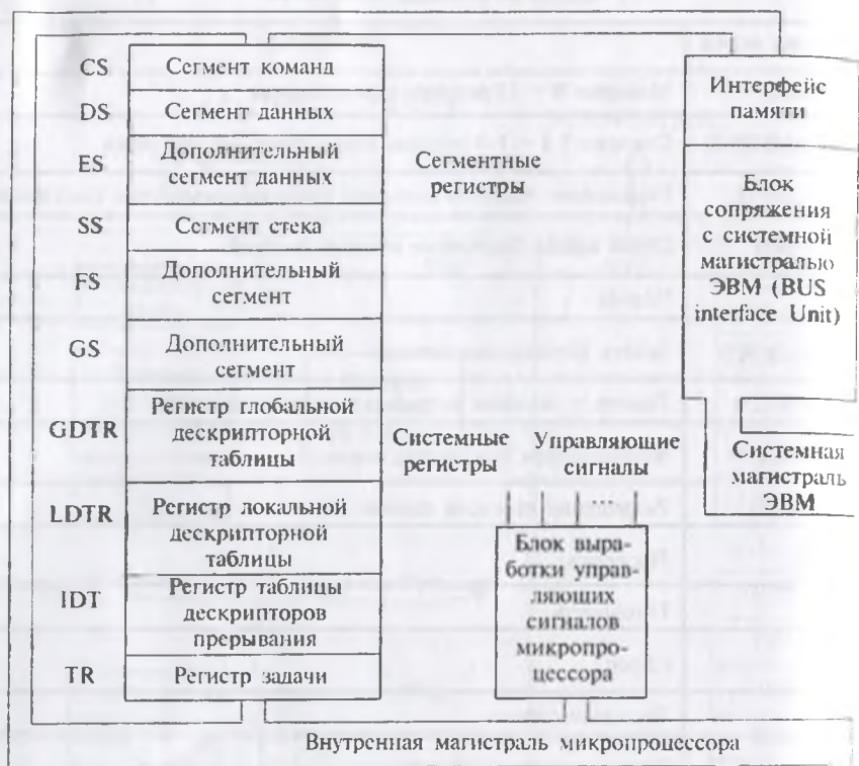


Рис.2.2. Обобщенная структурная схема 32-разрядного микропроцессора x86.

В исполнительном блоке находятся: арифметический блок (DATE CACHE, АЛУ, регистр флагов), регистры общего назначения (РОН) EAX, EBX, ECX, EDX; общие регистры ESI, EDI, ESP, EBP.

В регистре флагов каждый разряд имеет строго определенное назначение. Обычно разряды регистра флагов устанавливаются аппаратно при выполнении очередной операции в зависимости от получаемого в АЛУ результата. При этом фиксируются такие свойства получаемого результата, как нулевой результат, отрицательное число, переполнение разрядной сетки АЛУ и т.д. Но некоторые разряды регистра флагов могут устанавливаться по специальным командам: установки флагов (STC, STD, STI), сброса (CLC, CLD, CLI), инвертирования (команда CMC). Некоторые разряды имеют чисто служебное назначение (например, хранят разряд, «выпавший» из АЛУ во время сдвига) или являются резервными (т.е. не используются).

Регистры общего назначения EAX, EBX, ECX, EDX имеют длину по 32 бита. Каждый из них делится на два 16-битных регистра, младший из которых имеет свое имя (что обеспечивает совместимость с 16-разрядными версиями микропроцессоров). Так, в регистре EAX содержится 16-битный регистр AX, в регистре EBX содержится регистр BX, в ECX — регистр CX, в EDX — регистр DX. Каждый из 16-битных регистров в свою очередь состоит из двух 8-битных регистров, имеющих свои имена. Так, AX (аккумулятор) делится на AH и AL, регистр базы (Base Register) — BX делится на BH и BL. Счетчик (Count Register) CX включает регистры CH и CL. Регистр данных (Data Register) DX содержит регистры DH и DL. Каждый из коротких регистров может использоваться самостоятельно или в составе регистровой пары. Условные названия (аккумулятор, регистр базы, счетчик, регистр данных) не ограничивают применения этих регистров — эти названия говорят о наиболее частом их использовании или об особенности использования того или иного регистра в той или иной команде. Буква «H» в имени 8-битного регистра означает верхний (или старший) регистр, буква «L» — младший (т.е. младший байт 16-битного регистра или младший однобайтный регистр двухбайтного регистра).

Эти регистры используются для операций с данными, такими, как сравнение, математические операции или запись данных в память. Регистр CX чаще всего используется как счетчик циклов. AH в DOS-программах часто используется как определитель, какой сервис будет необходим при вызове программного прерывания по команде INT.

Общие регистры — ESP, EBP, ESI, EDI также 32-битные, младшая половина которых доступна как регистры SP, BP, SI, DI.

Регистр ESP указывает на адрес вершины стека (адрес, куда будет заноситься следующая переменная командой PUSH).

Регистр EBP содержит адрес базы, который может использоваться при работе со стеком.

Регистр ESI — адрес источника, обычно содержит адрес начала блока информации для операций типа «переместить блок», а регистр EDI — адрес приемника (назначения) в этой операции.

### *Арифметико-логическое устройство (АЛУ)*

Arithmetric and Logical Unit (ALU) — компонента процессора, выполняющая арифметические и логические операции над данными.

АЛУ реализует важную часть процесса обработки данных. Она заключается в выполнении набора простых операций. Арифметической операцией называют процедуру обработки данных, аргументы и результат которой являются числами (сложение, вычитание, умножение, деление). Логической операцией именуют процедуру, осуществляющую построение сложного высказывания (операции И, ИЛИ, НЕ, ...). АЛУ состоит из регистров, сумматора с соответствующими логическими схемами и блока управления выполняемым процессом. Устройство работает в соответствии с сообщаемыми ему именами (кодами) операций, которые при пересылке данных нужно выполнить над переменными, помещаемыми в регистры.

АЛУ классифицируются следующим образом:

1. *По способу действий над операндами:*

- АЛУ последовательного действия;
- параллельного действия.

В последовательных АЛУ действия над операндами производятся последовательно — разряд за разрядом, начиная с младшего. В параллельных АЛУ все разряды операндов обрабатываются одновременно.

2. *По виду обрабатываемых чисел АЛУ* могут производить операции над двоичными числами с фиксированной или плавающей запятой и над двоично-десятичными числами.

АЛУ при действии над двоично-десятичными числами должны содержать схему десятичной коррекции. Схема десятичной коррекции преобразует полученный результат таким образом, чтобы каждый двоично-десятичный разряд не содержал цифру больше 9.

При записи числа с фиксированной запятой запятая фиксируется после младшего разряда, если число целое, и перед старшим, если число меньше 1.

При записи чисел с плавающей запятой выделяется целая часть, которая называется мантиссой, и показатель степени, который характеризует положение запятой.

### 3. По организации действий над операндами:

- блочные;
- многофункциональные АЛУ.

В блочных АЛУ отдельные блоки предназначены для действий над двоично-десятичными числами, отдельно для действий над числами с фиксированной запятой, отдельно с плавающей запятой.

В многофункциональных АЛУ одни и те же блоки обрабатывают числа с фиксированной запятой, плавающей запятой и двоично-десятичные числа (рис. 2.3).

Клапаны K1 и K2 объединяют сумматоры 1, 2 и 3 для действий над числами с фиксированной запятой.

Для действий над числами с плавающей запятой клапан K2 объединяет сумматоры 2 и 3 для обработки мантисс, а клапан K1 отсоединяет первый сумматор от второго. Сумматор 1 обрабатывает порядки.

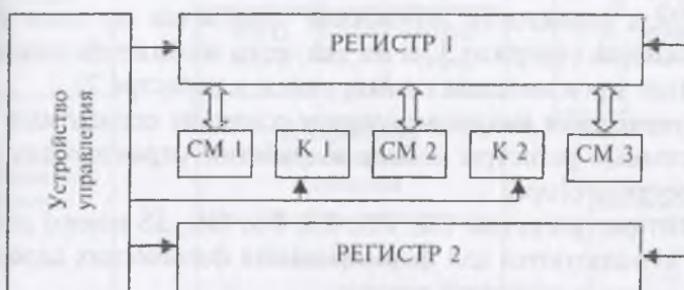


Рис.2.3. Многофункциональное АЛУ.

### 4. По структуре:

- АЛУ с непосредственными связями;
- многосвязные.

В многосвязных АЛУ входы и выходы регистров приемников и источников информации подсоединяются к одной шине. Распределение входных и выходных сигналов происходит под действием управляющих сигналов.

В АЛУ с непосредственной связью вход регистра приемника связан с выходом регистра источника операндов и регистра, в котором происходит обработка (рис. 2.4).

Например, в этой схеме суммирование происходит так: операнды подаются в регистр 1. Регистр 2 является накапливающим сумматором или автоматом с памятью. Он суммирует слагаемые, поступающие в разные моменты времени и передает результат в регистр 3.

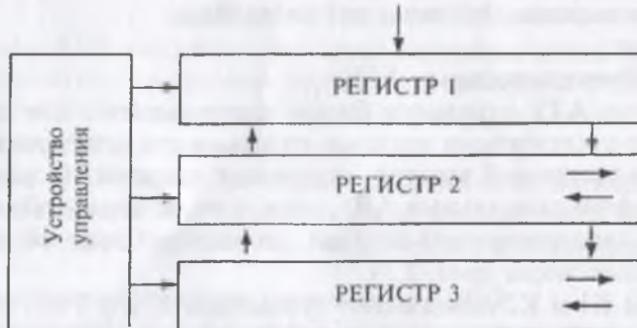


Рис.2.4. АЛУ с непосредственной связью.

Умножение в этом АЛУ происходит так: множимое помещают в регистр 3, множитель — в регистр 1. Регистры 2 и 3 являются, кроме того, сдвигающими регистрами. В зависимости от содержимого разряда множителя, множимое сдвигается на один разряд, если множитель содержит 1, и на два, если множитель содержит 0. Эти частные произведения суммируются в регистре 2.

**Блок управления микропроцессором** содержит сегментные регистры, системные регистры и блок выработки управляющих сигналов микропроцессора.

Сегментные регистры CS, DS, ES, FS, GS, SS имеют длину по 16 бит и используются для формирования физических адресов команд и данных в основной памяти.

CS — сегмент кода исполняемой в данный момент программы.

DS — сегмент данных исполняемой программы, т.е. константы, строковые ссылки и т.д.

SS — сегмент стека исполняемой программы.

ES, FS, GS — дополнительные сегменты, которые в некоторых программах могут не использоваться.

Системные регистры GDTR и LDTR являются регистрами глобальной и локальной дескрипторных таблиц. GDTR имеет длину 48J бит, LDTR — 16 бит (точнее, 16 бит — это только «видимая» часть этого регистра).

Регистр IDTR (таблицы дескрипторов прерываний — DTR) имеет длину 48 бит, 32 из которых являются базовым адресом таблицы дескрипторов прерываний (IDT), а 16 — смещением этого адреса (пределом).

Регистр TR является регистром состояния задачи. «Видимая» его часть имеет длину 16 бит и содержит селектор дескриптора. «Скрытая» часть регистра содержит загружаемые автоматически

базовый адрес точки входа, предел и атрибуты задачи. Структурная схема блока выработки управляющих сигналов микропроцессора приведена на рис 2.5.

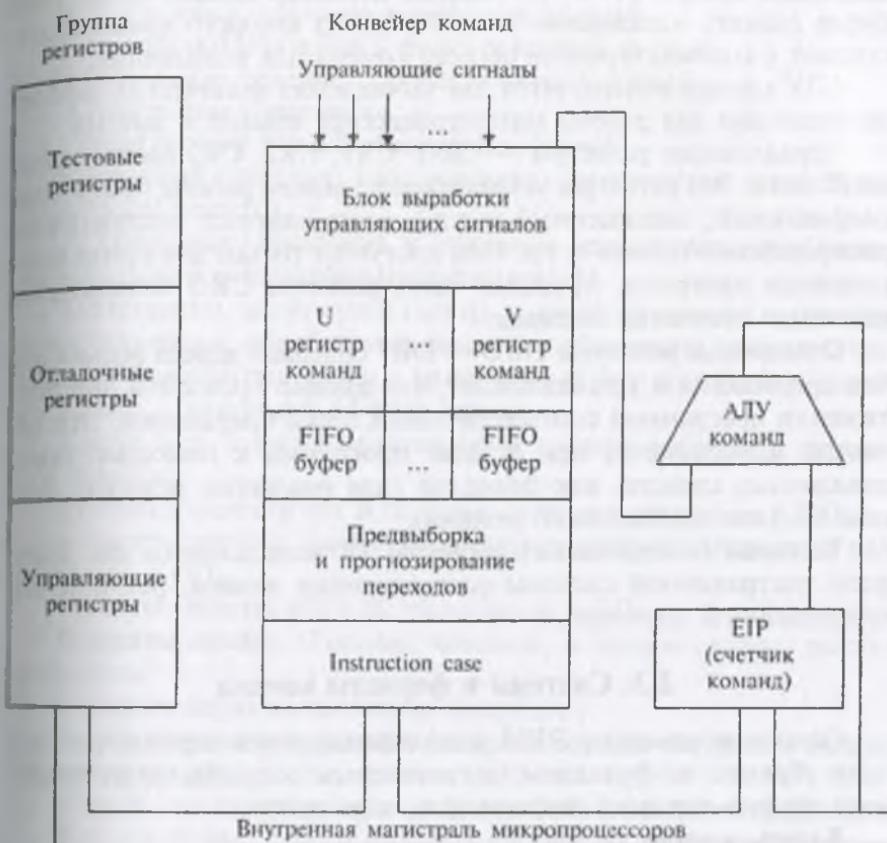


Рис.2.5. Блок формирования управляющих сигналов микропроцессора.

Основу его составляют счетчик команд, АЛУ, конвейер команд и группа управляющих, отладочных и тестовых регистров.

Регистр ЕIP является указателем адреса команды (Instruction Pointer), которая будет выбираться в конвейер команд в качестве очередной команды (в отечественной литературе такое устройство называется «счетчик команд»).

Конвейер команд МП хранит несколько команд, что позволяет при выполнении линейных программ совместить подготовку очередной команды с выполнением текущей. Команды в конвейер команд поступают с внутренней магистралью микропроцессора и накапливаются в кэше команд. Блок предвыборки и прогнозирования

переходов осуществляет трансляцию команд x86 в RISC-команды, прогнозирует последовательность исполнения команд и направляет полученные последовательности команд в соответствующие ветви конвейера команд ( $U, V, \dots$ ). Каждый конвейер команд имеет свой буфер (память магазинного типа FIFO), из которого команды поступают в соответствующий регистр команд для исполнения.

АЛУ команд используется для вычисления физических адресов необходимых для работы микропроцессора команд и данных.

Управляющие регистры — CR0, CR1, CR2, CR3 имеют длину по 32 бита. Эти регистры устанавливают режим работы процессора (нормальный, защищенный и т.д.), контролируют постраничное распределение памяти и т.д. Они доступны только для привилегированных программ. Младшая часть регистра CR0 используется как слово состояния машины.

Отладочные регистры DR0 — DR7 содержат адреса восьми токов прерывания и устанавливают, что должно произойти при достижении программой соответствующей точки прерывания. Эти регистры используются при отладке программы с помощью таких отладочных средств, как debug.exe (для реального режима), или ntsd.exe (для защищенного режима).

Тестовые (контрольные) регистры TR используются для контроля постраничной системы распределения памяти, реализуемой операционной системой.

### 2.3. Системы и форматы команд

Основные команды ЭВМ классифицируются вкратце следующим образом: по функциям (выполняемым операциям), направлению приема-передачи информации, адресности.

#### Классы команд

1. Команды обработки данных, в том числе (01 — первый операнд, 02 — второй).

1.1. Короткие операции (один такт).

1.1.1. Логические:

- логическое сложение (для каждого бита 01 и 02 осуществляется операция ИЛИ);

- логическое умножение (для каждого бита 01 и 02 осуществляется операция И);

- инверсия (в 01 все единицы заменяются на нули, и наоборот);

- сравнение логическое (если 01=02, то некий флаг или регистр устанавливается в «1», иначе в «0»);

1.1.2. Арифметические:

- сложение операндов;

- вычитание (сложение в обратном коде);
- сравнение арифметическое (если 01>02, или 01=02, или 01<02, то некий флаг или регистр устанавливается в 1, иначе – в 0);

1.2. Длинные операции (несколько тактов):

- сложение/вычитание с фиксированной точкой;
- умножение/деление с фиксированной точкой.

2. Операции управления:

- безусловный переход (ветвление, branch);
- условный переход (по условию, результатам вычислений (conditional branch)).

3. Операции обращения к внешним устройствам (требование на запись или считывание информации).

Естественно, могут существовать и другие операции – *десятичная арифметика*, обработка *символьной* информации, работа с числами *половинной* (полуслово – например, 16 бит) или *двойной* (двойное слово – например, 64 бит) длины.

Кроме того, команды различаются по типу выборки и пересылок данных:

- регистр–регистр (01 и 02 размещаются в регистрах АЛУ);
- память–регистр (регистр–память) – один из операндов размещается в ОП;
- память–память (01 и 02 размещены в ОП).

**Форматы команд.** Типовая команда, в общем случае, должна указывать:

- подлежащую выполнению операцию;
- адреса исходных данных (операндов), над которыми выполняется операция;
- адрес, по которому должен быть помещен результат операции.

В соответствии с этим команда состоит из двух частей: операционной и адресной:

Операционная часть	Адресная часть
--------------------	----------------

Формат команды определяет ее структуру, то есть количество двоичных разрядов, отводимых под всю команду, а также количество и расположение отдельных полей команды. *Полем* называется совокупность двоичных разрядов, кодирующих составную часть команды.

Известны одно-, двух- и трехадресные машины (системы команд). *Типовая структура трехадресной команды:*

КОП	A1	A2	A3
-----	----	----	----

где A2 и A3 – адреса ячеек (регистров), где расположены соответственно первое и второе числа, участвующие в операции; A1 – адрес ячейки (регистра), куда следует поместить число, полученное в результате выполнения операции.

*Типовая структура двухадресной команды:*

КОП	A1	A2
-----	----	----

где A1 – это обычно адрес ячейки (регистра), где хранится первое из чисел, участвующих в операции, и куда после завершения операции должен быть записан результат операции; A2 – обычно адрес ячейки (регистра), где хранится второе участвующее в операции число.

*Типовая структура одноадресной команды:*

КОП	A1
-----	----

где A1 в зависимости от модификации команды может обозначать либо адрес ячейки (регистра), в которой хранится одно из чисел, участвующих в операции, либо адрес ячейки (регистра), куда следует поместить число – результат операции.

Наибольшее применение нашли *двуадресные системы* команд.

Таким образом, программирование в машинных адресах требует знания системы команд конкретной ЭВМ и их адресности. При этом реализация даже довольно несложных вычислений требует разложения их на простые операции, что значительно увеличивает общий объем программы и затрудняет ее чтение и отладку.

В качестве примера рассмотрим последовательность реализации вычисления по формуле  $y = (a + b)^2 - c/d$ .

План последовательности машинных операций, выполнение которой приведет к нужному результату, в данном случае следующий:

rl = a + b; — операция сложения;  
r2 = rl\*rl; — операция умножения;  
r3 = c/d; — операция деления;  
y = r2 - r3; — операция вычитания;  
Стоп. — завершение обработки.

## 2.4. Методы адресации

Количество методов адресации в различных процессорах может быть от 4 до 16. Рассмотрим несколько типичных методов адресации операндов, используемых сейчас в большинстве микропроцессоров.

*Непосредственная адресация* (рис. 2.6) предполагает, что операнд (входной) находится в памяти непосредственно за кодом команды. Операнд обычно представляет собой константу, которую надо куда-то переслать, к чему-то прибавить и т.д. Например, команда может состоять в том, чтобы прибавить число 6 к содержимому какого-то внутреннего регистра процессора. Это число 6 будет располагаться в памяти, внутри программы в адресе, следующем за кодом данной команды сложения.



Рис.2.6. Непосредственная адресация.

*Прямая (она же абсолютная) адресация* (рис. 2.7) предполагает, что операнд (входной или выходной) находится в памяти по адресу, код которого находится внутри программы сразу же за кодом команды. Например, команда может состоять в том, чтобы очистить (сделать нулевым) содержимое ячейки памяти с адресом 1000000. Код этого адреса 1000000 будет располагаться в памяти, внутри программы в следующем адресе за кодом данной команды очистки.

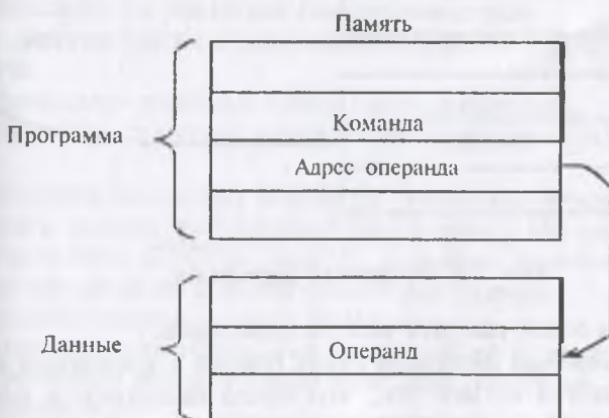


Рис.2.7. Прямая адресация.

*Регистровая адресация* (рис. 2.8) предполагает, что операнд (входной или выходной) находится во внутреннем регистре процессора. Например, команда может состоять в том, чтобы переслать число из нулевого регистра в первый. Номера обоих регистров (0 и 1) будут определяться кодом команды пересылки.

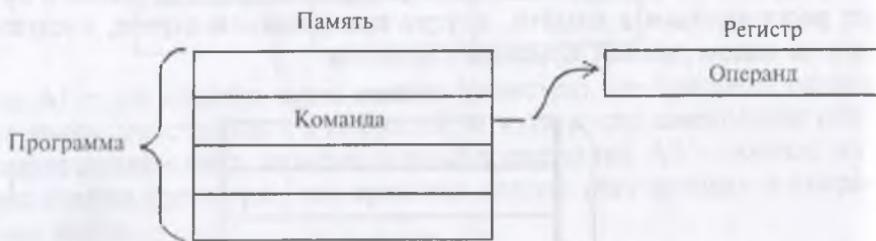


Рис. 2.8. Регистровая адресация.

*Косвенно-регистровая (она же косвенная) адресация* предполагает, что во внутреннем регистре процессора находится не сам операнд, а его адрес в памяти (рис. 2.9). Например, команда может состоять в том, чтобы очистить ячейку памяти с адресом, находящимся в нулевом регистре. Номер этого регистра (0) будет определяться кодом команды очистки.

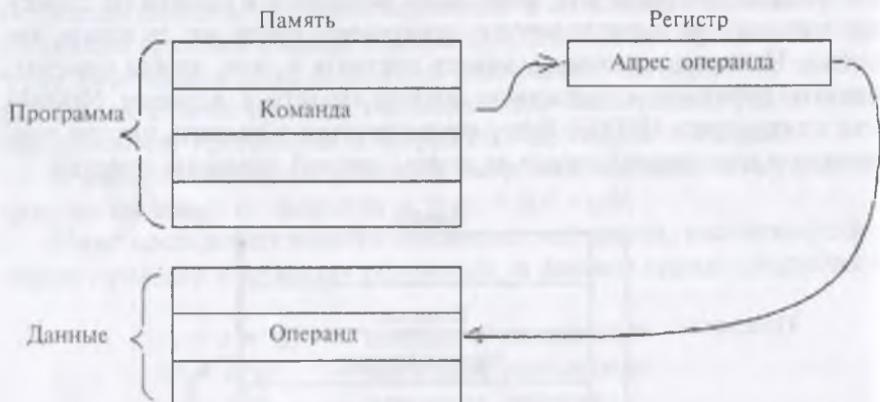


Рис. 2.9. Косвенная адресация.

Реже встречаются еще два метода адресации.

*Автоинкрементная адресация* очень близка к косвенной адресации, но отличается от нее тем, что после выполнения команды содержимое используемого регистра увеличивается на единицу или на два. Этот метод адресации очень удобен, например, при после-

довательной обработке кодов из массива данных, находящегося в памяти. После обработки какого-то кода адрес в регистре будет указывать уже на следующий код из массива. При использовании косвенной адресации в данном случае пришлось бы увеличивать содержимое этого регистра отдельной командой.

*Автодекрементная адресация* работает похоже на автоАнкрементную, но только содержимое выбранного регистра уменьшается на единицу или на два перед выполнением команды. Эта адресация также удобна при обработке массивов данных. Совместное использование автоАнкрементной и автодекрементной адресаций позволяет организовать память стекового типа.

Из других распространенных методов адресации можно упомянуть об индексных методах, которые предполагают для вычисления адреса операнда прибавление к содержимому регистра заданной константы (индекса). Код этой константы располагается в памяти непосредственно за кодом команды.

Отметим, что выбор того или иного метода адресации в значительной степени определяет время выполнения команды. Самая быстрая адресация — это регистровая, так как она не требует дополнительных циклов обмена по магистрали. Если же адресация требует обращения к памяти, то время выполнения команды будет увеличиваться за счет длительности необходимых циклов обращения к памяти. Понятно, что чем больше внутренних регистров у процессора, тем чаще и свободнее можно применять регистровую адресацию, и тем быстрее будет работать система в целом.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Расскажите об эволюции микропроцессоров.
2. Расскажите кратко о микропроцессорах, их структурах, задачах и параметрах.
3. Перечислите основные блоки микропроцессора.
4. Объясните структуру арифметико-логического устройства и ее задачи.
5. Приведите блок-схему устройства управления микропроцессора и расскажите о задачах, выполняемых входящими в его состав блоками.
6. Перечислите регистры памяти, входящие в микропроцессор, и дайте краткие сведения о выполняемых ими задачах.
7. Что такое система команд ЭВМ?
8. Какие критерии используются при выборе формата команд?
9. Перечислите типовые схемы адресации операндов, используемые в микропроцессоре.
10. В каких случаях используется автоАнкрементная, а в каких автодекрементная адресация?

## ГЛАВА 3. ОРГАНИЗАЦИЯ ПАМЯТИ ЭВМ

### 3.1. Концепция многоуровневой памяти

Известно, что память ЭВМ предназначена для хранения программ и данных, причем эффективность работы ЭВМ во многом определяется характеристиками ее памяти. Во все времена к памяти предъявлялись три основных требования: большой *объем*, высокое *быстродействие* и низкая (умеренная) *стоимость*.

Все перечисленные выше требования к памяти являются взаимно-противоречивыми, поэтому пока невозможно реализовать один тип ЗУ, отвечающий всем названным требованиям. В современных ЭВМ организуют комплекс разнотипных ЗУ, взаимодействующих между собой и обеспечивающих приемлемые характеристики памяти ЭВМ для каждого конкретного применения. В основе большинства ЭВМ лежит трехуровневая организация памяти: сверхоперативная (СОЗУ) – оперативная (ОЗУ) – внешняя (ВЗУ). СОЗУ и ОЗУ могут непосредственно взаимодействовать с процессором, ВЗУ взаимодействует только с ОЗУ.

СОЗУ обладает максимальным быстродействием (равным процессорному), небольшим объемом ( $10^5$  –  $10^7$  байтов) и располагается, как правило, на кристалле процессорной БИС. Для обращения к СОЗУ не требуются магистральные (машинные) циклы. В СОЗУ размещаются наиболее часто используемые на данном участке программы данные, а иногда – и фрагменты программы.

Быстродействие ОЗУ может быть ниже процессорного (не более чем на порядок), а объем составляет  $10^5$  –  $10^7$  байтов. В ОЗУ располагаются подлежащие выполнению программы и обрабатываемые данные. Связь между процессором и ОЗУ осуществляется по системному или специализированному интерфейсу и требует для своего осуществления машинных циклов.

Информация, находящаяся в ВЗУ, не может быть непосредственно использована процессором. Для использования программ и данных, расположенных в ВЗУ, их необходимо предварительно переписать в ОЗУ. Процесс обмена информацией между ВЗУ и ОЗУ осуществляется средствами специального канала или (реже) – непосредственно под управлением процессора. Объем ВЗУ практически неограничен, а быстродействие на 3 – 6 порядков ниже процессорного.

Схематически взаимодействие между процессором и уровнями памяти представлено на рис. 3.1.

Следует помнить, что положение ЗУ в иерархии памяти ЭВМ определяется не элементной базой запоминающих ячеек (известны

случаи реализации ВЗУ на БИС – «электронный диск» и, наоборот, организация оперативной памяти на электромеханических ЗУ – магнитных барабанах), а возможностью доступа процессора к данным, расположенным в этом ЗУ.

При организации памяти современных ЭВМ особое внимание уделяется сверхоперативной памяти и принципам обмена информацией между ОЗУ и ВЗУ.

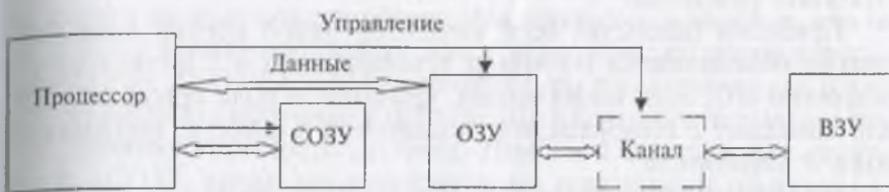


Рис. 3.1. Взаимодействие ЗУ различных уровней в составе ЭВМ.

### 3.2. Сверхоперативная память

Для эффективного применения сверхоперативного запоминающего устройства (СОЗУ) следует таким образом распределять информацию по уровням памяти ЭВМ, чтобы в СОЗУ всегда располагались наиболее часто используемые в данный момент коды.

Принято различать СОЗУ по способу доступа к хранимой в нем информации. Известны два основных класса СОЗУ по этому признаку:

- с прямым доступом;
- с ассоциативным доступом.

СОЗУ с прямым доступом (РОН – регистры общего назначения) получило широкое распространение в большинстве современных ЭВМ. Фактически РОН – это небольшая регистровая память, доступ к которой осуществляется специальными командами. Стrатегия размещения данных в РОН целиком определяется программистом (компилятором). Обычно в РОН размещают многократно используемые адреса (базы, индексы), счетчики циклов, данные активного фрагмента задачи, что повышает вероятность обращения в ячейки РОН по сравнению с ячейками ОЗУ.

Применение СОЗУ с ассоциативным доступом (АСОЗУ) позволяет автоматизировать процесс размещения данных в СОЗУ, обеспечивая «подмену» активных в данный момент ячеек ОЗУ ячейками СОЗУ. Эффективность такого подхода существенно зависит от выбранной стратегии замены информации в СОЗУ, причем использование ассоциативного СОЗУ имеет смысл только при условии  $T_c \ll T_0$ .

Принцип ассоциативного доступа состоит в следующем. Накопитель ассоциативного запоминающего устройства (АЗУ) разбит на два поля — информационное и признаков.

Структура информационного поля накопителя соответствует структуре обычного ОЗУ, а запоминающий элемент поля признаков, помимо функции записи, чтения и хранения бита, обеспечивает сравнение хранимой информации с поступающей и выдачу признака равенства.

Признаки равенства всех элементов одной ячейки поля признаков объединяются по «И» и устанавливаются в 1 индикатор совпадения ИС, если информация, хранимая в поле признака ячейки, совпадает с информацией, подаваемой в качестве признака на вход  $P$  накопителя.

Во второй фазе обращения (при чтении) на выход данных  $D$  последовательно поступает содержимое информационных полей тех ячеек, индикаторы совпадения которых установлены в 1 (если такие найдутся).

Способ использования АЗУ в качестве сверхоперативного иллюстрирует рис. 3.2. В информационном поле ячеек АСОЗУ — копия информации некоторых ячеек ОЗУ, а в поле признаков — адреса этих ячеек ОЗУ. Когда процессор генерирует обращение к ОЗУ, он одновременно (или прежде) инициирует процедуру опроса АСОЗУ, выдавая в качестве признака адрес ОЗУ.



Рис.3.2. СОЗУ с ассоциативным доступом.

Если имеет место совпадение признака ячейки с запрограммированным адресом (не более одного раза, алгоритм загрузки АСОЗУ не предусматривает возможности появления одинаковых признаков), то процессор обращается (по чтению или по записи) в информаци-

онное поле этой ячейки АСОЗУ, при этом блокируется обращение к ОЗУ. Если требуемый адрес не найден в АСОЗУ, инициируется (или продолжается) обращение к ОЗУ, причем в АСОЗУ создается копия ячейки ОЗУ, к которой обратился процессор. Повторное обращение процессора по этому адресу будет реализовано в АСОЗУ (на порядок быстрее, чем в ОЗУ).

Таким образом, в АСОЗУ создаются копии тех ячеек ОЗУ, к которым в данный момент обращается процессор в надежде, что «в ближайшее время» произойдет новое обращение по этому адресу.

Наличие АСОЗУ в ЭВМ позволяет (при достаточном его объеме и правильно выбранной стратегии загрузки) значительно увеличить производительность системы. При этом наличие или отсутствие АСОЗУ никак не отражается на построении программы. АСОЗУ не является программно-доступным объектом, оно скрыто от пользователя. Недаром в литературе для обозначения АСОЗУ часто используется термин «кэш-память» (cache — тайник).

### 3.3. Модули основной памяти

Упрощенная структурная схема модуля основной памяти при матричной его организации представлена на рис. 3.3.

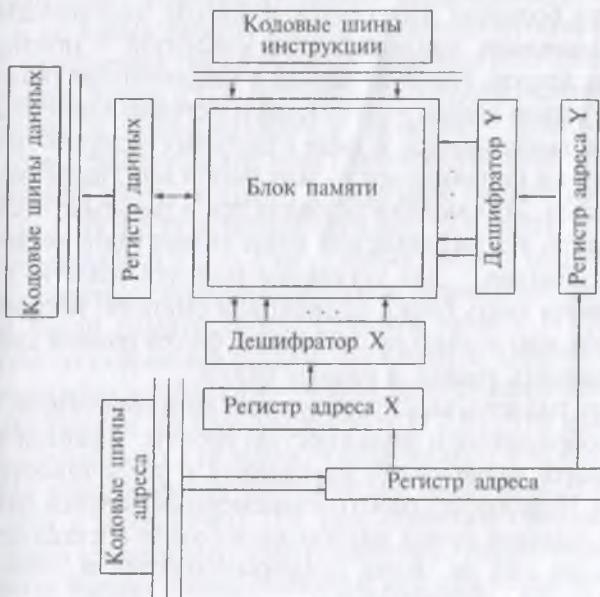


Рис.3.3. Структурная схема модуля основной памяти.

При матричной организации адрес ячейки, поступающий в регистр адреса, например по 20-разрядным кодовым шинам адреса, делится на две 10-разрядные части, поступающие, соответственно, в *регистр адреса X* и *регистр адреса Y*. Из этих регистров коды полуадресов поступают в *десифратор X* и *десифратор Y*, каждый из которых в соответствии с полученным адресом выбирает одну из 1024 шин. По выбранным шинам подаются сигналы записи-считывания в ячейку памяти, находящуюся на пересечении этих шин. Таким образом адресуется  $10^6$  (точнее,  $1024^2$ ) ячеек.

Считываемая или записываемая информация поступает в *регистр данных*, непосредственно связанный с кодовыми шинами данных. Управляющие сигналы, определяющие, какую операцию следует выполнить, поступают по кодовым шинам инструкций. *Куб памяти* содержит набор запоминающих элементов — собственно ячеек памяти.

**Основная память** (ОП) содержит оперативное (RAM — Random Access Memory) и постоянное (ROM — Read Only Memory) запоминающие устройства. **Оперативное запоминающее устройство** (ОЗУ) предназначено для хранения информации (программ и данных), непосредственно участвующей в вычислительном процессе в текущий интервал времени. ОЗУ — энергозависимая память: при отключении напряжения питания информация, хранящаяся в ней, теряется.

Основу ОЗУ составляют микросхемы динамической памяти DRAM. Это большие интегральные схемы, содержащие матрицы полупроводниковых запоминающих элементов — полупроводниковых конденсаторов. Наличие заряда в конденсаторе обычно означает «1», отсутствие заряда — «0». Конструктивно элементы оперативной памяти выполняются в виде отдельных модулей памяти — небольших плат с напаянными на них одной или, чаще, несколькими микросхемами. Эти модули вставляются в разъемы — слоты на системной плате. На материнской плате может быть несколько групп разъемов — банков — для установки модулей памяти; в один банк можно ставить лишь блоки одинаковой емкости, например, только по 16 Мбайт или только по 64 Мбайт; блоки разной емкости можно устанавливать только в разные банки.

Модули памяти характеризуются конструктивом, емкостью временем обращения и надежностью работы. Важным параметром модуля памяти является его надежность и устойчивость к возможным сбоям. Надежность работы современных модулей памяти весьма высокая — среднее время наработки на отказ составляет сотни тысяч часов, но тем не менее предпринимаются и дополнительные меры повышения надежности.

**Постоянное запоминающее устройство** (ПЗУ, или ROM — Read Only Memory, память только для чтения) также строится на основ-

установленных на материнской плате модулей (кассет) и используется для хранения неизменяемой информации: загрузочных программ операционной системы, программ тестирования устройств компьютера и некоторых драйверов базовой системы ввода-вывода (BIOS) и т. д.

К ПЗУ принято относить энергонезависимые постоянные и «полупостоянные» запоминающие устройства, из которых оперативно можно только считывать информацию, запись информации в ПЗУ выполняется вне ПК в лабораторных условиях или при наличии специального программатора и в компьютере. По технологии записи информации можно выделить ПЗУ следующих типов:

- микросхемы, программируемые только при изготовлении, — классические или масочные ПЗУ или ROM;
- микросхемы, программируемые однократно в лабораторных условиях, — программируемые ПЗУ (ППЗУ), или programmable ROM (PROM);
- микросхемы, программируемые многократно, — перепрограммируемые ПЗУ, или erasable PROM (EPROM). Среди них следует отметить электрически перепрограммируемые микросхемы EEPROM (Electrical Erasable PROM), в том числе **флеш-память**.

Устанавливаемые на системной плате ПК модули и кассеты ПЗУ имеют емкость, как правило, не превышающую 128 Кбайт. Быстродействие у постоянной памяти меньшее, чем у оперативной, поэтому для повышения производительности содержимое ПЗУ копируется в ОЗУ, и во время работы непосредственно используется только эта копия, называемая также *теневой памятью* ПЗУ (Shadow ROM).

В настоящее время в ПК используются «полупостоянные», перепрограммируемые запоминающие устройства — флеш-память. Модули или карты флеш-памяти могут устанавливаться прямо в разъемы материнской платы и имеют следующие параметры: емкость до 512 Мбайт (в ПЗУ BIOS используются до 128 Кбайт), время обращения по считыванию 0,035-0,2 мкс, время записи одного байта 2-10 мкс. Флеш-память — энергонезависимое запоминающее устройство. Примером такой памяти может служить память NVRAM — Non Volatile RAM со скоростью записи 500 Кбайт/с. Обычно для перезаписи информации необходимо подать на специальный вход флеш-памяти напряжение программирования (12 В), что исключает возможность случайного стирания информации. Перепрограммирование флеш-памяти может выполняться непосредственно с гибкого диска или с клавиатуры ПК при наличии специального контроллера либо с внешнего программатора, подключаемого к ПК. Флеш-память бывает очень полезной как для создания весьма быстродействующих компактных альтернативных НМД запоминаю-

щих устройств — «твердотельных дисков», так и для замены ПЗУ, хранящего программы BIOS, позволяя прямо с «дискеты» обновлять и заменять эти программы на более новые версии при модернизации ПК.

### 3.4. Внешние запоминающие устройства

Устройства внешней памяти, или, иначе, *внешние запоминающие устройства* (ВЗУ), весьма разнообразны. Их можно классифицировать по целому ряду признаков: по виду носителя, по типу конструкции, по принципу записи и считывания информации, по методу доступа и т. д. При этом под *носителем* понимается материальный объект, способный хранить информацию.

Один из возможных вариантов классификации ВЗУ приведен на рис. 3.4.

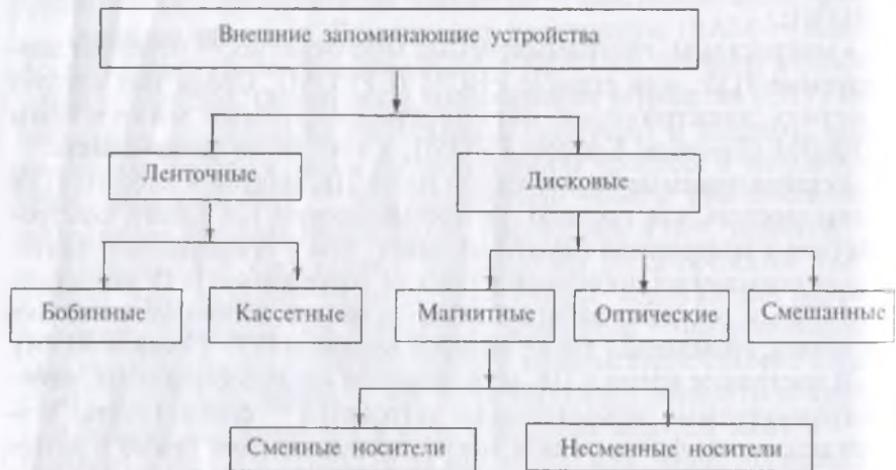


Рис.3.4. Классификация внешних запоминающих устройств.

В зависимости от типа носителя все ВЗУ можно подразделить на накопители на магнитной ленте и дисковые накопители.

Накопители на магнитной ленте, в свою очередь, бывают двух видов: накопители на бобинной ленте и накопители на кассетной ленте (стримеры). В ПК используются только стримеры.

Накопители на дисках более разнообразны (таблица 3.1):

- накопители на гибких магнитных дисках (*НГМД*) — накопители на флоппи-дисках, или дискетах;
- накопители на жестких магнитных дисках (*НЖМД*) — винчестеры;
- накопители на сменных жестких магнитных дисках, использующие эффект Бернулли;

- накопители на флоптических дисках – *floptical-накопители*;
- накопители сверхвысокой плотности записи (Very High Density) – *VHD-накопители*;
- накопители на оптических компакт-дисках (Compact Disk ROM) – *CD-ROM*;
- накопители на оптических дисках с однократной записью и многократным чтением (Continuous Composite Write Once, Read Many) – *CC WORM*;
- накопители на магнитооптических дисках – *НМОД*;
- накопители на цифровых видеодисках (Digital Versatile Disk) – *DVD* и др.

Таблица 3.1.

### Сравнительные характеристики дисковых накопителей

Тип накопителя	Емкость, Мбайт	Время доступа <sup>1</sup> , мс	Трансфер <sup>2</sup> , Кбайт/с	Вид доступа
НГМД	1,2; 1,44	65-100	55-150	Чтение-запись
Жесткий диск	1000-300000	5-30	500-6000	Чтение-запись
Бернумли	20-230	20	500-2000	Чтение-запись
Floptical	20-120	65	100-1000	Чтение-запись
VHD	120-240	65	200-1000	Чтение-запись
DVD	4700-17000	150-200	1380	Чтение-запись
CD-ROM	250-1500	50-300	150-3000	Чтение
CD-RW	120-1000	50-150	300-6000	Чтение-запись
НМОД	128-2600	50-150	300-6000	Чтение-запись
Flash	32-4000	$10^{-4}$	512-80000	Чтение-запись

<sup>1</sup> Время доступа – средний временной интервал, в течение которого накопитель находит требуемые данные.

<sup>2</sup> Трансфер – скорость передачи данных при последовательном чтении.

Магнитные диски (МД) относятся к магнитным машинным носителям информации. В качестве запоминающей среды в них используются магнитные материалы со специальным свойством (прямоугольная петля гистерезиса), позволяющим фиксировать два направления намагниченности, которым ставятся в соответствие двоичные цифры 0 и 1.

Накопители на МД (*НМД*) являются наиболее распространенными внешними запоминающими устройствами в ПК. Они бывают жесткими и гибкими, сменными и встроенными в ПК. Все диски, и магнитные, и оптические, характеризуются своим диамет-

ром, или, иначе, *форм-фактором*. Наибольшее распространение получили диски с форм-фактором 3,5 дюйма (89 мм). Но существуют диски и с форм-факторами 5,25 дюйма (133 мм), 2,5 дюйма (64 мм), 1,8 дюйма (45 мм) и др.

Информация на магнитные диски записывается и считывается магнитными головками вдоль концентрических окружностей — дорожек (треков). Количество дорожек на МД и их информационная емкость зависят от типа МД, конструкции накопителя на МД, качества магнитных головок и магнитного покрытия. Совокупность дорожек МД, находящихся на разных пластинах-дисках и на одинаковом расстоянии от его центра, называется *цилиндром*. При записи и чтении информации МД вращается вокруг своей оси, а механизм управления магнитной головкой подводит ее к дорожке, выбранной для записи или чтения информации. Устройство для чтения и записи информации на магнитном диске называется *дисководом*.

Кроме основной своей характеристики — *информационной емкости* — дисковые накопители характеризуются и двумя другими показателями:

- временем доступа;
- скоростью считывания последовательно расположенных байтов.

*Время доступа* (access time) к информации на диске, то есть время, которое дисковод тратит до начала чтения-записи данных, складывается из нескольких составляющих:

- времени перемещения магнитной головки на нужную дорожку (seek time);
- времени установки головки и затухания ее колебаний (setting time);
- времени ожидания вращения (rotation latency) — ожидания момента, когда из-за вращения диска нужный сектор окажется под головкой.

Диски относятся к машинным носителям информации с прямым доступом. Понятие *прямой доступ* применительно к диску означает, что ПК может «обратиться» к дорожке, на которой начинается участок с искомой информацией или куда нужно записать новую информацию, непосредственно, где бы ни находилась головка записи-чтения накопителя. После доступа к информации происходит ее последовательное считывание — хорошие дисководы обеспечивают *скорость считывания* (transfer rate) 2 Мбайт/с и выше.

Рассматривая организацию данных на внешних носителях, следует различать *физическую* и *логическую* организацию. *Физическая организация* определяет правила размещения данных на внешних носителях, *логическая* — описывает взаимные связи между данными и способом

бы доступа к ним. Поскольку при работе на компьютере пользователь практически взаимодействует только с *файлами*, необходимо подробнее познакомиться с организацией файловой системы.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каковы причины иерархического построения запоминающих устройств в ЭВМ?
2. Обоснуйте использование сверхоперативных запоминающих устройств (кэш-памяти).
3. Какие виды памяти может включать в своей структуре основная память ЭВМ?
4. Определите назначение оперативного запоминающего устройства.
5. Приведите классификацию постоянных запоминающих устройств по технологии записи информации.
6. Что вы понимаете под термином «модуль памяти»?
7. Как осуществляется перепрограммирование флеш-памяти?
8. Приведите классификацию внешних запоминающих устройств.
9. Дайте краткую характеристику CD и DVD-накопителям на оптических дисках.

## ГЛАВА 4. ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА И ПРЕРЫВАНИЙ

### 4.1. Организация ввода-вывода в ЭВМ

Вычислительная машина содержит помимо процессора (процессоров) и основной памяти, образующих ее ядро, многочисленные и разнообразные по выполняемым функциям и принципам действия *внешние устройства* (ВУ), предназначенные для хранения больших объемов информации (*внешние запоминающие устройства*) и для ввода в ЭВМ и вывода из нее информации, в том числе для ее регистрации и отображения (*устройства ввода-вывода*).

Передача информации с внешнего устройства в ядро ЭВМ (память и процессор) называется *операцией ввода*, а передача из ядра ЭВМ в периферийное устройство – *операцией вывода*.

Производительность и эффективность использования ЭВМ определяются не только возможностями ее процессора и характеристиками основной памяти, но в очень большой степени составом ее ВУ, их техническими данными и способом организации их совместной работы с ядром (процессором и основной памятью) ЭВМ.

В системах ввода-вывода используются два основных способа организации передачи данных между памятью и периферийными устройствами: программно-управляемая передача и прямой доступ к памяти (ПДП).

Программно-управляемая передача данных (рис. 4.1) осуществляется при непосредственном участии и под управлением процессора, который при этом выполняет специальную подпрограмму ввода-вывода. Данные между памятью и периферийным устройством пересылаются через процессор. Операция ввода-вывода инициируется текущей командой программы или запросом прерывания от периферийного устройства.

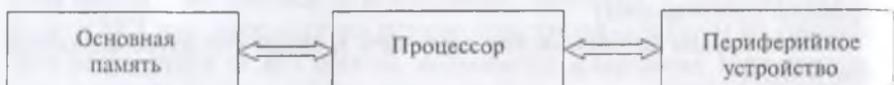


Рис. 4.1. Программно-управляемая передача данных.

При программно-управляемой передаче данных процессор на все время этой операции отвлекается от выполнения основной программы решения задачи. Операция пересылки данных логически слишком проста, чтобы эффективно загружать логически сложную быстродействующую аппаратуру процессора. В результате при использовании программно-управляемой передачи данных снижается производительность вычислительной машины.

Прямыми доступом к памяти называется способ обмена данными, обеспечивающий автономно от процессора установление связи и передачу данных между ОП и ПУ (рис. 4.2).

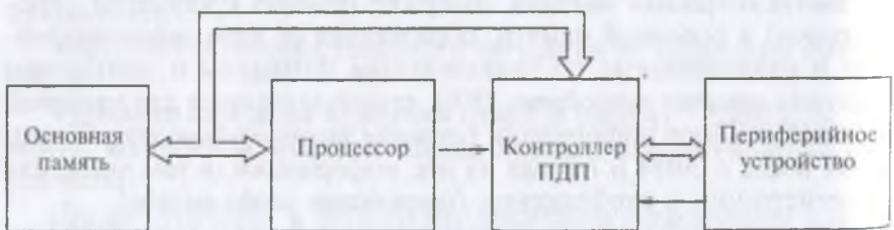


Рис. 4.2. Прямой доступ к памяти при использовании процессором и ПДП одной шины для связи с памятью.

Прямой доступ к памяти освобождает процессор или микропроцессор от управления операциями ввода-вывода, позволяет осуществлять параллельно во времени выполнение процессором (микропроцессором) программы с обменом данными между периферийным устройством и ОП, производить этот обмен со скоростью, ограниченной только пропускной способностью ОП или ПУ. Таким образом, ПДП, разгружая процессор (микропроцессор) от обслуживания операций ввода-вывода, способствует возрастанию

общей производительности ЭВМ или микроЭВМ. Повышение предельной скорости ввода-вывода информации делает машину более приспособленной для работы в системах реального времени. Прямым доступом к памяти управляет *контроллер ПДП*.

## 4.2. Внешние устройства ЭВМ

**Клавиатура** — важнейшее для пользователя устройство, с помощью которого осуществляется ввод данных, команд и управляющих воздействий в ПК. На клавиших нанесены буквы латинского и национального алфавитов, десятичные цифры, математические, графические и специальные служебные символы, знаки препинания, наименования некоторых команд, функций и т. д. В зависимости от типа ПК назначение клавиш, их обозначение и размещение может варьироваться. Чаще всего клавиатура содержит 101 клавишу, но встречаются еще и старые клавиатуры с 84 клавишами и новые, удобные для использования в системе Windows клавиатуры с 104 клавишами. Имеются клавиатуры с встроенными манипуляторами типа трекбол (trackball) и т. д. Появилось сообщение фирмы Data Hand Systems о разработке эргономичной, сокращающей движения руки 5-клавишной клавиатуры: 4 клавиши для ввода букв и цифр и 1 клавиша манипулятора. Каждая клавиша имеет 5 направлений движения: влево, вправо, вперед, назад и вниз. При работе кисть руки удобно лежит в специальном углублении, а клавишами управляют лишь кончики пальцев.

Типичная адаптированная под русский алфавит клавиатура IBM PC, содержащая 101 клавишу, показана на рис. 4.3.

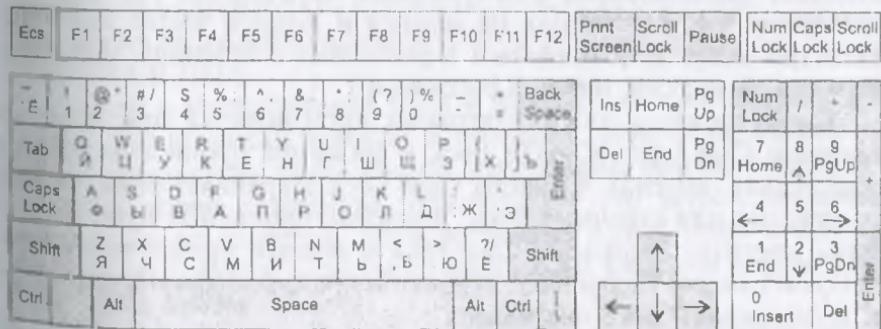


Рис.4.3. Адаптированная под русский алфавит клавиатура IBM PC.

Все клавиши можно разбить на следующие группы:

- буквенно-цифровые клавиши, предназначенные для ввода текстов и чисел;

- клавиши управления курсором, эта группа клавиш может быть использована также для ввода числовых данных, просмотра и редактирования текста на экране;
- специальные управляющие клавиши: переключение регистров, прерывание работы программы, вывод содержимого экрана на печать, перезагрузка ПК и т. д.;
- функциональные клавиши, широко используемые в сервисных программах в качестве управляющих клавиш.

Буквенно-цифровые клавиши занимают центральную часть клавиатуры. Расположение букв и цифр на клавишиах соответствует расположению их на клавиатуре пишущей машинки. Латинские буквы на клавиатуре расположены по стандарту **QWERTY**, названному так по последовательности первых шести букв в **верхнем** ряду буквенно-цифровых клавиш соответствует расположению **клавиш** на пишущих машинках с русским шрифтом – стандарт **ЙЦУКЕН** (первые шесть букв в **верхнем** ряду буквенно-цифровой клавиатуры). Для обеспечения ввода с клавиатуры русских букв необходим соответствующий драйвер, который должен быть предварительно загружен в оперативную память и оставаться в ней резидентно. **Переключение** клавиатуры в режим ввода русских букв (символов кириллицы) и **обратный** переход на ввод латинских букв осуществляется **нажатием** одной или двух специальных клавиш: для разных драйверов **по-разному**, но чаще всего **Ctrl** или **Shift**. Для алфавитно-цифровых клавиш существует понятие регистра, то есть режима их **использования**. Имеются две пары регистров: **верхний/нижний** и **латиница/кириллица**. В **верхнем** регистре вводятся прописные (заглавные) буквы, а в **нижнем** – строчные (маленькие); а также специальные **символы** и цифры, помещенные, соответственно, на верхней и нижней **частях** клавиши. Регистры могут использоваться в различных **сочетаниях**, например **верхний латинский, нижний русский** и т. п.

Выбор режима **нижний/верхний** производится при помощи клавиши **Caps Lock** (**Capitals Lock** – фиксация прописных букв) и **Shift** (сдвиг, замена). Клавиша **Caps Lock** закрепляет режим ввода прописных или строчных букв. В режиме прописных букв («**верхний**») светится индикатор **Caps Lock** в верхней **правой** части **клавишной панели**. Клавиша **Shift** изменяет режим **клавиатуры** на **противоположный**, пока она нажата.

Клавиши управления курсором расположены **в** правой **части** панели клавиатуры. Для удобства работы они **продублированы** и состоят из трех групп:

- малая цифровая клавиатура;
- клавиши просмотра текста на экране и его **редактирования**;
- клавиши управления курсором.

Клавиши малой цифровой клавиатуры могут быть использованы в двух режимах:

- в режиме управления курсором;
- в режиме ввода цифр, знаков математических операций и точки.

Выбор режима производится при помощи клавиши Num Lock (Number Lock – фиксация цифр) и Shift. Клавиша Num Lock закрепляет режим ввода цифр, а Shift изменяет режим клавиатуры на противоположный, пока она нажата.

**Мышь (mouse)** представляет собой электронно-механическое или электронное устройство, с помощью которого осуществляется дистанционное управление курсором на экране монитора. При перемещении манипулятора типа мышь по столу или другой поверхности на экране монитора соответствующим образом передвигается и курсор. Принцип работы электромеханической мыши основан на преобразовании вращательного движения шарика по двум осям через оптический или электрический конвертор в серию цифровых сигналов (импульсов), пропорциональных скорости передвижения.

Все большую популярность приобретают оптические мыши: у них отсутствует механическая часть, а «считывание» движения мыши по поверхности оптическое: в них оптический датчик анализирует отраженный от поверхности луч света и преобразует его параметры в последовательность импульсов. Мыши бывают двухкнопочные и трехкнопочные. Для большинства видов программ достаточно двух кнопок. Имеются мыши, специально ориентированные для работы в Интернете, с дополнительной третьей кнопкой (колесиком), применяемой для вертикального скроллинга (прокрутки) страницы в окне программы. В настоящее время выпускаются мыши с интерфейсами COM, PS/2, USB, Bluetooth и IrDA.

Мыши с интерфейсами IrDA и Bluetooth являются беспроводными: они не имеют «хвоста» и передают сигналы на подключенный к компьютеру приемник по оптическому или радиоканалу.

**Печатающие устройства** (принтеры) – это устройства вывода данных из компьютера, преобразующие ASCII-коды и битовые последовательности в соответствующие им символы и фиксирующие их на бумаге.

Принтеры являются наиболее развитой группой ВУ ПК, насчитывающей до 1000 различных модификаций. Принтеры различаются между собой:

- по цветности (черно-белые и цветные);
- по способу формирования символов (знакопечатающие и знакосинтезирующие);

- по принципу действия (матричные, струйные, лазерные, термические и др.);
- по способам печати (ударные, безударные) и формирования строк (последовательные, параллельные);
- по ширине каретки (широкой (375–450 мм) и узкой (250 мм));
- по длине печатной строки (80 и 132-136 символов);
- по набору символов;
- по скорости печати;
- по разрешающей способности и т. д.

Внутри ряда групп можно выделить по несколько разновидностей принтеров; например, широко применяемые матричные знакосинтезирующие принтеры по принципу действия могут быть ударными, термографическими, электрографическими, электростатическими, магнитографическими и т. д.; собственно говоря, и струйные принтеры также являются матричными.

Среди ударных принтеров наиболее распространены игольчатые (матричные), но в локальном варианте, без компьютера, еще встречаются и литерные, шаровидные, лепестковые (типа «ромашка») и т. д.

Печать у принтеров может быть посимвольная, построчная, постраничная. Скорость печати варьируется от 10-300 знаков/с (ударные принтеры) до 500-1000 знаков/с и даже до 20 страниц в минуту (безударные лазерные принтеры); разрешающая способность — от 3-5 до 30-40 точек на мм (лазерные принтеры). Принтеры могут работать в двух режимах — текстовом и графическом:

- в текстовом режиме на принтер посылаются коды символов, которые следует распечатать, причем контуры символов выбираются из знакогенератора принтера;
- в графическом режиме на принтер пересылаются коды, определяющие последовательность и местоположение точек изображения.

Для текстовой печати в общем случае имеются следующие режимы, характеризующиеся различным качеством печати:

- режим черновой печати (Draft);
- режим печати, близкий к типографскому (NLQ — Near Letter Quality);
- режим с типографским качеством печати (LQ — Letter Quality);
- сверхкачественный режим (SLQ — Super Letter Quality).

В текстовом режиме принтеры обычно поддерживают несколько шрифтов и их гарнитур, среди которых получили широкое распространение roman (мелкий шрифт пишущей машинки), italic (прямой курсив), bold-face (полужирный), expanded (растянутый), elite (полужирный), condensed (сжатый), pica (пика или цицеро — прямой шрифт кеглем 12 пунктов), courier (курьер), san-serif (рубле-

ный шрифт сенсериф), serif (сериф), prestige elite (престиж-элита) и пропорциональный шрифт (ширина поля, отводимого под символ, зависит от ширины символа).

Желательно, чтобы принтер был русифицированным, то есть своими средствами обеспечивал печать русских букв — кириллицы, в противном случае в текстовом режиме потребуется подключение в ПК специальных драйверов.

Многие принтеры позволяют реализовать:

- эффективный вывод графической информации (с помощью символов псевдографики);

- сервисные режимы печати: плотная печать, печать с двойной шириной, с подчеркиванием, с верхними и нижними индексами, выделенная печать (каждый символ печатается дважды) и печать за два прохода (второй раз символ печатается с незначительным сдвигом);

- многоцветную (до 100 различных цветов и оттенков) печать.

Основными характеристиками принтеров являются:

- Разрешающая способность, или просто разрешение. Разрешение при печати чаще всего измеряется числом элементарных точек (dots), которые размещаются на одном дюйме {dpi — dots per inch, inch — дюйм, примерно 2,54 см} или на одном сантиметре (точек на сантиметр бумаги). Например, разрешение 1440 dpi означает, что на длине одного дюйма бумаги размещается 1440 точек. Чем больше разрешение, тем точнее воспроизводятся детали изображения. Однако при этом соответственно возрастает и время печати (исключением являются лазерные принтеры).

- Скорость печати. Единицей измерения скорости печати информации служит величина количества символов в секунду — cps (characters per second), а при листовой печати показатель страниц в минуту — ppr (pages per minute). Как правило, ppr указывается для страниц формата А4.

**Сканер** — это устройство ввода в компьютер информации непосредственно с бумажного документа. Это могут быть тексты, схемы, рисунки, графики, фотографии и другая информация. Сканер, подобно копировальному аппарату, создает копию изображения бумажного документа, но не на бумаге, а в электронном виде — формируется электронная копия изображения.

Сканеры являются важнейшим звеном электронных систем обработки документов и необходимым элементом любого «электронного стола». Записывая результаты своей деятельности в файлы и вводя информацию с бумажных документов в ПК с помощью сканера, объединенного с системой автоматического распознавания образов, можно сделать реальный шаг к созданию систем безбумажного делопроизводства.

Сканеры весьма разнообразны, и их можно классифицировать по целому ряду признаков. Прежде всего, сканеры бывают черно-белые и цветные.

Черно-белые сканеры позволяют считывать штриховые и полуточные изображения. Штриховые изображения не передают полутона, или, иначе, градаций серого. Полуточные способны распознать и передать 16, 64 или 256 уровней серого цвета.

В цветных сканерах используется цветовая модель RGB (Red – Green – Blue): сканируемое изображение освещается через вращающийся RGB-светофильтр или последовательно зажигаемыми тремя цветными лампами. Сигнал, соответствующий каждому основному цвету, обрабатывается отдельно: для этого имеются параллельные линейки датчиков, каждая из которых воспринимает свой цвет.

Число передаваемых цветов колеблется от 256 до 65 536 (стандарт High Color), и даже до 16,7 млн (стандарт True Color).

Разрешающая способность сканеров измеряется в количестве различаемых точек на дюйм изображения и составляет от 300 до 9600 dpi.

Качество сканирования зависит от типа используемых в сканере датчиков – элементов, воспринимающих оптическое изображение оригинала.

По конструктивному исполнению сканеры делятся на ручные (hand-held) и настольные (desktop). Есть и комбинированные устройства, сочетающие в себе возможности обоих типов.

Ручные сканеры конструктивно самые простые – они состоят из линейки светодиодов и источника света, помещенных в единый корпус. Перемещение по изображению такого сканера выполняется вручную.

С их помощью за один проход вводится лишь небольшое количество строчек изображения (полоса захвата обычно не превышает 105 мм). У ручных сканеров имеется индикатор, предупреждающий оператора о превышении допустимой скорости сканирования. Эти сканеры имеют малые габариты и низкую стоимость. Скорость сканирования составляет 5-50 мм/с (зависит от разрешающей способности).

Например, ручные сканеры Mustek: GS-400L – черно-белый полуточновый, CG-8400T – цветной.

Планшетные сканеры – самые распространенные, в них сканирующая головка (линейка светодиодов) перемещается относительно оригинала автоматически; они позволяют обрабатывать и листовые, и сброшюрованные (книги) документы.

Принцип работы планшетных сканеров заключается в следующем. Сканируемый оригинал помещается на прозрачном неподвижном стекле. Вдоль него передвигается сканирующая головка с

источником света. Оптическая система планшетного сканера проецирует световой поток, отражаемый от сканируемого оригинала, на сканирующую головку, состоящую из трех параллельных линеек светочувствительных элементов (CCD-матрица). Каждая линейка принимает информацию о своем цвете — красном (Red), зеленом (Green), синем (Blue). В сканирующей головке уровни освещенности преобразуются в уровни напряжения (аналоговый сигнал). Затем, после коррекции и обработки, аналоговый сигнал преобразуется в цифровой с помощью аналого-цифрового преобразователя. Цифровой сигнал поступает в компьютер, где данные, соответствующие изображению оригинала, обрабатываются и преобразовываются под управлением драйвера сканера.

Скорость сканирования: 2-10 с на страницу (формат А4).

Примером могут служить цветные сканеры: Mustek Paragon 1200, Epson ES1200, HP ScanJet 5S и 5P, HP ScanJet 11CX (следует заметить, что подавляющее количество продаж сканеров на российском рынке в последние годы приходится на продукцию фирмы Hewlett-Packard).

Среди сканеров, работающих с документами большого формата, следует выделить популярные сканеры фирмы AGFA, например AGFA Argus II, имеющий физическое разрешение 600 x 1200 dpi (логическое разрешение при использовании интерполирующей технологии UltraView достигает 2400 x 2400 dpi), передающий 4096 оттенков и масштабирующий изображение в 7-9 раз.

Листовые сканеры (их также называют страничными, протяжными) наиболее автоматизированы; в них оригинал автоматически перемещается относительно неподвижной сканирующей головки, часто имеется автоматическая подача документов, но сканируемые документы только листовые.

Достоинствами листовых сканеров являются низкая стоимость и более высокое, по сравнению с ручными сканерами, качество получаемого изображения. К недостаткам относятся проблемы выравнивания листов и сложности работы с нестандартными листами.

Примером может служить сканер Mustek SF-630 со скоростью 10 страниц за секунду.

Барабанные сканеры чаще всего содержат один датчик, в качестве которого выступает фотоэлектронный умножитель (ФЭУ). Сканируемый оригинал закрепляется на поверхности прозрачного цилиндра, который вращается с большой скоростью. Датчик последовательно, пиксель за пиксельом считывает оригинал.

Достоинством барабанного сканера является самое высокое качество получаемого изображения, основным недостатком — высокая стоимость.

Проекционные сканеры внешне напоминают фотоувеличитель, но внизу лежит сканируемый документ, а наверху находится сканирующая головка. Сканер оптически снимает информацию с документа и вводит ее в память компьютера.

### 4.3. Организация прерываний

**Подсистема прерываний** – совокупность аппаратных и программных средств, обеспечивающих реакцию программы на события, происходящие вне программы. Такие события возникают, как правило, случайно и асинхронно по отношению к программе и требуют прекращения (чаще временного) выполнения текущей программы и переход на выполнение другой программы (подпрограммы), соответствующей возникшему событию.

Различают внутренние и внешние (по отношению к процессору) события, требующие реакции подсистемы прерываний. К внутренним событиям относятся переполнение разрядной сетки при выполнении арифметических операций, попытка деления на 0, извлечение корня четной степени из отрицательного числа, появление несуществующего кода команды, обращение программы в область памяти, для нее не предназначеннную, сбой при выполнении передачи данных или операции в АЛУ и многое другое. Внутренние прерывания должны обеспечиваться развитой системой аппаратного контроля процессора, поэтому они не получили широкого распространения в простых 8- и 16-разрядных МП.

Внешние прерывания могут возникать во внешней по отношению к процессору среде и отмечать как аварийные ситуации (кончилась бумага на принтере, температура в реакторе превысила допустимый уровень, исполнительный орган робота дошел до предельного положения и т. п.), так и нормальные рабочие события, которые происходят в случайные моменты времени (нажата клавиша, исчерпан буфер принтера или ВЗУ и т. п.). Во всех этих случаях требуется прервать выполнение текущей программы и перейти на выполнение другой программы (подпрограммы), обслуживающей данное событие.

С точки зрения реализации внутренние и внешние прерывания функционируют одинаковым образом, хотя при работе подсистемы с внешними прерываниями возникают дополнительные проблемы идентификации источника прерывания. Поэтому ниже остановимся на рассмотрении внешних прерываний.

Анализ состояния внешней среды можно осуществлять путем программного сканирования – считывания через определенные промежутки времени слов состояния всех возможных источников прерываний, выделения признаков отслеживаемых событий и переход (при необходимости) на прерывающую подпрограмму (часто называют *обработчиком прерывания*).

Однако такой способ не обеспечивает для большинства применений приемлемого времени реакции системы на события, особенно при необходимости отслеживания большого числа событий. К тому же при коротком цикле сканирования большой процент процессорного времени тратится на проверку (чаще безрезультатную) состояния внешней среды.

Гораздо эффективней организовать взаимодействие с внешней средой таким образом, чтобы всякое изменение состояния среды, требующее реакции МПС, вызывало появление на специальном входе МП сигнала прерывания текущей программы. Организация прерываний должна быть обеспечена определенными аппаратными и программными средствами, которые мы и называем *подсистемой прерываний*.

Подсистема прерываний должна обеспечивать выполнение следующих функций:

- обнаружение изменения состояния внешней среды (запрос на прерывание);
- идентификация источника прерывания;
- разрешение конфликтной ситуации в случае одновременного возникновения нескольких запросов (приоритет запросов);
- определение возможности прерывания текущей программы (приоритет программ);
- фиксация состояния прерываемой (текущей) программы;
- переход к программе, соответствующей обслуживаемому прерыванию;
- возврат к прерванной программе после окончания работы прерывающей программы.

Рассмотрим варианты реализации в МПС перечисленных выше функций.

**Обнаружение изменения состояния внешней среды.** Фиксация изменения состояния внешней среды может осуществляться различными схемами: двоичными датчиками, компараторами, схемами формирования состояний и др. Будем полагать, что все эти схемы формируют в конечном итоге логические сигналы запроса на прерывание, причем для определенности будем считать, что активное состояние этого сигнала передается уровнем логической единицы (Н-уровень).

Количество источников запросов в МПС может быть различно, и довольно велико. Дефицит внешних выводов МП в общем

исключает возможность передачи каждого запроса от ВУ по «собственной» линии интерфейса. Обычно на одну линию запроса подключается несколько источников прерываний (по функции ИЛИ), а иногда и все источники запросов – на единственный вход. Управляющий автомат процессора должен периодически анализировать состояние линии (линий) запросов на прерывания. Ка-

ким образом выбирается период проверки? С одной стороны, этот период должен быть коротким, чтобы обеспечить быструю реакцию системы на события. С другой стороны, при переходе на обслуживание прерывания требуется сохранить текущее состояние процессора на момент прерывания, с тем, чтобы, завершив программу «обработчик», продолжить выполнение прерванной программы «с того же места», на котором произошло прерывание.

Напомним, что в основе работы процессора лежит *командный цикл*, состоящий, в свою очередь, из *машинных циклов*, каждый из которых длится несколько *тактов*. Осуществлять прерывание в произвольном такте невозможно, т. к. при этом пришлось бы сохранять в качестве контекста прерванной программы состояние всех элементов памяти процессора.

Прерывание по завершению текущего машинного цикла требует сохранение текущего состояния незавершенной команды. Например, при возникновении прерывания в том месте командного цикла, когда из памяти выбраны код команды и первый операнд, а для второго операнда только сформирован исполнительный адрес следует сохранить: содержимое программного счетчика РС, код команды, первый операнд и адрес второго операнда. Сохранение этой информации требует как дополнительных аппаратных, так и временных затрат. Очевидно, при возврате к прерванной программе придется начать выполнение текущей команды заново. В этом случае сохранять при прерывании достаточно лишь значение РС. Поэтому в большинстве случаев процессоры анализируют состояния линий запросов в конце каждого командного цикла.

**Идентификация источника прерывания.** Различают два типа входов запросов на прерывания – *радиальные* и *векторные*. Получив запрос на прерывание, процессор должен идентифицировать его источник, т. е. в конечном счете определить начальный адрес обрабатывающей это прерывание программы. Способ идентификации зависит от типа входа, на который поступил запрос.

Каждый радиальный вход связан с определенным адресом памяти, по которому размещается указатель на обрабатывающую программу или сама программа. Если радиальный вход связан с несколькими источниками запросов, то необходимо осуществить программную идентификацию путем последовательного (в порядке убывания приоритетов) опроса всех связанных с этим входом источников прерывания.

Этот способ не требует дополнительных аппаратных затрат и одновременно решает проблему приоритета запросов, однако время реакции системы на запрос может оказаться недопустимо большим, особенно при большом числе источников прерываний.

Гораздо чаще в современных МПС используется т. н. *векторная подсистема прерываний*. В такой системе микропроцессор, получив запрос на векторном входе INT, выдает на свою выходную линию сигнал подтверждения прерывания INTA, поступающий на все возможные источники прерывания. Источник, не выставивший запроса, никак не реагирует на сигнал INTA. Источник, выставивший запрос, получая сигнал INTA, выдает на системную шину данных «вектор прерывания» – свой номер или адрес обслуживающей программы, или, чаще, адрес памяти, по которому расположен указатель на обслуживающую программу. Время реакции МПС на запрос векторного прерывания минимально (1 – 3 машинных цикла) и не зависит от числа источников.

**Приоритет запросов.** Для исключения конфликтов при одновременном возникновении нескольких запросов на векторном входе ответный сигнал INTA подается на источники запросов не параллельно, а последовательно – в порядке убывания приоритетов запросов. Источник, не выставивший запроса, транслирует сигнал INTA со своего входа на выход, а источник, выставивший запрос, блокирует дальнейшее распространение сигнала INTA. Таким образом, только один источник, выставивший запрос, получит от процессора сигнал INTA и выдаст по нему свой вектор на шину данных.

Более гибко решается проблема организации приоритетов запросов при использовании в МПС специальных контроллеров прерываний.

Конфликты на радиальном входе исключаются самим порядком программного опроса источников.

**Приоритет программ.** Прерывание в общем случае может возникать не только при решении «фоновой» задачи, но и в момент работы другой прерывающей программы, причем не всякую прерывающую программу допустимо прерывать любым запросом. В фоновой задаче также могут встречаться участки, при работе которых прерывания (все или некоторые) недопустимы. В общем случае в каждый момент времени работы процессора должно быть выделено подмножество запросов, которым разрешено прерывать текущую программу.

В МПС эта задача решается на нескольких уровнях. В процессоре обычно предусматривается программно-доступный флаг разрешения/запрещения прерывания, значение которого определяет возможность или невозможность всех прерываний. Для создания более гибкой системы приоритетов программ на каждом источнике прерываний может быть предусмотрен специальный программно-доступный триггер разрешения формирования запроса. В таком случае возможно формирование произвольного подмножества разрешенных в данный момент источников прерываний.

При использовании контроллера внешних прерываний, в нем обычно предусматривают специальный программно-доступный регистр, разряды которого *маскируют* соответствующие линии запросов на прерывание, запрещая контроллеру вырабатывать сигнал прерывания процессору, если запросы от ВУ поступают по замаскированным линиям. Однако замаскированные запросы сохраняются в контроллере и в дальнейшем, при изменении состояния регистра маски, могут быть переданы на обслуживание.

**Обработка прерывания.** К обработке прерывания отнесем фиксацию состояния прерываемой программы, переход к программе, соответствующей обслуживаемому прерыванию, и возврат к прерванной программе после окончания работы прерывающей программы.

Выше мы определили, что большинство процессоров может прервать выполнение текущей программы и переключиться на реализацию обработчика прерывания только после завершения очередной команды. При этом в качестве контекста прерванной программы необходимо сохранить текущее состояние счетчика команд РС, а в РС загрузить новое значение – адрес программы обработчика прерывания. Очевидно, адрес возврата в прерванную программу (содержимое РС на момент прерывания) следует размещать в стеке, что позволит при необходимости осуществлять вложенные прерывания (когда в процессе обслуживания одного прерывания получен запрос на обслуживание другого).

Можно вспомнить, что подобный механизм реализован в системах команд многих процессоров для выполнения команд вызова подпрограммы (сил, JSR). В этих командах адрес вызываемой подпрограммы содержится в коде команды.

В случае вызова обработчика прерывания его адрес необходимо связать либо со входом, на который поступил запрос (радиальные прерывания), либо с номером источника прерываний, сформировавшего запрос (векторные прерывания). В первом случае не требуется никаких внешних процедур для идентификации источника. сразу можно запускать связанный со входом обработчик. Понятно, здесь идет речь об отсутствии необходимости в аппаратных процедурах идентификации источника запроса. Если на радиальный вход «работают» несколько источников, то выбор осуществляется программными способами.

В случае векторных прерываний адрес перехода связывают с информацией, поступающей от источника запроса по шине данных в машинном цикле обслуживания прерывания – *вектором прерывания*.

Напомним, что любой командный цикл процессора начинается с чтения команды из памяти. В первом машинном цикле командного цикла процессор выдает на шину адреса содержимое РС, фор-

мирует управляющий сигнал RDM и помещенное памятью на шину данных слово интерпретирует как команду (или ее начальную часть, если длина команды превышает длину машинного слова).

Если в конце очередного командного цикла процессор обнаруживает (не замаскированный) запрос на векторном входе, он начинает следующий командный цикл с небольшими изменениями: содержимое РС по-прежнему выдается на шину адреса (чтобы не нарушать общности цикла), но вместо сигнала RDM формирует сигнал INTA. Источник запроса (чаще – контроллер прерываний) в ответ на сигнал INTA формирует на шину данных код команды вызова подпрограммы, в адресной части которой размещается адрес обработчика соответствующего прерывания.

Такой простой способ реализации векторных прерываний, с использованием уже существующего механизма вызова подпрограмм, был реализован, например, в микропроцессоре i8080 с контроллером прерываний i8259. Однако этот механизм, как, впрочем, и все остальное, допускает дальнейшее совершенствование.

Прежде всего, желание иметь возможность располагать подпрограммы в произвольной области памяти приводит к необходимости размещать в поле адреса команды вызова полноразрядный адрес (16 – 20 – 32 бита). В этом случае длина команды превышает длину машинного слова и ее ввод требует нескольких машинных циклов (например, в i8080 – трех), что увеличивает время реакции системы на запрос прерывания.

Для преодоления этого недостатка в систему команд процессора включают дополнительно «укороченные» команды вызова длиной в одно машинное слово. Эти команды в процессорах 8080 и x86 имеют мнемокод INT. В микропроцессоре i8080 имеется 8 таких команд длиной в 1 байт, адресующих под программы по фиксированным адресам памяти: 0000h, 0008h, 0010h, ..., 003811.

В процессорах x86 имеется 256 вариантов двухбайтовых команд INT INT 00h, ...INT FFh, байт поля адреса которых (называемый вектором) после умножения на 4 указывает на четырехбайтовую структуру, определяющую произвольный адрес в адресном пространстве памяти.

Напомним, что доступ в память процессоров x86 (в реальном режиме) осуществляется только в рамках сегментов размером в 64 Кбайт. Положение начала сегмента в адресном пространстве памяти определяется содержимым 16-разрядного сегментного регистра, а положение адресуемого байта внутри сегмента – 16-разрядным смещением. Среди команд передачи управления различают *короткие* и *длинные переходы* (вызовы). При коротком вызове подпрограмма должна располагаться в текущем сегменте кода, и ее вызов сопровождается только изменением счетчика команд (в x86 он обо-

значается, как IP). При длинном вызове новое значение загружается как в IP, так и в сегментный регистр кода CS. Таким образом, для осуществления длинного вызова (перехода) в адресном поле команды необходимо разместить 4 байта.

Механизм векторных прерываний в процессорах x86 в реальном режиме реализован следующим образом. В начальных адресах 00000h, ..., 003FFh пространства памяти размещается таблица векторов прерываний объемом 1 Кбайт, включающая 256 строк таблицы — четырехбайтовых структур CS:IP, которые определяют адреса соответствующих обработчиков прерываний. В цикле обработки векторного прерывания (запрос по входу INT) процессор получает от источника байт — номер строки таблицы векторов прерываний, из которой и загружаются новые значения CS и IP. Старые значения CS:IP (адрес возврата) размещаются в стеке.

Запросу по радиальному входу NMI соответствует вектор 2, поэтому появление активного значения не вызывает машинного цикла обслуживания прерывания, а сразу вызывается обработчик по адресу из ячеек памяти 00008h, ..., 0000Bh. Кстати, любой обработчик прерывания (независимо от значения маскирующих флагов) можно вызвать программно с помощью команды  $INT_{nn}$ , где nn — номер строки таблицы векторов прерываний.

Таким образом, команда INT отличается от команды *CALL*, во-первых, способом адресации вызываемой подпрограммы (прямой адрес — в команде *CALL*, косвенный — в INT), во-вторых, при реализации INT в стек, помимо CS и IP) помещается содержимое регистра признаков процессора — FLAGS. Соответственно, завершаться подпрограмма, вызываемая командой *IRET*, должна командой *INT* («возврат из прерывания»). Действие *IRET* отличается от действия от извлечением из стека дополнительного слова в регистр FLAGS.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Расскажите о влиянии методов организации устройств ввода-вывода на эффективность использования ЭВМ.
2. За счет чего повышается быстродействие ЭВМ при использовании прямого доступа к памяти машины?
3. Перечислите группы клавиш на клавиатуре ЭВМ типа PC.
4. Охарактеризуйте типы компьютерных мышек и их задачи.
5. Дайте классификацию основных типов принтеров.
6. Дайте классификацию основных типов сканеров.
7. Что вы понимаете под определением «прерывание» программ?
8. Перечислите причины появления прерываний программ.
9. Охарактеризуйте задачи системы прерываний.
10. Дайте характеристики и особенности радиального и векторного способов прерываний.

## ГЛАВА 5. ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ. ОСНОВНЫЕ СВЕДЕНИЯ

### 5.1. Вычислительные сети — частный случай распределенных систем

Компьютерные сети относятся к распределенным (или децентрализованным) вычислительным системам. Поскольку основным признаком распределенной вычислительной системы является наличие нескольких центров обработки данных, то наряду с компьютерными сетями к распределенным системам относят также мультипроцессорные компьютеры и многомашинные вычислительные комплексы.

**Мультипроцессорные компьютеры.** В мультипроцессорных компьютерах имеется несколько процессоров, каждый из которых может относительно независимо от остальных выполнять свою программу. В мультипроцессоре существует общая для всех процессоров операционная система, которая оперативно распределяет вычислительную нагрузку между процессорами. Взаимодействие между отдельными процессорами организуется наиболее простым способом — через общую оперативную память.

Сам по себе процессорный блок не является законченным компьютером и поэтому не может выполнять программы без остальных блоков мультипроцессорного компьютера — памяти и периферийных устройств. Все периферийные устройства являются для всех процессоров мультипроцессорной системы общими. Территориальную распределенность мультипроцессор не поддерживает — все его блоки располагаются в одном или нескольких близко расположенных конструктивах, как и у обычного компьютера.

Основное достоинство мультипроцессора — его высокая производительность, которая достигается за счет параллельной работы нескольких процессоров. Так как при наличии общей памяти взаимодействие процессоров происходит очень быстро, мультипроцессоры могут эффективно выполнять даже приложения с высокой степенью связи по данным.

Еще одним важным свойством мультипроцессорных систем является отказоустойчивость, то есть способность к продолжению работы при отказах некоторых элементов, например процессоров или блоков памяти. При этом производительность, естественно, снижается, но не до нуля, как в обычных системах, в которых отсутствует избыточность.

**Многомашинные системы.** Многомашинная система — это вычислительный комплекс, включающий в себя несколько компьютеров (каждый из которых работает под управлением собственной

операционной системы), а также программные и аппаратные средства связи компьютеров, которые обеспечивают работу всех компьютеров комплекса как единого целого.

Работа любой многомашинной системы определяется двумя главными компонентами: высокоскоростным механизмом связи процессоров и системным программным обеспечением, которое предоставляет пользователям и приложениям прозрачный доступ к ресурсам всех компьютеров, входящих в комплекс. В состав средств связи входят программные модули, которые занимаются распределением вычислительной нагрузки, синхронизацией вычислений и реконфигурацией системы. Если происходит отказ одного из компьютеров комплекса, его задачи могут быть автоматически переназначены и выполнены на другом компьютере. Если в состав многомашинной системы входят несколько контроллеров внешних устройств, то в случае отказа одного из них другие контроллеры автоматически подхватывают его работу. Таким образом достигается высокая отказоустойчивость комплекса в целом.

Помимо повышения отказоустойчивости многомашинные системы позволяют достичь высокой производительности за счет организации параллельных вычислений. По сравнению с мультипроцессорными системами возможности параллельной обработки в многомашинных системах ограничены: эффективность распараллеливания резко снижается, если параллельно выполняемые задачи тесно связаны между собой по данным. Это объясняется тем, что связь между компьютерами многомашинной системы менее тесная, чем между процессорами в мультипроцессорной системе, так как основной обмен данными осуществляется через общие многовходовые периферийные устройства. Говорят, что в отличие от мультипроцессоров, где используются сильные программные и аппаратные связи, в многомашинных системах аппаратные и программные связи между обрабатывающими устройствами являются более слабыми. Территориальная распределенность в многомашинных комплексах не обеспечивается, так как расстояния между компьютерами определяются длиной связи между процессорным блоком и дисковой подсистемой.

**Вычислительные сети.** В вычислительных сетях программные и аппаратные связи являются еще более слабыми, а автономность обрабатывающих блоков проявляется в наибольшей степени — основными элементами сети являются стандартные компьютеры, не имеющие ни общих блоков памяти, ни общих периферийных устройств. Связь между компьютерами осуществляется с помощью специальных периферийных устройств — сетевых адаптеров, соединенных относительно протяженными каналами связи. Каждый компьютер

работает под управлением собственной операционной системы, а какая-либо «общая» операционная система, распределяющая работу между компьютерами сети, отсутствует. Взаимодействие между компьютерами сети происходит за счет передачи сообщений через сетевые адаптеры и каналы связи. С помощью этих сообщений один компьютер обычно запрашивает доступ к локальным ресурсам другого компьютера. Такими ресурсами могут быть как данные, хранящиеся на диске, так и разнообразные периферийные устройства — принтеры, модемы, факс-аппараты и т. д. Разделение локальных ресурсов каждого компьютера между всеми пользователями сети — основная цель создания вычислительной сети.

Каким же образом оказывается на пользователе тот факт, что его компьютер подключен к сети? Прежде всего, он может пользоваться не только файлами, дисками, принтерами и другими ресурсами своего компьютера, но и аналогичными ресурсами других компьютеров, подключенных к той же сети. Правда, для этого недостаточно снабдить компьютеры сетевыми адаптерами и соединить их кабельной системой. Необходимы еще некоторые добавления к операционным системам этих компьютеров. На тех компьютерах, ресурсы которых должны быть доступны всем пользователям сети, необходимо добавить модули, которые постоянно будут находиться в режиме ожидания запросов, поступающих по сети от других компьютеров. Обычно такие модули называются программными *серверами (server)*, так как их главная задача — обслуживать (serv) запросы на доступ к ресурсам своего компьютера. На компьютерах, пользователи которых хотят получать доступ к ресурсам других компьютеров, также нужно добавить к операционной системе некоторые специальные программные модули, которые должны вырабатывать запросы на доступ к удаленным ресурсам и передавать их по сети на нужный компьютер. Такие модули обычно называют программными *клиентами (client)*. Собственно же сетевые адаптеры и каналы связи решают в сети достаточно простую задачу — они передают сообщения с запросами и ответами от одного компьютера к другому, а основную работу по организации совместного использования ресурсов выполняют клиентские и серверные части операционных систем.

Пара модулей «клиент-сервер» обеспечивает совместный доступ пользователей к определенному типу ресурсов, например к файлам. В этом случае говорят, что пользователь имеет дело с *файловой службой (service)*. Обычно сетевая операционная система поддерживает несколько видов сетевых служб для своих пользователей — файловую службу, службу печати, службу электронной почты, службу удаленного доступа и т. п.

Термины «клиент» и «сервер» используются не только для обозначения программных модулей, но и компьютеров, подключенных к сети. Если компьютер предоставляет свои ресурсы другим компьютерам сети, то он называется сервером, а если он их потребляет — клиентом. Иногда один и тот же компьютер может одновременно играть роли и сервера, и клиента.

*Распределенные программы.* Сетевые службы всегда представляют собой распределенные программы. *Распределенная программа* — это программа, которая состоит из нескольких взаимодействующих частей (в приведенном на рис. 5.1 примере — из двух), причем каждая часть, как правило, выполняется на отдельном компьютере сети.

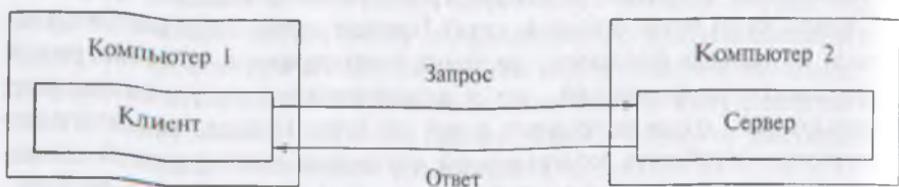


Рис. 5.1. Взаимодействие частей распределенной программы-приложения.

До сих пор речь шла о системных распределенных программах. Однако в сети могут выполняться и распределенные пользовательские программы — приложения. Распределенное приложение также состоит из нескольких частей, каждая из которых выполняет какую-то определенную законченную работу по решению прикладной задачи. Например, одна часть приложения, выполняющаяся на компьютере пользователя, может поддерживать специализированный графический интерфейс, вторая — работать на мощном выделенном компьютере и заниматься статистической обработкой введенных пользователем данных, а третья — заносить полученные результаты в базу данных на компьютере с установленной стандартной СУБД. Распределенные приложения в полной мере используют потенциальные возможности распределенной обработки, предоставляемые вычислительной сетью, и поэтому часто называются *сетевыми приложениями*.

Следует подчеркнуть, что не всякое приложение, выполняемое в сети, является сетевым. Существует большое количество популярных приложений, которые не являются распределенными и целиком выполняются на одном компьютере сети. Тем не менее и такие приложения могут использовать преимущества сети за счет встроенных в операционную систему сетевых служб. Значительная часть истории локальных сетей связана как раз с использованием таких нераспределенных приложений. Рассмотрим, например, как происходила работа

пользователя с известной в свое время СУБД dBase. Обычно файлы базы данных, с которыми работали все пользователи сети, располагались на файловом сервере. Сама же СУБД хранилась на каждом клиентском компьютере в виде единого программного модуля.

Программа dBase была рассчитана на обработку только локальных данных, то есть данных, расположенных на том же компьютере, что и сама программа. Пользователь запускал dBase на своем компьютере, и она искала данные на локальном диске, совершенно не принимая во внимание существование сети. Чтобы обрабатывать с помощью dBase данные на удаленном компьютере, пользователь обращался к услугам файловой службы, которая доставляла данные с сервера на клиентский компьютер и создавала для СУБД эффект их локального хранения.

Большинство приложений, используемых в локальных сетях в середине 80-х годов, являлись обычными, нераспределенными приложениями. И это понятно – они были написаны для автономных компьютеров, а потом просто были перенесены в сетевую среду. Создание же распределенных приложений, хотя и сулило много преимуществ (уменьшение сетевого трафика, специализация компьютеров), оказалось делом совсем не простым. Нужно было решать множество дополнительных проблем – на сколько частей разбить приложение, какие функции возложить на каждую часть, как организовать взаимодействие этих частей, чтобы в случае сбоев и отказов оставшиеся части корректно завершали работу, и т. д., и т. п. Поэтому до сих пор только небольшая часть приложений является распределенными, хотя очевидно, что именно за этим классом приложений будущее, так как они в полной мере могут использовать потенциальные возможности сетей по распараллеливанию вычислений.

## 5.2. Аппаратно-программное обеспечение сети

Даже в результате достаточно поверхностного рассмотрения работы в сети становится ясно, что вычислительная сеть – это сложный комплекс взаимосвязанных и согласованно функционирующих программных и аппаратных компонентов. Изучение сети в целом предполагает знание принципов работы ее отдельных элементов:

- компьютеров;
- коммуникационного оборудования;
- операционных систем;
- сетевых приложений.

Весь комплекс программно-аппаратных средств сети может быть описан многослойной моделью. В основе любой сети лежит аппаратный слой стандартизованных компьютерных платформ. В на-

стоящее время в сетях широко и успешно применяются компьютеры различных классов — от персональных компьютеров до мэйнфреймов и суперЭВМ. Набор компьютеров в сети должен соответствовать набору разнообразных задач, решаемых сетью.

Второй слой — это коммуникационное оборудование. Хотя компьютеры и являются центральными элементами обработки данных в сетях, в последнее время не менее важную роль стали играть коммуникационные устройства. Кабельные системы, повторители, мосты, коммутаторы, маршрутизаторы и модульные концентраторы из вспомогательных компонентов сети превратились в основные наряду с компьютерами и системным программным обеспечением как по влиянию на характеристики сети, так и по стоимости. Сегодня коммуникационное устройство может представлять собой сложный специализированный мультипроцессор, который нужно конфигурировать, оптимизировать и администрировать. Изучение принципов работы коммуникационного оборудования требует знакомства с большим количеством протоколов, используемых как в локальных, так и глобальных сетях.

Третим слоем, образующим программную платформу сети, являются операционные системы (ОС). От того, какие концепции управления локальными и распределенными ресурсами положены в основу сетевой ОС, зависит эффективность работы всей сети. При проектировании сети важно учитывать, насколько просто данная операционная система может взаимодействовать с другими ОС сети, насколько она обеспечивает безопасность и защищенность данных, до какой степени она позволяет наращивать число пользователей, можно ли перенести ее на компьютер другого типа и многие другие соображения.

Самым верхним слоем сетевых средств являются различные сетевые приложения, такие как сетевые базы данных, почтовые системы, средства архивирования данных, системы автоматизации коллективной работы и др. Очень важно представлять диапазон возможностей, предоставляемых приложениями для различных областей применения, а также знать, насколько они совместимы с другими сетевыми приложениями и операционными системами.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем различие между многопроцессорными и многомашинными системами?
2. Каковы задачи вычислительных сетей?
3. Каковы задачи программных серверов?
4. В чем заключается смысл терминов «клиент» и «сервер»?
5. Расскажите об аппаратно-программных средствах как основе многослойной модели сети.

## ГЛАВА 6. ОСНОВНЫЕ ПРОБЛЕМЫ ПОСТРОЕНИЯ СЕТЕЙ

При создании вычислительных сетей их разработчикам пришлось решить много проблем. В этом разделе мы рассмотрим только наиболее важные из них, причем в той последовательности, в которой они естественно возникали в процессе развития и совершенствования сетевых технологий.

Механизмы взаимодействия компьютеров в сети многое позаимствовали у схемы взаимодействия компьютера с периферийными устройствами, поэтому начнем рассмотрение принципов работы сети с этого «досетевого» случая.

### 6.1. Связь компьютера с периферийными устройствами

Для обмена данными между компьютером и периферийным устройством (ПУ) в компьютере предусмотрен внешний *интерфейс* (рис. 6.1), то есть набор проводов, соединяющих компьютер и периферийное устройство, а также набор правил обмена информацией по этим проводам (иногда вместо термина *интерфейс* употребляется термин *протокол* — подробней об этих важных терминах мы еще поговорим).

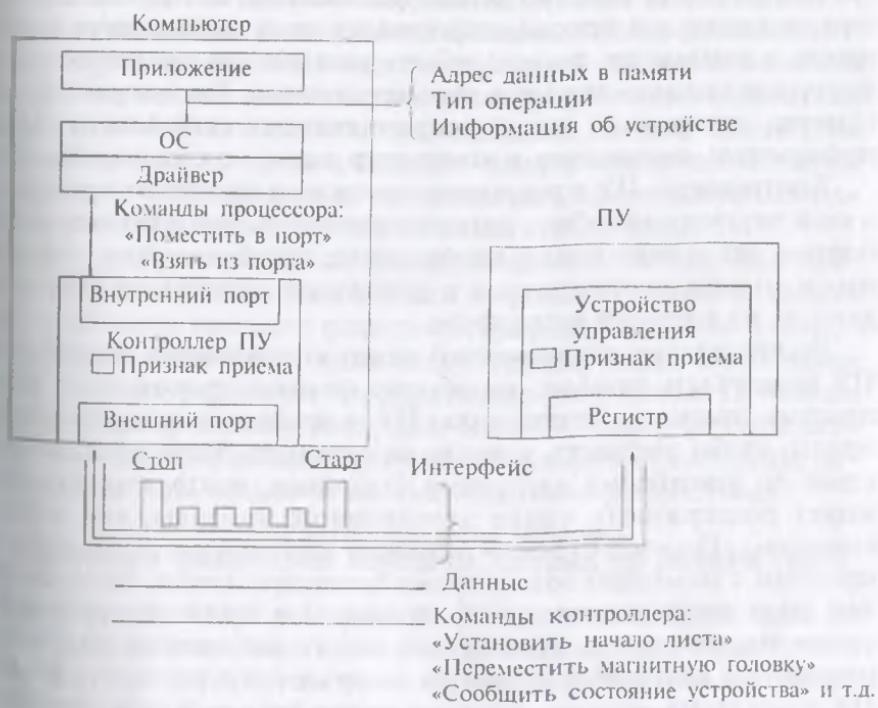


Рис.6.1. Связь компьютера с периферийным устройством.

Примерами интерфейсов, используемых в компьютерах, являются параллельный интерфейс Centronics, предназначенный, как правило, для подключения принтеров, и последовательный интерфейс RS-232C, через который подключаются мышь, модем и много других устройств. Интерфейс реализуется со стороны компьютера совокупностью аппаратных и программных средств: контроллером ПУ и специальной программой, управляющей этим контроллером, которую часто называют *драйвером* соответствующего периферийного устройства.

Со стороны ПУ интерфейс чаще всего реализуется аппаратным устройством управления, хотя встречаются и программно-управляемые периферийные устройства.

Программа, выполняемая процессором, может обмениваться данными с помощью команд ввода/вывода с любыми модулями, подключенными к внутреннейшине компьютера, в том числе и с контроллерами ПУ.

Периферийные устройства могут принимать от компьютера как данные, например байты информации, которую нужно распечатать на бумаге, так и команды управления, в ответ на которые ПУ может выполнить специальные действия, например, перевести головку диска на требуемую дорожку или же вытолкнуть лист бумаги из принтера. Периферийное устройство использует внешний интерфейс компьютера не только для приема информации, но и для передачи информации в компьютер, то есть обмен данными по внешнему интерфейсу, как правило, является двунаправленным. Так, например, даже принтер, который по своей природе является устройством вывода информации, возвращает в компьютер данные о своем состоянии.

Контроллеры ПУ принимают команды и данные от процессора в свой внутренний буфер, который часто называется регистром или портом, затем выполняют необходимые преобразования этих данных и команд в соответствии с форматами, понятными ПУ, и выдают их на внешний интерфейс.

Распределение обязанностей между контроллером и драйвером ПУ может быть разным, но обычно контроллер выполняет набор простых команд по управлению ПУ, а драйвер использует эти команды, чтобы заставить устройство совершать более сложные действия по некоторому алгоритму. Например, контроллер принтера может поддерживать такие элементарные команды, как «Печать символа», «Перевод строки», «Возврат каретки» и т. п. Драйвер же принтера с помощью этих команд организует печать строк символов, разделение документа на страницы и другие более высоконивневые операции. Для одного и того же контроллера можно разработать различные драйверы, которые будут управлять данным ПУ по-разному — одни лучше, а другие хуже — в зависимости от опыта и способностей программистов, их разработавших.

Рассмотрим схему передачи одного байта информации от прикладной программы на периферийное устройство. Программа, которой потребовалось выполнить обмен данными с ПУ, обращается к драйверу этого устройства, сообщая ему в качестве параметра адрес байта памяти, который нужно передать. Драйвер загружает значение этого байта в буфер контроллера ПУ, который начинает последовательно передавать биты в линию связи, представляя каждый бит соответствующим электрическим сигналом. Чтобы устройству управления ПУ стало понятно, что начинается передача байта, перед передачей первого бита информации контроллер ПУ формирует стартовый сигнал специфической формы, а после передачи последнего информационного бита — стоповый сигнал. Эти сигналы *синхронизируют* передачу байта.

Кроме информационных бит контроллер может передавать бит контроля четности для повышения достоверности обмена. Устройство управления, обнаружив на соответствующей линии стартовый бит, выполняет подготовительные действия и начинает принимать информационные биты, формируя из них байт в своем приемном буфере. Если передача сопровождается битом четности, то выполняется проверка правильности передачи: при правильно выполненной передаче в соответствующем регистре устройства управления устанавливается признак завершения приема информации.

Обычно на драйвер возлагаются наиболее сложные функции протокола (например, подсчет контрольной суммы последовательности передаваемых байтов, анализ состояния периферийного устройства, проверка правильности выполнения команды). Но даже самый примитивный драйвер контроллера должен поддерживать как минимум две операции: «Взять данные из контроллера в оперативную память» и «Передать данные из оперативной памяти в контроллер».

Существуют как весьма специализированные интерфейсы, пригодные для подключения узкого класса устройств (например, графических мониторов высокого разрешения фирмы Vista), так и интерфейсы общего назначения, являющиеся стандартными и позволяющие подключать различные периферийные устройства. Примером такого интерфейса является интерфейс RS-232C, который поддерживается многими терминалами, принтерами, графопостроителями, манипуляторами типа «мышь» и многими другими устройствами.

## 6.2. Проблемы физической передачи данных по линиям связи

Даже при рассмотрении простейшей сети, состоящей всего из двух машин, можно увидеть многие проблемы, присущие любой вычислительной сети, в том числе проблемы, связанные с физической передачей сигналов по линиям связи, без решения которой невозможен любой вид связи.

В вычислительной технике для представления данных используется двоичный код. Внутри компьютера единицам и нулям данных соответствуют дискретные электрические сигналы. Представление данных в виде электрических или оптических сигналов называется **кодированием**. Существуют различные способы кодирования двоичных цифр 1 и 0, например, потенциальный способ, при котором единице соответствует один уровень напряжения, а нулю — другой или импульсный способ, когда для представления цифр используются импульсы различной или одной полярности.

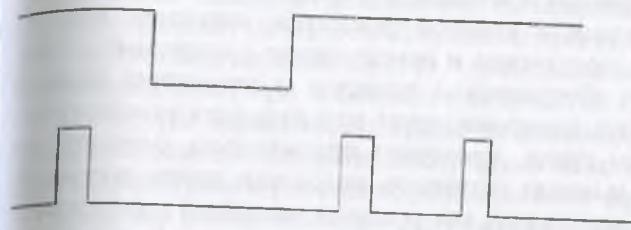
Аналогичные подходы могут быть использованы для кодирования данных и при передаче их между двумя компьютерами по линиям связи. Однако эти линии связи отличаются по своим электрическим характеристикам от тех, которые существуют внутри компьютера. Главное отличие внешних линий связи от внутренних состоит в их гораздо большей протяженности, а также в том, что они проходят вне экранированного корпуса по пространствам, зачастую подверженным воздействию сильных электромагнитных помех. Все это приводит к значительно большим искажениям прямоугольных импульсов (например, «затягиванию» фронтов), чем внутри компьютера. Поэтому для надежного распознавания импульсов на приемном конце линии связи при передаче данных внутри и вне компьютера не всегда можно использовать одни и те же скорости и способы кодирования. Например, медленное нарастание фронта импульса из-за высокой емкостной нагрузки линии требует передачи импульсов с меньшей скоростью (чтобы передний и задний фронты соседних импульсов не перекрывались и импульс успел дорастить до требуемого уровня).

В вычислительных сетях применяют как потенциальное, так импульсное кодирование дискретных данных, а также специфический способ представления данных, который никогда не используется внутри компьютера, — **модуляцию** (рис. 6.2). При модуляции дискретная информация представляется синусоидальным сигналом той частоты, которую хорошо передает имеющаяся линия связи.

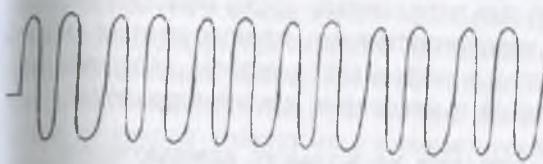
Потенциальное или импульсное кодирование применяется в каналах высокого качества, а модуляция на основе синусоидальных сигналов предпочтительнее в том случае, когда канал вносит сильные искажения в передаваемые сигналы. Обычно модуляция используется в глобальных сетях при передаче данных через аналоговые телефонные каналы связи, которые были разработаны для передачи голоса в аналоговой форме и поэтому плохо подходят для непосредственной передачи импульсов.

На способ передачи сигналов влияет и количество проводов линиях связи между компьютерами. Для сокращения стоимости линий связи в сетях обычно стремятся к сокращению количества проводов и из-за этого используют не параллельную передачу

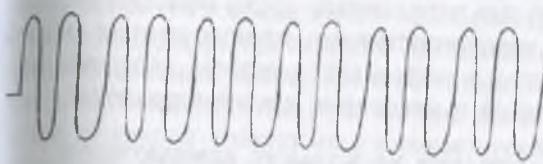
бит одного байта или даже нескольких байт, как это делается внутри компьютера, а последовательную, побитную передачу, требующую всего одной пары проводов.



Потенциальное кодирование



Импульсное кодирование



Модуляция

Рис. 6.2. Примеры представления дискретной информации.

Еще одной проблемой, которую нужно решать при передаче сигналов, является проблема взаимной **синхронизации** передатчика одного компьютера с приемником другого. При организации взаимодействия модулей внутри компьютера эта проблема решается очень просто, так как в этом случае все модули синхронизируются от общего тактового генератора. Проблема синхронизации при связи компьютеров может решаться разными способами, как с помощью обмена специальными тактовыми синхроимпульсами по отдельной линии, так и с помощью периодической синхронизации заранее обусловленными кодами или импульсами характерной формы, отличающейся от формы импульсов данных.

Несмотря на предпринимаемые меры — выбор соответствующей скорости обмена данными, линий связи с определенными характеристиками, способа синхронизации приемника и передатчика, — существует вероятность искажения некоторых бит передаваемых данных. Для повышения надежности передачи данных между компьютерами часто используется стандартный прием — подсчет **контрольной суммы** и передача ее по линиям связи после каждого байта или после некоторого блока байтов. Часто в протокол обмена данными включается как обязательный элемент **сигнал-квитанция**, который подтверждает правильность приема данных и посыпается от получателя отправителю.

Задачи надежного обмена двоичными сигналами, представляемыми соответствующими электромагнитными сигналами, в вычислительных сетях решает определенный класс оборудования. В локальных сетях это *сетевые адаптеры*, а в глобальных сетях — аппаратура передачи данных, к которой относятся, например, устройства, выполняющие модуляцию и демодуляцию дискретных сигналов, — *модемы*. Это оборудование кодирует и декодирует каждый информационный бит, синхронизирует передачу электромагнитных сигналов по линиям связи, проверяет правильность передачи по контрольной сумме и может выполнять некоторые другие операции. Сетевые адаптеры рассчитаны, как правило, на работу с определенной *передающей средой* — коаксиальным кабелем, витой парой, оптоволокном и т. п. Каждый тип передающей среды обладает определенными электрическими характеристиками, влияющими на способ использования данной среды, и определяет скорость передачи сигналов, способ их кодирования и некоторые другие параметры.

### 6.3. Проблемы объединения нескольких компьютеров

До сих пор мы рассматривали вырожденную сеть, состоящую всего из двух машин. При объединении в сеть большего числа компьютеров возникает целый комплекс новых проблем.

**Топология физических связей.** В первую очередь необходимо выбрать способ организации физических связей, то есть *топологию*. Под топологией вычислительной сети понимается конфигурация графа, вершинам которого соответствуют компьютеры сети (иногда и другое оборудование, например концентраторы), а ребрам — физические связи между ними. Компьютеры, подключенные к сети, часто называют *станциями* или *узлами сети*.

Заметим, что конфигурация физических связей определяется электрическими соединениями компьютеров между собой и может отличаться от конфигурации логических связей между узлами сети. Логические связи представляют собой маршруты передачи данных между узлами сети и образуются путем соответствующей настройки коммуникационного оборудования.

Выбор топологии электрических связей существенно влияет на многие характеристики сети. Например, наличие резервных связей повышает надежность сети и делает возможным балансирование нагрузки отдельных каналов. Простота присоединения новых узлов, свойственная некоторым топологиям, делает сеть легко расширяемой. Экономические соображения часто приводят к выбору топологий, которых характерна минимальная суммарная длина линий связи.

Рассмотрим некоторые, наиболее часто встречающиеся топологии.

**Полносвязная топология** (рис. 6.3, а) соответствует сети, в которой каждый компьютер сети связан со всеми остальными. Несмотря

на логическую простоту, этот вариант оказывается громоздким и неэффективным. Действительно, каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров сети. Для каждой пары компьютеров должна быть выделена отдельная электрическая линия связи. Полносвязные топологии применяются редко, так как не удовлетворяют ни одному из приведенных выше требований. Чаще всего этот вид топологии используется в многомашинных комплексах или глобальных сетях при небольшом количестве компьютеров.

Все другие варианты основаны на неполносвязных топологиях, когда для обмена данными между двумя компьютерами может потребоваться промежуточная передача данных через другие узлы сети.

**Ячеистая топология** (mesh) получается из полносвязной путем удаления некоторых возможных связей (рис. 6.3, б). В сети с ячеистой топологией непосредственно связываются только те компьютеры, между которыми происходит интенсивный обмен данными, а для обмена данными между компьютерами, не соединенными прямыми связями, используются транзитные передачи через промежуточные узлы. Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для глобальных сетей.

**Общая шина** (рис. 6.3, в) является очень распространенной (а до недавнего времени самой распространенной) топологией для локальных сетей. В этом случае компьютеры подключаются к одному коаксиальному кабелю по схеме «монтажного ИЛИ». Передаваемая информация может распространяться в обе стороны. Применение общей шины снижает стоимость проводки, унифицирует подключение различных модулей, обеспечивает возможность почти мгновенного широковещательного обращения ко всем станциям сети. Таким образом, основными преимуществами такой схемы являются дешевизна и простота разводки кабеля по помещениям. Самый серьезный недостаток общей шины заключается в ее низкой надежности: любой дефект кабеля или какого-нибудь из многочисленных разъемов полностью парализует всю сеть. К сожалению, дефект коаксиального разъема редкостью не является. Другим недостатком общей шины является ее невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные в сеть. Поэтому пропускная способность канала связи всегда делится здесь между всеми узлами сети.

**Топология звезда** (рис. 6.3, г). В этом случае каждый компьютер подключается отдельным кабелем к общему устройству, называемому *концентратором*, который находится в центре сети. В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. Главное преимущество этой топологии перед общей шиной — существенно

большая надежность. Любые неприятности с кабелем касаются лишь того компьютера, к которому этот кабель присоединен, и только неисправность концентратора может вывести из строя всю сеть. Кроме того, концентратор может играть роль интеллектуального фильтра информации, поступающей от узлов в сеть, и при необходимости блокировать запрещенные администратором передачи.

К недостаткам топологии типа звезды относится более высокая стоимость сетевого оборудования из-за необходимости приобретения концентратора. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора. Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой связями типа звезды (рис. 6.3, д). В настоящее время иерархическая звезда является самым распространенным типом топологии связей как в локальных, так и глобальных сетях.

В сетях с *кольцевой* конфигурацией (рис. 6.3, е) данные передаются по кольцу от одного компьютера к другому, как правило, в одном направлении. Если компьютер распознает данные как «свои», то он копирует их себе во внутренний буфер. В сети с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прервался канал связи между остальными станциями. Кольцо представляет собой очень удобную конфигурацию для организации обратной связи — данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому этот узел может контролировать процесс доставки данных адресату. Часто это свойство кольца используется для тестирования связности сети и поиска узла, работающего некорректно. Для этого в сеть посылаются специальные тестовые сообщения.

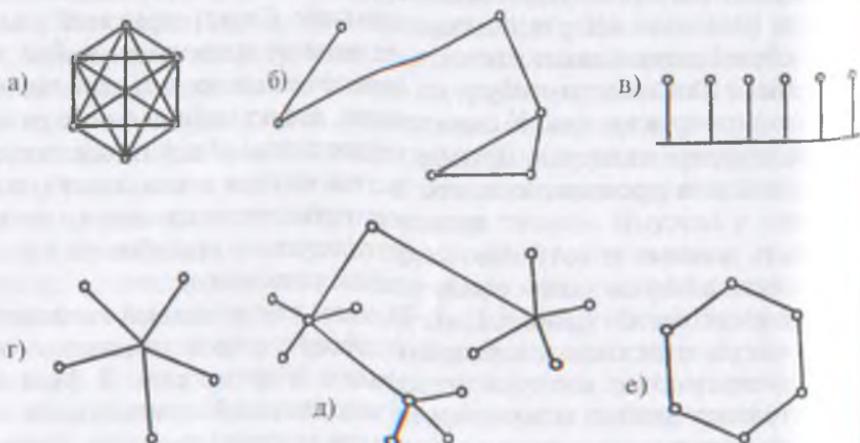


Рис. 6.3. Типовые топологии сети.

В то время как небольшие сети, как правило, имеют типовую топологию — звезда, кольцо или общая шина, для крупных сетей характерно наличие произвольных связей между компьютерами. В таких сетях можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со смешанной топологией (рис. 6.4).

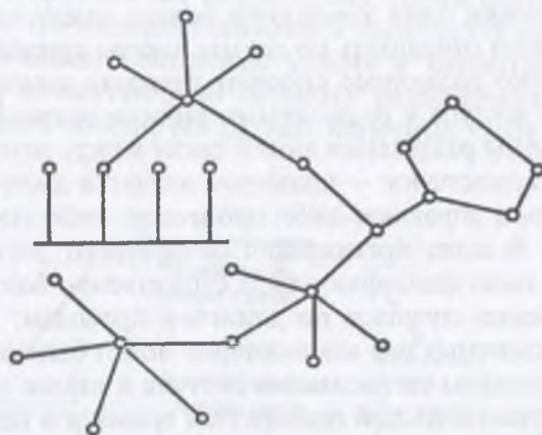


Рис. 6.4. Смешанная топология.

**Организация совместного использования линий связи.** Только в сети с полносвязной топологией для соединения каждой пары компьютеров имеется отдельная линия связи. Во всех остальных случаях неизбежно возникает вопрос о том, как организовать совместное использование линий связи несколькими компьютерами сети. Как и всегда при разделении ресурсов, главной целью здесь является удешевление сети.

В вычислительных сетях используют как *индивидуальные* линии связи между компьютерами, так и *разделяемые (shared)*, когда одна линия связи пополам используется несколькими компьютерами. В случае применения разделяемых линий связи (часто используется также термин *разделяемая среда передачи данных — shared media*) возникает комплекс проблем, связанных с их совместным использованием, который включает как чисто электрические проблемы обеспечения нужного качества сигналов при подключении к одному и тому же проводу нескольких приемников и передатчиков, так и логические проблемы разделения во времени доступа к этим линиям.

Классическим примером сети с разделяемыми линиями связи являются сети с топологией «общая шина», в которых один кабель совместно используется всеми компьютерами сети. Ни один из ком-

пьютеров сети в принципе не может индивидуально, независимо от всех других компьютеров сети, использовать кабель, так как при одновременной передаче данных сразу несколькими узлами сигналы смешиваются и искажаются. В топологиях «кольцо» или «звезда» индивидуальное использование линий связи, соединяющих компьютеры, принципиально возможно, но эти кабели часто также рассматривают как разделяемые для всех компьютеров сети, так что, например, только один компьютер кольца имеет право в данный момент времени отправлять по кольцу пакеты другим компьютерам.

Существуют различные способы решения задачи организации совместного доступа к разделяемым линиям связи. Внутри компьютера проблемы разделения линий связи между различными модулями также существуют — примером является доступ к системной шине, которым управляет либо процессор, либо специальный арбитр шины. В сетях организация совместного доступа к линиям связи имеет свою специфику из-за существенно большего времени распространения сигналов по длинным проводам, к тому же это время для различных пар компьютеров может быть различным. Из-за этого процедуры согласования доступа к линии связи могут занимать слишком большой промежуток времени и приводить к значительным потерям производительности сети.

Несмотря на все эти сложности, в локальных сетях разделяемые линии связи используются очень часто. Этот подход, в частности, реализован в широко распространенных классических технологиях Ethernet и Token Ring. Однако в последние годы наметилась тенденция отказа от разделяемых сред передачи данных и в локальных сетях. Это связано с тем, что за достигаемое таким образом удешевление сети приходится расплачиваться производительностью.

Сеть с разделяемой средой при большом количестве узлов будет работать всегда медленнее, чем аналогичная сеть с индивидуальными линиями связи, так как пропускная способность индивидуальной линии связи достается одному компьютеру, а при ее совместном использовании — делится на все компьютеры сети. Часто с такой потерей производительности мирятся ради увеличения экономической эффективности сети. Не только в классических, но и в совсем новых технологиях, разработанных для локальных сетей, сохраняется режим разделяемых линий связи. Например, разработчики технологии Gigabit Ethernet, принятой в 1998 году в качестве нового стандарта, включили режим разделения передающей среды в свои спецификации наряду с режимом работы по индивидуальным линиям связи.

При использовании индивидуальных линий связи в полносвязных топологиях конечные узлы должны иметь по одному порту на каждую линию связи. В звездообразных топологиях конечные узлы

могут подключаться индивидуальными линиями связи к специальному устройству — коммутатору. В глобальных сетях коммутаторы использовались уже на начальном этапе, а в локальных сетях — с начала 90-х годов. Коммутаторы приводят к существенному удешевлению локальной сети, поэтому пока их применение ограничено, но по мере снижения стоимости коммутации этот подход, возможно, вытеснит применение разделяемых линий связи. Необходимо подчеркнуть, что индивидуальными в таких сетях являются только линии связи между конечными узлами и коммутаторами сети, а связи между коммутаторами остаются разделяемыми, так как по ним передаются сообщения разных конечных узлов (рис. 6.5)

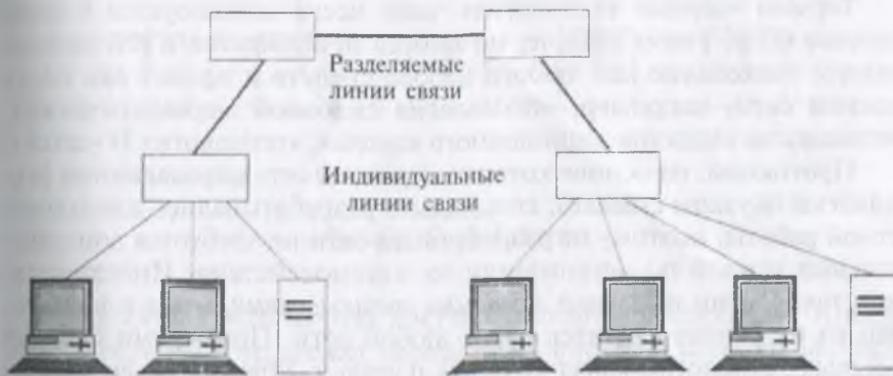


Рис.6.5. Индивидуальные и разделяемые линии связи в сетях на основе коммутаторов.

В глобальных сетях отказ от разделяемых линий связи объясняется техническими причинами. Здесь большие временные задержки распространения сигналов принципиально ограничивают применимость техники разделения линии связи. Компьютеры могут затратить больше времени на переговоры о том, кому сейчас можно использовать линию связи, чем непосредственно на передачу данных по этой линии связи. Однако это не относится к линиям связи типа «коммутатор-коммутатор». В этом случае только два коммутатора борются за доступ к линии связи, и это существенно упрощает задачу организации совместного использования линии.

#### 6.4. Ethernet — пример стандартного решения сетевых проблем

Рассмотрим, каким образом описанные выше общие подходы к решению наиболее важных проблем построения сетей воплощены в наиболее популярной сетевой технологии — *Ethernet*.

*Сетевая технология* — это согласованный набор стандартных протоколов и реализующих их программно-аппаратных средств (например, сетевых адаптеров, драйверов, кабелей и разъемов), достаточный для построения вычислительной сети. Эпитет «достаточный» подчеркивает то обстоятельство, что этот набор представляет собой минимальный набор средств, с помощью которых можно построить работоспособную сеть. Возможно, эту сеть можно улучшить, например, за счет выделения в ней подсетей, что сразу потребует кроме протоколов стандарта Ethernet применения протокола IP, а также специальных коммуникационных устройств — маршрутизаторов. Улучшенная сеть будет, скорее всего, более надежной и быстродействующей, но за счет надстроек над средствами технологии Ethernet, которая составила базис сети.

Термин «сетевая технология» чаще всего используется в описанном выше узком смысле, но иногда применяется и его расширенное толкование как любого набора средств и правил для построения сети, например, «технология сквозной маршрутизации», «технология создания защищенного канала», «технология IP-сетей».

Протоколы, на основе которых строится сеть определенной технологии (в узком смысле), специально разрабатывались для совместной работы, поэтому от разработчика сети не требуется дополнительных усилий по организации их взаимодействия. Иногда сетевые технологии называют *базовыми технологиями*, имея в виду то, что на их основе строится базис любой сети. Примерами базовых сетевых технологий могут служить наряду с Ethernet такие известные технологии локальных сетей, как Token Ring и FDDI, или же технологии территориальных сетей X.25 и frame relay. Для получения работоспособной сети в этом случае достаточно приобрести программные и аппаратные средства, относящиеся к одной базовой технологии — сетевые адаптеры с драйверами, концентраторы, коммутаторы, кабельную систему и т. п., — и соединить их в соответствии с требованиями стандарта на данную технологию.

Стандарт Ethernet был принят в 1980 году. Число сетей, построенных на основе этой технологии, к настоящему моменту оценивается в 5 миллионов, а количество компьютеров, работающих в таких сетях, — в 50 миллионов.

Основной принцип, положенный в основу Ethernet, — *случайный метод доступа* к разделяемой среде передачи данных. В качестве такой среды может использоваться толстый или тонкий коаксиальный кабель, витая пара, оптоволокно или радиоволны (кстати, первой сетью, построенной на принципе случайного доступа к разделяемой среде, была радиосеть Aloha Гавайского университета).

В стандарте Ethernet строго зафиксирована топология электрических связей. Компьютеры подключаются к разделяемой среде в

соответствии с типовой структурой «общая шина» (рис. 6.6). С помощью разделяемой во времени шины любые два компьютера могут обмениваться данными. Управление доступом к линии связи осуществляется специальными контроллерами — сетевыми адаптерами Ethernet. Каждый компьютер, а более точно, каждый сетевой адаптер, имеет уникальный адрес. Передача данных происходит со скоростью 10 Мбит/с. Эта величина является пропускной способностью сети Ethernet.

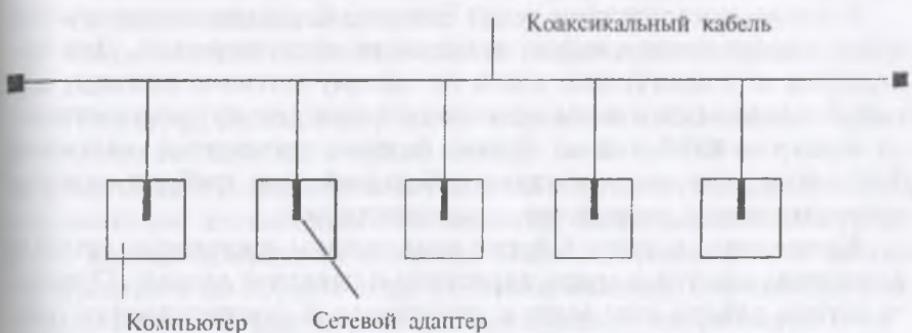


Рис. 6.6. Сеть Ethernet.

Суть случайного метода доступа состоит в следующем. Компьютер в сети Ethernet может передавать данные по сети, только если сеть свободна, то есть если никакой другой компьютер в данный момент не занимается обменом. Поэтому важной частью технологии Ethernet является процедура определения доступности среды.

После того как компьютер убедился, что сеть свободна, он начинает передачу, при этом «захватывает» среду. Время монопольного использования разделяемой среды одним узлом ограничивается временем передачи одного кадра. *Кадр* — это единица данных, которыми обмениваются компьютеры в сети Ethernet. Кадр имеет фиксированный формат и наряду с полем данных содержит различную служебную информацию, например адрес получателя и адрес отправителя.

Сеть Ethernet устроена так, что при попадании кадра в разделяемую среду передачи данных все сетевые адаптеры одновременно начинают принимать этот кадр. Все они анализируют адрес назначения, располагающийся в одном из начальных полей кадра, и, если этот адрес совпадает с их собственным адресом, кадр помещается во внутренний буфер сетевого адаптера. Таким образом, компьютер-адресат получает предназначенные ему данные.

Иногда может возникать ситуация, когда одновременно два или более компьютера решают, что сеть свободна, и начинают переда-

вать информацию. Такая ситуация, называемая *коллизией*, препятствует правильной передаче данных по сети. В стандарте Ethernet предусмотрен алгоритм обнаружения и корректной обработки коллизий. Вероятность возникновения коллизии зависит от интенсивности сетевого трафика.

После обнаружения коллизии сетевые адаптеры, которые пытались передать свои кадры, прекращают передачу и после паузы случайной длительности пытаются снова получить доступ к среде и передать тот кадр, который вызвал коллизию.

Главным достоинством сетей Ethernet, благодаря которому они стали такими популярными, является их экономичность. Для построения сети достаточно иметь по одному сетевому адаптеру для каждого компьютера плюс один физический сегмент коаксиального кабеля нужной длины. Другие базовые технологии, например Token Ring, для создания даже небольшой сети требуют наличия дополнительного устройства — концентратора.

Кроме того, в сетях Ethernet реализованы достаточно простые алгоритмы доступа к среде, адресации и передачи данных. Простота логики работы сети ведет к упрощению и, соответственно, удешевлению сетевых адаптеров и их драйверов. По той же причине адаптеры сети Ethernet обладают высокой надежностью.

И, наконец, еще одним замечательным свойством сетей Ethernet является их хорошая расширяемость, то есть легкость подключения новых узлов.

Другие базовые сетевые технологии — Token Ring, FDDI, 100VGAny-LAN, хотя и обладают многими индивидуальными чертами, в то же время имеют много общих свойств с Ethernet. В первую очередь — это применение регулярных фиксированных топологий (иерархическая звезда и кольцо), а также разделяемых сред передачи данных. Существенные отличия одной технологии от другой связаны с особенностями используемого метода доступа к разделяемой среде. Так, отличия технологии Ethernet от технологии Token Ring во многом определяются спецификой заложенных в них методов разделения среды — случайного алгоритма доступа в Ethernet и метода доступа путем передачи маркера в Token Ring.

## 6.5. Структурирование как средство построения больших сетей

В сетях с небольшим (10–30) количеством компьютеров чаще всего используется одна из типовых топологий — общая шина, кольцо, звезда или полносвязная сеть. Все перечисленные топологии обладают свойством однородности, то есть все компьютеры в такой сети имеют одинаковые права в отношении доступа к другим компьютерам (за исключением центрального компьютера при

соединении звезды). Такая однородность структуры делает простой процедуру наращивания числа компьютеров, облегчает обслуживание и эксплуатацию сети.

Однако при построении больших сетей однородная структура связей превращается из преимущества в недостаток. В таких сетях использование типовых структур порождает различные ограничения, важнейшими из которых являются:

- ограничения на длину связи между узлами;
- ограничения на количество узлов в сети;
- ограничения на интенсивность трафика, порождаемого узлами сети.

Например, технология Ethernet на тонком коаксиальном кабеле позволяет использовать кабель длиной не более 185 метров, к которому можно подключить не более 30 компьютеров. Однако, если компьютеры интенсивно обмениваются информацией между собой, иногда приходится снижать число подключенных к кабелю компьютеров до 20, а то и до 10, чтобы каждому компьютеру доставалась приемлемая доля общей пропускной способности сети.

Для снятия этих ограничений используются специальные методы структуризации сети и специальное структурообразующее оборудование — повторители, концентраторы, мосты, коммутаторы, маршрутизаторы. Оборудование такого рода также называют коммуникационным, имея в виду, что с помощью него отдельные сегменты сети взаимодействуют между собой.

**Физическая структуризация сети.** Простейшее из коммуникационных устройств — повторитель (repeater) — используется для физического соединения различных сегментов кабеля локальной сети с целью увеличения общей длины сети. Повторитель передает сигналы, приходящие из одного сегмента сети, в другие ее сегменты (рис. 6.7). Повторитель позволяет преодолеть ограничения на длину линий связи за счет улучшения качества передаваемого сигнала — восстановления его мощности и амплитуды, улучшения фронтов и т. п.



Рис. 6.7. Повторитель позволяет увеличить длину сети Ethernet.

Повторитель, который имеет несколько портов и соединяет несколько физических сегментов, часто называют *концентратором* (*concentrator*) или *хабом* (*hub*). Эти названия (*hub* — основа, центр деятельности) отражают тот факт, что в данном устройстве сосредоточиваются все связи между сегментами сети.

Концентраторы характерны практически для всех базовых технологий локальных сетей — Ethernet, ArcNet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, 100VG-AnyLAN<sup>1</sup>.

Нужно подчеркнуть, что в работе концентраторов любых технологий много общего — они повторяют сигналы, пришедшие с одного из своих портов, на других своих портах. Разница состоит в том, на каких именно портах повторяются входные сигналы. Так, концентратор Ethernet повторяет входные сигналы на всех своих портах, кроме того, с которого сигналы поступают (рис. 6.8, а). А концентратор Token Ring (рис. 6.8, б) повторяет входные сигналы, поступающие с некоторого порта, только на одном порту — на том, к которому подключен следующий в кольце компьютер.

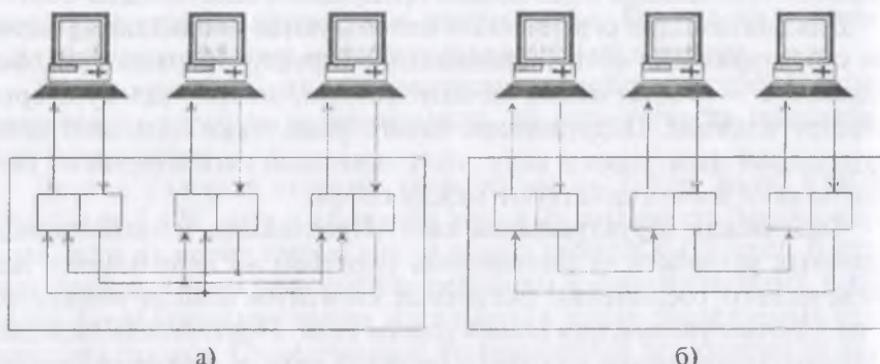


Рис. 6.8. Концентраторы различных технологий.

Напомним, что под физической топологией понимается конфигурация связей, образованных отдельными частями кабеля, а под логической — конфигурация информационных потоков между компьютерами сети. Во многих случаях физическая и логическая топология сети совпадают. Например, сеть, представленная на рис. 6.9, а, имеет физическую топологию кольцо. Компьютеры этой сети получают доступ к кабелям кольца за счет передачи друг другу специального кадра — маркера, причем этот маркер также передается последо-

<sup>1</sup> Концентратор всегда изменяет физическую топологию сети, но при этом оставляет без изменения ее логическую топологию.

вательно от компьютера к компьютеру в том же порядке, в котором компьютеры образуют физическое кольцо, то есть компьютер А передает маркер компьютеру В, компьютер В — компьютеру С и т. д. Сеть, показанная на рис. 6.9, б, демонстрирует пример несовпадения физической и логической топологии. Физически компьютеры соединены по топологии общая шина. Доступ же к шине происходит не по алгоритму случайного доступа, применяемому в технологии Ethernet, а путем передачи маркера в кольцевом порядке: от компьютера А — компьютеру В, от компьютера В — компьютеру С и т. д. Здесь порядок передачи маркера уже не повторяет физические связи, а определяется логическим конфигурированием драйверов сетевых адаптеров. Ничто не мешает настроить сетевые адаптеры и их драйверы так, чтобы компьютеры образовали кольцо в другом порядке, например: В, А, С... При этом физическая структура сети никак не изменяется.

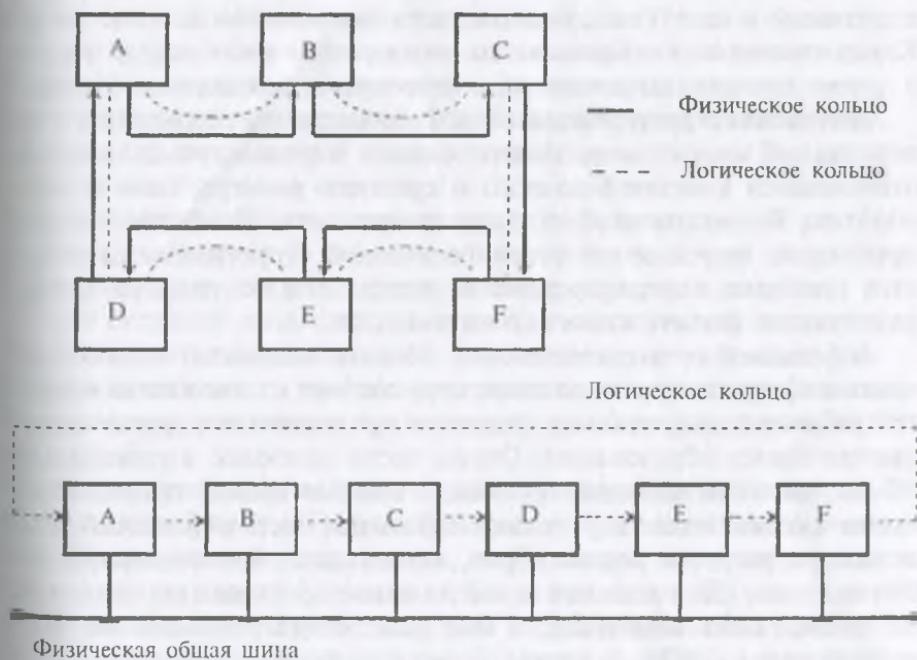


Рис.6.9. Логическая и физическая топологии сети.

Другим примером несовпадения физической и логической топологии сети является уже рассмотренная сеть на рис. 6.8, а. Концентратор Ethernet поддерживает в сети физическую топологию звезды. Однако логическая топология сети осталась без изменений — это общая шина. Так как концентратор повторяет данные, пришедшие

с любого порта, на всех остальных портах, то они появляются одновременно на всех физических сегментах сети, как и в сети с физической общей шиной. Логика доступа к сети совершенно не меняется: все компоненты алгоритма случайного доступа — определение незанятости среды, захват среды, распознавание и отработка коллизий — остаются в силе.

Физическая структуризация сети с помощью концентраторов полезна не только для увеличения расстояния между узлами сети, но и для повышения ее надежности. Например, если какой-либо компьютер сети Ethernet с физической общей шиной из-за сбоя начинает непрерывно передавать данные по общему кабелю, то вся сеть выходит из строя, и для решения этой проблемы остается только один выход — вручную отсоединить сетевой адаптер этого компьютера от кабеля. В сети Ethernet, построенной с использованием концентратора, эта проблема может быть решена автоматически — концентратор отключает свой порт, если обнаруживает, что присоединенный к нему узел слишком долго монопольно занимает сеть. Концентратор может блокировать некорректно работающий узел и в других случаях, выполняя роль некоторого управляющего узла.

**Логическая структуризация сети.** Физическая структуризация сети полезна во многих отношениях, однако в ряде случаев, обычно относящихся к сетям большого и среднего размера, невозможно обойтись без логической структуризации сети. Наиболее важной проблемой, не решаемой путем физической структуризации, остается проблема перераспределения передаваемого трафика между различными физическими сегментами сети.

В большой сети естественным образом возникает неоднородность информационных потоков: сеть состоит из множества подсетей рабочих групп, отделов, филиалов предприятия и других административных образований. Очень часто наиболее интенсивный обмен данными наблюдается между компьютерами, принадлежащими к одной подсети, и только небольшая часть обращений происходит к ресурсам компьютеров, находящихся вне локальных рабочих групп. (До недавнего времени такое соотношение трафиков не подвергалось сомнению, и был даже сформулирован эмпирический закон «80/20», в соответствии с которым в каждой подсети 80 % трафика является внутренним и только 20 % — внешним.) Сейчас характер нагрузки сетей во многом изменился, широко внедряется технология intranet, на многих предприятиях имеются централизованные хранилища корпоративных данных, активно используемые всеми сотрудниками предприятия. Все это не могло не повлиять на распределение информационных потоков. И теперь не редки ситуации, когда интенсивность внешних обращений выше

интенсивности обмена между «соседними» машинами. Но независимо от того, в какой пропорции распределяются внешний и внутренний трафик, для повышения эффективности работы сети неоднородность информационных потоков необходимо учитывать.

Сеть с типовой топологией (шина, кольцо, звезда), в которой все физические сегменты рассматриваются в качестве одной разделяемой среды, оказывается неадекватной структуре информационных потоков в большой сети. Например, в сети с общей шиной взаимодействие любой пары компьютеров занимает ее на все время обмена, поэтому при увеличении числа компьютеров в сети шина становится узким местом. Компьютеры одного отдела вынуждены ждать, когда окончит обмен пара компьютеров другого отдела, и это при том, что необходимость в связи между компьютерами двух разных отделов возникает гораздо реже и требует совсем небольшой пропускной способности.

Этот случай иллюстрирует рис. 6.10, а. Здесь показана сеть, построенная с использованием концентраторов. Пусть компьютер А, находящийся в одной подсети с компьютером В, посыпает ему данные. Несмотря на разветвленную физическую структуру сети, концентраторы распространяют любой кадр по всем ее сегментам. Поэтому кадр, посыпаемый компьютером А компьютеру В, хотя и не нужен компьютерам отделов 2 и 3, в соответствии с логикой работы концентраторов поступает на эти сегменты тоже. И до тех пор, пока компьютер В не получит адресованный ему кадр, ни один из компьютеров этой сети не сможет передавать данные.

Такая ситуация возникает из-за того, что логическая структура данной сети осталась однородной — она никак не учитывает увеличение интенсивности трафика внутри отдела и предоставляет всем парам компьютеров равные возможности по обмену информацией (рис. 6.10, б).

Решение проблемы состоит в отказе от идеи единой однородной разделяемой среды. Например, в рассмотренном выше примере желательно было бы сделать так, чтобы кадры, которые передают компьютеры отдела 1, выходили бы за пределы этой части сети в том и только в том случае, если эти кадры направлены какому-либо компьютеру из других отделов. С другой стороны, в сеть каждого из отделов должны попадать те и только те кадры, которые адресованы узлам этой сети. При такой организации работы сети ее производительность существенно повыситься, так как компьютеры одного отдела не будут простоять в то время, когда обмениваются данными компьютеры других отделов.

Нетрудно заметить, что в предложенном решении мы отказались от идеи общей разделяемой среды в пределах всей сети, хотя

и оставили ее в пределах каждого отдела. Пропускная способность линий связи между отделами не должна совпадать с пропускной способностью среды внутри отделов. Если трафик между отделами составляет только 20 % трафика внутри отдела (как уже отмечалось, эта величина может быть другой), то и пропускная способность линий связи и коммуникационного оборудования, соединяющего отделы, может быть значительно ниже внутреннего трафика сети отдела.

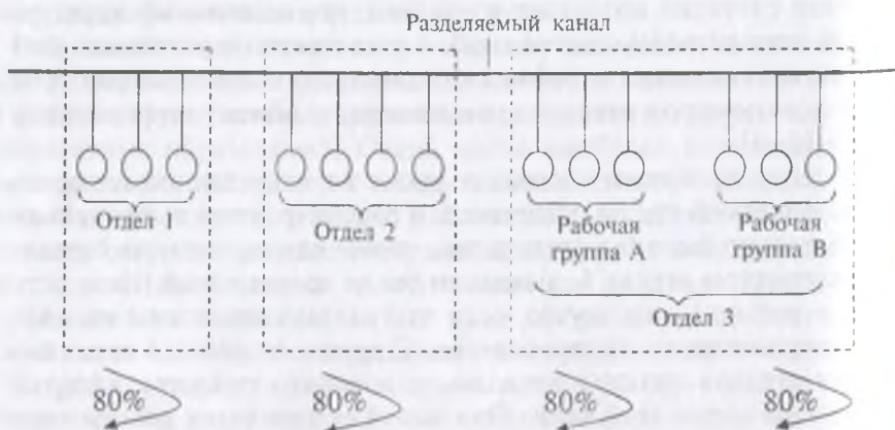
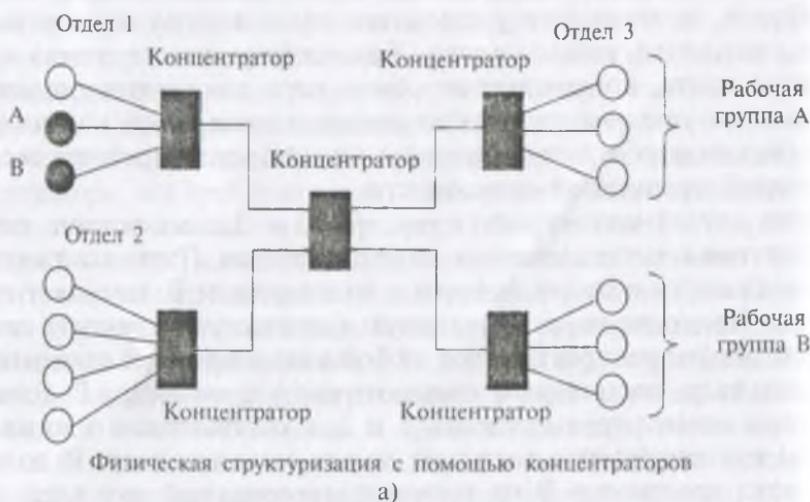


Рис.6.10. Противоречие между логической структурой сети и структурой информационных потоков.

Распространение трафика, предназначенного для компьютеров некоторого сегмента сети, только в пределах этого сегмента, называется *локализацией трафика*. *Логическая структуризация* сети — это процесс разбивки сети на сегменты с локализованным трафиком.

Для логической структуризации сети используются такие коммуникационные устройства, как мосты, коммутаторы, маршрутизаторы и шлюзы.

*Мост (bridge)* делит разделяемую среду передачи сети на части (часто называемые логическими сегментами), передавая информацию из одного сегмента в другой только в том случае, если такая передача действительно необходима, то есть если адрес компьютера назначения принадлежит другой подсети. Тем самым мост изолирует трафик одной подсети от трафика другой, повышая общую производительность передачи данных в сети. Локализация трафика не только экономит пропускную способность, но и уменьшает возможность несанкционированного доступа к данным, так как кадры не выходят за пределы своего сегмента и их сложнее перехватить злоумышленнику.

На рис. 6.11 показана сеть, которая была получена из сети с центральным концентратором (см. рис. 6.10) путем его замены на мост. Сети 1-го и 2-го отделов состоят из отдельных логических сегментов, а сеть отдела 3 — из двух логических сегментов. Каждый логический сегмент построен на базе концентратора и имеет простейшую физическую структуру, образованную отрезками кабеля, связывающими компьютеры с портами концентратора.

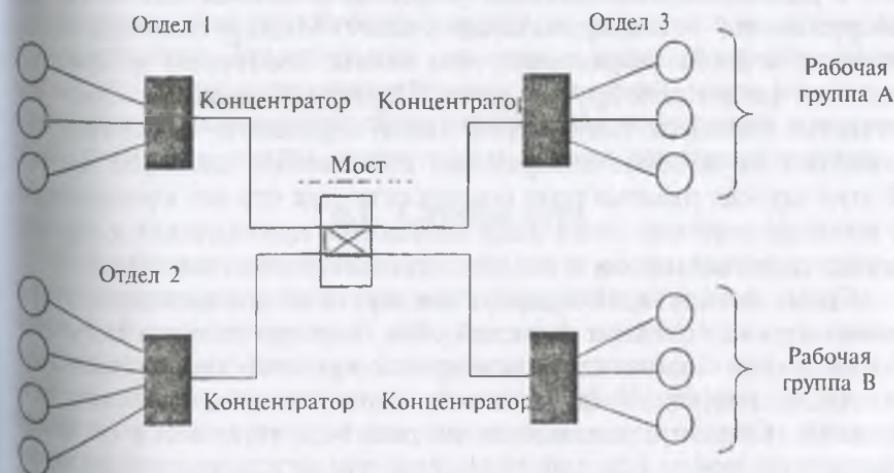


Рис. 6.11. Логическая структуризация сети с помощью моста.

Мосты используют для локализации трафика аппаратные адреса компьютеров. Это затрудняет распознавание принадлежности того или иного компьютера к определенному логическому сегменту — сам адрес не содержит никакой информации по этому поводу. Поэтому мост достаточно упрощенно представляет деление сети на сегменты — он запоминает, через какой порт на него поступил кадр данных от каждого компьютера сети, и в дальнейшем передает кадры, предназначенные для этого компьютера, на этот порт. Точной топологии связей между логическими сегментами мост не знает. Из-за этого применение мостов приводит к значительным ограничениям на конфигурацию связей сети — сегменты должны быть соединены таким образом, чтобы в сети не образовывались замкнутые контуры.

*Коммутатор (switch, switching hub)* по принципу обработки кадров ничем не отличается от моста. Основное его отличие от моста состоит в том, что он является своего рода коммуникационным мультипроцессором, так как каждый его порт оснащен специализированным процессором, который обрабатывает кадры по алгоритму моста независимо от процессоров других портов. За счет этого общая производительность коммутатора обычно намного выше производительности традиционного моста, имеющего один процессорный блок. Можно сказать, что коммутаторы — это мосты нового поколения, которые обрабатывают кадры в параллельном режиме.

Ограничения, связанные с применением мостов и коммутаторов — по топологии связей, а также ряд других, — привели к тому, что в ряду коммуникационных устройств появился еще один тип оборудования — *маршрутизатор (router)*. Маршрутизаторы более надежно и более эффективно, чем мосты, изолируют трафик отдельных частей сети друг от друга. Маршрутизаторы образуют логические сегменты посредством явной адресации, поскольку используют не плоские аппаратные, а составные числовые адреса. В этих адресах имеется поле номера сети, так что все компьютеры, у которых значение этого поля одинаково, принадлежат к одному сегменту, называемому в данном случае *подсетью (subnet)*.

Кроме локализации трафика маршрутизаторы выполняют еще много других полезных функций. Так, маршрутизаторы могут работать в сети с замкнутыми контурами, при этом они осуществляют выбор наиболее рационального маршрута из нескольких возможных. Сеть, представленная на рис. 6.12, отличается от своей предшественницы (см. рис. 6.11) тем, что между подсетями отловов 1 и 2 проложена дополнительная связь, которая может использоваться как для повышения производительности сети, так и для повышения ее надежности.



Рис. 6.12. Логическая структуризация сети с помощью маршрутизаторов.

Другой очень важной функцией маршрутизаторов является их способность связывать в единую сеть подсети, построенные с использованием разных сетевых технологий, например Ethernet и X.25.

Кроме перечисленных устройств отдельные части сети может соединять *шлюз* (*gateway*). Обычно основной причиной, по которой в сети используют шлюз, является необходимость объединить сети с разными типами системного и прикладного программного обеспечения, а не желание локализовать трафик. Тем не менее шлюз обеспечивает и локализацию трафика в качестве некоторого побочного эффекта.

Крупные сети практически никогда не строятся без логической структуризации. Для отдельных сегментов и подсетей характерны типовые однородные топологии базовых технологий, и для их объединения всегда используется оборудование, обеспечивающее локализацию трафика, — мосты, коммутаторы, маршрутизаторы и шлюзы.

## 6.6. Службы сети

Для конечного пользователя сеть — это не компьютеры, кабели и концентраторы и даже не информационные потоки, для него сеть — это прежде всего тот набор сетевых служб, с помощью которых он получает возможность просмотреть список имеющихся в сети компьютеров, прочитать удаленный файл, распечатать документ на «чужом» принтере или послать почтовое сообщение. Именно совокупность предоставляемых возможностей — насколько широк их выбор, насколько они удобны, надежны и безопасны — определяет для пользователя облик той или иной сети.

Кроме собственно обмена данными, сетевые службы должны решать и другие, более специфические задачи, например, задачи, порождаемые распределенной обработкой данных. К таким задачам относятся обеспечение непротиворечивости нескольких копий данных, размещенных на разных машинах (служба репликации), или организация выполнения одной задачи параллельно на нескольких машинах сети (служба вызова удаленных процедур). Среди сетевых служб можно выделить административные, то есть такие, которые в основном ориентированы не на простого пользователя, а на администратора и служат для организации правильной работы сети в целом. Служба администрирования учетных записей о пользователях, которая позволяет администратору вести общую базу данных о пользователях сети, система мониторинга сети, позволяющая захватывать и анализировать сетевой трафик, служба безопасности, в функции которой может входить среди прочего выполнение процедуры логического входа с последующей проверкой пароля, — все это примеры административных служб.

Реализация сетевых служб осуществляется программными средствами. Основные службы — файловая служба и служба печати — обычно предоставляются сетевой операционной системой, а вспомогательные, например служба баз данных, факса или передачи голоса, — системными сетевыми приложениями или утилитами, работающими в тесном контакте с сетевой ОС. Вообще говоря, распределение служб между ОС и утилитами достаточно условно и меняется в конкретных реализациях ОС.

При разработке сетевых служб приходится решать проблемы, которые свойственны любым распределенным приложениям: определение протокола взаимодействия между клиентской и серверной частями, распределение функций между ними, выбор схемы адресации приложений и др.

Одним из главных показателей качества сетевой службы является ее удобство. Для одного и того же ресурса может быть разработано несколько служб, по-разному решающих в общем-то одну и ту же задачу. Отличия могут заключаться в производительности или в уровне удобства предоставляемых услуг. Например, файловая служба может быть основана на использовании команды передачи файла из одного компьютера в другой по имени файла, а это требует от пользователя знания имени нужного файла. Та же файловая служба может быть реализована и так, что пользователь монтирует удаленную файловую систему к локальному каталогу, а далее обращается к удаленным файлам как к своим собственным, что гораздо более удобно. Качество сетевой службы зависит и от качества пользовательского интерфейса — интуитивной понятности, наглядности, рациональности.

При определении степени удобства разделяемого ресурса часто употребляют термин «прозрачность». Прозрачный доступ — это такой доступ, при котором пользователь не замечает, где расположен нужный ему ресурс — на его компьютере или на удаленном. После того как он смонтировал удаленную файловую систему в свое дерево каталогов, доступ к удаленным файлам становится для него совершенно прозрачным. Сама операция монтирования также может иметь разную степень прозрачности — в сетях с меньшей прозрачностью пользователь должен знать и задавать в команде имя компьютера, на котором хранится удаленная файловая система, в сетях с большей степенью прозрачности соответствующий программный компонент сети производит поиск разделяемых томов файлов безотносительно мест их хранения, а затем предоставляет их пользователю в удобном для него виде, например в виде списка или набора пиктограмм.

Для обеспечения прозрачности важен способ адресации (именования) разделяемых сетевых ресурсов. Имена разделяемых сетевых ресурсов не должны зависеть от их физического расположения на том или ином компьютере. В идеале пользователь не должен ничего менять в своей работе, если администратор сети переместил том или каталог с одного компьютера на другой. Сам администратор и сетевая операционная система имеют информацию о расположении файловых систем, но от пользователя она скрыта. Такая степень прозрачности пока редко встречается в сетях, — обычно для получения доступа к ресурсам определенного компьютера сначала приходится устанавливать с ним логическое соединение. Такой подход применяется, например, в сетях Windows NT.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем разница в соединениях между компьютерами в сети и соединениями компьютера с внешними устройствами машины?
2. Каковы виды представления дискретной информации в компьютерной сети.
3. Перечислите основные недостатки соединений — полносвязных, с общей шиной, звездообразных и кольцевых соединений.
4. Каковы задачи совместного использования линий связи в сетях?
5. Начертите топологию одного Ethernet-сегмента, построенного на основе концентратора.
6. В чем разница между физической и логической структурами сети?
7. Какие задачи выполняют средства коммуникации сети — распределители, концентраторы, мосты, коммутаторы, маршрутизаторы?
8. Объясните смысл термина «прозрачность».

## ГЛАВА 7. ПРОБЛЕМЫ СТАНДАРТИЗАЦИИ ПРИ ПОСТРОЕНИИ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Универсальный тезис о пользе стандартизации, справедливый для всех отраслей, в компьютерных сетях приобретает особое значение. Суть сети – это соединение разного оборудования, а значит, проблема совместимости является одной из наиболее острых. Без принятия всеми производителями общепринятых правил построения оборудования прогресс в деле «строительства» сетей был бы невозможен. Поэтому все развитие компьютерной отрасли в конечном счете отражено в стандартах – любая новая технология только тогда приобретает «законный» статус, когда ее содержание закрепляется в соответствующем стандарте.

В компьютерных сетях идеологической основой стандартизации является многоуровневый подход к разработке средств сетевого взаимодействия. Именно на основе этого подхода была разработана стандартная семиуровневая модель взаимодействия открытых систем, ставшая своего рода универсальным языком сетевых специалистов.

### 7.1. Многоуровневый подход. Протокол. Интерфейс. Стек протоколов

Организация взаимодействия между устройствами в сети является сложной задачей. Как известно, для решения сложных задач используется универсальный прием – декомпозиция, то есть разбиение одной сложной задачи на несколько более простых задач-модулей (рис. 7.1). Процедура декомпозиции включает в себя четкое определение функций каждого модуля, решавшего отдельную задачу, и интерфейсов между ними. В результате достигается логическое упрощение задачи, а кроме того, появляется возможность модификации отдельных модулей без изменения остальной части системы.

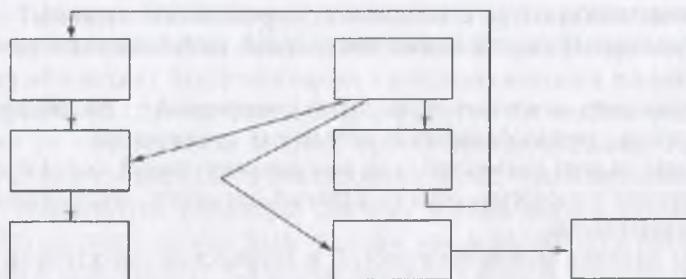


Рис. 7.1. Пример декомпозиции задачи.

При декомпозиции часто используют многоуровневый подход. Он заключается в следующем. Все множество модулей разбивают на уровни. Уровни образуют иерархию, то есть имеются вышележащие и нижележащие уровни (рис. 7.2). Множество модулей, составляющих каждый уровень, сформировано таким образом, что для выполнения своих задач они обращаются с запросами только к модулям непосредственно примыкающего нижележащего уровня. С другой стороны, результаты работы всех модулей, принадлежащих некоторому уровню, могут быть переданы только модулям соседнего вышележащего уровня. Такая иерархическая декомпозиция задачи предполагает четкое определение функций каждого уровня и интерфейсов между уровнями. Интерфейс определяет набор функций, которые нижележащий уровень предоставляет вышележащему. В результате иерархической декомпозиции достигается относительная независимость уровней, а значит, и возможность их легкой замены.

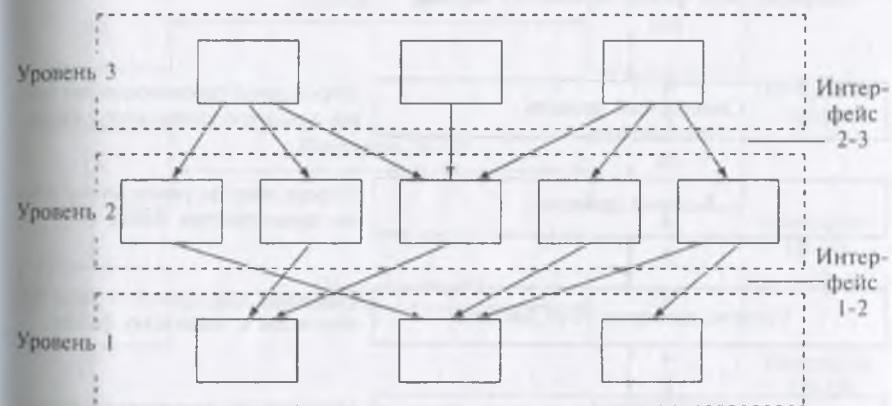


Рис. 7.2. Многоуровневый подход — создание иерархии задач.

Средства сетевого взаимодействия, конечно, тоже могут быть представлены в виде иерархически организованного множества модулей. При этом модули нижнего уровня могут, например, решать все вопросы, связанные с надежной передачей электрических сигналов между двумя соседними узлами. Модули более высокого уровня организуют транспортировку сообщений в пределах всей сети, пользуясь для этого средствами упомянутого нижележащего уровня. А на верхнем уровне работают модули, предоставляющие пользователям доступ к различным службам — файловой, печати и т. п. Конечно, это только один из множества возможных вариантов деления общей задачи организации сетевого взаимодействия на частные подзадачи.

Многоуровневый подход к описанию и реализации функций системы применяется не только в отношении сетевых средств. Такая модель функционирования используется, например, в локальных файловых системах, когда поступивший запрос на доступ к файлу последовательно обрабатывается несколькими программными уровнями (рис. 7.3). Запрос вначале анализируется верхним уровнем, на котором осуществляется последовательный разбор составного символьного имени файла и определение уникального идентификатора файла. Следующий уровень находит по уникальному имени все основные характеристики файла: адрес, атрибуты доступа и т. п. Затем на более низком уровне осуществляется проверка прав доступа к этому файлу, а далее, после расчета координат области файла, содержащей требуемые данные, выполняется физический обмен с внешним устройством с помощью драйвера диска.

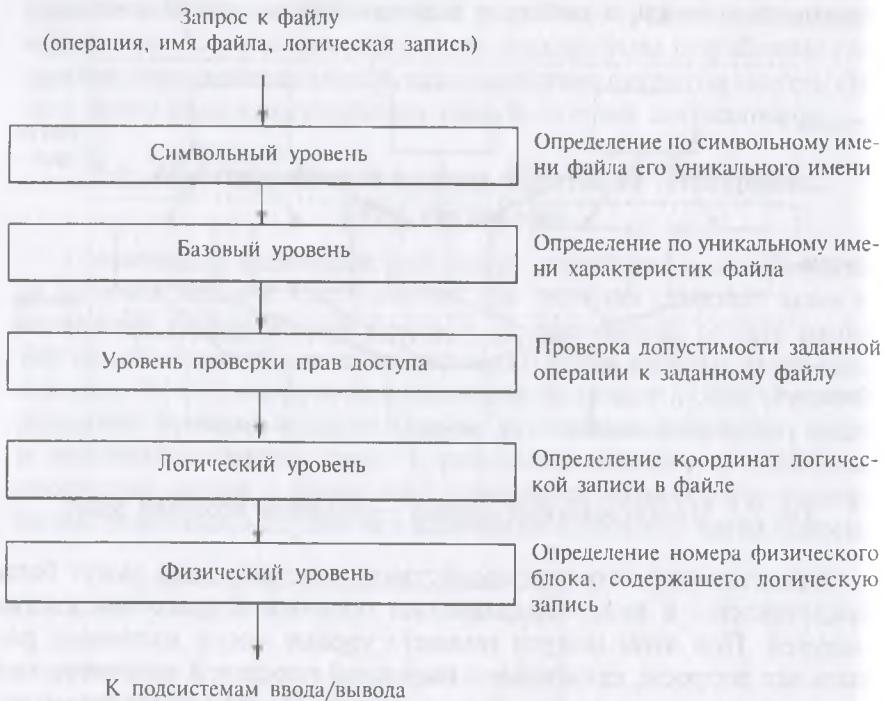


Рис. 7.3. Многоуровневая модель файловой системы.

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют *две* машины, то есть в данном случае необходимо организовать согласованную работу двух «иерархий». При передаче сообщений оба участника сетевого обмена должны при-

нять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения длины сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты для всех уровней, начиная от самого низкого – уровня передачи битов – до самого высокого, реализующего сервис для пользователей сети.

На рис. 7.4 показана модель взаимодействия двух узлов. С каждой стороны средства взаимодействия представлены четырьмя уровнями. Процедура взаимодействия этих двух узлов может быть описана в виде набора правил взаимодействия каждой пары соответствующих уровней обеих участвующих сторон. Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются *протоколом*.



Рис. 7.4. Взаимодействие двух узлов.

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. Эти правила принято называть *интерфейсом*. Интерфейс определяет набор сервисов, предоставляемый данным уровнем соседнему уровню. В сущности, протокол и интерфейс выражают одно и то же понятие, но традиционно в сетях за ними закрепили разные области действия: протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы – модулей соседних уровней в одном узле.

Средства каждого уровня должны отрабатывать, во-первых, свой собственный протокол, а во-вторых, интерфейсы с соседними уровнями.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется *стеком коммуникационных протоколов*.

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней — как правило, чисто программными средствами.

Программный модуль, реализующий некоторый протокол, часто для краткости также называют «протоколом». При этом соотношение между протоколом — формально определенной процедурой и протоколом — программным модулем, реализующим эту процедуру, аналогично соотношению между алгоритмом решения некоторой задачи и программой, решающей эту задачу.

Понятно, что один и тот же алгоритм может быть запрограммирован с разной степенью эффективности. Точно так же и протокол может иметь несколько программных реализаций. Именно поэтому при сравнении протоколов следует учитывать не только логику их работы, но и качество программных решений. Более того, на эффективность взаимодействия устройств в сети влияет качество всей совокупности протоколов, составляющих стек, в частности, насколько рационально распределены функции между протоколами разных уровней и насколько хорошо определены интерфейсы между ними.

Протоколы реализуются не только компьютерами, но и другими сетевыми устройствами — концентраторами, мостами, коммутаторами, маршрутизаторами и т. д. Действительно, в общем случае связь компьютеров в сети осуществляется не напрямую, а через различные коммуникационные устройства. В зависимости от типа устройства в нем должны быть встроенные средства, реализующие тот или иной набор протоколов.

Чтобы еще раз пояснить понятия «протокол» и «интерфейс», рассмотрим пример, не имеющий отношения к вычислительным системам, а именно обсудим взаимодействие двух предприятий А и В, связанных между собой деловым сотрудничеством. Между предприятиями существуют многочисленные договоренности и соглашения, такие, например, как регулярные поставки продукции одного предприятия другому. В соответствии с этой договоренностью начальник отдела продаж предприятия А регулярно в начале каждого месяца посыпает официальное сообщение начальнику отдела закупок предприятия В о том, сколько и какого товара может быть поставлено в этом месяце. В ответ на это сообщение начальник отдела заку-

ПОК предприятия В посыпает в ответ заявку установленного образца на требуемое количество продукции. Возможно, процедура взаимодействия этих начальников включает дополнительные согласования, в любом случае существует установленный порядок взаимодействия, который можно считать «протоколом уровня начальников». Начальники посыпают свои сообщения и заявки через своих секретарей. Порядок взаимодействия начальника и секретаря соответствует понятию многоуровневого интерфейса «начальник — секретарь». На предприятии А обмен документами между начальником и секретарем идет через специальную папку, а на предприятии В начальник общается с секретарем по факсу. Таким образом, интерфейсы «начальник — секретарь» на этих двух предприятиях отличаются.

После того как сообщения переданы секретарям, начальников не волнует, каким образом эти сообщения будут перемещаться дальше — обычной или электронной почтой, факсом или нарочным. Выбор способа передачи — это уровень компетенции секретарей, они могут решать этот вопрос, не уведомляя об этом своих начальников, так как их протокол взаимодействия связан только с передачей сообщений, поступающих сверху, и не касается содержания этих сообщений. На рис. 7.5 показано, что в качестве протокола взаимодействия «секретарь — секретарь» используется обмен письмами. При решении других вопросов начальники могут взаимодействовать по другим правилам-протоколам, но это не повлияет на работу секретарей, для которых не важно, какие сообщения отправлять, а важно, чтобы они дошли до адресата. Итак, в данном случае мы имеем дело с двумя уровнями — уровнем начальников и уровнем секретарей, и каждый из них имеет собственный протокол, который может быть изменен независимо от протокола другого уровня. Эта независимость протоколов друг от друга и делает привлекательным многоуровневый подход.



Рис. 7.5. Пример многоуровневого взаимодействия предприятий.

## 7.2. Модель OSI

Из того, что протокол является соглашением, принятым двумя взаимодействующими объектами, в данном случае двумя работающими в сети компьютерами, совсем не следует, что он обязательно является стандартным. Но на практике при реализации сетей стремятся использовать стандартные протоколы. Это могут быть фирменные, национальные или международные стандарты.

В начале 80-х годов ряд международных организаций по стандартизации — ISO, ITU-Т и некоторые другие — разработали модель, которая сыграла значительную роль в развитии сетей. Эта модель называется *моделью взаимодействия открытых систем (Open System Interconnection, OSI)* или моделью OSI. Модель OSI определяет различные уровни взаимодействия систем, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень. Модель OSI была разработана на основании большого опыта, полученного при создании компьютерных сетей, в основном глобальных, в 70-е годы. Полное описание этой модели занимает более 1000 страниц текста.

В модели OSI (рис. 7.6) средства взаимодействия делятся на семь уровней: прикладной, представительный, сеансовый, транспортный, сетевой, канальный и физический. Каждый уровень имеет дело с одним определенным аспектом взаимодействия сетевых устройств.

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Свои собственные протоколы взаимодействия приложения реализуют, обращаясь к системным средствам. Поэтому необходимо различать уровень взаимодействия приложений и прикладной уровень.

Следует также иметь в виду, что приложение может взять на себя функции некоторых верхних уровней модели OSI. Например, некоторые СУБД имеют встроенные средства удаленного доступа к файлам. В этом случае приложение, выполняя доступ к удаленным ресурсам, не использует системную файловую службу; оно обходит верхние уровни модели OSI и обращается напрямую к системным средствам, ответственным за транспортировку сообщений по сети, которые располагаются на нижних уровнях модели OSI.

Итак, пусть приложение обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. Обычное сообщение состоит из заголовка и поля данных. Заголовок содержит служебную информацию, которую необходимо передать через сеть прикладному уровню ма-

шины-адресата, чтобы сообщить ему, какую работу надо выполнить. В нашем случае заголовок, очевидно, должен содержать информацию о месте нахождения файла и о типе операции, которую необходимо над ним выполнить. Поле данных сообщения может быть пустым или содержать какие-либо данные, например те, которые необходимо записать в удаленный файл. Но для того чтобы доставить эту информацию по назначению, предстоит решить еще много задач, ответственность за которые несут нижележащие уровни.

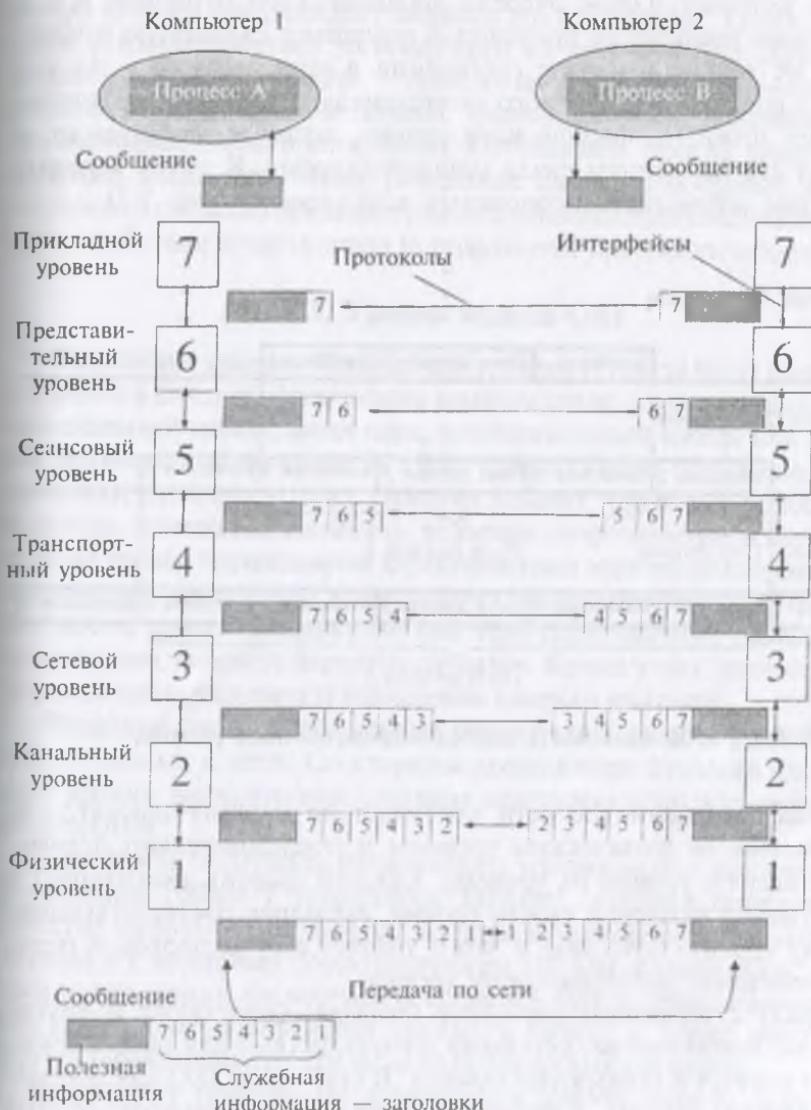


Рис. 7.6. Модель взаимодействия открытых систем ISO/OSI.

После формирования сообщения прикладной уровень направляет его вниз по стеку представительному уровню. Протокол представительного уровня на основании информации, полученной из заголовка прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию — заголовок представительного уровня, в котором содержатся указания для протокола представительного уровня машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который, в свою очередь, добавляет свой заголовок, и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в конце, в виде так называемого «концевика».) Наконец, сообщение достигает нижнего, физического уровня, который, собственно, и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 7.7).

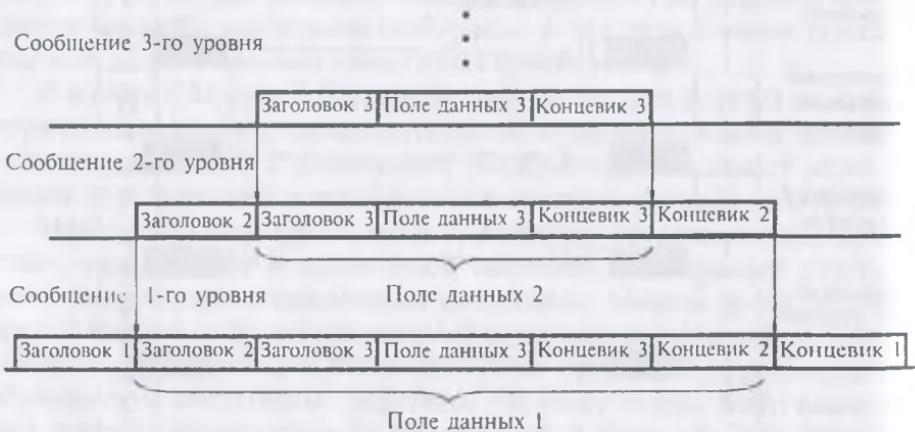


Рис. 7.7. Вложенность сообщений различных уровней.

Когда сообщение по сети поступает на машину-адресат, оно принимается ее физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие данному уровню функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

Наряду с термином *сообщение* (*message*) существуют и другие термины, применяемые сетевыми специалистами для обозначения единиц данных в процедурах обмена. В стандартах ISO для обозначения единиц данных, с которыми имеют дело протоколы разных уровней, используется общее название *протокольный блок данных*.

(Protocol Data Unit, PDU). Для обозначения блоков данных определенных уровней часто используются специальные названия: кадр (frame), пакет (packet), дейтаграмма (datagram), сегмент (segment).

В модели OSI различаются два основных типа протоколов. В протоколах *с установлением соединения* (*connection-oriented*) перед обменом данными отправитель и получатель должны сначала установить соединение и, возможно, выбрать некоторые параметры протокола, которые они будут использовать при обмене данными. После завершения диалога они должны разорвать это соединение. Телефон – это пример взаимодействия, основанного на установлении соединения. Вторая группа протоколов – протоколы *без предварительного установления соединения* (*connectionless*). Такие протоколы называются также *дейтаграммными* протоколами. Отправитель просто передает сообщение, когда оно готово. Опускание письма в почтовый ящик – это пример связи без предварительного установления соединения. При взаимодействии компьютеров используются протоколы обоих типов.

### 7.3. Уровни модели OSI

**Физический уровень.** Физический уровень (Physical layer) имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, передающих дискретную информацию, например, крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта.

Функции физического уровня реализуются во всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для предоставления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.

**Канальный уровень.** На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попарно

несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня (Data Link layer) является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые *кадрами (frames)*. Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, для его выделения, а также вычисляет контрольную сумму, обрабатывая все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит по сети, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка. Канальный уровень может не только обнаруживать ошибки, но и исправлять их за счет повторной передачи поврежденных кадров. Необходимо отметить, что функция исправления ошибок не является обязательной для канального уровня, поэтому в некоторых протоколах этого уровня она отсутствует, например, в Ethernet и frame relay.

В протоколах канального уровня, используемых в локальных сетях, заложена определенная структура связей между компьютерами и способы их адресации. Хотя канальный уровень и обеспечивает доставку кадра между любыми двумя узлами локальной сети, он это делает только в сети с совершенно определенной топологией связей, именно той топологией, для которой он был разработан. К таким типовым топологиям, поддерживаемым протоколами канального уровня локальных сетей, относятся общая шина, кольцо и звезда, а также структуры, полученные из них с помощью мостов и коммутаторов. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI, 100VG-AnyLAN.

В локальных сетях протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

В глобальных сетях, которые редко обладают регулярной топологией, канальный уровень часто обеспечивает обмен сообщениями только между двумя соседними компьютерами, соединенными индивидуальной линией связи. Примерами протоколов «точка-точка» (как часто называют такие протоколы) могут служить широко распространенные протоколы PPP и LAP-B. В таких случаях для

доставки сообщений между конечными узлами через всю сеть используются средства сетевого уровня. Именно так организованы сети X.25. Иногда в глобальных сетях функции канального уровня в чистом виде выделить трудно, так как в одном и том же протоколе они объединяются с функциями сетевого уровня. Примерами такого подхода могут служить протоколы технологий ATM и frame relay.

В целом канальный уровень представляет собой весьма мощный и законченный набор функций по пересылке сообщений между узлами сети. В некоторых случаях протоколы канального уровня оказываются самодостаточными транспортными средствами и могут допускать работу поверх них непосредственно протоколов прикладного уровня или приложений, без привлечения средств сетевого и транспортного уровней. Например, существует реализация протокола управления сетью SNMP непосредственно поверх Ethernet, хотя стандартно этот протокол работает поверх сетевого протокола IP и транспортного протокола UDP. Естественно, что применение такой реализации будет ограниченным — она не подходит для составных сетей разных технологий, например Ethernet и X.25, и даже для такой сети, в которой во всех сегментах применяется Ethernet, но между сегментами существуют петлевидные связи. А вот в двухсегментной сети Ethernet, объединенной мостом, реализация SNMP над канальным уровнем будет вполне работоспособна.

Тем не менее для обеспечения качественной транспортировки сообщений в сетях любых топологий и технологий функций канального уровня оказывается недостаточно, поэтому в модели OSI решение этой задачи возлагается на два следующих уровня — сетевой и транспортный.

**Сетевой уровень.** Сетевой уровень (Network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать совершенно различные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей. Функции сетевого уровня достаточно разнообразны. Начнем их рассмотрение на примере объединения локальных сетей.

Протоколы канального уровня локальных сетей обеспечивают доставку данных между любыми узлами только в сети с соответствующей типовой топологией, например топологией иерархической звезды. Это очень жесткое ограничение, которое не позволяет строить сети с развитой структурой, например, сети, объединяющие несколько сетей предприятия в единую сеть, или высокона-

дежные сети, в которых существуют избыточные связи между узлами. Можно было бы усложнять протоколы канального уровня для поддержания петлевидных избыточных связей, но принцип разделения обязанностей между уровнями приводит к другому решению. Чтобы, с одной стороны, сохранить простоту процедур передачи данных для типовых топологий, а с другой — допустить использование произвольных топологий, вводится дополнительный сетевой уровень.

На сетевом уровне сам термин *сеть* наделяют специфическим значением. В данном случае под сетью понимается совокупность компьютеров, соединенных между собой в соответствии с одной из стандартных типовых топологий и использующих для передачи данных один из протоколов канального уровня, определенный для этой топологии.

Внутри сети доставка данных обеспечивается соответствующим канальным уровнем, а вот доставкой данных между сетями занимается сетевой уровень, который и поддерживает возможность правильного выбора маршрута передачи сообщения даже в том случае, когда структура связей между составляющими сетями имеет характер, отличный от принятого в протоколах канального уровня.

Сети соединяются между собой специальными устройствами, называемыми маршрутизаторами. *Маршрутизатор* — это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересыпает пакеты сетевого уровня в сеть назначения. Чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество *транзитных передач* между сетями, или *хопов* (от *hop* — прыжок), каждый раз выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет.

На рис. 7.8. показаны четыре сети, связанные тремя маршрутизаторами. Между узлами А и В данной сети пролегают два маршрута: первый через маршрутизаторы 1 и 3, а второй через маршрутизаторы 1, 2 и 3.

Проблема выбора наилучшего пути называется *маршрутизацией*, и ее решение является одной из главных задач сетевого уровня. Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгорит-

мы маршрутизации пытаются приспособиться к изменению нагрузки, в то время как другие принимают решения на основе средних показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например надежности передачи.

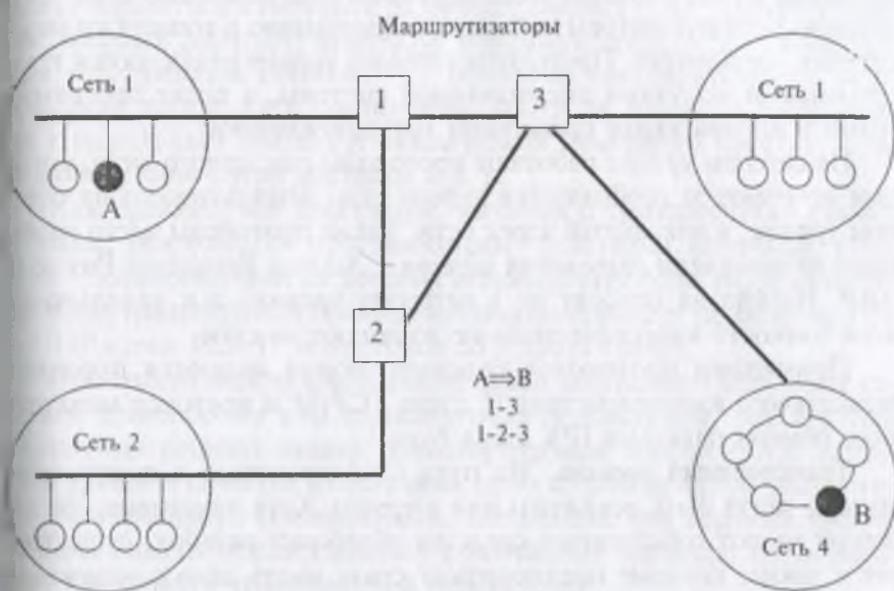


Рис. 7.8. Пример составной сети.

В общем случае функции сетевого уровня шире, чем функции передачи сообщений по связям с нестандартной структурой, которые мы сейчас рассмотрели на примере объединения нескольких локальных сетей. Сетевой уровень решает также задачи согласования разных технологий, упрощения адресации в крупных сетях и создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

Сообщения сетевого уровня принято называть *пакетами (packets)*. При организации доставки пакетов на сетевом уровне используется понятие «номер сети». В этом случае адрес получателя состоит из старшей части — номера сети и младшей — номера узла в этой сети. Все узлы одной сети должны иметь одну и ту же старшую часть адреса, поэтому термину «сеть» на сетевом уровне можно дать и другое, более формальное определение: сеть — это совокупность узлов, сетевой адрес которых содержит один и тот же номер сети.

На сетевом уровне определяются два вида протоколов. Первый вид — *сетевые протоколы (routed protocols)* — реализуют продвижение

ние пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых протоколами обмена маршрутной информацией или просто *протоколами маршрутизации (routing protocols)*. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений. Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.

На сетевом уровне работают протоколы еще одного типа, которые отвечают за отображение адреса узла, используемого на сетевом уровне, в локальный адрес сети. Такие протоколы часто называют *протоколами разрешения адресов – Address Resolution Protocol ARP*. Иногда их относят не к сетевому уровню, а к канальному, хотя тонкости классификации не изменяют их сути.

Примерами протоколов сетевого уровня являются протокол межсетевого взаимодействия IP стека TCP/IP и протокол межсетевого обмена пакетами IPX стека Novell.

**Транспортный уровень.** На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Транспортный уровень (*Transport layer*) обеспечивает приложениям или верхним уровням стека – прикладному и сеансовому – передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное – способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Выбор класса сервиса транспортного уровня определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких уровней, а с другой стороны, этот выбор зависит от того, насколько надежной является система транспортировки данных в сети, обеспечиваемая уровнями, расположенными ниже транспортного – сетевым, канальным и физическим. Так, например, если качество каналов передачи связи очень высокое и вероятность возникновения ошибок, не обнаруженных протоколами

более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня, не обремененных многочисленными проверками, квитированием и другими приемами повышения надежности. Если же транспортные средства нижних уровней изначально очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок, – с помощью предварительного установления логического соединения, контроля доставки сообщений по контрольным суммам и циклической нумерации пакетов, установления тайм-аутов доставки и т. п.

Как правило, все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети – компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протоколы TCP и UDP стека TCP/IP и протокол SPX стека Novell.

Протоколы нижних четырех уровней обобщенно называют сетевым транспортом или транспортной подсистемой, так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями. Остальные три верхних уровня решают задачи предоставления прикладных сервисов на основании имеющейся транспортной подсистемы.

**Сеансовый уровень.** Сеансовый уровень (*Session layer*) обеспечивает управление диалогом: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

**Представительный уровень.** Представительный уровень (*Presentation layer*) имеет дело с формой предоставления передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня предоставления информации, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На этом уровне может выполняться

шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол Secure Socket Layer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

**Прикладной уровень.** Прикладной уровень (Application layer) – это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например, с помощью протокола электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется *сообщением* (*message*).

Существует очень большое разнообразие служб прикладного уровня. Приведем в качестве примера хотя бы несколько наиболее распространенных реализаций файловых служб: NCP в операционной системе Novell NetWare, SMB в Microsoft Windows NT, NFS, FTP и TFTP, входящие в стек TCP/IP.

**Сетезависимые и сетенезависимые уровни.** Функции всех уровней модели OSI могут быть отнесены к одной из двух групп: либо к функциям, зависящим от конкретной технической реализации сети, либо к функциям, ориентированным на работу с приложениями.

Три нижних уровня – физический, канальный и сетевой – являются сетезависимыми, то есть протоколы этих уровней тесно связаны с технической реализацией сети и используемым коммуникационным оборудованием. Например, переход на оборудование FDDI означает полную смену протоколов физического и канального уровней во всех узлах сети.

Три верхних уровня – прикладной, представительный и сеансовый – ориентированы на приложения и мало зависят от технических особенностей построения сети. На протоколы этих уровней не влияют какие бы то ни было изменения в топологии сети, замена оборудования или переход на другую сетевую технологию. Так, переход от Ethernet на высокоскоростную технологию IOOVG-AnyLAN не потребует никаких изменений в программных средствах, реализующих функции прикладного, представительного и сеансового уровней.

Транспортный уровень является промежуточным, он скрывает все детали функционирования нижних уровней от верхних. Это позволяет разрабатывать приложения, не зависящие от технических средств непосредственной транспортировки сообщений.

На рис. 7.9. показаны уровни модели OSI, на которых работают различные элементы сети. Компьютер с установленной на нем се-

Компьютер А

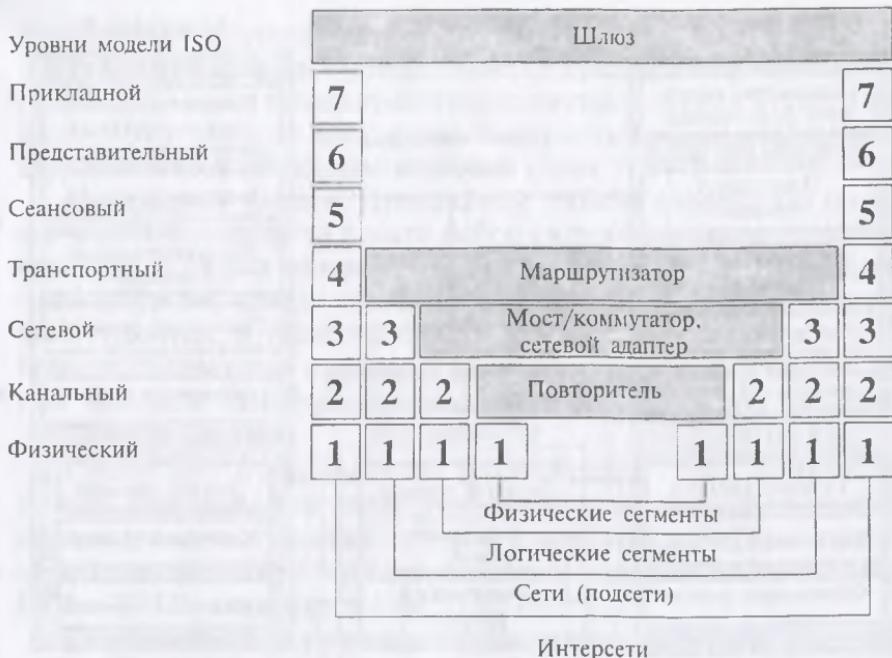
Компьютер В



Рис. 7.9. Сетевозависимые и систенезависимые уровни модели OSI.

тевой ОС взаимодействует с другим компьютером с помощью протоколов всех семи уровней. Это взаимодействие компьютеры осуществляют опосредованно — через различные коммуникационные устройства: концентраторы, модемы, мосты, коммутаторы, маршрутизаторы, мультиплексоры. В зависимости от типа коммуникационное устройство может работать либо только на физическом уровне (повторитель), либо на физическом и канальном (мост), либо на физическом, канальном и сетевом, иногда захватывая и транспортный уровень (маршрутизатор). На рис. 7.10 показано соответствие функций различных коммуникационных устройств уровням модели OSI.

Модель OSI представляет хотя и очень важную, но только одну из многих моделей коммуникаций. Эти модели и связанные с ними стеки протоколов могут отличаться количеством уровней, их функциями, форматами сообщений, службами, поддерживаемыми на верхних уровнях, и прочими параметрами.



Интерсети

Рис. 7.10. Соответствие функции различных устройств сети уровням модели ISO.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что вы понимаете под термином «открытая система»?
2. Что такое «протокол» и «интерфейс» и какая между ними разница?
3. Что стандартизируется при использовании OSI-модели?
4. Что стандартизируется при применении OSI-стека?
5. Для чего в модели OSI сформированы семь уровней?
6. Дайте краткую характеристику каждого уровня OSI-модели, приведите примеры использования стандартных протоколов на каждом из уровней.

## ГЛАВА 8. ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ СЕТИ

Для классификации компьютерных сетей используются различные признаки, но чаще всего сети делят на типы по территориальному признаку, то есть по величине территории, которую покрывает сеть. И для этого есть веские причины, так как отличия технологий локальных и глобальных сетей очень значительны, несмотря на их постоянное сближение.

## 8.1. Особенности локальных, глобальных и городских сетей

К локальным сетям – *Local Area Networks (LAN)* – относят сети компьютеров, сосредоточенные на небольшой территории (обычно в радиусе не более 1-2 км). В общем случае локальная сеть представляет собой коммуникационную систему, принадлежащую одной организации. Из-за коротких расстояний в локальных сетях имеется возможность использования относительно дорогих высококачественных линий связи, которые позволяют, применяя простые методы передачи данных, достигать высоких скоростей обмена данными порядка 100 Мбит/с. В связи с этим услуги, предоставляемые локальными сетями, отличаются широким разнообразием и обычно предусматривают реализацию в режиме on-line.

Глобальные сети – *Wide Area Networks (WAN)* – объединяют территориально рассредоточенные компьютеры, которые могут находиться в различных городах и странах. Так как прокладка высококачественных линий связи на большие расстояния обходится очень дорого, в глобальных сетях часто используются уже существующие линии связи, изначально предназначенные совсем для других целей. Например, многие глобальные сети строятся на основе телефонных и телеграфных каналов общего назначения. Из-за низких скоростей таких линий связи в глобальных сетях (десятки килобит в секунду) набор предоставляемых услуг обычно ограничивается передачей файлов, преимущественно не в оперативном, а в фоновом режиме, с использованием электронной почты. Для устойчивой передачи дискретных данных по некачественным линиям связи применяются методы и оборудование, существенно отличающиеся от методов и оборудования, характерных для локальных сетей. Как правило, здесь применяются сложные процедуры контроля и восстановления данных, так как наиболее типичный режим передачи данных по территориальному каналу связи связан со значительными искажениями сигналов.

Городские сети (или сети мегаполисов) – *Metropolitan Area Networks (MAN)* – являются менее распространенным типом сетей. Эти сети появились сравнительно недавно. Они предназначены для обслуживания территории крупного города – мегаполиса. В то время как локальные сети наилучшим образом подходят для разделения ресурсов на коротких расстояниях и широковещательных передач, а глобальные сети обеспечивают работу на больших расстояниях, но с ограниченной скоростью и небогатым набором услуг, сети мегаполисов занимают некоторое промежуточное положение. Они используют цифровые магистральные линии связи, часто оптоволоконные, со скоростями от 45 Мбит/с, и предназначены для связи локальных сетей в масштабах города и соединения локальных сетей в глобаль-

ными. Эти сети первоначально были разработаны для передачи данных, но сейчас они предоставляют и такие услуги, как видеоконференции и интегральную передачу голоса и текста. Развитие технологии сетей мегаполисов осуществлялось местными телефонными компаниями. Исторически сложилось так, что местные телефонные компании всегда обладали слабыми техническими возможностями и из-за этого не могли привлечь крупных клиентов. Чтобы преодолеть свою отсталость и занять достойное место в мире локальных и глобальных сетей, местные предприятия связи занялись разработкой сетей на основе самых современных технологий, например технологии коммутации ячеек SMDS или ATM. Сети мегаполисов являются общественными сетями, и поэтому их услуги обходятся дешевле, чем построение собственной (частной) сети в пределах города.

## 8.2. Отличия локальных сетей от глобальных

Рассмотрим основные отличия локальных сетей от глобальных более детально. Так как в последнее время эти отличия становятся все менее заметными, то будем считать, что в данном разделе мы рассматриваем сети конца XX века, когда эти отличия проявлялись весьма отчетливо, а современные тенденции сближения технологий локальных и глобальных сетей будут рассмотрены в следующем разделе.

*Протяженность, качество и способ прокладки линий связи.* Класс локальных вычислительных сетей по определению отличается от класса глобальных сетей небольшим расстоянием между узлами сети. Это в принципе делает возможным использование в локальных сетях качественных линий связи: коаксиального кабеля, витой пары, оптоволоконного кабеля, которые не всегда доступны (из-за экономических ограничений) на больших расстояниях, свойственных глобальным сетям. В глобальных сетях часто применяются уже существующие линии связи (телефрафные или телефонные), а в локальных сетях они прокладываются заново.

*Сложность методов передачи и оборудования.* В условиях низкой надежности физических каналов в глобальных сетях требуются более сложные, чем в локальных сетях, методы передачи данных и соответствующее оборудование. Так, в глобальных сетях широко применяются модуляция, асинхронные методы, сложные методы контрольного суммирования, квитирование и повторные передачи искаженных кадров. С другой стороны, качественные линии связи в локальных сетях позволили упростить процедуры передачи данных за счет применения немодулированных сигналов и отказа от обязательного подтверждения получения пакета.

*Скорость обмена данными.* Одним из главных отличий локальных сетей от глобальных является наличие высокоскоростных ка-

налов обмена данными между компьютерами, скорость которых (10,16 и 100 Мбит/с) сравнима со скоростями работы устройств и узлов компьютера — дисков, внутренних шин обмена данными и т. п. За счет этого у пользователя локальной сети, подключенного к удаленному разделяемому ресурсу (например, диску сервера), складывается впечатление, что он пользуется этим диском, как «своим». Для глобальных сетей типичны гораздо более низкие скорости передачи данных — 2400,9600, 28800,33600 бит/с, 56 и 64 Кбит/с и только на магистральных каналах — до 2 Мбит/с.

*Разнообразие услуг.* Локальные сети предоставляют, как правило, широкий набор услуг — это различные виды услуг файловой службы, услуги печати, услуги службы передачи факсимильных сообщений, услуги баз данных, электронная почта и другие, в то время как глобальные сети в основном предоставляют почтовые услуги и иногда файловые услуги с ограниченными возможностями — передачу файлов из публичных архивов удаленных серверов без предварительного просмотра их содержания.

*Оперативность выполнения запросов.* Время прохождения пакета через локальную сеть обычно составляет несколько миллисекунд, время же его передачи через глобальную сеть может достигать нескольких секунд. Низкая скорость передачи данных в глобальных сетях затрудняет реализацию служб для режима on-line, который является обычным для локальных сетей.

*Разделение каналов.* В локальных сетях каналы связи используются, как правило, совместно сразу несколькими узлами сети, а в глобальных сетях — индивидуально.

*Использование метода коммутации пакетов.* Важной особенностью локальных сетей является неравномерное распределение нагрузки. Отношение пиковой нагрузки к средней может составлять 100:1 и даже выше. Такой трафик обычно называют *пульсирующим*. Из-за этой особенности трафика в локальных сетьях для связи узлов применяется метод коммутации пакетов, который для пульсирующего трафика оказывается гораздо более эффективным, чем традиционный ; для глобальных сетей — метод коммутации каналов. Эффективность метода коммутации пакетов состоит в том, что сеть в целом передает в единицу времени больше данных своих абонентов. В глобальных сетях метод коммутации пакетов также используется, но наряду с ним часто применяется и метод коммутации каналов, а также некоммутуемые каналы — как унаследованные технологии некомпьютерных сетей.

*Масштабируемость.* «Классические» локальные сети обладают плохой масштабируемостью из-за жесткости базовых топологий, определяющих способ подключения станций и длину линии. При

использовании многих базовых топологий характеристики сети резко ухудшаются при достижении определенного предела по количеству узлов или протяженности линий связи. Глобальным же сетям присуща хорошая масштабируемость, так как они изначально разрабатывались в расчете на работу с произвольными топологиями.

### 8.3. Тенденция к сближению локальных и глобальных сетей

Если принять во внимание все перечисленные выше различия локальных и глобальных сетей, то становится понятным, почему так долго могли существовать раздельно два сообщества специалистов, занимающиеся этими двумя видами сетей. Но за последние годы ситуация резко изменилась.

Специалисты по локальным сетям, перед которыми встали задачи объединения нескольких локальных сетей, расположенных в разных, географически удаленных друг от друга пунктах, были вынуждены начать освоение чуждого для них мира глобальных сетей и телекоммуникаций. Тесная интеграция удаленных локальных сетей не позволяет рассматривать глобальные сети в виде «черного ящика», представляющего собой только инструмент транспортировки сообщений на большие расстояния. Поэтому все, что связано с глобальными связями и удаленным доступом, стало предметом повседневного интереса многих специалистов по локальным сетям.

С другой стороны, стремление повысить пропускную способность, скорость передачи данных, расширить набор и оперативность служб, другими словами, стремление улучшить качество предоставляемых услуг – все это заставило специалистов по глобальным сетям обратить пристальное внимание на технологии, используемые в локальных сетях.

Таким образом, в мире локальных и глобальных сетей явно на-метилось движение навстречу друг другу, которое уже сегодня привело к значительному взаимопроникновению технологий локальных и глобальных сетей.

Одним из проявлений этого сближения является появление сетей масштаба большого города (MAN), занимающих промежуточное положение между локальными и глобальными сетями. При достаточно больших расстояниях между узлами они обладают качественными линиями связи и высокими скоростями обмена, даже более высокими, чем в классических локальных сетях. Как и в случае локальных сетей, при построении MAN уже существующие линии связи не используются, а прокладываются заново.

Сближение в методах передачи данных происходит на платформе оптической цифровой (немодулированной) передачи данных по оптоволоконным линиям связи. Из-за резкого улучшения качества

каналов связи в глобальных сетях начали отказываться от сложных и избыточных процедур обеспечения корректности передачи данных. Примером могут служить сети frame relay. В этих сетях предполагается, что искажение бит происходит настолько редко, что ошибочный пакет просто уничтожается, а все проблемы, связанные с его потерей, решаются программами прикладного уровня, которые непосредственно не входят в состав сети frame relay.

За счет новых сетевых технологий и, соответственно, нового оборудования, рассчитанного на более качественные линии связи, скорости передачи данных в уже существующих коммерческих глобальных сетях нового поколения приближаются к традиционным скоростям локальных сетей (в сетях frame relay сейчас доступны скорости 2 Мбит/с), а в глобальных сетях ATM и превосходят их, достигая 622 Мбит/с.

В результате службы для режима on-line становятся обычными и в глобальных сетях. Наиболее яркий пример — гипертекстовая информационная служба World Wide Web, ставшая основным поставщиком информации в сети Internet. Ее интерактивные просто позаимствовали эту службу у глобальных сетей. Процесс переноса служб и технологий из глобальных сетей в локальные приобрел такой массовый характер, что появился даже специальный термин — intranet-технологии (intra — внутренний), обозначающий применение служб внешних (глобальных) сетей во внутренних — локальных.

Локальные сети перенимают у глобальных сетей и транспортные технологии. Все новые скоростные технологии (Fast Ethernet, Gigabit Ethernet, 100VG-AnyLAN) поддерживают работу по индивидуальным линиям связи наряду с традиционными для локальных сетей разделямыми линиями. Для организации индивидуальных линий связи используется специальный тип коммуникационного оборудования — коммутаторы. Коммутаторы локальных сетей соединяются между собой по иерархической схеме, подобно тому, как это делается в телефонных сетях: имеются коммутаторы нижнего уровня, к которым непосредственно подключаются компьютеры сети, коммутаторы следующего уровня соединяют между собой коммутаторы нижнего уровня и т. д. Коммутаторы более высоких уровней обладают, как правило, большей производительностью и работают с более скоростными каналами, уплотняя данные нижних уровней. Коммутаторы поддерживают не только новые протоколы локальных сетей, но и традиционные — Ethernet и Token Ring.

В локальных сетях в последнее время уделяется такое же большое внимание методам обеспечения защиты информации от несанкционированного доступа, как и в глобальных сетях. Такое внимание обусловлено тем, что локальные сети перестали быть изо-

лированными, чаще всего они имеют выход в «большой мир» через глобальные связи. При этом часто используются те же методы — шифрование данных, аутентификация пользователей, введение защитных барьеров, предохраняющих от проникновения в сеть извне.

И наконец, появляются новые технологии, изначально предназначенные для обоих видов сетей. Наиболее ярким представителем нового поколения технологий является технология ATM, которая может служить основой не только локальных и глобальных компьютерных сетей, но и телефонных сетей, а также широковещательных видеосетей, объединяя все существующие типы трафика в одной транспортной сети.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое глобальная, региональная (городская), локальная сеть? Дайте краткую характеристику каждой из них.
2. Чем отличается локальная сеть от глобальной компьютерной сети?
3. Что мы имеем в виду, когда говорим о сближении локальной и глобальной сети?

## ГЛАВА 9. СЕТИ ОТДЕЛОВ, КАМПУСОВ И КОРПОРАЦИЙ

Еще одним популярным способом классификации сетей является их классификация по масштабу производственного подразделения, в пределах которого действует сеть. Различают сети отделов, сети кампусов и корпоративные сети.

### 9.1. Сети отделов

*Сети отделов* — это сети, которые используются сравнительно небольшой группой сотрудников, работающих в одном отделе предприятия. Эти сотрудники решают некоторые общие задачи, например ведут бухгалтерский учет или занимаются маркетингом. Считается, что отдел может насчитывать до 100–150 сотрудников.

Главной целью сети отдела является разделение локальных ресурсов, таких как приложения, данные, лазерные принтеры и модемы. Обычно сети отделов имеют один или два файловых сервера и не более тридцати пользователей (рис. 9.1). Сети отделов обычно не разделяются на подсети. В этих сетях локализуется большая часть трафика предприятия. Сети отделов обычно создаются на основе какой-либо одной сетевой технологии — Ethernet, Token Ring. Для такой сети характерен один или, максимум, два типа операционных систем. Чаще всего — это сеть с выделенным сервером, напри-

мер NetWare, хотя небольшое количество пользователей делает возможным использование одноранговых сетевых ОС, таких, например, как Windows 95.



Рис. 9.1. Пример сети масштаба отдела.

Задачи управления сетью на уровне отдела относительно просты: добавление новых пользователей, устранение простых отказов, инсталляция новых узлов и установка новых версий программного обеспечения. Такой сетью может управлять сотрудник, посвящающий обязанностям администратора только часть своего времени. Чаще всего администратор сети отдела не имеет специальной подготовки, но является тем человеком в отделе, который лучше всех разбирается в компьютерах, и само собой получается так, что он занимается администрированием сети.

Существует и другой тип сетей, близкий к сетям отделов, — *сети рабочих групп*. К таким сетям относят совсем небольшие сети, включающие до 10—20 компьютеров. Характеристики сетей рабочих групп практически не отличаются от описанных выше характеристик сетей отделов. Такие свойства, как простота сети и однородность, здесь проявляются в наибольшей степени, в то время как сети отделов могут приближаться в некоторых случаях к следующему по масштабу типу сетей — сетям кампусов.

## 9.2. Сети кампусов

*Сети кампусов* получили свое название от английского слова *campus* — студенческий городок. Именно на территории университетских городков часто возникла необходимость объединения нескольких мелких сетей в одну большую сеть. Сейчас это название не связывают со студенческими городками, а используют для обозначения сетей любых предприятий и организаций.

Главными особенностями сетей кампусов является то, что сети этого типа объединяют множество сетей различных отделов одного предприятия в пределах отдельного здания или в пределах одной территории, покрывающей площадь в несколько квадратных километров. При этом глобальные соединения в сетях кампусов не используются. Службы такой сети включают взаимодействие между сетями отделов, доступ к общим базам данных предприятия, доступ к общим факс-серверам, высокоскоростным модемам и высокоскоростным принтерам. В результате сотрудники каждого отдела предприятия получают доступ к некоторым файлам и ресурсам сетей других отделов. Важной службой, предоставляемой сетями кампусов, стал доступ к корпоративным базам данных независимо от того, на каких типах компьютеров они располагаются (рис. 9.2).

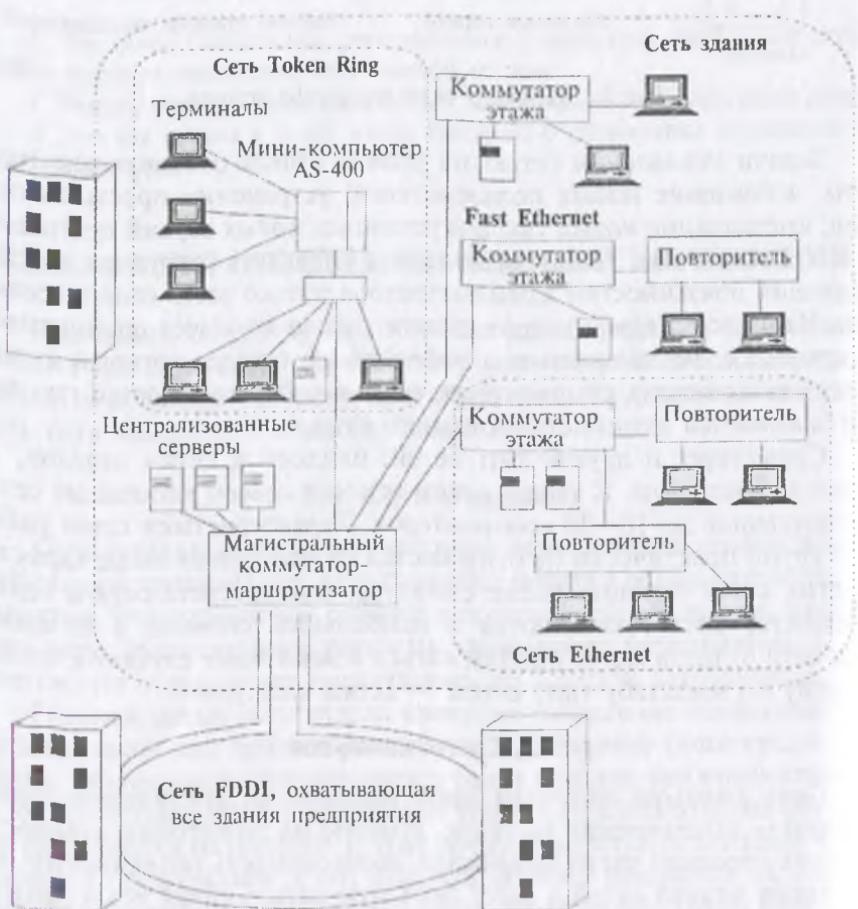


Рис.9.2. Пример сети кампusa.

Именно на уровне сети кампуса возникают проблемы интеграции неоднородного аппаратного и программного обеспечения. Типы компьютеров, сетевых операционных систем, сетевого аппаратного обеспечения могут отличаться в каждом отделе. Отсюда вытекают сложности управления сетями кампусов. Администраторы должны быть в этом случае более квалифицированными, а средства оперативного управления сетью — более совершенными.

### 9.3. Корпоративные сети

*Корпоративные сети* называют также сетями масштаба предприятия, что соответствует дословному переводу термина «enterprise-wide networks», используемого в англоязычной литературе для обозначения этого типа сетей. Сети масштаба предприятия (корпоративные сети) объединяют большое количество компьютеров на всех территориях отдельного предприятия. Они могут быть сложно связаны и покрывать город, регион или даже континент. Число пользователей и компьютеров может измеряться тысячами, а число серверов — сотнями, расстояния между сетями отдельных территорий могут оказаться такими, что становится необходимым использование глобальных связей (рис. 9.3). Для соединения удаленных локальных сетей и отдельных компьютеров в корпоративной сети применяются разнообразные телекоммуникационные средства, в том числе телефонные каналы, радиоканалы, спутниковая связь. Корпоративную сеть можно представить в виде «островков локальных сетей», плавающих в телекоммуникационной среде.

Непременным атрибутом такой сложной и крупномасштабной сети является высокая степень гетерогенности — нельзя удовлетворить потребности тысяч пользователей с помощью однотипных программных и аппаратных средств. В корпоративной сети обязательно будут использоваться различные типы компьютеров — от мэйнфреймов до персоналок, несколько типов операционных систем и множество различных приложений. Неоднородные части корпоративной сети должны работать как единое целое, предоставляя пользователям по возможности прозрачный доступ ко всем необходимым ресурсам.

Появление корпоративных сетей — это хорошая иллюстрация известного философского постулата о переходе количества в качество. При объединении отдельных сетей крупного предприятия, имеющего филиалы в разных городах и даже странах, в единую сеть многие количественные характеристики объединенной сети превосходят некоторый критический порог, за которым начинается новое качество. В этих условиях существующие методы и подходы к решению традиционных задач сетей меньших масштабов для корпоративных сетей оказались непригодными. На первый план вышли та-

кие задачи и проблемы, которые в сетях рабочих групп, отделов и даже кампусов либо имели второстепенное значение, либо вообще не проявлялись. Примером может служить простейшая (для небольших сетей) задача — ведение учетных данных о пользователях сети.

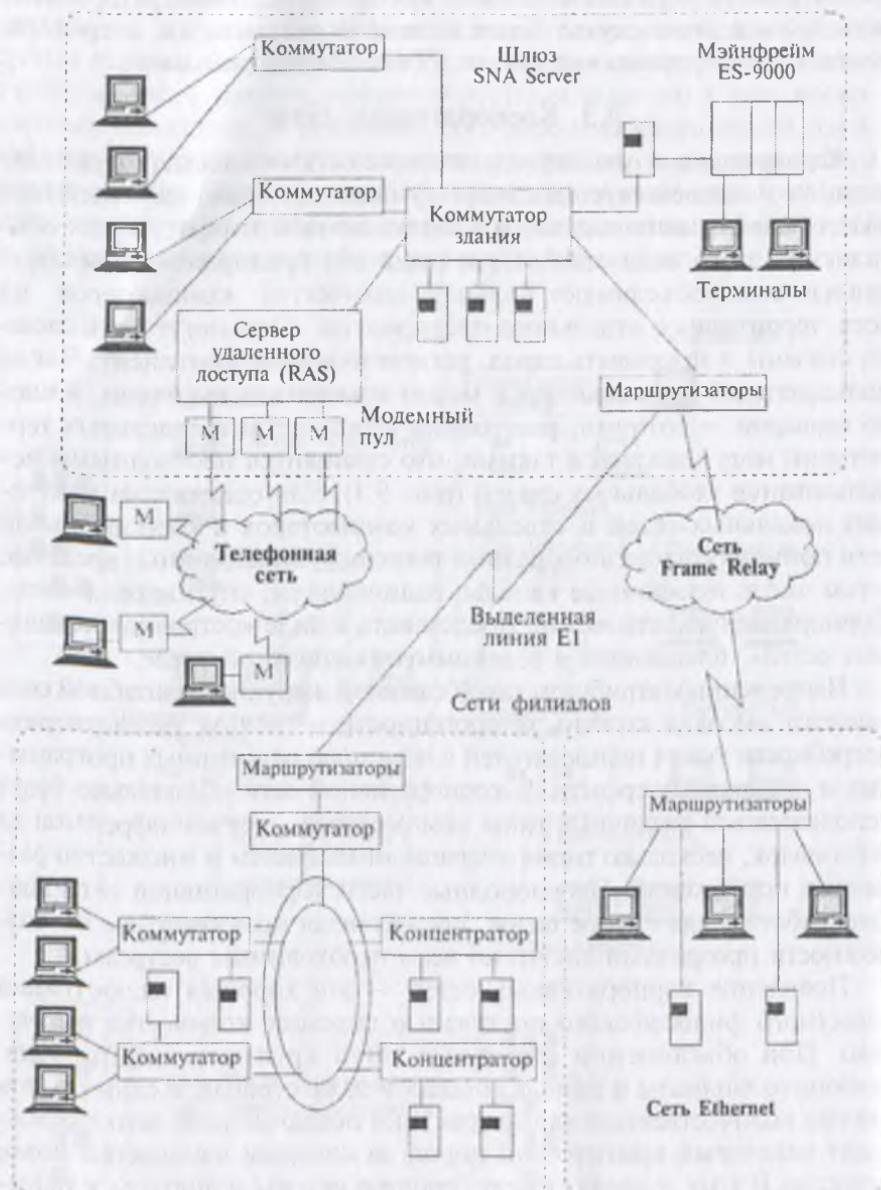


Рис. 9.3. Пример корпоративной сети.

Наиболее простой способ ее решения — помещение учетных данных каждого пользователя в локальную базу учетных данных каждого компьютера, к ресурсам которого пользователь должен иметь доступ. При попытке доступа эти данные извлекаются из локальной учетной базы и на их основе доступ предоставляется или не предоставляется. Для небольшой сети, состоящей из 5-10 компьютеров и примерно такого же количества пользователей, такой способ работает очень хорошо. Но если в сети насчитывается несколько тысяч пользователей, каждому из которых нужен доступ к нескольким десяткам серверов, то, очевидно, это решение становится крайне неэффективным. Администратор должен повторить несколько десятков раз операцию занесения учетных данных пользователя. Сам пользователь также вынужден повторять процедуру логического входа каждый раз, когда ему нужен доступ к ресурсам нового сервера. Хорошее решение этой проблемы для крупной сети — использование централизованной справочной службы, в базе данных которой хранятся учетные записи всех пользователей сети. Администратор один раз выполняет операцию занесения данных пользователя в эту базу, а пользователь один раз выполняет процедуру логического входа, причем не в отдельный сервер, а в сеть целиком.

При переходе от более простого типа сетей к более сложному — от сетей отдела к корпоративной сети — сеть должна быть все более надежной и отказоустойчивой, при этом требования к ее производительности также существенно возрастают. По мере увеличения масштабов сети увеличиваются и ее функциональные возможности. По сети циркулирует все возрастающее количество данных, и сеть должна обеспечивать их безопасность и защищенность наряду с доступностью. Соединения, обеспечивающие взаимодействие, должны быть более прозрачными. При каждом переходе на следующий уровень сложности компьютерное оборудование сети становится все более разнообразным, а географические расстояния увеличиваются, делая достижение целей более сложным; более проблемным и дорогостоящим становится управление такими соединениями.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каковы основные задачи сети отдела?
2. Дайте основные характеристики сети кампуса.
3. Чем характеризуются корпоративные сети?

## ГЛАВА 10. ЭКСПЛУАТАЦИЯ ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И СЕТЕЙ

Как и всякое вычислительное устройство, ЭВМ характеризуется *надежностью*, т. е. свойством выполнять установленные функции, сохраняя при этом характеристики, определяемые условиями эксплуатации. Надежность ЭВМ зависит от безотказности, ремонтопригодности и достоверности функционирования.

Под *безотказностью* понимается свойство ЭВМ функционировать без отказов, т. е. без событий, нарушающих нормальное функционирование ЭВМ, устранение которых требует вмешательства обслуживающего персонала для ремонта (замены, регулировки) неисправного узла. В том случае, если причина, вызвавшая функционирование ЭВМ, самоустранилась без вмешательства обслуживающего персонала, считают, что произошел сбой ЭВМ. Сбои в работе ЭВМ могут привести к получению неверных результатов при вычислениях. Для оценки влияния сбоев на работу ЭВМ служит понятие *достоверности* функционирования, характеризующееся безошибочностью производимых вычислений. Для повышения достоверности функционирования служат аппаратные и программные средства контроля ЭВМ. Под *ремонтопригодностью* понимается приспособленность ЭВМ к обнаружению и устранению отказов.

Для повышения достоверности функционирования электронных вычислительных машин и сетей используются аппаратные и программные средства контроля.

### 10.1. Аппаратные средства контроля ЭВМ

Аппаратные средства контроля вводятся в состав устройств ЭВМ и позволяют осуществлять контроль при передаче информации, выполнении над ней логических и арифметических операций, хранении ее.

*Контроль при передаче информации* обычно осуществляется на основе избыточных кодов, т. е. таких кодов, в которых кроме информационных разрядов имеется один или несколько контрольных разрядов. В ЭВМ наиболее широко применяются коды с дополнением до нечетности. В этом случае в контрольном разряде находится «1» или «0» в зависимости от числа единиц в коде: если число единиц в коде четное, то в контрольном разряде будет «1», в противоположном случае — «0». Обычно контрольный разряд присваивается каждому байту информации. Код с дополнением до нечетности позволяет обнаруживать одиноч-

ные ошибки в байте и все случаи возникновения нечетного числа ошибок. Недостатком этого кода является невозможность обнаружения двойных ошибок (когда в байте информации одновременно искажаются два разряда), а также четного числа ошибок. Код с дополнением до нечетности имеет небольшую избыточность, т. е. небольшое число контрольных разрядов, и, следовательно, не требует больших затрат оборудования на реализацию схем контроля. Наиболее широко этот код применяется при передаче информации с регистра на регистр, причем как внутри устройства ЭВМ, так и между ними.

Для построения схем определения числа единиц в байте информации, т. е. четности или нечетности числа, применяют логические элементы с парафазными входами, выполняющие операцию сложения по модулю два (обозначение  $M_2$  или (рис. 10.1, а, б)). На рис. 10.2 показана схема контроля байта информации. При этом каждый разряд байта задается прямым КР,  $x0-x7$ , и инверсным кодом КР,  $x0-\bar{x}-x7$  (где  $x$  – информационный, КР – контрольный разряды). Организация контроля слова или двойного слова строится по аналогичной схеме. Сначала раздельно производится контроль по байтам, а затем контроль слова или двойного слова.

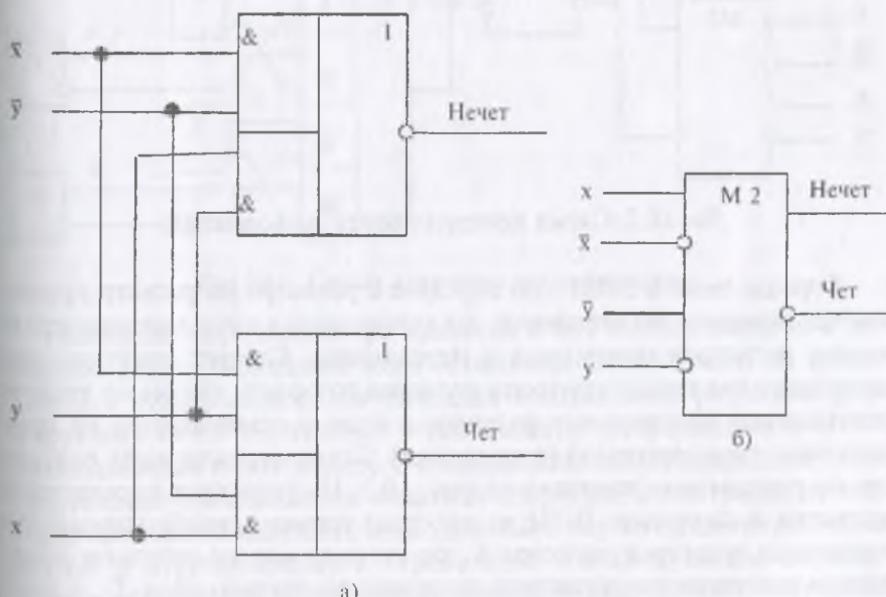


Рис. 10.1. Логические элементы сложения по модулю два.

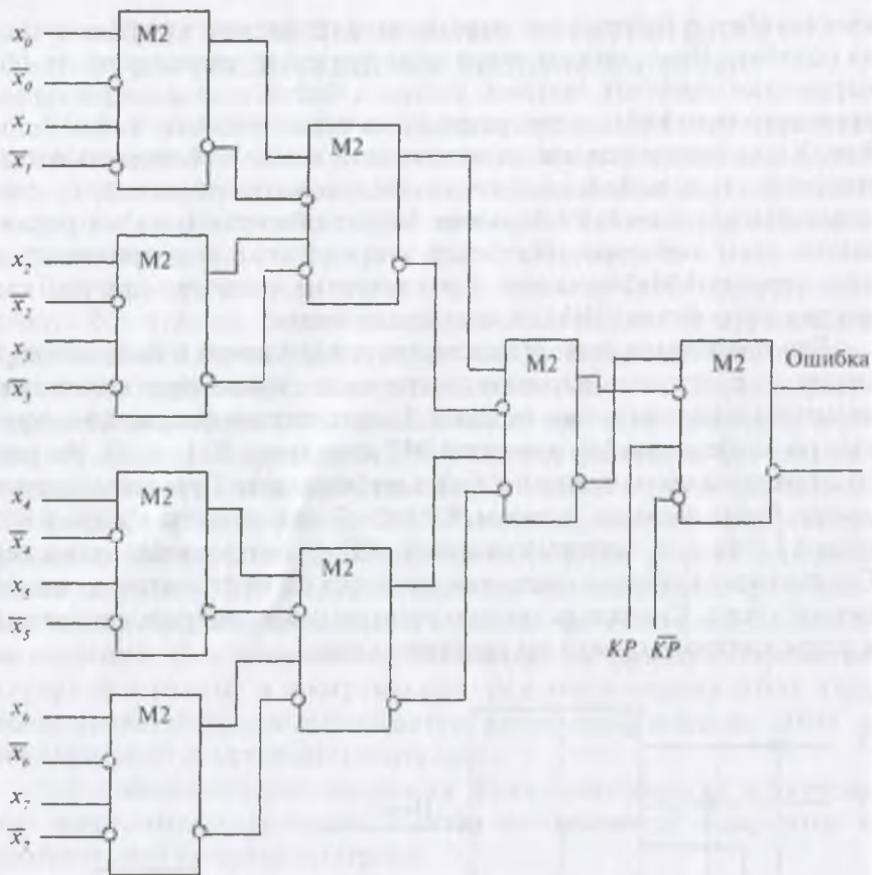


Рис. 10.2. Схема контроля байта информации.

Гораздо реже в ЭВМ при передаче с регистра на регистр применяется *контроль по совпадению*, заключающийся в поразрядном сравнении регистров приемника и передатчика. Следует отметить, что достоинством такого контроля является тот факт, что он не требует применения контрольных разрядов в коде, а основывается на применении схем сравнения (совпадения). Схемная реализация контроля по совпадению показана на рис. 10.3. Информация передается с регистра А на регистр В. Через интервал времени, необходимый для установки триггеров регистра В, по сигналу опроса результат сравнения содержимого регистров заносится на триггер сбоя  $T_{cb}$ .

Контроль по совпадению сложно применять в тех случаях, когда смена информации в регистрах производится с большой частотой, что характерно для современных ЭВМ. Это обстоятельство приводит к применению *контроля по четности*.

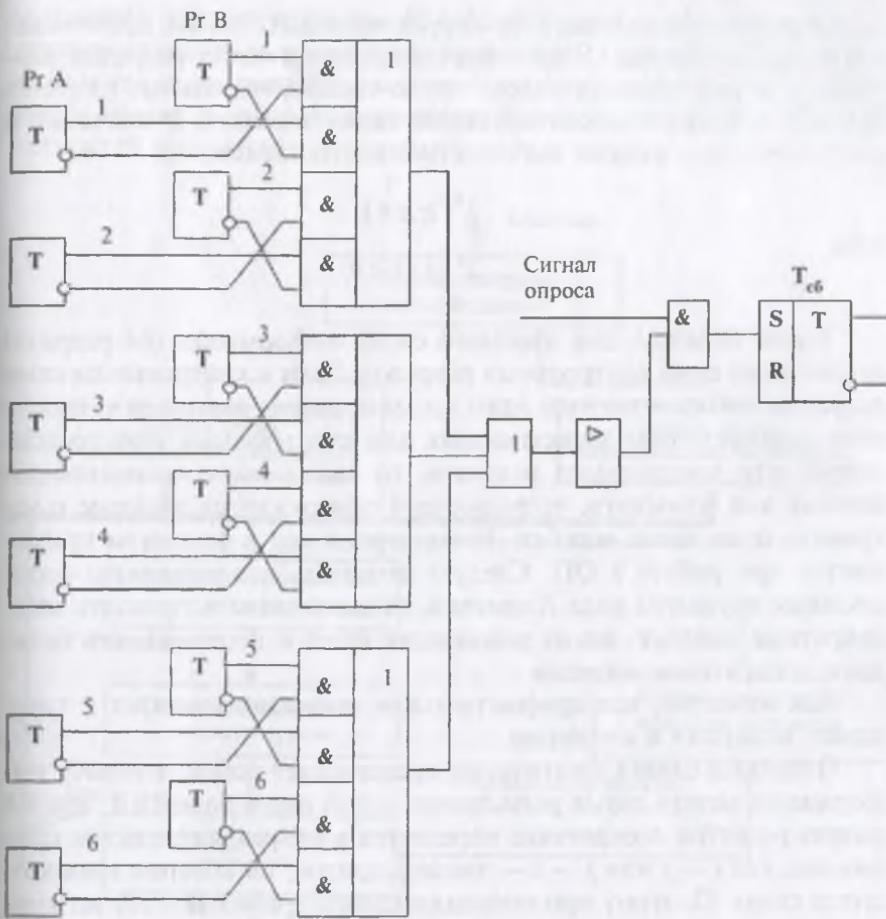


Рис. 10.3. Схема контроля по совпадению.

Контроль информации при хранении в ОП осуществляется с помощью корректирующего кода Хэмминга, основанного на добавлении к информационным разрядам контрольных разрядов, формируемых непосредственно перед записью информации в ОП и записываемых в ОП вместе с информационными разрядами. При считывании производится обратная операция, а контрольные схемы позволяют обнаружить и исправить не только одиночную ошибку, но и многоразрядную. Происходит это следующим образом. После считывания контрольные схемы формируют из считанных информационных и контрольных разрядов корректирующее число, указывающее в случае обнаружения ошибки на неверный разряд в коде. Обнаруженная ошибка исправляется инверсией значения неверного разряда на противоположное.

Определение разрядности корректирующего числа происходит следующим образом. Если число информационных разрядов равно  $m$ , а общее число разрядов —  $n$ , то число контрольных разрядов  $k = n - m$ . Всего в корректирующем числе возможно  $2^k$  значений и соответственно должно выполняться соотношение:

$$2^k \geq n + 1$$

или:

$$2^k - k - 1 \geq m$$

Таким образом, для двойного слова информации (64 разряда) необходимо семь контрольных разрядов. Если к контрольным семи разрядам добавляется еще один восьмой разряд контроля четности всех одновременно записываемых или считываемых информационных или контрольных разрядов, то имеет место модернизированный код Хэмминга, позволяющий обнаруживать двойные и устранять одиночные ошибки. Именно этот код в основном применяется при работе с ОП. Следует отметить, что возможны более сложные варианты кода Хэмминга, позволяющие исправлять многократные ошибки, но их реализация ведет к неоправданно большим аппаратным затратам.

Как известно, все арифметические операции сводятся к трем: сдвиг, инверсия и сложение.

Операция сдвига фактически представляет собой передачу информации между двумя регистрами, с той лишь разницей, что  $i$ -й разряд регистра передатчика передается в  $j$ -й разряд регистра приемника, где  $i = j$  или  $j = i$  — числа разрядов, на которые производится сдвиг. Поэтому при операции сдвига применяют тот же контроль, что и при передаче информации. Операция инверсии контролируется кодами с проверкой на четность. При четном числе «1» в информационных разрядах байта четно и число информационных разрядов, имеющих значение «0», а если число «1» в информационных разрядах нечетно, то и число «0» нечетно. Это правило и лежит в основе контроля с проверкой на четность, так как при инверсии четность не изменяется.

*Операция сложения.* Здесь контроль усложняется тем, что в сумматорах поразрядные суммы и переносы формируются порознь, и если сбой в схеме формирования поразрядных сумм приводит к одиночным ошибкам, то сбой в схеме формирования поразрядных переносов приводит к распространению ошибок. Для борьбы с этим широко применяют дополнительный контроль формирования переносов, который обычно реализуется за счет дублирования схем формирования переносов. Упрощенная схема контроля за работой

сумматора приведена на рис. 10.4. Слагаемые одновременно поступают на схемы формирования переноса и схему формирования суммы, затем проверяется совпадение переносов и формируется четность суммы и четность переносов. Далее производится проверка четностей на специальной схеме.

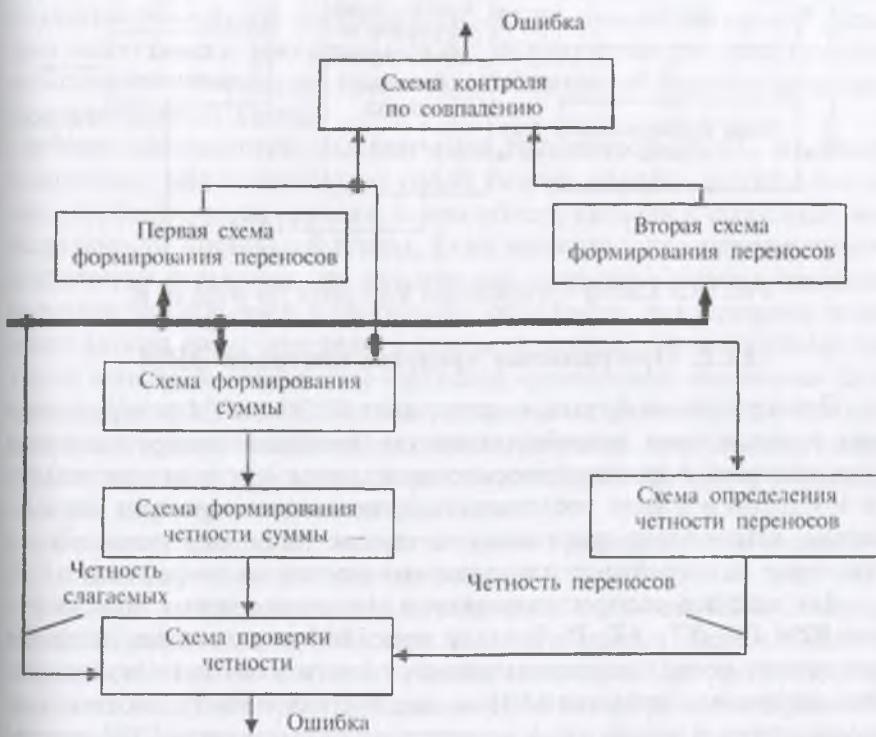


Рис. 10.4. Схема контроля сумматора.

Контроль за выполнением операции умножения, сложения и вычитания может производиться с помощью контрольного кода, использующего остаток от деления на модуль  $R$ . Здесь действует правило:

$$R(A^*B) = R/R(A) * R(B)],$$

где  $R(X)$  – остаток от деления числа  $X$  по модулю  $R$ ;  $*$  – операция умножения, сложения или вычитания.

Организация контроля по модулю R показана на рис. 10.5.

Контроль выполнения логических операций происходит аналогично контролю выполнения арифметических операций.



Рис. 10.5. Схема организации контроля по модулю R.

## 10.2. Программные средства контроля ЭВМ

Предприятия и фирмы — изготовители ЭА, ЭВМ и персональных компьютеров разрабатывают специальные инструкции для пользователей и диагностические программы, которые прилагаются к изделиям в виде технического описания, инструкции пользователя, встроенного программного обеспечения или специальных программ на магнитных или лазерных носителях информации.

Для широко распространенных в настоящее время компьютеров ЮМ PC, XT, AT, PS/2 и для всех IBM-совместимых моделей существует несколько разновидностей диагностических программ. Эти программы применяют при диагностике неисправностей как компьютеров в целом, так и их отдельных компонентов. Во многих случаях такие программы могут проделать большую часть работы по определению дефектного узла. Их можно условно подразделить на три группы: POST (Power-On Self Test — процедура самопроверки при включении), специализированные и общего назначения. Сложность программ и их потенциальные возможности на каждой последующей ступени, как правило, возрастают.

Программа POST предназначена для повышения надежности ЭВМ наряду с контролем на четность памяти. Эта программа представляет собой последовательность коротких программ, «зашитых» в ПЗУ BIOS (Basic Input Output System — базовая система ввода-вывода) на системной плате и предназначенных для проверки всех основных компонентов системы непосредственно после ее включения. Процедура POST выполняется перед загрузкой операционной системы и запускается всякий раз, когда включают компьютер.

*Процесс самопроверки при включении (POST).* Автоматически выполняется некоторая последовательность операций по проверке различных компонентов компьютера. Обычно проверяются центральный процессор, ПЗУ, вспомогательные схемы системной платы, оперативная память и основные периферийные устройства. Эти тесты выполняются быстро и не слишком тщательно по сравнению с диагностическими программами, записанными на дисках. Если при выполнении процедуры POST обнаруживается неисправный компонент системы, то выдается сообщение об ошибке или предупредительный сигнал.

Хотя диагностика, выполняемая процедурой POST, не очень подробная, она представляет собой первую ступень защиты системы, особенно в тех случаях, когда обнаруживаются серьезные неисправности системной платы. Если окажется, что неисправность достаточно серьезная, то дальнейшая загрузка системы приостанавливается и выдается сообщение об ошибке, по которому чаще всего можно сразу определить причину возникшей неисправности. Такие неисправности иногда называют «фатальными ошибками» (fatal errors). Процедурой POST обычно предусматривается три способа индикации неисправности: звуковые сигналы, сообщения, выводимые на экран монитора, и шестнадцатеричные коды, посылаемые по адресам портов ввода/вывода.

*Специализированные диагностические программы предприятий и фирм изготавителей* выпускают большинство солидных предприятий и фирм — изготавителей компьютеров, таких как IBM, Compaq, Hewlett-Packard и т. п. для своих вычислительных систем. Обычно это наборы тестов для «тотальной» проверки всех компонентов компьютера. Фирменная программа IBM для общего тестирования систем PS/2 записывается на установочной дискете, а для компьютеров других моделей — на отдельном диагностическом диске. Фирмы Compaq и Hewlett-Packard также выпускают программы для технических специалистов, предназначенные для поисков неисправностей в соответствующих системах. Такие же программы можно найти в компьютерных справочных системах (BBS — Bulletin Board System) соответствующих фирм. Некоторые компании, например Dell и Gateway, выпускают сокращенные варианты таких программ, приспособленные для обслуживания их компьютеров.

*Диагностические программы фирм-изготавителей* обычно предусмотрены двух уровней. Первый уровень — это общая диагностика, которая в основном ориентирована на пользователей. Так как процедуры поиска неисправностей в большинстве современных систем достаточно просты, у пользователей обычно не возникает сложностей при работе с программами общей диагностики — даже при

отсутствии соответствующей документации. Второй уровень — технический и рассчитан на специалистов.

Рассмотрим диагностические программы фирмы IBM.

*Расширенная диагностика.* Для проведения диагностики на техническом уровне фирма IBM выпускает для каждой системы руководства по эксплуатации и техническому обслуживанию с соответствующими программами расширенной диагностики. Для ПЭВМ PS/2 программа расширенной диагностики прилагается в установочной диске (Reference Disk).

Сообщения об ошибках выводятся в виде чисел, по которым можно определить причину неисправности или сузить круг ее поисков. Используемые коды идентичны кодам процедуры POST, программы общей диагностики и вообще всех диагностических программ IBM.

*Диагностические программы общего назначения* выпускают несколько фирм, среди которых можно отметить пакеты программ Norton Utilities компании Symantec, MicroScope (Micro 2000), QAPlus (Diagsoft), PC-Probe (Landmark) и других фирм.

Большинство пользователей в основном имеют дело с программой POST и одной из программ общей диагностики. Хотя эти диагностические программы дорогие, они наиболее полные и работают с предназначенными для них компьютерами.

В настоящее время существует много диагностических программ для тестирования памяти, накопителей на гибких и жестких дисках, видеосистем и т. п. Они существенно расширены по сравнению со стандартной диагностической программой IBM. С их помощью удастся точнее определить местоположение неисправности в системе (особенно в IBM-совместимых ПЭВМ). Для проверки последовательных и параллельных портов имеются тест-разъемы. Многие из этих программ можно запускать в режиме командной строки (в пакетном режиме), поэтому, без вмешательства оператора, может быть выполнена целая серия тестов. С помощью таких программ можно проверить все типы памяти — основную, расширенную и дополнительную, а также определить неисправности с точностью до отдельной микросхемы или разряда модуля SIMM.

Рассмотрим диагностические программные пакеты некоторых фирм.

*AMIDIAG.* Компания AMI (American Megatrends Inc.) выпускает наиболее популярные BIOS. В большинстве IBM-совместимых ПЭВМ используется AMI BIOS. Эта фирма выпускает также расширенный (дисковый) вариант диагностической программы AMIDIAG, которая является полноценной программой общей диагностики, рассчитанной на использование во всех IBM-совместимых системах, а не только в тех, где установлено ПЗУ с AMI BIOS.

*Checkit Pro.* Этот пакет фирмы Touchstone Software Corp. состоит из набора программ, предназначенных для тестирования центрального процессора, основной, расширенной и дополнительной памяти, накопителей на жестких и гибких дисках, мыши, клавиатуры, а также видеоплаты и монитора (в том числе устройств, выполненных в стандарте VESA). Используется несколько версий пакета Checkit. Наиболее полной из них является комплект Checkit Pro Deluxe. Checkit Pro Analyst предназначен для работы в среде Windows. Некоторые фирмы-изготовители ПЭВМ чаще всего включают в комплекты своих систем пакет Checkit Plus – простейшую и наименее полную из версий. Для проверки быстродействия и «аттестации» системы возможности Checkit Pro Deluxe достаточно ограничены, но при этом можно получить детальную информацию о полном объеме установленной памяти, типе накопителя на жестком диске и его емкости, текущем распределении памяти (в том числе и в области верхней памяти – UMA), доступных используемых прерываниях, быстродействии факс-модема и другие сведения, полезные при поиске неисправностей в ПЭВМ. В пакет входит встроенный текстовый редактор, с помощью которого можно оперативно внести изменения в файлы CONFIG.SYS и AUTOEXEC.BAT. При работе в среде Windows точно так же могут быть отредактированы файлы SYSTEM.INI и WIN.INI.

*Micro-Scope.* Пакет Micro-Scope (Micro 2000) предназначен для IBM-совместимых ПЭВМ. Являясь полноценной диагностической программой общего назначения, она единственная из всех «знает» все о PS/2. С ее помощью удается отформатировать стандартные ESDI-накопители на жестких дисках, подключенные к ESDI-контроллеру PS/2. Программа Micro-Scope работает в обход DOS и BIOS. В ней есть своя операционная система, и при необходимости тестирование проходит без участия системной BIOS. Программа полезна для специалистов, занимающихся обслуживанием ПЭВМ, работающих под управлением не DOS, а, например, UNIX или в сетях Novell.

Кроме вышеперечисленных существует много диагностических программ, в числе которых наиболее известные: Norton Diagnostics (NDIAGS), входящая в состав версии 8.0 пакета Norton Utilities; программа PC Technician фирмы Windsor Technologies; QAPlus/FE компании Diagsoft для тестирования ПЭВМ с процессорами 386, 486, Pentium и системы PS/2s; Service Diagnostics фирмы Landmark. Кроме того, имеются программы для диагностирования дисков и эталонные диски, предназначенные для оценки качества дисководов и их настройки. Это цифровой диагностический диск (DDD), диагностический диск высокого разрешения (HRD – High-Resolution Diagnostic) и аналоговый диск для настройки позиционирования (AAD).

Для Windows 3.0, 3.1 и Windows 95 разработан пакет Norton Desktop, в который входит программа SYSINFO, позволяющая диагностировать накопители, память, видеоадаптер, принтер и т. д. и проводить аттестацию ПЭВМ, подсчитав ее общую производительность. Подобными свойствами обладают программы WinSleuth, WinSleuth Gold. Одной из наиболее полных диагностических программ для Windows является программа Winprobe фирмы Landmark.

### 10.3. Обслуживание и ремонт персональных ЭВМ

Рассмотрим три уровня поиска неисправностей и ремонта персональных ЭВМ (ПЭВМ): платы, ИС и схемы. Каждый уровень имеет свои цели при поиске дефектного компонента или соединения с последующей заменой или ремонтом.

На уровне плат заменяют подозрительную плату; на уровне ИС определяют и заменяют дефектную ИС или компонент; на уровне схемы определяют точную причину неисправности.

Проще всего заменить всю дефектную плату. При замене ИС используют два способа. Первый способ заключается в замене ИС по одной до тех пор, пока не будет обнаружена дефектная. Труднее всего точно найти дефектную ИС и заменить только ее. К сожалению, в ПЭВМ этот уровень сложнее, чем в РЭА. Прежде всего в ПЭВМ только небольшое число ИС находится в гнездах. Многие ИС впаяны в плату, что усложняет их замену. Второй способ заключается в определении подозреваемых ИС и замене их по одной до обнаружения дефектной ИС.

*Замена плат.* Когда ПЭВМ выходит из строя, пользователя не интересует, почему это случилось, ему нужна работающая ПЭВМ.

В ПЭВМ типа IBM PC бывает шесть и более плат (основная плата и съемные дочерние платы). В этом случае технику необходимо отыскивать дефектную плату. Он должен проанализировать симптомы неисправности, определить дефектную плату и заменить ее. Операция диагностирования и замены ПП повторяется до успешного завершения ремонта.

Если имеются конкретные симптомы, указывающие на одну или две платы, например, на дисплее изображение мелькает или нарушается строчная (кадровая) синхронизация, неисправными считаются плата монохроматического адаптера или цветная графическая плата. Их нужно заменять по одной, и изображение восстановится. Если этого не происходит, значит, неисправность находится в мониторе.

Еще один симптом, относящийся к данным платам, связан с отсутствием графических изображений на экране монитора. Этот симптом также требует замены плат. Если неисправность исчезает, то неисправна одна из ИС на дефектной плате. Когда на экране отсутству-

ет текст, а графика выводится, неисправность находится в цветной или графической плате, так как именно она задает графический или текстовой режим. При замене платы неисправность обычно исчезает.

Симптом, относящийся к цветной (графической) плате, заключается в отсутствии цвета или плохом цвете. Именно эта плата формирует цвета. Если новая плата не восстанавливает цвета, то неисправен цветной монитор.

Схемы клавиатуры, естественно, вызывают неисправности, относящиеся к клавиатуре. Например, при нажатии клавиши на экране появляются неправильные символы. Возможно, вместо прописных букв появляются строчные или наоборот. Клавиатура в ШМ РС сама по себе является небольшим компьютером со своим процессором и ПЗУ. Для устранения таких неисправностей необходимо заменить плату клавиатуры. Неисправности на этих платах вызывают невозможность обращения к накопителям или ошибки в операциях считывания и записи.

Тщательный анализ симптомов позволяет определить возможную причину неисправности одной или двух плат. Если, например, один накопитель не проводит считывание и запись, то неисправна аналоговая плата накопителя. Когда не считывают и не записывают оба накопителя, дефект находится на плате адаптера дисковых накопителей. Если к накопителям вообще невозможно обратиться, подозрительны обе платы. Кроме конкретных неисправностей есть такие общие симптомы, как отсутствие изображения или «мусор» на экране. К сожалению, эти симптомы вызываются многими платами, исключая платы дисковых накопителей. В этом случае приходится проверять клавиатуру, основную плату и две дисплейные платы.

Несмотря на дороговизну замены плат, для сокращения времени ремонта во многих случаях пользуются этим способом.

*Обслуживание на уровне микросхем.* Если представить ПП небольшим электронным городом, то ИС следует считать домами. Обычно неисправность возникает только в одной ИС или поддерживающих компонентах. Наиболее сложным при ремонте оказывается поиск дефектной ИС или компонента.

При тщательном изучении симптома (признака) неисправности определяется подозрительная ИС. Каждая ИС выполняет конкретные функции. Эти функции могут быть простыми или сложными, но все они важны для работы ПЭВМ. Печатная плата с десятками ИС чрезвычайно сложна, но только из-за большого числа схем. Разобраться в каждой ИС не составляет труда. К счастью, нет необходимости разбираться с работой каждого транзистора и даже отдельных узлов, составленных из них, таких как триггер, регистр или дешифратор. Даже если определено, что не работает какой-

либо разряд регистра, заключенного в БИС, то все равно необходимо заменить целиком всю БИС. Поэтому необходимо знать, какие сигналы должны поступать на входы ИС, что с ними происходит в ИС и какие сигналы в результате работы должны появиться на выходе. Этой информации достаточно для того, чтобы можно было отремонтировать ПЭВМ.

Все ИС на ПП расположены в определенном порядке. Для обслуживания на уровне ИС необходима диаграмма, показывающая неисправность, которая возникает при выходе той или иной ИС из строя. При неисправности появляется симптом, и диаграмма показывает, какая ИС соответствует данному симпту. Когда из диаграммы известны подозрительные ИС, необходимо найти дефектную ИС.

Схема размещения ИС должна показывать физическое местонахождение ИС на плате, общий номер каждой ИС, ключ для нумерации контактов и все важные контрольные точки на плате. Такими точками служат указания портов, предохранителей, сетевого переключателя и других элементов. Сама схема может быть оформлена как рисунок, фотография, топология платы и даже сама плата. В простейшем случае схема представляет собой чертеж размещения ИС.

*Работа со схемой размещения.* Типичный поиск неисправности ПЭВМ происходит следующим образом. При включении большинство ПЭВМ выполняют диагностические тесты микросхем. Тесты проводятся при инициализации процессором различных регистров ИС. Процессор по указанию операционной системы заставляет дефектную ИС выполнять несложные действия. Если ИС не проходит тест, устанавливается флагок, и на экране появляется сообщение о неисправности.

Предположим, что после включения ПЭВМ на экране появилось сообщение «Микросхема ЗУПВ номер 4 банка 2 системной памяти неисправна» или соответствующий код, по которому можно узнать о неисправности. После этого снимается кожух ПЭВМ и находится основная плата. На схеме размещения расположены ИС четырех банков 0-3 ЗУПВ по девять ИС в банке. По схеме в банке 2 находят ИС 4 в гнезде и определяют ее общий номер 4164. Извлекают дефектную ИС из гнезда и устанавливают на ее место новую согласно указаниям инструкции.

Для более полного понимания сути неисправности дополнительную информацию можно получить из блок-схемы компьютера. Она позволяет перейти от чисто механического ремонта к логическому анализу неисправности и выявить истинную причину отказа.

Большинство ИС для ПЭВМ выпускают в корпусе DIP с двусторонним расположением контактов, БИС и СБИС чаще располагают в корпусах типа PGA, PQFP или PLCC. Для проведения полного

тестирования на контактах ИС требуется принципиальная схема. Если потребуется схема внутренней организации ИС, то следует обращаться к справочной литературе по интегральным микросхемам.

Схема размещения, блок-схема и принципиальная схема по-разному показывают одни и те же ИС. Схема размещения сообщает физическое расположение микросхем. Ее можно использовать для быстрых проверок, которые позволяют отремонтировать ПЭВМ примерно в 50 % случаев. Блок-схема придает смысл схеме размещения, без блок-схемы схема размещения довольно ограничена по информативности. С помощью блок-схемы и схемы размещения можно осуществить ремонт еще примерно 20 % случаев отказов. Принципиальная схема детализирует блок-схему. Три эти схемы содержат всю необходимую информацию по обслуживанию. С их помощью можно поставить диагноз, найти подозрительную ИС и провести измерения на ее контактах.

#### 10.4. Контроль локальных сетей

Постоянный контроль за работой локальной сети, составляющей основу любой компьютерной сети, необходим для поддержания ее в работоспособном состоянии. Контроль — это необходимый первый этап, который должен выполняться при управлении сетью. Ввиду важности этой функции ее часто отделяют от других функций систем управления и реализуют специальными средствами. Такое разделение функций контроля и собственно управления полезно для небольших и средних сетей, для которых установка интегрированной системы управления экономически нецелесообразна. Использование автономных средств контроля помогает администратору сети выявить проблемные участки и устройства сети, а их отключение или реконфигурацию он может выполнять в этом случае вручную. Процесс контроля работы сети обычно делят на два этапа — мониторинг и анализ. На этапе мониторинга выполняется более простая процедура — процедура сбора первичных данных о работе сети: статистики о количестве циркулирующих в сети кадров и пакетов различных протоколов, состоянии портов концентраторов, коммутаторов и маршрутизаторов и т. п.

Далее выполняется этап анализа, под которым понимается более сложный и интеллектуальный процесс осмысливания собранной на этапе мониторинга информации, сопоставления ее с данными, полученными ранее, и выработки предположений о возможных причинах замедленной или ненадежной работы сети. Задачи мониторинга решаются программными и аппаратными измерителями, тестерами, сетевыми анализаторами, встроенными средствами мониторинга коммуникационных устройств, а также агентами систем уп-

равления. Задача анализа требует более активного участия человека и использования таких сложных средств, как экспертные системы, аккумулирующие практический опыт многих сетевых специалистов.

Все многообразие средств, применяемых для анализа и диагностики вычислительных сетей, можно разделить на несколько крупных классов.

*Агенты систем управления*, поддерживающие функции одной из стандартных баз MIB и поставляющие информацию по протоколу SNMP или CMIP. Для получения данных от агентов обычно требуется наличие системы управления, собирающей данные от агентов в автоматическом режиме.

*Встроенные системы диагностики и управления* (*Embedded systems*). Эти системы выполняются в виде программно-аппаратных модулей, устанавливаемых в коммуникационное оборудование, а также в виде программных модулей, встроенных в операционные системы. Они выполняют функции диагностики и управления только одним устройством, и в этом их основное отличие от централизованных систем управления. Примером средств этого класса может служить модуль управления многосегментным повторителем Ethernet, реализующий функции автосегментации портов при обнаружении неисправностей, приписывания портов внутренним сегментам повторителя и некоторые другие. Как правило, встроенные модули управления «по совместительству» выполняют роль SNMP-агентов, поставляющих данные о состоянии устройства для систем управления.

*Анализаторы протоколов* (*Protocol analyzers*). Представляют собой программные или аппаратно-программные системы, которые в отличие от систем управления ограничиваются лишь функциями мониторинга и анализа трафика в сетях. Хороший анализатор протоколов может захватывать и декодировать пакеты большого количества протоколов, применяемых в сетях, — обычно несколько десятков. Анализаторы протоколов позволяют установить некоторые логические условия для захвата отдельных пакетов и выполняют полное декодирование захваченных пакетов, то есть показывают в удобной для специалиста форме вложенность пакетов протоколов разных уровней друг в друга с расшифровкой содержания отдельных полей каждого пакета.

*Экспертные системы*. Этот вид систем аккумулирует знания технических специалистов о выявлении причин аномальной работы сетей и возможных способах приведения сети в работоспособное состояние. Экспертные системы часто реализуются в виде отдельных подсистем различных средств мониторинга и анализа сетей: систем управления сетями, анализаторов протоколов, сетевых анализаторов. Простейшим вариантом экспертной системы является контекстно-зависимая система помощи. Более сложные экспертные системы представляют собой

так называемые базы знаний, обладающие элементами искусственного интеллекта. Примерами таких систем являются экспертные системы, встроенные в систему управления Spectrum компании Cabletron и анализатора протоколов Sniffer компании Network General. Работа экспертных систем состоит в анализе большого числа событий для выдачи пользователю краткого диагноза о причине неисправности сети.

*Оборудование для диагностики и сертификации кабельных систем.* Условно это оборудование можно поделить на четыре основные группы: сетевые мониторы, приборы для сертификации кабельных систем, кабельные сканеры и тестеры.

*Сетевые мониторы* (называемые еще сетевыми анализаторами) предназначены для тестирования кабелей различных категорий. Сетевые мониторы собирают также данные о статистических показателях трафика: средней интенсивности общего трафика сети, средней интенсивности потока пакетов с определенным типом ошибки и т. п. Эти устройства являются наиболее интеллектуальными устройствами из всех четырех групп устройств данного класса, так как работают не только на физическом, но и на канальном, а иногда и на сетевом уровнях.

*Устройства для сертификации кабельных систем* выполняют сертификацию в соответствии с требованиями одного из международных стандартов на кабельные системы.

*Кабельные сканеры* используются для диагностики медных кабельных систем.

*Тестеры* предназначены для проверки кабелей на отсутствие физического разрыва.

*Многофункциональные портативные устройства анализа и диагностики.* В связи с развитием технологий больших интегральных схем появилась возможность производства портативных приборов, которые совмещали бы функции нескольких устройств: кабельных сканеров, сетевых мониторов и анализаторов протоколов.

*Анализаторы протоколов.* Анализатор протоколов представляет собой либо специализированное устройство, либо персональный компьютер, обычно переносной, класса Notebook, оснащенный специальной сетевой картой и соответствующим программным обеспечением. Применяемые сетевая карта и программное обеспечение должны соответствовать технологии сети (Ethernet, Token Ring, FDDI, Fast Ethernet). Анализатор подключается к сети точно так же, как и обычный узел. Отличие состоит в том, что анализатор может принимать все пакеты данных, передаваемые по сети, в то время как обычная станция — только адресованные ей. Для этого сетевой адаптер анализатора протоколов переводится в режим «беспорядочного» захвата (promiscuous mode).

Программное обеспечение анализатора состоит из ядра, поддерживающего работу сетевого адаптера и программного обеспечения, декодирующего протокол канального уровня, с которым работает сетевой адаптер, а также наиболее распространенные протоколы верхних уровней, например IP, TCP, ftp, telnet, HTTP, IPX, NCP, NetBEUI, DECnet и т. п. В состав некоторых анализаторов может входить также экспертная система, которая позволяет выдавать пользователю рекомендации о том, какие эксперименты следует проводить в данной ситуации, что могут означать те или иные результаты измерений, как устранить некоторые виды неисправности сети.

Анализаторы протоколов имеют некоторые общие свойства.

*Возможность (помимо захвата пакетов) измерения среднестатистических показателей трафика в сегменте локальной сети*, в котором установлен сетевой адаптер анализатора. Обычно измеряется коэффициент использования сегмента, матрицы перекрестного трафика узлов, количество хороших и плохих кадров, прошедших через сегмент.

*Возможность работы с несколькими агентами, поставляющими захваченные пакеты из разных сегментов локальной сети*. Эти агенты чаще всего взаимодействуют с анализатором протоколов по собственному протоколу прикладного уровня, отличному от SNMP или CMIP.

*Наличие развитого графического интерфейса*, позволяющего представить результаты декодирования пакетов с разной степенью детализации.

*Фильтрация захватываемых и отображаемых пакетов*. Условия фильтрации задаются в зависимости от значения адресов назначения и источника, типа протокола или значения определенных полей пакета. Пакет либо игнорируется, либо записывается в буфер захвата. Использование фильтров значительно ускоряет и упрощает анализ, так как исключает захват или просмотр ненужных в данный момент пакетов.

*Использование триггеров*. Триггеры – это задаваемые администратором некоторые условия начала и прекращения процесса захвата данных из сети. Такими условиями могут быть: время суток, продолжительность процесса захвата, появление определенных значений в кадрах данных. Триггеры могут использоваться совместно с фильтрами, позволяя более детально и тонко проводить анализ, а также продуктивнее расходовать ограниченный объем буфера захвата.

*Многоканальность*. Некоторые анализаторы протоколов позволяют проводить одновременную запись пакетов от нескольких сетевых адаптеров, что удобно для сопоставления процессов, происходящих в разных сегментах сети.

Возможности анализа проблем сети на физическом уровне у анализаторов протоколов минимальные, поскольку всю информацию

они получают от стандартных сетевых адаптеров. Поэтому они передают и обобщают информацию физического уровня, которую сообщает им сетевой адаптер, а она во многом зависит от типа сетевого адаптера. Некоторые сетевые адаптеры сообщают более детальные данные об ошибках кадров и интенсивности коллизий в сегменте, а некоторые вообще не передают такую информацию верхним уровням протоколов, на которых работает анализатор протоколов.

С распространением серверов Windows NT все более популярным становится анализатор Network Monitor фирмы Microsoft. Он является частью сервера управления системой SMS, а также входит в стандартную поставку Windows NT Server, начиная с версии 4.0 (версия с усеченными функциями). Network Monitor в версии SMS является многоканальным анализатором протоколов, поскольку может получать данные от нескольких агентов Network Monitor Agent, работающих в среде Windows NT Server, однако в каждый момент времени анализатор может работать только с одним агентом, так что сопоставить данные разных каналов с его помощью не удастся. Network Monitor поддерживает фильтры захвата (достаточно простые) и дисплейные фильтры (более сложные), отображающие нужные кадры после захвата. Экспертной системой Network Monitor не располагает.

*Сетевые анализаторы.* Сетевые анализаторы представляют собой эталонные измерительные приборы для диагностики и сертификации кабелей и кабельных систем. Они могут с высокой точностью измерять все электрические параметры кабельных систем, а также работают на более высоких уровнях стека протоколов. Сетевые анализаторы генерируют синусоидальные сигналы в широком диапазоне частот, что позволяет измерять на приемной паре амплитудно-частотную характеристику и перекрестные наводки, затухание и суммарное затухание. Сетевой анализатор представляет собой лабораторный прибор больших размеров, достаточно сложный в обращении.

Многие производители дополняют сетевые анализаторы функциями статистического анализа трафика — коэффициента использования сегмента, уровня широковещательного трафика, процента ошибочных кадров, а также функциями анализатора протоколов, которые обеспечивают захват пакетов разных протоколов в соответствии с условиями фильтров и декодирование пакетов.

*Кабельные сканеры и тестеры.* Основное назначение кабельных сканеров — измерение электрических и механических параметров кабелей: длины кабеля, параметра NEXT, затухания, импеданса, схемы разводки пар проводников, уровня электрических шумов в кабеле. Точность измерений, произведенных этими устройствами, ниже, чем у сетевых анализаторов, но вполне достаточна для оценки соответствия кабеля стандарту.

Для определения местоположения неисправности кабельной системы (обрыва, короткого замыкания, неправильно установленного разъема и т. д.) используется метод «отраженного импульса» (Time Domain Reflectometry, TDR). Суть этого метода состоит в том, что сканер излучает в кабель короткий электрический импульс и измеряет время задержки до прихода отраженного сигнала. По полярности отраженного импульса определяется характер повреждения кабеля (короткое замыкание или обрыв). В правильно установленном и подключенном кабеле отраженный импульс почти отсутствует.

Точность измерения расстояния зависит от того, насколько точно известна скорость распространения электромагнитных волн в кабеле. В различных кабелях она будет разной. Скорость распространения электромагнитных волн в кабеле (Nominal Velocity of Propagation, NVP) обычно задается в процентах от скорости света в вакууме. Современные сканеры содержат в себе электронную таблицу данных о NVP для всех основных типов кабелей, что дает возможность пользователю устанавливать эти параметры самостоятельно после предварительной калибровки.

Кабельные сканеры – это портативные приборы, которые обслуживающий персонал может постоянно носить с собой.

*Кабельные тестеры* – наиболее простые и дешевые приборы для диагностики кабеля. Они позволяют определить непрерывность кабеля, однако в отличие от кабельных сканеров не дают ответа на вопрос о том, в каком месте произошел сбой.

*Многофункциональные портативные приборы мониторинга*. В последнее время начали выпускаться многофункциональные портативные приборы, которые объединяют в себе возможности кабельных сканеров, анализаторов протоколов и даже некоторые функции систем управления, сохраняя в то же время такое важное свойство, как портативность. Многофункциональные приборы мониторинга (МПМ) имеют специализированный физический интерфейс, позволяющий выявлять проблемы и тестировать кабели на физическом уровне, который дополняется микропроцессором с программным обеспечением для выполнения высококачественных функций.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какими показателями можно охарактеризовать надежность функционирования ЭВМ?
2. Как организуется аппаратный контроль функционирования ЭВМ?
3. Как осуществляется контроль при хранении, передаче информации и при выполнении арифметических и логических операций?
4. Перечислите уровни поиска неисправностей и ремонта ПЭВМ.
5. Перечислите виды диагностических программ.
6. Как осуществляется контроль работы локальных сетей?

## ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации. СПб. ПИТЕР. 2003.
2. Жмакин А.П. Архитектура ЭВМ. Уч.пособие. Санкт-Петербург. БХВ-Петербург, 2006
3. Максимов Н.В., Партика Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем. М. ФОРУМ-ИНФАРМ. 2006.
4. Максимов Н.В., Попов И.И. Компьютерные сети. Учебное пособие. М. ФОРУМ-ИНФАРМ. 2003.
5. Пятибратов А.П. и др. Вычислительные системы, сети и телекоммуникации. М. «Финансы и статистика». 2005.
6. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. Учебник для ВУЗов, 2-е издание. СПб. Питер. 2005.
7. <http://www.intuit.ru/department/hardware/mpbasics/>. Основы микропроцессорной техники.

## ОГЛАВЛЕНИЕ

<b>Предисловие</b> .....	3
<b>Глава 1. Электронные вычислительные машины. Общие сведения</b>	
1.1. Базовая структура ЭВМ. Классификация.....	6
1.2. Устройства электронных вычислительных машин.....	11
1.3. Особенности представления информации в персональных компьютерах.....	14
<b>Глава 2. Архитектура микропроцессоров</b>	
2.1. Структура 16-разрядного микропроцессора i8086.....	17
2.2. Структура 32-разрядного микропроцессора x86 (серии Pentium).....	19
2.3. Системы и форматы команд.....	26
2.4. Методы адресации.....	28
<b>Глава 3. Организация памяти ЭВМ</b>	
3.1. Концепция многоуровневой памяти.....	32
3.2. Сверхоперативная память.....	33
3.3. Модули основной памяти.....	35
3.4. Внешние запоминающие устройства.....	38
<b>Глава 4. Организация ввода-вывода и прерываний</b>	
4.1. Организация ввода-вывода в ЭВМ.....	41
4.2. Внешние устройства ЭВМ.....	43
4.3. Организация прерываний.....	50
<b>Глава 5. Вычислительные сети. Основные сведения</b>	
5.1. Вычислительные сети – частный случай распределенных систем.....	57
5.2. Аппаратно-программное обеспечение сети.....	61
<b>Глава 6. Основные проблемы построения сетей</b>	
6.1. Связь компьютера с периферийными устройствами.....	63
6.2. Проблемы физической передачи данных по линиям связи.....	65
6.3. Проблемы объединения нескольких компьютеров.....	68
6.4. Ethernet – пример стандартного решения сетевых проблем.....	73
6.5. Структурирование как средство построения больших сетей.....	76
6.6. Службы сети.....	85
<b>Глава 7. Проблемы стандартизации при построении вычислительных сетей</b>	
7.1. Многоуровневый подход. Протокол. Интерфейс. Стек протоколов.....	88
7.2. Модель OSI.....	94
7.3. Уровни модели OSI.....	97

<b>Глава 8. Локальные и глобальные сети</b>	
8.1. Особенности локальных, глобальных и городских сетей.....	107
8.2. Отличия локальных сетей от глобальных.....	108
8.3. Тенденция к сближению локальных и глобальных сетей... ..	110
<b>Глава 9. Сети отделов, кампусов и корпораций</b>	
9.1. Сети отделов.....	112
9.2. Сети кампусов.....	113
9.3. Корпоративные сети.....	115
<b>Глава 10. Эксплуатация электронных вычислительных машин и сетей</b>	
10.1. Аппаратные средства контроля ЭВМ.....	118
10.2. Программные средства контроля ЭВМ.....	124
10.3. Обслуживание и ремонт персональных ЭВМ.....	128
10.4. Контроль локальных сетей.....	131
<b>Использованная литература.....</b>	137

**Салим Каримович Ганиев  
Тахир Анварович Кучкаров**

**ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ  
МАШИНЫ И СЕТИ**

*Учебное пособие для профессионально-технических колледжей*

Редактор *Л.А.Казакова*

Художник *Н.А. Попов*

Художественный редактор *А.А.Бобров*

Технический редактор *Т.И.Смирнова*

Компьютерная верстка *А.Т. Сулейманова*

ИБ № 4576

Подписано в печать 29.12.2007г. Формат 60x90 1/16. Гарнитура тип Таймс.

Офсетная печать. Усл. печ. л. 8,75. Уч-изд. л. 9,5. Тираж 1860.

Заказ № 16. Цена договорная. Договор № 115-2007.

Издательско-полиграфический творческий дом имени Гафура Гуляма  
Узбекского агентства по печати и информации.

100129, Ташкент, ул. Навои, 30 // 100128, Ташкент, ул. Шайхонтаур, 86.

**Наш интернет-адрес: [www.iptdgulom.uz](http://www.iptdgulom.uz)**



:0000h, 0008h, 0010h, ...,  
INT.  
...INT F8h,  
i8080



ISBN 978-9943-03-118-0

9 789943 031180