

А. В. КУЗИН, М. А. ЖАВОРОНКОВ

# МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

*Допущено*

*Министерством образования Российской Федерации  
в качестве учебника для студентов образовательных  
учреждений среднего профессионального образования,  
обучающихся по группам специальностей 2200 «Информатика  
и вычислительная техника», 1800 «Электротехника»*

3-е издание, стереотипное

БИБЛИОТЕКА ГОУ СПО "ВГЭТК"  
Инв. № 5307



Москва  
Издательский центр «Академия»  
2007

ЧИТАЛЬНЫЙ ЗАЛ

УДК 681.326.32(075.32)

ББК 32.973.26я722

К89

Рецензенты:

доц. кафедры ИИТ МЭИ (ТУ) *Ю. Н. Евланов*;  
зам. зав. кафедрой ИИТ МЭИ (ТУ), доц. *Н. А. Серов*;  
директор Красногорского оптико-электронного колледжа,  
д-р пед. наук *В. М. Демин*

**Кузин А. В.**

**К89** Микропроцессорная техника : учебник для студ. сред. проф. образования / А.В.Кузин, М.А.Жаворонков. — 3-е изд., стер. — М. : Издательский центр «Академия», 2007. — 304 с. ISBN 978-5-7695-4278-7

Рассматриваются основные сведения о микропроцессорах и микроЭВМ, информационно-логические основы электронно-вычислительной техники, типовые логические элементы и узлы ЭВМ, структура и функционирование процесса, периферийные устройства ЭВМ, тенденции развития архитектуры и аппаратного обеспечения электронных вычислительных систем, программное (математическое) обеспечение ЭВМ, алгоритмизация и основы программирования.

Для студентов средних профессиональных учебных заведений.

УДК 681.326.32(075.32)

Учебное издание

ББК 32.973.26я722

**Кузин Александр Владимирович,  
Жаворонков Михаил Анатольевич**

**Микропроцессорная техника**

**Учебник**

3-е издание, стереотипное

Редактор *Н. С. Лепина-Лавринович*

Технический редактор *О. С. Александрова*

Компьютерная верстка: *С. Ф. Блудова*

Корректоры: *Е. Ю. Куринских, М. В. Дьяконова*

Изд. № 103105437. Подписано в печать 30.05.2007. Формат 60×90/16. Гарнитура «Таймс». Бумага тип. № 2. Печать офсетная. Усл. печ. л. 19,0. Тираж 2 000 экз. Заказ № 19231.

Издательский центр «Академия». [www.academia-moscow.ru](http://www.academia-moscow.ru)

Санитарно-эпидемиологическое заключение № 77.99.02.953.Д.004796.07.04 от 20.07.2004. 117342, Москва, ул. Бутлерова, 17-Б, к. 360. Тел./факс: (495) 330-1092, 334-8337.

Отпечатано в ОАО «Саратовский полиграфкомбинат».

410004, г. Саратов, ул. Чернышевского, 59. [www.sarprk.ru](http://www.sarprk.ru).

*Оригинал-макет данного издания является собственностью  
Издательского центра «Академия», и его воспроизведение любым способом  
без согласия правообладателя запрещается*

© Кузин А. В., Жаворонков М. А., 2004

© Образовательно-издательский центр «Академия», 2004

ISBN 978-5-7695-4278-7

© Оформление. Издательский центр «Академия», 2004

## ПРЕДИСЛОВИЕ

Микропроцессорная техника получила широкое применение в системах управления технологическим и контрольно-испытательным оборудованием, транспортными средствами, космическими аппаратами, бытовыми приборами и т. д. Малые размеры, масса и энергоемкость микропроцессоров позволяют встраивать их непосредственно в объект управления. На базе микропроцессоров создаются различные типы микроЭВМ, контроллеров, программаторов и других устройств автоматики и вычислительной техники.

В учебнике рассмотрены основные сведения о микропроцессорах и микроЭВМ, информационно-логические основы электронно-вычислительной техники, типовые логические элементы и узлы ЭВМ, структура и функционирование процессора, периферийные устройства ЭВМ, тенденции развития архитектуры и аппаратного обеспечения электронных вычислительных систем, программное (математическое) обеспечение ЭВМ, алгоритмизация и основы программирования. Рассмотрены методы совершенствования электронных вычислительных средств (ЭВС): конвейеризация обработки, распараллеливание операций, компьютеры с сокращенным набором команд (КСНК, или RISC), векторные процессоры и др.

В данном учебнике сделана попытка изложить в сжатой и доступной форме все разделы программы курса. Каждая глава учебника снабжена контрольными вопросами, позволяющими проверить усвоение материала. Материал скомпонован таким образом, что каждая последующая глава является логическим продолжением предыдущих.

# Глава 1

## ОСНОВНЫЕ СВЕДЕНИЯ О МИКРОПРОЦЕССОРАХ И МИКРОЭВМ

### 1.1. Классификация ЭВМ

**Классификация ЭВМ по принципу действия.** *Электронная вычислительная машина, компьютер* — комплекс технических средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач.

По принципу действия вычислительные машины делятся на три больших класса: цифровые (ЦВМ), аналоговые (АВМ) и гибридные (ГВМ). Критерием деления вычислительных машин на эти три класса является форма представления информации, с которой они работают.

*Цифровые вычислительные машины* — вычислительные машины дискретного действия, работают с информацией, представленной в дискретной, а точнее, в цифровой форме.

*Аналоговые вычислительные машины* — вычислительные машины непрерывного действия, работают с информацией, представленной в непрерывной (аналоговой) форме, т.е. в виде непрерывного ряда значений какой-либо физической величины (чаще всего электрического напряжения).

Аналоговые вычислительные машины весьма просты и удобны в эксплуатации; программирование задач для решения на них, как правило, нетрудоемкое; скорость решения задач изменяется по желанию оператора и может быть сделана сколь угодно большой (больше, чем у ЦВМ), но точность решения задач очень низкая (относительная погрешность 2...5%). На АВМ наиболее эффективно решать математические задачи, содержащие дифференциальные уравнения, не требующие сложной логики.

*Гибридные вычислительные машины* — 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.

Наиболее широкое применение получили ЦВМ с электрическим представлением дискретной информации — электронные цифровые вычислительные машины.

**Классификация ЭВМ по этапам создания.** По этапам создания и используемой элементной базе ЭВМ условно подразделяются:

- на 1-е поколение, 1950-е гг.: ЭВМ на электронных вакуумных лампах;
- 2-е поколение, 1960-е гг.: ЭВМ на дискретных полупроводниковых приборах (транзисторах);
- 3-е поколение, 1970-е гг.: ЭВМ на полупроводниковых интегральных схемах с малой и средней степенью интеграции (сотни — тысячи транзисторов в одном корпусе). *Интегральная схема (ИС)* — электронная схема специального назначения, выполненная в виде единого полупроводникового кристалла, объединяющего большое число диодов и транзисторов;
- 4-е поколение, 1980-е гг.: ЭВМ на больших и сверхбольших интегральных схемах-микропроцессорах (от десятков тысяч — до миллионов транзисторов в одном кристалле);
- 5-е поколение, 1990-е гг.: ЭВМ с многими десятками параллельно работающих микропроцессоров, позволяющих строить эффективные системы обработки знаний; ЭВМ на сверхсложных микропроцессорах с параллельно-векторной структурой, одновременно выполняющих десятки последовательных команд программы;
- 6-е и последующие поколения: оптоэлектронные ЭВМ с массовым параллелизмом и нейронной структурой — с распределенной сетью большого числа (десятки тысяч) несложных микропроцессоров, моделирующих архитектуру нейронных биологических систем.

Каждое следующее поколение ЭВМ имеет по сравнению с предшествующим существенно лучшие характеристики. Так, производительность ЭВМ и емкость всех запоминающих устройств увеличиваются, как правило, больше чем на порядок.

**Классификация ЭВМ по назначению.** По назначению ЭВМ можно разделить на три группы: универсальные (общего назначения), проблемно-ориентированные и специализированные.

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

Характерными чертами универсальных ЭВМ являются:

- высокая производительность;
- разнообразие форм обрабатываемых данных: двоичных, десятичных, символьных 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 ЭВМ служат 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 ЭВМ аппаратными и программными ресурсами. К проблемно-ориентированным ЭВМ можно отнести, в частности, всевозможные управляющие вычислительные комплексы.

Специализированные ЭВМ используются для решения узкого круга задач или реализации строго определенной группы функций. Такая узкая ориентация ЭВМ позволяет четко специализировать их структуру, существенно снизить сложность и стоимость ЭВМ при сохранении высокой производительности и надежности их работы. К специализированным ЭВМ 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.

**Классификация ЭВМ по размерам и функциональным возможностям.** По размерам и функциональным возможностям ЭВМ можно разделить на сверхмалые (микроЭВМ), малые (мини-ЭВМ), большие, и сверхбольшие (суперЭВМ). Функциональные возможности ЭВМ определяются их техническими характеристиками, основные из которых рассматриваются в следующем подразделе.

## 1.2. Базовые параметры и технические характеристики ЭВМ

ЭВМ — самые сложные системы, и сравнение их может быть проведено весьма относительно. Провести сравнительную оценку ЭВМ разных типов возможно, рассмотрев совокупность различных показателей. Главными из них являются технические характеристики, стоимость приобретения, стоимость эксплуатации. Для специализированных ЭВМ это может быть еще и масса.

Основные технические характеристики — это производительность, емкость оперативного запоминающего устройства (ОЗУ),

пропускная способность подсистемы ввода-вывода, надежность функционирования.

**Производительность** — это показатель эффективности ЭВМ или вычислительного средства, использующий некоторые характеристики скорости работы системы. Измеряется обычно в числе наиболее часто повторяющихся либо средних по длительности операций в секунду (опер/с), либо в числе миллионов операций над числами с плавающей точкой в секунду MIPS (Million Instructions Per Second). Значения производительности изменяются от сотен опер/с у персональных компьютеров, микроЭВМ и микропроцессоров до  $10^9$  опер/с и выше у суперЭВМ. Производительность зависит не только от самой ЭВМ, но и от особенностей обрабатываемой информации, таких, как разрядность слов, форма представления чисел — с плавающей или фиксированной точкой, частоты повторения различных операций в общем потоке выполняемых программ и др. Поэтому оценка производительности осуществляется для тестовых наборов задач и предварительно выявляется процентное содержание команд различного типа. Значения производительности могут использоваться для ориентировочной оценки реальной производительности при решении конкретных задач. Иногда производительность удобно оценивать как число выполняемых команд в минуту, число выполняемых заданий в день и т. д.

**Емкость ОЗУ и внешнего запоминающего устройства (ВЗУ)** определяет возможности ЭВМ по выполнению сложных программ с обработкой больших объемов информации. Обозначается  $C_{OЗУ}$ , измеряется в битах, байтах, словах, килобайтах (Кбайт), мега- и гигабайтах (Мбайт и Гбайт).

**Разрядность** — число двоичных разрядов, которыми оперирует ЭВМ.

**Пропускная способность** подсистемы ввода-вывода позволяет определить возможности ЭВМ при обмене информацией с различными внешними устройствами или другими ЭВМ. Измеряется максимальным количеством единиц информации, передаваемых через подсистему ввода-вывода за единицу времени. Единицы измерения — бит/с, равный 1 Бод; байт/с; Кбайт/с; Мбайт/с. Значения пропускной способности подсистемы ввода-вывода измеряются от сотен байт/с до сотен мегабайт/с. В справочной литературе обычно указывается количество каналов ввода-вывода и скорость работы каждого канала.

**Надежность функционирования** ЭВМ выражается показателями, имеющими вероятностный характер, и основывается на значениях  $\lambda$ -характеристик интенсивностей отказов составляющих элементов. Основные показатели надежности функционирования:

- вероятность безотказной работы в течение заданного интервала времени  $\tau$ ,  $P(\tau)$ ;



• время наработки до первого отказа  $T_n$  иногда обозначается как  $T_o$ ;

• среднее время восстановления работоспособности  $T_b$ ;

• коэффициент готовности  $K_r = T_n / (T_n + T_b)$ ;

• живучесть системы — способность системы к сохранению своих основных функций хотя бы при пониженной эффективности системы при воздействии факторов катастрофического характера — в отличие от надежности как способности системы выполнять свои функции в нормальных, заранее рассчитанных условиях.

Четкое разделение ЭВМ на классы затруднено, так как микроэлектроника развивается быстро и ЭВМ по одним признакам относится, например, к мини-ЭВМ, а по другим — к ЭВМ высокой производительности (табл. 1.1).

Исторически первыми появились *большие ЭВМ*, элементная база которых прошла путь от электронных ламп до интегральных схем со сверхвысокой степенью интеграции.

Производительность больших ЭВМ оказалась недостаточной для ряда задач: прогнозирования метеообстановки, управления сложными оборонными комплексами, моделирования экологических систем и др. Это явилось предпосылкой для разработки и создания *суперЭВМ*, самых мощных вычислительных систем, интенсивно развивающихся и в настоящее время.

Появление в 1970-х гг. *малых ЭВМ* обусловлено, с одной стороны, прогрессом в области электронной элементной базы, а с другой — избыточностью ресурсов больших ЭВМ для ряда приложений. Малые ЭВМ используются чаще всего для управления технологическими процессами. Они более компактны и значительно дешевле больших ЭВМ.

Изобретение в 1969 г. микропроцессора (МП) привело к появлению в 1970-х гг. еще одного класса ЭВМ — *микроЭВМ*. Именно наличие МП послужило первоначально определяющим призна-

Таблица 1.1

Сравнительные параметры классов ЭВМ

Параметр	МикроЭВМ	Малые ЭВМ	Большие ЭВМ	СуперЭВМ
Производительность, MIPS	1 ... 100	1 ... 100	10 ... 1000	1000 ... 100 000
Емкость оперативной памяти, Мбайт	4 ... 256	4 ... 512	64 ... 10000	2000 ... 10 000
Емкость ВЗУ, Гбайт	0,5 ... 10	2 ... 100	50 ... 1000	500 ... 5000
Разрядность, бит	16 ... 64	16 ... 64	32 ... 64	64 ... 128

ком микроЭВМ. Сейчас микропроцессоры используются во всех без исключения классах ЭВМ.

Условно микроЭВМ можно разделить на многопользовательские (серверы), однопользовательские (персональные, рабочие станции).

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

*Серверы (server)* — многопользовательские мощные микроЭВМ в вычислительных сетях, выделенные для обработки запросов от всех станций сети.

*Персональные компьютеры (ПК)* — однопользовательские микроЭВМ, удовлетворяющие требованиям общедоступности и универсальности применения.

*Рабочие станции (work station)* представляют собой однопользовательские мощные микроЭВМ, специализированные для выполнения определенного вида работ (графических, инженерных, издательских и др.).

Конечно, вышеприведенная классификация весьма условна, ибо мощный современный ПК, оснащенный проблемно-ориентированным программным и аппаратным обеспечением, может использоваться и как полноправная рабочая станция, и как многопользовательская микроЭВМ, и как хороший сервер, по своим характеристикам почти не уступающий малым ЭВМ. Рассмотрим кратко современное состояние некоторых классов ЭВМ.

### 1.3. Архитектура вычислительной системы. Структура аппаратной части и назначение основных функциональных узлов ЭВМ

*Вычислительная система (ВС)* — взаимосвязанная совокупность средств вычислительной техники как программных, так и аппаратных, включающая не менее двух процессоров или вычислительных машин, причем хотя бы один из этих процессоров выполняет роль основного, центрального (ЦП).

ВС может включать в себя несколько процессоров, различающихся по выполняемым функциям. Для производства вычислений, задаваемых оператором — абонентом, применяется центральный процессор; для управления вводом-выводом данных — периферийный процессор; для переработки информации объекта управления — управляющий процессор; для решения задач определенного типа — специализированный процессор и т.д.

*Микропроцессор* — полупроводниковая микросхема или комплект-набор микросхем, на которых реализуется центральный про-

цессор. Первый микропроцессор Intel 4004 на четырех микросхемах появился в 1971 г., тогда много спорили о его применимости и покупательском спросе на него. Он был построен фирмой Intel для калькулятора, реализующего простейший набор команд аппаратным способом, но при этом имелась возможность хранить сложные последовательности команд в постоянном запоминающем устройстве (ПЗУ). Один из кристаллов набора представлял собой память с произвольной выборкой.

*Архитектура ВС* — система основных функциональных средств, доступных пользователю, и принципов организации процесса переработки информации в ЭВМ на уровне операций над массивами и задачами в целом. Как видно из этого определения, архитектура объединяет аппаратные, микропрограммные и программные средства вычислительной техники в систему, т.е. архитектура — логическая организация системы. В контексте разработки вычислительной системы и проектирования ее аппаратных средств термин «архитектура» используется для описания принципа действия, конфигурации и взаимного соединения основных логических узлов ЭВМ. Обычно частью, а зачастую и основой такого описания служит подробная структурная схема конкретной реальной машины.

Под *микропроцессорной системой* (МПС) будем понимать вычислительную либо управляющую систему, построенную на микропроцессорных наборах.

В свою очередь, основным рабочим элементом вычислительного устройства, ЭВМ является ЦП, который состоит из арифметико-логического устройства (АЛУ) и устройства управления, а иногда и внутренней памяти. По мере того, как функции вычислительной системы становятся все более распределенными и автономными, этот термин — «центральный процессор» приобретает более широкое толкование.

Центральный процессор ВС выполняет вычисления, предусмотренные алгоритмами решаемых задач. В отличие от него вспомогательный процессор (сопроцессор) применяется для предварительной обработки информации, например, для «сжатия» информации, диспетчирования, не предусматриваемой этими алгоритмами; либо для выполнения неосновных действий — редактирование вводимой и выводимой информации, поиск, накопление и подготовка исходных данных для решаемой задачи и др. Вспомогательный процессор может быть частью вычислительного устройства, ЭВМ либо отдельной машиной. В последнем случае его сопряжение с основным процессором (ЦП) является достаточным для того, чтобы эта совокупность также называлась вычислительной системой.

Управляющая МПС входит как подсистема в состав систем автоматического управления (САУ), предназначенных для автоматического поддержания желаемого режима работы объекта.

Назначение средств вычислительной техники — ввод, хранение, обработка, передача и вывод информации. Соответственно, над информацией производятся следующие действия: ее можно создавать, передавать, запоминать, искать, принимать, копировать, обрабатывать и разрушать.

Для выполнения своего назначения — проведения действий над информацией аппаратная часть ВС должна включать в себя основные узлы, показанные на рис. 1.1. В структурной схеме можно проследить два типа информационных потоков. Первый поток — данные входные и выходные. Широкими стрелками (см. рис. 1.1) показаны потоки входных, выходных данных и адресов, тонкими — потоки управляющих сигналов  $V_i$ . Это собственно данные в виде чисел, слов и адресов, по которым хранятся или передаются данные. В МПС данные передаются по шине данных (ШД), адреса — по адресной шине (ША). Второй тип информационного потока — команды управления узлами и устройствами, соответствующие выполняемым операциям программы. Сигналы управления в МПС передаются по управляющей шине (ШУ).

Информация, подлежащая обработке, поступает либо от оператора с помощью устройства ввода (УВв), либо, в случае управляющей системы, — с датчиков объекта управления. В последнем случае устройство ввода будет представлять собой один или несколько аналого-цифровых преобразователей. Назначение УВв — преобразование информации без изменения ее содержания в вид, удобный для ЭВМ, т.е. в последовательность сигналов двоичного цифрового кода.

Результаты обработки информации — выходные данные — с помощью устройства вывода (УВыв) преобразуются в вид, удобный

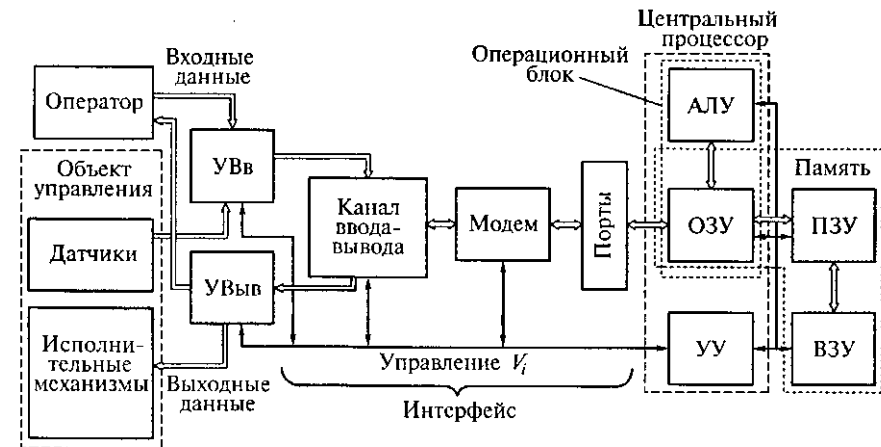


Рис. 1.1. Обобщенная структурная схема вычислительной МПС

для восприятия оператором или исполнительными органами объекта управления. В САУ устройством вывода будут являться один или несколько цифроаналоговых преобразователей.

Устройства ввода-вывода информации (УВВ) в отличие от центральных устройств (процессора, памяти, устройства управления) называют внешними (ВУ) или периферийными устройствами (ПУ). Информация от внешних устройств к центральным и наоборот передается по каналу ввода-вывода. Это специализированный процессор, в котором реализованы аппаратные и программные средства пересылки данных и управления операциями ввода-вывода. Совокупность унифицированных аппаратных, программных и конструктивных средств, необходимых для реализации взаимодействия различных функциональных узлов в ВС при условиях, предписанных стандартом по обеспечению информационной, электрической и конструктивной совместимости этих узлов, называется *стандартным интерфейсом*.

Интерфейс может включать в себя несколько каналов. В случае большой удаленности терминалов от центральных устройств интерфейс включает в себя и канал связи. Это коаксиальные кабели, телефонные линии, волоконно-оптические, лазерные и спутниковые каналы. Все они имеют конечную скорость передачи данных и задержку распространения сигналов. Каналы связи подвержены влиянию помех, приводящих к возникновению ошибок.

Для осуществления передачи информации по каналам связи необходимо преобразовать (модулировать) поток битов двоичного цифрового кода в аналоговые сигналы, пригодные для передачи, а принимаемые аналоговые сигналы необходимо преобразовать обратно в цифровую форму — демодулировать. Для этого подключаются устройства модуляции-демодуляции — *модемы*.

К ВУ относится и внешнее запоминающее устройство, используемое для хранения справочной информации, больших массивов данных, не участвующих непосредственно в обработке в данное время, а также программ задач, подлежащих выполнению. ВЗУ, хотя и связано с центральными устройствами и управляется ими, в конструктивном отношении выполнено отдельно. Когда термин «ВЗУ» используется в абсолютном смысле, то обычно имеется в виду накопитель на магнитном диске. ВЗУ может коллективно использоваться несколькими ЭВМ.

Перейдем к рассмотрению центральных устройств МПС.

АЛУ является частью центрального процессора, формирующей функции двух входных переменных операндов, порождает одну выходную переменную. Эти функции обычно состоят: из простой арифметической операций — сложения; простых логических операций — И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ; операций сдвига арифметического, логического или циклического.

ОЗУ, или внутренняя или основная память, непосредственно связано с процессором. В нем хранятся программные команды и данные, участвующие в данное время в вычислениях. В него записываются результаты вычислений перед пересылкой их в ВЗУ или УВВ.

В быстродействующих МПС в качестве буфера между АЛУ и ОЗУ используется сверхоперативное запоминающее устройство (СОЗУ), или кэш-память.

Наиболее сложным устройством является устройство управления (УУ), вырабатывающее из потока поступающей информации последовательность управляющих сигналов, координирующих совместную работу всех узлов и обеспечивающих управление перемещением информации между памятью, АЛУ и другими частями ЭВМ. Основной функцией УУ является управление выполнением команд и операций.

Выборка команд из ОЗУ осуществляется естественным (в порядке очередности) или принудительным (по адресу, содержащемуся в предыдущей команде) способом. Команды выполняются по частям, называемым микрокомандами. Последовательность микрокоманд, реализующая заданный алгоритм, образует микропрограмму. Каждая микрокоманда соответствует одной или нескольким микрооперациям. Микропрограммы хранятся в ПЗУ.

#### Контрольные вопросы

1. Каково назначение средств вычислительной техники?
2. В чем отличие вычислительной микропроцессорной системы от управляющей МПС?
3. В чем заключается отличие универсальной ЭВМ от специализированной?
4. Какие по меньшей мере пять функциональных устройств составляют структуру цифровой ЭВМ?
5. Перечислите типы процессоров по функциональному назначению.
6. Каков состав ЦП и каковы его функции?
7. Каково назначение АЛУ?
8. Каково назначение устройства управления ЦП?
9. В какой памяти располагаются микрокоманды?
10. В какой памяти располагаются временные, часто сменяющиеся программы и данные?
11. Каково назначение внешних устройств ВС?
12. Какие два типа информационных потоков вводятся в ЭВМ и помещаются в память?
13. Перечислите основные технические характеристики ЭВМ и порядок их значений для микроЭВМ.
14. На какие классы подразделяются ЭВМ по размерам и функциональным возможностям?



## Глава 2

# ИНФОРМАЦИОННО-ЛОГИЧЕСКИЕ ОСНОВЫ ЭВМ

### 2.1. Системы счисления

В цифровых системах, к которым относятся ЭВМ, информация представляется в двоичном (бинарном) коде, состоящем из двух символов: логического нуля «0» и логической единицы «1». Наличие двух символов определяет применение в цифровых устройствах *двоичной системы счисления*.

Системы счисления, в которых значение каждой цифры определяется не только символом, но и разрядом (позицией в числе  $N$ ), называют *позиционными*. Нетрудно заметить, что римская система счисления является непозиционной. Как известно, любое число  $N$  в позиционной системе счисления с основанием  $q$  можно представить в виде полинома

$$N_{(q)} = a_{m-1}q^{m-1} + a_{m-2}q^{m-2} + \dots + a_1q^1 + a_0q^0 + a_{-1}q^{-1} + a_{-2}q^{-2} + \dots + a_{-k}q^{-k}, \quad (2.1)$$

где  $q$  — основание системы счисления;  $a_i$  — коэффициент  $i$ -го разряда (элемента числа  $N_{(q)}$ ), принимающий значения 0, 1, ...,  $(q-1)$  — в зависимости от величины  $N_{(q)}$ ;  $m$  — номер разряда целой части, отчитываемый от нулевого;  $k$  — число цифр в дробной части числа.

#### Пример 2.1

Число  $7526_{(10)}$  в обычной десятичной системе ( $q = 10$ ) представляют согласно формуле (2.1):

$$N_{(10)} = 7526_{(10)} = 7 \cdot 10^3 + 5 \cdot 10^2 + 2 \cdot 10^1 + 6 \cdot 10^0.$$

Как видно, коэффициент нулевого разряда ( $a_0 = 6$ ) определяет количество единиц в числе  $N_{(10)}$ , коэффициент первого разряда ( $a_1 = 2$ ) — число десятков и т.д.

Для представления числа в двоичной системе счисления принимаем в выражении (2.1) основание  $q = 2$ . Тогда

$$N_{(2)} = a_m 2^m + a_{m-1} 2^{m-1} + \dots + a_1 2^1 + a_0 2^0, \quad (2.2)$$

где  $a_i$  — коэффициент при разряде, принимающий одно из двух значений («0» или «1») в зависимости от величины  $N_{(2)}$ .

#### Пример 2.2

Число  $26_{(10)}$  в двоичной системе согласно (2.2) можно представить в виде

$$26_{(10)} = \frac{1 \cdot 2^4}{16} + \frac{1 \cdot 2^3}{8} + \frac{0 \cdot 2^2}{0} + \frac{1 \cdot 2^1}{2} + \frac{0 \cdot 2^0}{0} = 11010_{(2)}.$$

В цифровых вычислительных системах используют также комбинированную, *десятично-двоичную*, систему счисления, облегчающую запись больших чисел с применением двоичного кода. В этом случае каждый разряд десятичного числа записывают двоичным кодом, используя для этого соответствующие тетрады, т.е. четырехразрядные двоичные элементы (числа 8 и 9 иначе представить нельзя).

#### Пример 2.3

Число  $197_{(10)}$  в десятично-двоичной системе имеет вид

$$197_{(10)} = \underbrace{0001}_1 \underbrace{1001}_9 \underbrace{0111}_7. \quad (2.3)$$

Недостатком рассматриваемой системы является ее избыточность для чисел 7 и менее (недоиспользуются многие двоичные разряды). Для устранения этого недостатка в ЭВМ используют *восьмеричную систему счисления* ( $q = 8$ ), которую можно затем записать в двоичном коде (2.2) с использованием для каждой цифры только трех разрядов — триад.

Восьмеричное число в соответствии с (2.1) представляют в виде

$$N_{(8)} = a_m 8^m + a_{m-1} 8^{m-1} + \dots + a_1 8^1 + a_0 8^0.$$

#### Пример 2.4

Анализируемое ранее число  $197_{(10)}$  в восьмеричной системе имеет вид

$$197_{(10)} = \frac{3 \cdot 8^2}{192} + \frac{0 \cdot 8^1}{0} + \frac{5 \cdot 8^0}{5} = 305_{(8)}. \quad (2.4)$$

Важнейшее свойство восьмеричной системы состоит в следующем: при записи каждого из разрядов восьмеричной системы триадой двоичного кода полученное выражение представляется в двоичной системе счисления.

Для конкретизации этого важнейшего свойства восьмеричной системы, обеспечивающей ей широкое использование в вычис-

лительной технике, вновь воспользуемся числом  $305_{(8)}$ , записанном в виде (2.4):

$$305_{(8)} = \underbrace{011}_{3} \underbrace{000}_{0} \underbrace{101}_{5} = \underbrace{011000101}_{197}_{(2)} = 197_{(10)}$$

Восьмеричные системы применяют в ЭВМ для кодирования адресов и команд. Для этого сначала составляют в восьмеричной системе соответствующую программу, а затем переводят ее в двоичную систему, которую и вводят в вычислительную машину.

Еще более удобна *шестнадцатеричная (цифробуквенная) система счисления*, которая образуется из десяти цифровых (0, 1, ..., 9) и шести буквенных (A, B, ..., F) символов. При этом буквы A, B, ..., F изображают соответственно числа 10, 11, ..., 15.

#### Пример 2.5

Число  $B7E_{(16)}$  обозначает

$$N_{(16)} = B7E_{(16)} = \underbrace{11 \cdot 16^2}_{2816} + \underbrace{7 \cdot 16^1}_{112} + \underbrace{14 \cdot 16^0}_{14} = 2942_{(10)}$$

Заметим, что при записи каждого из разрядов шестнадцатеричного числа тетрадами двоичного кода получают значение этого числа в двоичной системе счисления.

Для удобства сопоставления рассмотренных систем счисления в табл. 2.1 приведены первые 22 числа натурального ряда чисел.

Таблица 2.1

Натуральный ряд чисел в различных системах

Десятичная	Шестнадцатеричная	Восьмеричная	Двоичная	Десятичная	Шестнадцатеричная	Восьмеричная	Двоичная
0	0	0	0	11	B	13	1011
1	1	1	1	12	C	13	1100
2	2	2	10	13	D	15	1101
3	3	3	11	14	E	16	1110
4	4	4	100	15	F	17	1111
5	5	5	101	16	10	20	10000
6	6	6	110	17	11	21	10001
7	7	7	111	18	12	22	10010
8	8	10	1000	19	13	23	10011
9	9	11	1001	20	14	24	10100
10	A	12	1010	21	15	25	10101

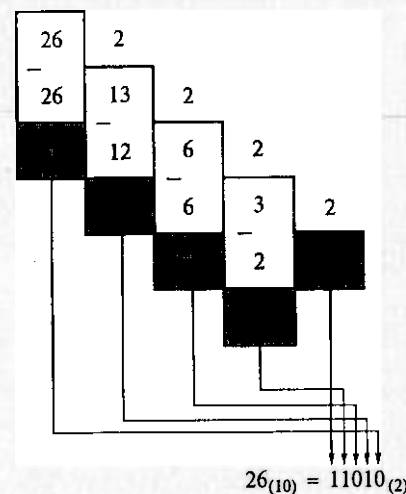
### 2.1.1. Перевод целых чисел из десятичной системы счисления в двоичную

Правило перевода числа из десятичной системы счисления в двоичную заключается в последовательном его делении на 2 (основание двоичной системы) до тех пор, пока частное от деления не станет равным 1. Тогда искомое число в двоичной форме можно сформировать из этого частного и всех остатков, начиная с последнего (справа налево).

#### Пример 2.6

Перевести число 26 в двоичную систему.

Выполняя последовательное деление 26 на 2, получаем



### 2.1.2. Перевод дробных чисел

Дробное число с основанием  $q_1$  переводится в систему счисления с основанием  $q_2$  путем последовательного умножения  $A_{q_1}$  на основание  $q_2$ , записанное в виде числа с основанием  $q_1$ .

При каждом умножении целая часть произведения берется в виде очередной цифры соответствующего разряда, а оставшаяся дробная часть принимается за новое множимое.

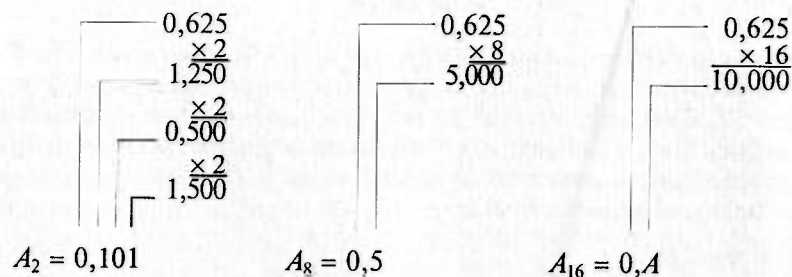
Число умножений определяет разрядность полученного результата, представляющего число  $A_{q_1}$  в системе счисления  $q_2$ .

#### Пример 2.7

$$A_{10} = 0,625 \quad A_2 = ? \quad A_8 = ? \quad A_{16} = ?$$

БИБЛИОТЕКА ГОУ СПО "ВГЭТК" 17  
Инв. № 5307

Решение.



Так как двоичная, восьмеричная и шестнадцатеричная системы связаны через степени числа 2, то преобразования между ними можно выполнять другим более простым способом. Для перевода из шестнадцатеричной (восьмеричной) системы счисления в двоичную достаточно двоичным кодом записать шестнадцатеричные коды цифр тетрадами (по 4 двоичных разряда) и триадами (по 3 двоичных разряда) — для восьмеричных цифр. Обратный перевод из двоичного кода производится в обратном порядке: двоичное число разбивается влево и вправо от границы целой и дробной частей на тетрады — для последующей записи цифр в шестнадцатеричном представлении, на триады — для записи их значений восьмеричными цифрами.

### 2.1.3. Представление информации в ЭВМ

ЭВМ относятся к разряду цифровых устройств. Эти устройства предназначены для обработки цифровой информации, представляемой в двоичной форме. В этом случае цифровой сигнал может принимать только два значения: логические «0» и «1».

Понятия «1» и «0» являются условными, поскольку они отражают два состояния цифровых систем: «включено» и «выключено». При этом, если «1» представляется высоким потенциалом (уровнем электрического напряжения), а «0» — низким, то имеют положительную потенциальную логику. Соответственно при представлении «1» низким потенциалом, а «0» высоким получают отрицательную потенциальную логику. Потенциальная логика широко применяется в интегральной микроэлектронике, реализующей устройства ЭВМ, поскольку при ее реализации можно избежать использования трансформаторов, конденсаторов и других устройств, трудно изготавливаемых средствами микроэлектронной технологии.

Пусть представление информации обеспечивается посредством алфавита, составленного из  $n$  символов, каждый из которых имеет равную вероятность появления. Согласно теории передачи инфор-

мации максимальное количество информации  $I_i$ , передаваемое одним символом алфавита, равно двоичному логарифму числа  $n$ , т. е.

$$I_i = \log_2 n.$$

В этом случае величину  $I_i$  измеряют в единицах информации — битах (от англ. *binary digit* — двоичная цифра). Действительно, при двоичном алфавите ( $n = 2$ )

$$I_i = \log_2 2 = 1 \text{ бит.}$$

Таким образом, мера информации 1 бит выражает такое ее количество, которое может передать один символ (цифра) двоичного алфавита.

В цифровой электронике это понятие распространяют на двоичные системы с любым числом разрядов, полагая при этом информационную емкость каждого разряда равной одному биту.

В большинстве ЭВМ число разрядов в группе цифровых символов, несущих некоторую порцию информации (так называемое машинное слово), кратно 8. Поэтому информационная емкость машинного слова кратна 8 бит. Единицу количества информации достоинством 8 бит именуют байтом (от англ. *byte* — слог), т. е. 1 байт = 8 бит. Единицы измерения байт и особенно килобайт (Кбайт) находят широкое применение при оценке информационных показателей ЭВМ (объемов памяти и т. п.).

Следует отметить, что под приставкой «кило» в цифровой технике понимают увеличение не в 1000, а в 1024 раз — число, которое представляет собой десятичную степень числа 2. Так,

$$1 \text{ Кбайт} = 2^{10} \text{ байт} = 1024 \text{ байт};$$

$$1 \text{ Мбайт} = 2^{20} \text{ байт} = 1\,048\,576 \text{ байт.}$$

В вычислительных машинах применяются две формы представления двоичных чисел:

- естественная форма или форма с фиксированной запятой (точкой);
- нормальная форма или форма с плавающей запятой (точкой).

С фиксированной запятой все числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть от дробной.

Эта форма наиболее проста, естественна, но имеет небольшой диапазон представления чисел и поэтому не всегда приемлема при вычислениях. Диапазон значащих чисел  $N$  в системе счисления с основанием  $q$  при наличии  $m$  разрядов в целой части и  $s$  разрядов в дробной части числа (без учета знака числа) имеет вид

$$q^{-s} \leq N \leq q^m.$$



При  $q = 2$ ,  $m = 10$  и  $s = 6$ :

$$0,015 \leq N \leq 1024.$$

Если в результате операции получится число, выходящее за допустимый диапазон, происходит переполнение разрядной сетки, и дальнейшие вычисления теряют смысл. В современных ЭВМ естественная форма представления используется как вспомогательная и только для целых чисел.

С плавающей запятой каждое число изображается в виде двух групп цифр. Первая группа цифр называется *мантиссой*, вторая — *порядком*, причем абсолютная величина мантиссы должна быть меньше 1, а порядок — целым числом. В общем виде число в форме с плавающей запятой может быть представлено в виде

$$N = \pm Mq^{\pm p},$$

где  $M$  — мантисса числа ( $|M| < 1$ );  $q$  — основание системы счисления;  $p$  — порядок числа (целое число).

Нормальная форма представления имеет огромный диапазон отображения чисел и является основной в современных ЭВМ. Диапазон значащих чисел в системе счисления с основанием  $q$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		▶		0	@	P	.	p	A	P	a	█	└	┘	p	Ё
1	⊙	◀	!	1	A	Q	a	q	Б	С	б	▬	┌	┐	с	ё
2	⊙	↕	"	2	B	R	b	r	В	Т	в	■	└	┘	т	ё
3	♥	!!	#	3	C	S	c	s	Г	У	г		└	┘	у	ё
4	♦		\$	4	D	T	d	t	Д	Ф	д	└	-	┘	ф	ї
5	♣	\$	%	5	E	U	e	u	Е	Х	е	└	+	┘	х	і
6	♠	-	&	6	F	V	f	v	Ж	Ц	ж	└	└	┘	ц	ў
7	•	↕	'	7	G	W	g	w	З	Ч	з	└	└	┘	ч	ў
8	■	↑	(	8	H	X	h	x	И	Ш	и	└	└	┘	ш	°
9	○	↓	)	9	I	Y	i	y	Й	Щ	й	└	└	┘	щ	•
A	■	→	*	:	J	Z	j	z	К	Ъ	к		└	└	ъ	•
B	♠	←	+	;	K	[	k	{	Л	Ы	л	└	└	┘	ы	√
C	♀	└	.	<	L	\	l		М	Ь	м	└	└	┘	ь	№
D	♠	↔	-	=	M	]	m	}	Н	Э	н	└	=	┘	э	□
E	♠	▲	.	>	N	^	n	~	О	Ю	о	└	└	┘	ю	•
F	♠	▼	/	?	O	-	o	△	П	Я	п	└	└	┘	я	

Рис. 2.1. Таблица кодирования ASCII

при наличии  $m$  разрядов у мантиссы и  $s$  разрядов у порядка (без учета знаковых разрядов порядка и мантиссы) следующий:

$$q^{-m}q^{-(q^s-1)} \leq N \leq (1 - q^{-m})q^{-(q^s-1)}.$$

При  $q = 2$ ,  $m = 10$  и  $s = 6$  диапазон чисел простирается примерно от  $10^{-19}$  до  $10^{19}$ .

Знак числа обычно кодируется двоичной цифрой, при этом код 0 означает знак «+», код 1 — знак «-».

Для представления символьной информации в персональных компьютерах применяется код ASCII. Код ASCII (от англ. *American Standard Code for Information Interchange* — Американский стандартный код для обмена информацией) имеет основной стандарт и его расширение (рис. 2.1). Основным стандартом для кодирования символов использует шестнадцатеричные коды 00—7F, расширение стандарта — 80—FF. Основным стандартом является международным и используется для кодирования управляющих символов, цифр и букв латинского алфавита; в расширении стандарта кодируются символы псевдографики и буквы национального алфавита (естественно, в разных странах свои).

## 2.2. Арифметические основы ЭВМ

### 2.2.1. Машинные коды

Для алгебраического представления чисел (т. е. для представления положительных и отрицательных чисел) в машинах используются специальные коды: прямой, обратный и дополнительный. Причем два последних позволяют заменить неудобную для ЭВМ операцию вычитания на операцию сложения с отрицательным числом; дополнительный код обеспечивает более быстрое выполнение операций, поэтому в ЭВМ применяется чаще именно он.

Прямой код двоичного числа образуется из абсолютного значения этого числа и кода знака (ноль или единица) перед его старшим числовым разрядом.

Пример 2.8

$$\begin{aligned} A_{10} &= +10; & A_2 &= +1010; & [A_2]_n &= 01010; \\ B_{10} &= -14; & B_2 &= -1110; & [B_2]_n &= 11110. \end{aligned}$$

Обратный код двоичного числа образуется по следующему правилу. Обратный код положительных чисел совпадает с их прямым кодом. Обратный код отрицательного числа содержит единицу в знаковом разряде числа, а значащие разряды числа заменяются на инверсные, т. е. нули заменяются единицами, а единицы — нулями.

Пример 2.9

$$A_{10} = +4; \quad A_2 = +100; \quad [A_2]_n = [A_2]_o = 0100;$$

$$B_{10} = -12; \quad B_2 = -1100; \quad [B_2]_o = 10011.$$

Дополнительный код положительных чисел совпадает с их прямым кодом. Дополнительный код отрицательного числа представляет собой результат суммирования обратного кода числа с единицей младшего разряда ( $2^0$  — для целых чисел,  $2^{-k}$  — для дробных).

Пример 2.10

$$A_{10} = +18; \quad A_2 = +10010; \quad [A_2]_n = [A_2]_o = [A_2]_d = 010010;$$

$$B_{10} = -12; \quad B_2 = -1100; \quad [B_2]_d = [B_2]_o + 2^0 = 10011 + 1 = 10100.$$

Модифицированные обратные и дополнительные коды двоичных чисел отличаются соответственно от обратных и дополнительных кодов удвоением значений знаковых разрядов. Знак «+» в этих кодах кодируется двумя нулевыми знаковыми разрядами, а знак «-» — двумя единичными разрядами.

Пример 2.11

$$A_{10} = 8; \quad A_2 = +100; \quad [A_2]_n = [A_2]_o = [A_2]_d = 0100;$$

$$[A_2]_{mo} = [A_2]_{md} = 00100;$$

$$B_{10} = -8; \quad B_2 = -100; \quad [B_2]_o = 1011; \quad [B_2]_d = 1100;$$

$$[B_2]_{mo} = 11011; \quad [B_2]_{md} = 11100.$$

Целью введения модифицированных кодов являются фиксация и обнаружение случаев получения неправильного результата, т.е. когда его значение превышает максимально возможный результат в отведенной разрядной сетке машины. В этом случае перенос из значащего разряда может исказить значение младшего знакового разряда. Значение знаковых разрядов «01» свидетельствует о положительном переполнении разрядной сетки, а «10» — об отрицательном переполнении. В настоящее время практически во всех моделях ЭВМ роль удвоенных разрядов для фиксации переполнения разрядной сетки играют переносы, идущие в знаковый и из знакового разряда.

### 2.2.2. Арифметические операции над числами с фиксированной точкой

**Сложение (вычитание).** Операция вычитания приводится к операции сложения путем преобразования чисел в обратный или дополнительный код. Пусть даны числа  $A \geq 0$  и  $B \geq 0$ , тогда операция алгебраического сложения выполняется в соответствии с табл. 2.2.

Таблица преобразования кодов при алгебраическом сложении

Требуемая операция	Необходимое преобразование
$A + B$	$A + B$
$A - B$	$A + (-B)$
$-A + B$	$(-A) + B$
$-A - B$	$(-A) + (-B)$

Скобки в представленных выражениях указывают на замену операции вычитания операцией сложения с обратным или дополнительным кодом соответствующего числа. Сложение двоичных чисел осуществляется последовательно, поразрядно. При выполнении сложения цифр необходимо соблюдать следующие правила.

1. Слагаемые должны иметь одинаковое число разрядов. Для выравнивания разрядной сетки слагаемых можно дописывать незначащие нули слева к целой части числа и незначащие нули справа к дробной части числа.

2. Знаковые разряды чисел участвуют в сложении так же, как и значащие.

3. Необходимые преобразования кодов производятся с изменением знаков чисел. Приписанные незначащие нули изменяют свое значение при преобразованиях по общему правилу.

4. При образовании единицы переноса из старшего знакового разряда, в случае использования обратного кода, эта единица складывается с младшим числовым разрядом. При использовании дополнительного кода единица переноса теряется. Знак результата формируется автоматически, результат представляется в том коде, в котором представлены исходные слагаемые.

Пример 2.12

Сложить два числа  $A_{10} = 6; B_{10} = 16$ .

$$A_2 = +110 = +0110;$$

$$B_2 = +10000.$$

Исходные числа имеют различную разрядность, необходимо провести выравнивание разрядной сетки:

$$[A_2]_n = [A_2]_o = [A_2]_d = 000110;$$

$$[B_2]_n = [A_2]_o = [A_2]_d = 010000.$$

Сложение в обратном или дополнительном коде дает один и тот же результат  $C$

$$\begin{array}{r} 000110 \\ + 010000 \\ \hline C_2 = 010110; \\ C_{10} = +22. \end{array}$$

**Умножение.** Умножение двоичных чисел наиболее просто реализуется в прямом коде. Рассмотрим, каким образом оно приводится к операциям сложения и сдвигам.

*Пример 2.13*

Умножить два числа  $A_{10} = 6$ ;  $B_{10} = 5$ .

Перемножим эти числа, представленные прямыми двоичными кодами, так же, как это делается в десятичной системе.

$$\begin{array}{r} [A_2]_n = 110 \text{ — множимое;} \\ \times \\ [B_2]_n = \underline{101} \text{ — множитель;} \\ \hline 110 \\ + 000 \\ \hline 110 \\ \hline [C_2]_n = 11110; \\ C_{10} = 35. \end{array}$$

Нетрудно видеть, что произведение получается путем сложения частных произведений, представляющих собой разряды множимого, сдвинутые влево в соответствии с позициями разрядов множителя. Частные произведения, полученные умножением на нуль, игнорируются. Важной особенностью операции умножения  $n$ -разрядных сомножителей является увеличение разрядности произведения до  $n + n = 2n$ . Знак произведения формируется путем сложения знаковых разрядов сомножителей. Возможные переносы из знакового разряда игнорируются.

**Деление.** Операция деления, как и в десятичной арифметике, является обратной операции умножения. Покажем, что и эта операция приводится к последовательности операций сложения и сдвига.

*Пример 2.14*

Разделить два числа  $A_{10} = 40$ ;  $B_{10} = 5$ .

$$\begin{array}{r} [A_2]_n = 101000; \\ [B_2]_n = 101; \\ [C_2]_n = 1000; \\ \hline \begin{array}{r} 101000 \quad | \quad 101 \\ \underline{101} \quad \quad \quad 1000 \\ 0 \end{array} \\ \hline C_{10} = 8. \end{array}$$

Деление произведено так же, как это делается обычно в десятичной системе. Сначала проверяется, можно ли вычесть значение делителя из старших разрядов делимого. Если возможно, то в разряде частного записывается единица и определяется частная разность. В противном случае в частное записывается нуль и разряды делителя сдвигаются вправо на один разряд по отношению к

разрядам делимого. К полученной предыдущей разности сносится очередная цифра делимого, и данный процесс повторяется, пока не будет получена необходимая точность. Если учесть, что все вычитания в ЭВМ заменяются сложением в обратном или в дополнительном коде, то действительно операция деления приводится к операциям сложения и сдвигам вправо разрядов делителя относительно разрядов делимого. Отметим, что делимое перед операцией деления должно быть приведено к  $2n$ -разрядной сетке. Только в этом случае при делении на  $n$ -разрядный делитель получается  $n$ -разрядное частное.

Знак частного формируется также путем сложения знаковых разрядов делимого и делителя, как это делалось при умножении.

### 2.2.3. Арифметические операции над двоичными числами с плавающей точкой

В современных ЭВМ числа с плавающей точкой хранятся в памяти машин, имея мантиссу и порядок (характеристику) в прямом коде и нормализованном виде. Все арифметические действия над этими числами выполняются так же, как это делается с ними, если они представлены в полулогарифмической форме (мантисса и десятичный порядок), в десятичной системе счисления. Порядок и мантиссы обрабатываются отдельно.

**Сложение (вычитание).** Операция сложения (вычитания) производится в следующей последовательности.

1. Сравниваются порядки (характеристики) исходных чисел путем их вычитания  $\Delta p = p_1 - p_2$ . При выполнении этой операции определяется, одинаковый ли порядок имеют исходные слагаемые.
2. Если разность порядков равна нулю, то это значит, что одноименные разряды мантисс имеют одинаковые веса (двоичный порядок). В противном случае должно проводиться выравнивание порядков.
3. Для выравнивания порядков число с меньшим порядком сдвигается вправо на разницу порядков  $\Delta p$ . Младшие выталкиваемые разряды при этом теряются.
4. После выравнивания порядков мантиссы чисел можно складывать (вычитать) в зависимости от требуемой операции. Операция вычитания заменяется операцией сложения в соответствии с данными (см. табл. 2.2). Действия над слагаемыми производятся в обратном или дополнительном коде по общим правилам.
5. Порядок результата берется равным большему порядку.
6. Если мантисса результата не нормализована, то осуществляются нормализация и коррекция значений порядка.

*Пример 2.15*

Сложить два числа  $A_{10} = +1,375$ ;  $B_{10} = -0,625$ .



$$A_2 = +1,011 = 01011 \cdot 10^1; B_2 = -0,101 = -0101 \cdot 10^0.$$

В нормализованном виде эти числа будут иметь вид

Порядок	Мантисса
$[A_2]_n = 01$	01011
$[B_2]_n = 00$	1101

1. Вычитаем порядки  $\Delta p = 1 - 0 = 1$ . В машине эта операция требует операции сложения с преобразованием порядка чисел в дополнительный код

$$\begin{array}{r} p_1 = 01; \quad [p_1]_d = 01 \\ p_2 = 00; \quad + [p_2]_d = 00 \\ \hline \Delta p = 01. \end{array}$$

2. Порядок первого числа больше порядка второго числа на единицу. Требуется выравнивание порядков.

3. Для выравнивания порядков необходимо второе число сдвинуть вправо на один разряд. После сдвига

$$\begin{array}{l} [B'_2]_n = 01100101; \\ [M'_B]_d = 11011. \end{array}$$

4. Складываем мантиссы.

$$\begin{array}{r} [M_A]_d = 01011 \\ + [M'_B]_d = 11011 \\ \hline [M_C]_d = 00110. \end{array}$$

5. Порядок числа  $C$  равен порядку числа с большим порядком, т.е.  $p_C = +1$ .

$$[C_2]_n = 0100110.$$

Видно, что мантисса результата не нормализована, так как старшая цифра мантиссы равна нулю.

6. Нормализуем результат путем сдвига мантиссы на один разряд влево и соответственно вычитаем из значения порядка единицу

$$\begin{array}{l} [C_2]_n = 000110; \\ C_{10} = +0,75. \end{array}$$

**Умножение (деление).** Операция умножения (деления) чисел с плавающей точкой также требует разных действий над порядками и мантиссами. Алгоритмы этих операций выполняются в следующей последовательности.

1. При умножении (делении) порядки складываются (вычитаются) так, как это делается над числами с фиксированной точкой.

2. При умножении (делении) мантиссы перемножаются (делятся).

3. Знаки произведения (частного) формируются путем сложения знаковых разрядов сомножителей (делимого и делителя). Возможные переносы из знакового разряда игнорируются.

## 2.3. Логические основы ЭВМ

### 2.3.1. Основные сведения из алгебры логики

Алгебра логики, разработанная в середине IX в. ирландским математиком Д. Булем, является научной основой работы цифровых устройств. В ней действуют принципы (правила), схожие с обычной алгеброй, но буквами (символами) обозначаются не числа, а высказывания. В алгебре Буля переменные принимают только два дискретных значения: логическая «1» приписывается истинному высказыванию и логический «0» — ложному (неистинному). Символы нельзя рассматривать как арифметические числа, т.е. алгебра логики является алгеброй состояний, а не чисел. Аппарат алгебры логики используют как для анализа, так и проектирования (синтеза) логических устройств любой сложности в системах цифровой обработки информации. В этом случае можно проводить все исследования строго математически.

**Анализ логических устройств** проводят, рассматривая входные сигналы  $x_1, x_2, \dots, x_n$  в качестве аргументов и представляя соответствующие выходные сигналы логического устройства (ЛУ) в виде функции  $y_i$ , как показано на рис. 2.2, а.

В этом случае аналитическое соотношение

$$y_i = f_i(x_1, x_2, \dots, x_n)$$

устанавливает в явном виде соответствие между значением функции и всевозможными значениями комбинаций аргументов. Нетрудно заметить, что для  $n$  бинарных (принимающих только два значения) аргументов возможное число комбинаций  $B_n$  типа «0», «1»:

$$B_n = 2^n.$$

Например, при  $n = 2$  имеем четыре следующие комбинации из двух элементов: 00; 01; 10; 11. При  $n = 3$  имеем восемь комбинаций из трех элементов: 000; 001; 010; 100; 011; 101; 110; 111. Тогда для рассматриваемого устройства с одним бинарным вы-

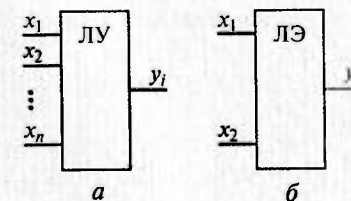


Рис. 2.2. Логическое устройство с  $n$  входами (а) и логический элемент с двумя входами (б)

Формы отображения основных логических операций  
(по ГОСТ 2.743—72)

Наименование формы	Вид логической операции																																						
	Дизъюнкция	Конъюнкция	Инверсия																																				
Символическая	$\vee ; +$	$\wedge ; \cdot$	$\bar{x}$																																				
Буквенная	«ИЛИ»	«И»	«НЕ»																																				
Условная																																							
Аналитическая	$y_д = x_1 \vee x_2$	$y_к = x_1 \wedge x_2$	$y_и = \bar{x}$																																				
Табличная (истинности)	<table border="1"> <tr><td><math>x_1</math></td><td><math>x_2</math></td><td><math>y_д</math></td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	$x_1$	$x_2$	$y_д$	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <tr><td><math>x_1</math></td><td><math>x_2</math></td><td><math>y_к</math></td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	$x_1$	$x_2$	$y_к$	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <tr><td><math>x</math></td><td><math>y_и</math></td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	$x$	$y_и$	0	1	1	0
$x_1$	$x_2$	$y_д$																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	1																																					
$x_1$	$x_2$	$y_к$																																					
0	0	0																																					
0	1	0																																					
1	0	0																																					
1	1	1																																					
$x$	$y_и$																																						
0	1																																						
1	0																																						
Контактная																																							
Электронная (схемотехническая)																																							

ходом (см. рис. 2.2, а) общее число различных логических функций (вариантов) составляет

$$i_{[n]} = 2^{2^n}.$$

На практике для упрощения процедуры анализа сложных ЛУ их представляют в виде комбинации простейших логических элементов (ЛЭ) (рис. 2.2, б) по аналогии с элементарными звеньями в структурных схемах автоматики. Как видно, для ЛЭ с двумя входными сигналами  $x_1$  и  $x_2$  и одним выходом  $y_i$

$$i_{[2]} = 2^{2^2} = 16. \quad (2.5)$$

В табл. 2.3 представлены все возможные варианты логических функций для случая (2.5).

В случае представления ЛУ в виде комбинации ЛЭ каждому элементу приписывают одну определенную операцию над входными комбинациями  $x_1, x_2, \dots, x_n$ , соответствующую, например, одному из столбцов (см. табл. 2.3) для простейшего случая функции двух переменных  $x_1, x_2$ .

**Основные логические операции** включают следующие элементарные преобразования двоичных сигналов.

1. Логическое сложение или дизъюнкция (от англ. *disjunction* — разъединение), обозначаемое символом  $\vee$  и называемое также операцией ИЛИ. Эта операция описывается для простейшей функции двух переменных  $x_1$  и  $x_2$  (см. рис. 2.2, б) в виде логической формулы

$$y_д = x_1 \vee x_2. \quad (2.6)$$

Соотношение (2.6) означает, что функция  $y_д$  равна «1», если хотя бы один из аргументов ( $x_1$  или  $x_2$ ) равен «1».

Условное обозначение, таблица истинности и другие показатели этой логической функции приведены в табл. 2.4.

Заметим, что таблицей истинности называют функциональную взаимосвязь значений выходной величины  $y_i$  логического

Таблица 2.3

Варианты функций  $y_i$  для простейшего логического элемента

Комбинации $x_2x_1$	$y_1$	$y_2$ (ИЛИ)	$y_3$	$y_4$	$y_5$	$y_6$	$y_7$	$y_8$	$y_9$	$y_{10}$	$y_{11}$	$y_{12}$	$y_{13}$	$y_{14}$	$y_{15}$ (И)	$y_{16}$
00	1	0	1	1	1	1	0	0	1	1	0	1	0	0	0	0
01	1	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0
10	1	1	1	0	1	0	1	1	0	1	0	0	0	1	0	0
11	1	1	1	1	0	0	0	1	1	0	1	0	0	0	1	0

устройства с каждой из возможных  $i$ -х комбинаций входных переменных  $x_1, x_2, \dots, x_n$ , представленных в табличной форме. Как отмечалось, для функции  $n$  переменных таких комбинаций будет  $B_n = 2^n$ . В простейшем случае двух переменных  $x_1$  и  $x_2$  таблица истинности будет насчитывать  $B_n = 2^2 = 4$  комбинации этих переменных.

Анализируя табл. 2.3 и 2.4 можно отметить, что столбец  $y_2$  соответствует операции логического сложения. Наиболее просто эту операцию можно реализовать с помощью контактной цепи с двумя параллельно включенными контактами. Сигнал  $y_д$  на выходе такой цепи появится только в том случае, если хотя бы один из контактов замкнут.

В цифровой электронике операцию логического сложения легко реализуют с помощью двух диодов (с независимыми входами),

работающих на одно нагрузочное устройство с сопротивлением  $R$ . Принципиальную схему такой электронной цепи см. в табл. 2.4. Как видно из рисунка, сигнал на выходе цепи, соответствующий логической «1», имеет место только в том случае, если на входе хотя бы одного из диодов также существует сигнал, соответствующий логической «1». Этот сигнал открывает диод, в результате чего в нагрузочном устройстве появляется ток, обеспечивающий соответствующее логической «1» выходное напряжение цепи.

2. Логическое умножение или конъюнкция (от англ. *conjunction* — соединение), обозначаемое символом  $\wedge$  и называемое операцией И. Условное обозначение & конъюнкции на логических схемах именуют *амперсандом*. Для удобства записи сложных логических функций символ конъюнкции можно условно отождествлять со знаком обычного умножения. Для функции двух переменных в этом случае имеем

$$y_k = x_1 \wedge x_2 = x_1 \cdot x_2. \quad (2.7)$$

Соотношение (2.7) показывает, что  $y_k = 1$  только в том случае, когда оба аргумента ( $x_1$  и  $x_2$ ) становятся равными «1».

Условное обозначение и другие показатели функции  $y_k$  представлены в третьем столбце (см. табл. 2.4). Вновь анализируя табл. 2.3, отметим, что столбец с  $y_{15}$  отвечает операции логического умножения. Эта операция может быть реализована контактной цепью, принципиальная схема которой приведена в табл. 2.4, а принципиальная схема электрической цепи, действие которой аналогично контактной, показана в следующей ячейке таблицы. Сигнал на выходе электронной цепи, равный примерно  $U_n$ , что соответствует  $y_k = 1$ , можно получить только в том случае, если оба диода заперты, т.е. на их катоды подан высокий потенциал (по отношению к аноду), соответствующий входным сигналам  $x_1 = 1$  и  $x_2 = 1$ .

3. Логическое отрицание или инверсия, обозначаемое черточкой над переменной и называемое операцией НЕ. Эта операция записывается

$$y_n = \bar{x}.$$

Как видно, операция выполняется над одной переменной  $x$  и значение  $y_n$  всегда противоположно значению этой переменной. Условное обозначение и другие показатели функции  $y_n$  приведены в четвертом столбце (см. табл. 2.4).

Реализация логической операции НЕ может быть также осуществлена контактной цепью, но (в отличие от цепей, рассмотренных ранее) с помощью нормально замкнутых контактов электромагнитного реле. Отсутствие напряжения на обмотке реле ( $x = 0$ ) предполагает замыкание цепи и появление сигнала на ее выходе,

соответствующего логической «1» ( $y_n = 1$ ). При наличии напряжения (логической «1») на обмотке реле ( $x = 1$ ) цепь разомкнута и сигнал на выходе цепи отсутствует ( $y_n = 0$ ).

Логическая операция инверсии сравнительно легко реализуется в электронике цепью простейшего усилителя при включении транзистора в схему с общим эмиттером (ОЭ), которая обладает инвертирующим свойством. Действительно, когда транзистор работает в режиме насыщения (при входном напряжении, соответствующем логической «1»), выходной сигнал  $y_n = U_{кэ} \cong 0$  ( $U_{кэ}$  — напряжение между коллектором и эмиттером транзистора). Когда же транзистор заперт (при отсутствии входного сигнала:  $x = 0$ ),  $y_n = U_{кэ} \cong U_n$ , что соответствует логической «1».

Сопоставляя таблицы истинности для операций дизъюнкции и конъюнкции (см. табл. 2.4), можно обосновать следующие соотношения алгебры Буля, имеющие большое практическое значение.

*Принцип дуальности*, который удобно выразить в виде двух положений:

$$\text{если } x_1 \vee x_2 = y_d, \text{ то } \bar{x}_1 \vee \bar{x}_2 = \bar{y}_d; \quad (2.8)$$

$$\text{если } x_1 \wedge x_2 = y_k, \text{ то } \bar{x}_1 \wedge \bar{x}_2 = \bar{y}_k. \quad (2.9)$$

*Правило де Моргана* вытекает как следствие принципа дуальности и формулируется в виде двух логических соотношений:

$$\begin{aligned} \overline{x_1 \vee x_2} &= \bar{x}_1 \wedge \bar{x}_2 = \overline{x_1 \cdot x_2}; \\ \overline{x_1 \cdot x_2} &= \bar{x}_1 \vee \bar{x}_2 = \overline{x_1 \vee x_2}. \end{aligned} \quad (2.10)$$

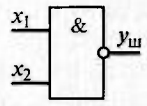
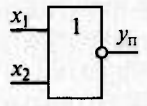
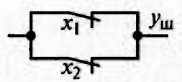
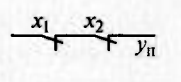
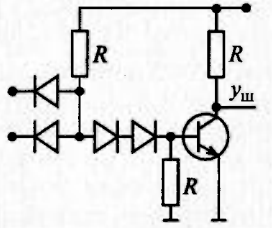
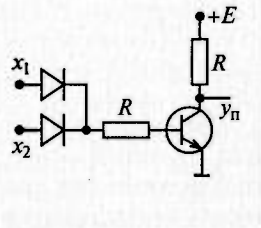
Приведенные соотношения (2.8) ... (2.10) можно легко обобщить для  $n$  входных сигналов  $x_1, x_2, \dots, x_n$ . Их широко используют для преобразования сложных логических функций к более простому виду (минимизации функций) при проектировании (синтезе) логических устройств цифровой электроники.

Рассмотренные выше основные логические операции И, ИЛИ и НЕ образуют функционально полный набор, т.е. позволяют реализовать любые логические функции (преобразования) комбинационной логики. Строго говоря, для этой цели можно ограничиться даже двумя операциями, например ИЛИ и НЕ. Однако при использовании только этих трех элементов не всегда удается получить логические устройства наипростейшего вида. Поэтому в логических системах находят применение и другие типовые элементы, реализующие иные логические операции (см. табл. 2.3).

Особое значение в цифровой микроэлектронике уделяется двум универсальным логическим операциям, каждая из которых способна самостоятельно образовать функционально полный набор. Как известно, в случае применения только одного базового эле-



Формы отображения универсальных логических операций  
(по ГОСТ 2.743—72)

Наименование формы	Вид логической операции																															
	Функция Шеффера	Функция Пирса																														
Символическая		↓																														
Буквенная	«И-НЕ»	«ИЛИ-НЕ»																														
Условная																																
Аналитическая	$y_{ш} = x_1   x_2$	$y_{п} = x_1 \downarrow x_2$																														
Табличная (истинности)	<table border="1" data-bbox="555 495 737 682"> <tr><td><math>x_1</math></td><td><math>x_2</math></td><td><math>y_{ш}</math></td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	$x_1$	$x_2$	$y_{ш}$	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1" data-bbox="850 495 1031 682"> <tr><td><math>x_1</math></td><td><math>x_2</math></td><td><math>y_{п}</math></td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	$x_1$	$x_2$	$y_{п}$	0	0	1	0	1	0	1	0	0	1	1	0
$x_1$	$x_2$	$y_{ш}$																														
0	0	1																														
0	1	1																														
1	0	1																														
1	1	0																														
$x_1$	$x_2$	$y_{п}$																														
0	0	1																														
0	1	0																														
1	0	0																														
1	1	0																														
Контактная																																
Электронная (схемотехническая)																																

мента наблюдается заметное усложнение проектируемых логических устройств. Однако в интегральной технологии удобство изготовления одного базового элемента имеет решающее значение. Поэтому универсальные логические элементы составляют основу большинства интегральных цифровых микросхем.

**Универсальные логические операции**, реализуемые базовыми элементами, включают две следующие разновидности (для удобства сохраним имеющийся порядок нумерации операций).

4. Функция Шеффера, обозначаемая символически вертикальной черточкой | (штрих Шеффера), отображает операцию И-НЕ. Этой операции соответствует столбец  $y_5$  (см. табл. 2.3). Для простейшей функции двух переменных  $x_1$  и  $x_2$  в этом случае получают

$$x_1 | x_2 = \overline{x_1 \wedge x_2} = \overline{x_1 \cdot x_2} = y_{ш}. \quad (2.11)$$

Формула (2.11) указывает на то, что функция  $y_{ш} = 0$  тогда и только тогда, когда  $x_1 = x_2 = 1$ .

5. Функция Пирса, обозначаемая символически вертикальной стрелкой (стрелка Пирса), выражает операцию ИЛИ-НЕ. Этой операции соответствует столбец  $y_{12}$  (см. табл. 2.3). Для функции двух переменных  $x_1$  и  $x_2$  она записывается в виде

$$x_1 \downarrow x_2 = \overline{x_1 \vee x_2} = y_{п}. \quad (2.12)$$

Соотношение (2.12) означает, что  $y_{п} = 1$  тогда и только тогда, когда  $x_1 = x_2 = 0$ .

Важнейшие показатели универсальных логических операций представлены в табл. 2.5.

Реализацию операций И-НЕ и ИЛИ-НЕ не представляет труда осуществить также в контактной цепи, применяя для этой цели электромагнитные реле с нормально замкнутыми (в отсутствие сигнала на входе управления реле, соответствующее отсутствию напряжения на его обмотке) контактами. Для реализации операции И-НЕ электромагнитные реле включают в цепь параллельно (см. табл. 2.5), а в случае операции ИЛИ-НЕ — последовательно.

### 2.3.2. Законы алгебры логики

Законы булевой алгебры отражают связи, существующие между операциями, выполняемыми над логическими переменными. Сформулируем основные из них.

- |   |  |
|---|--|
| 1. $x + 0 = x$ ;                                | $x \cdot 0 = 0$ .  |
| 2. $x + 1 = 1$ ;                                | $x \cdot 1 = x$ .  |
| 3. $x + x = x$ ;                                | $x \cdot x = x$ .  |
| 4. $x + \bar{x} = 1$ ;                          | $x \cdot \bar{x} = 0$ .  |
| 5. $\bar{\bar{x}} = x$ .                        |  |
| 6. $x + y = y + x$ ;                            | $x \cdot y = y \cdot x$ (коммутативный закон).                     |
| 7. $(x + y) + z = x + (y + z)$ ;                | $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ (ассоциативный закон). |
| 8. $\overline{x + y} = \bar{x} \cdot \bar{y}$ ; | $\overline{x \cdot y} = \bar{x} + \bar{y}$ (правило де Моргана).   |
| 9. $x + x \cdot y = x$ ;                        | $x(x + y) = x$ (закон поглощения).                                 |
| 10. $x + y \cdot z = (x + y)(x + z)$ ;          | $x(y + z) = xy + xz$ .   |
| 11. $x + \bar{x}y = x + y$ ;                    | $x(\bar{x} + y) = xy$ (закон свертки).                             |
| 12. $xy + x\bar{y} = x$ ;                       | $(x + y)(x + \bar{y}) = x$ (закон склеивания).                     |

Справедливость всех перечисленных законов может быть доказана подстановкой.

### 2.3.3. Понятие минимизации логических функций

**Представление функции алгебры логики (ФАЛ) математическим выражением.** При представлении ФАЛ математическим выражением используют два вида ее представления.

Дизъюнктивной нормальной формой (ДНФ) называется логическая сумма элементарных логических произведений, в каждое из которых аргумент или его отрицание входят один раз. ДНФ может быть получена из таблицы истинности следующим образом: для каждого набора аргументов, на котором функция равна «1», записывают элементарные произведения переменных, причем переменные, значение которых равно нулю, записывают с инверсией. Полученные произведения, которые носят название *конституента единицы*, или *минтермов*, суммируют.

#### Пример 2.16

Задана логическая функция 3-х переменных, которая равна единице в случае, если хотя бы две из входных переменных равны единице. Запишем эту функцию в виде таблицы истинности (табл. 2.6).

Таблица 2.6

Таблица истинности				
$x_2$	$x_1$	$x_0$	$y$	
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Для данной логической функции ДНФ имеет вид

$$y(x_2, x_1, x_0) = \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0.$$

ДНФ, полученная суммированием конституента единицы, называется совершенной (СДНФ).

Конъюнктивной нормальной формой (КНФ) называется логическое произведение элементарных сумм, в каждую

из которых аргумент или его отрицание входят один раз. КНФ может быть получена из таблицы истинности: для каждого набора аргументов, на котором функция равна «0», составляют элементарную сумму, причем переменные, значение которых равно «1», записываются с отрицанием. Полученные суммы, которые носят название конституента нуля или макстермов, объединяют операцией логического умножения.

#### Пример 2.17

Для функции из примера 2.16:

КНФ:

$$y(x_2, x_1, x_0) = (x_2 + x_1 + x_0)(x_2 + x_1 + \bar{x}_0)(x_2 + \bar{x}_1 + x_0)(\bar{x}_2 + x_1 + x_0).$$

Эта КНФ также называется совершенной (СКНФ), так как каждая элементарная сумма содержит все переменные.

Иногда удобнее пользоваться не самой ФАЛ, а ее инверсией. В этом случае при использовании вышеописанных методик для записи СДНФ надо использовать нулевые, а для записи СКНФ единичные значения функции.

#### Пример 2.18

Для функции из примера 2.16:

$$\text{СДНФ: } \bar{y}(x) = \bar{x}_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 \bar{x}_1 x_0 + \bar{x}_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 \bar{x}_0;$$

$$\text{СКНФ: } \bar{y}(x) = (x_2 + \bar{x}_1 + \bar{x}_0)(\bar{x}_2 + x_1 + \bar{x}_0)(\bar{x}_2 + \bar{x}_1 + x_0)(\bar{x}_2 + \bar{x}_1 + \bar{x}_0).$$

Иногда для сокращения записи ФАЛ представляют последовательностью десятичных чисел. Для представления ФАЛ последовательностью чисел задают десятичные значения конституента единицы или нуля.

#### Пример 2.19

Для функции из примера 2.16 можно записать:

$$\text{СДНФ: } y(x_2, x_1, x_0) = \sum(3, 5, 6, 7) = \vee(3, 5, 6, 7);$$

$$\text{СКНФ: } y(x_2, x_1, x_0) = \Pi(0, 1, 2, 4) = \wedge(0, 1, 2, 4) = \&(0, 1, 2, 4).$$

**Минимизация ФАЛ.** Ранее было показано, что логическую схему, реализующую заданный алгоритм обработки сигналов, можно синтезировать непосредственно по выражению, представленному в виде СДНФ или СКНФ. Однако полученная таким образом схема, как правило, не оптимальна с точки зрения ее практической реализации. Поэтому исходные выражения минимизируют.

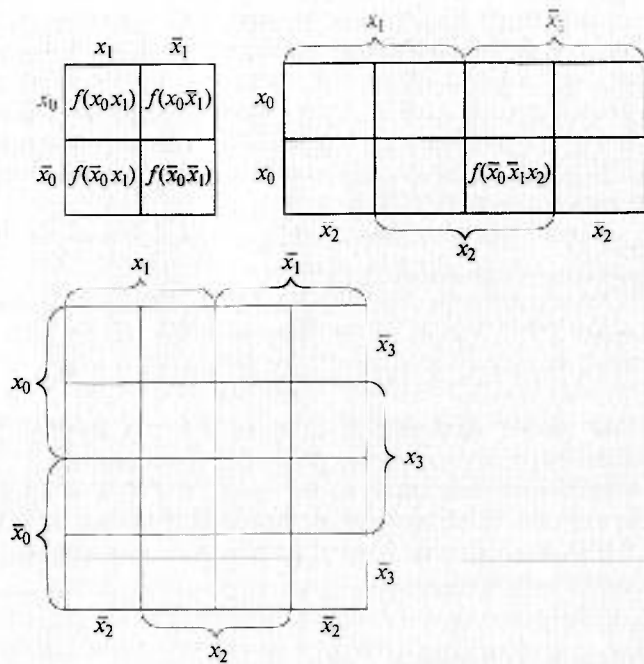


Рис. 2.3. Вид карты Вейча (Карно) для функций 2, 3 и 4 переменных

Целью минимизации логической функции является минимизация стоимости ее технической реализации. Критерии, в соответствии с которыми осуществляют минимизацию ФАЛ, не однозначны. Основным критерием минимизации ФАЛ является критерий уменьшения количества элементарных логических элементов при использовании только однородных элементов, например, только типа И-НЕ (ИЛИ-НЕ).

Метод минимизации ФАЛ с использованием карт Вейча (Карно) базируется на табличном виде представления ФАЛ. Он широко используется для ручной, без применения ЭВМ минимизации ФАЛ, число переменных в которых не превышает 4...5.

Картой Вейча (Карно) называется таблица, число клеток которой для функции  $n$  переменных равно  $2^n$ , причем каждому минтерму соответствует своя клетка карты (рис. 2.3).

Из приведенных рисунков видно, что минтерм представляется минимальным участком площади — одной клеткой на картах Карно (картах минтермов).

Макстерм занимает максимальную площадь на картах Карно (рис. 2.4) и охватывает все квадраты, кроме одного  $x_0 + \bar{x}_1$ .

Если требуется представить на карте Вейча (Карно) логическую функцию, заданную в виде СДНФ, в соответствующие клетки заносятся единицы. Остальные клетки остаются незаполнен-

ными или заполняются нулями. Логическая функция на карте Вейча (Карно) представляется совокупностью клеток, заполненных единицами, а инверсия функции представляется совокупностью пустых или заполненных нулями клеток.

При минимизации ФАЛ используются либо ее единичные, либо нулевые значения. В обоих случаях получаются равносильные выражения, которые, однако, могут отличаться по числу членов и выполненным логическим операциям.

Алгоритм минимизации сводится к следующему:

1. На карте Вейча выделяют прямоугольные области, объединяющие выбранные значения функции («1» или «0»). Причем каждая область должна содержать  $2^k$  клеток, где  $k$  может принимать значения 0; 1; 2; 4, т.е. 1; 2; 4; 8; 16 и т.д. клеток. Выделенные области могут пересекаться, т.е. одна клетка может входить в несколько разных областей.

2. Каждая из выделенных областей является самостоятельным произведением переменных, значения которых в рамках выделенной области остаются постоянными. Каждое произведение содержит  $n - k$  переменных и носит название импликанты, где  $n$  число аргументов исходной логической функции.

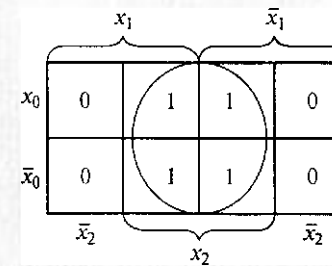
3. Из полученного множества выделенных областей выбирают минимальное число максимально больших областей, включающих все клетки с выбранным значением ФАЛ. Сумма полученных произведений образует минимальную ДНФ.

При объединении клеток с нулевыми значениями ФАЛ получим ФАЛ, инверсную заданной.

**Пример 2.20**

Минимизируем ФАЛ:

$$y = x_2x_1x_0 + x_2\bar{x}_1x_0 + x_2x_1\bar{x}_0 + x_2\bar{x}_1\bar{x}_0;$$



$$y = x_2;$$

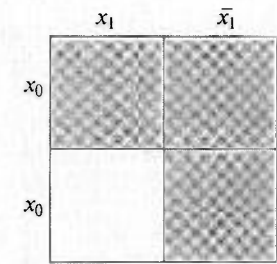
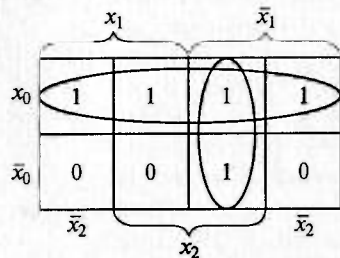


Рис. 2.4. Вид макстерма на карте Карно



$$y = \bar{x}_2 x_1 x_0 + x_2 x_1 x_0 + x_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 \bar{x}_1 x_0;$$

объединяем по «1»



$$y = x_0 + x_2 \bar{x}_1;$$

объединяем по «0»

$$\bar{y} = x_1 \bar{x}_0 + \bar{x}_2 \bar{x}_0$$

Применяя теорему де Моргана,

$$\bar{\bar{y}} = \overline{x_1 \bar{x}_0 + \bar{x}_2 \bar{x}_0} = \overline{x_1 \bar{x}_0} \cdot \overline{\bar{x}_2 \bar{x}_0} = (\bar{x}_1 + x_0)(x_2 + x_0).$$

Из последнего примера видно, что объединение областей по единичным и нулевым значениям функции может привести хотя и к равносильным, но различным минимальным выражениям. Следовательно, может отличаться и схема, реализующая заданный алгоритм. Поэтому для получения минимально простой технической реализации желательно проводить минимизацию с использованием как нулевых, так и единичных значений функции и из полученных минимальных форм выбрать простейшую.

Следует также отметить, что, поскольку карты Вейча (Карно) функций 3 и 4-х переменных представляют собой объемные фигуры, при формировании областей необходимо помнить, что в одну область объединяются крайние столбцы карты 3-х переменных, а также четыре угловых клетки карты 4-х переменных (рис. 2.5).

**Минимизация недоопределенной ФАЛ.** Недоопределенной называется ФАЛ, часть значений которой не задана, т. е. может быть

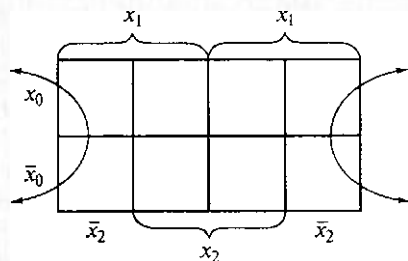


Рис. 2.5. Объединение крайних областей в единую область

произвольной. При различном доопределении ФАЛ могут быть получены и различные минимальные формы. Поэтому доопределение исходной функции должно выполняться таким образом, чтобы в его результате было выделено минимальное число максимально больших областей, что приводит к получению наиболее простой технической реализации заданного алгоритма.

*Пример 2.21*

ФАЛ задана таблицей истинности (табл. 2.7).

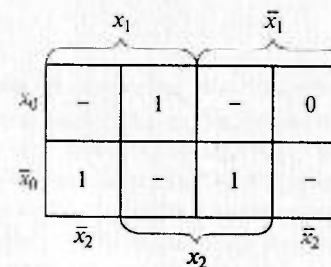
Таблица 2.7

Таблица истинности ФАЛ

$x_2$	$x_1$	$x_0$	$y$
0	0	0	—
0	0	1	0
0	1	0	1
0	1	1	—
1	0	0	1
1	0	1	—
1	1	0	—
1	1	1	1

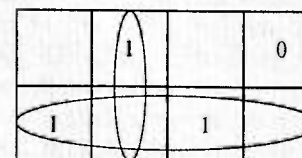
Для минимизации по «1» исходное выражение

$$y = \bar{x}_0 x_1 \bar{x}_2 + \bar{x}_0 \bar{x}_1 x_2 + x_0 x_1 x_2.$$

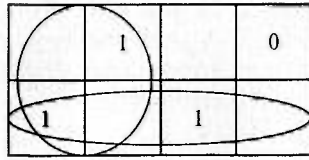


Доопределяем

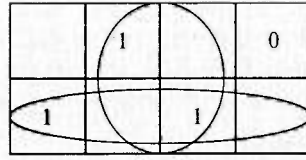
$$y = \bar{x}_0 + x_1 x_2;$$



$$y = \bar{x}_0 + x_1;$$

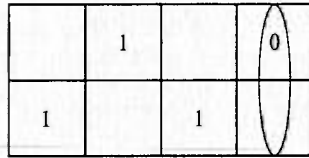


$$y = \bar{x}_0 + x_2.$$

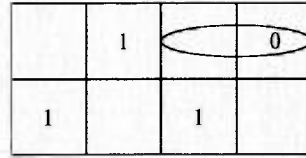


Для минимизации по «0» исходное выражение

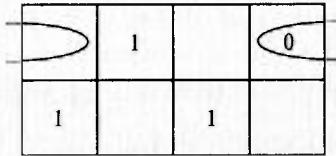
$$\bar{y} = x_0 x_1 \bar{x}_2 \Rightarrow y = \bar{x}_0 + x_1 + x_2;$$



$$\bar{y} = \bar{x}_1 \bar{x}_2 \Rightarrow y = x_1 + x_2;$$



$$\bar{y} = x_0 \bar{x}_1 \Rightarrow y = \bar{x}_0 + x_1;$$



$$\bar{y} = x_0 \bar{x}_2 \Rightarrow y = \bar{x}_0 + x_2.$$

**Синтез логических устройств в заданном базисе логических элементов.** До сих пор для построения структуры логических устройств мы пользовались функционально полной системой логических элементов, реализующих три основные логические операции И, ИЛИ, НЕ. Однако на практике в целях уменьшения номенклатуры используемых микросхем часто пользуются функционально полной системой логических элементов в составе двух, выполняющих операции И-НЕ, ИЛИ-НЕ. Любую ФАЛ можно записать в заданном базисе логических элементов. Если задан базис И-НЕ, то путем двойного инвертирования исходного выражения или его части и применения теорем де Моргана ФАЛ приводится к виду, содержащему только операции логического умножения и инвертирования. Если же задан базис ИЛИ-НЕ, исходную ФАЛ теми же приемами приводят к виду, содержащему только операции логического сложения и инверсии. Далее логическое выражение записывается через условные обозначения выбранных операций.

**Пример 2.22**

Задана минимальная ДНФ, приведем ее в базис И-НЕ:

$$y = x_1 x_3 + \bar{x}_0 x_2 \bar{x}_3 + x_0 \bar{x}_1 \bar{x}_2 = x_1 x_3 + \bar{x}_0 x_2 \bar{x}_3 + x_0 \bar{x}_1 \bar{x}_2 = (x_1 x_3) (\bar{x}_0 x_2 \bar{x}_3) (x_0 \bar{x}_1 \bar{x}_2) = (x_1 | x_3) | (\bar{x}_0 | x_2 | \bar{x}_3) | (x_0 | \bar{x}_1 | \bar{x}_2).$$

**Пример 2.23**

Задана минимальная КНФ, приведем ее в базис ИЛИ-НЕ:

$$y = (x_0 + x_3) (\bar{x}_0 + x_1 + x_2) (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) = (x_0 + x_3) + (\bar{x}_0 + x_1 + x_2) + (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) = (x_0 \downarrow x_3) \downarrow (\bar{x}_0 \downarrow x_1 \downarrow x_2) \downarrow (\bar{x}_1 \downarrow \bar{x}_2 \downarrow \bar{x}_3).$$

**Пример 2.24**

Задана логическая функция, приведем ее к базису И-НЕ:

$$y = x_0 x_3 + (\bar{x}_0 + x_2 + \bar{x}_3) (x_1 + \bar{x}_2);$$

$$y = y_1 + \bar{y}_2 y_3 = y_1 + \bar{\bar{y}_2} y_3 = y_1 + (\bar{y}_2 | y_3) = y_1 + (\bar{\bar{y}_2} | y_3) = \bar{y}_1 (\bar{y}_2 | y_3) = \bar{y}_1 | (\bar{y}_2 | y_3);$$

$$\bar{y}_1 = x_0 x_3 = x_0 | x_3;$$

$$\bar{y}_2 = \bar{x}_0 + x_2 + \bar{x}_3 = x_0 \bar{x}_2 x_3 = x_0 \bar{x}_2 x_3 = x_0 | \bar{x}_2 | x_3;$$

$$y_3 = x_1 + \bar{x}_2 = x_1 + \bar{x}_2 = \bar{x}_1 x_2 = \bar{x}_1 | x_2;$$

$$y = (x_0 | x_3) | ((x_0 | \bar{x}_2 | x_3) | (\bar{x}_1 | x_2)).$$

**Пример 2.25**

Приведем к базису ИЛИ-НЕ функцию из примера 2.24:

$$y = y_1 + \bar{y}_2 y_3 = y_1 + \bar{\bar{y}_2} y_3 = y_1 + (\bar{y}_2 | y_3) = y_1 + (\bar{\bar{y}_2} | y_3) = y_1 \downarrow (\bar{y}_2 \downarrow y_3);$$

$$y_1 = x_0 x_3 = x_0 \bar{x}_3 = \bar{x}_0 + \bar{x}_3 = \bar{x}_0 \downarrow \bar{x}_3;$$

$$y_2 = \bar{x}_0 + x_2 + \bar{x}_3 = \bar{x}_0 + x_2 + \bar{x}_3 = \bar{x}_0 \downarrow x_2 \downarrow \bar{x}_3;$$

$$y_3 = x_1 + \bar{x}_2 = x_1 + \bar{x}_2 = x_1 \downarrow \bar{x}_2 \Rightarrow \bar{y}_3 = x_1 \downarrow \bar{x}_2;$$

$$y = (\bar{x}_0 \downarrow \bar{x}_3) \downarrow ((\bar{x}_0 \downarrow x_2 \downarrow \bar{x}_3) \downarrow (x_1 \downarrow \bar{x}_2)).$$

### 2.3.4. Техническая реализация логических функций

Для построения логической схемы, реализующей логическую функцию, необходимо логические элементы, предназначенные для выполнения логических операций, указанных в ФАЛ, располагать начиная от входа в порядке, указанном в булевом выражении.

#### Пример 2.26

Построить структуру логического устройства, реализующего ФАЛ:

$$F = \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0.$$

Логическая схема представлена на рис. 2.6.

Логические схемы, иллюстрирующие пример 2.24 представлены ниже (рис. 2.7, 2.8, 2.9).

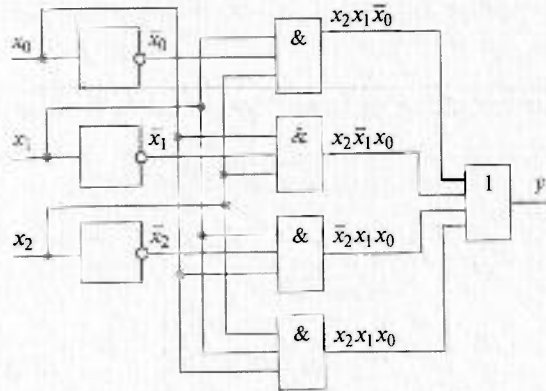


Рис. 2.6. Пример технической реализации логической функции

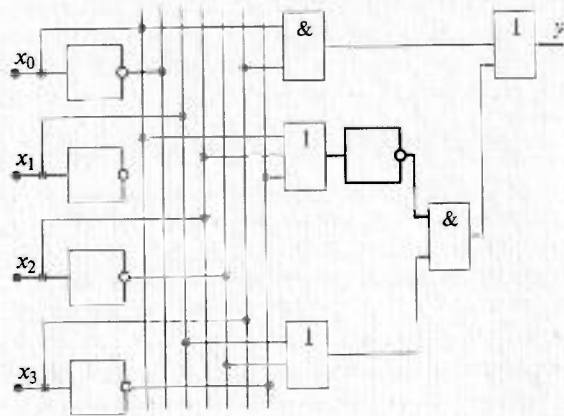


Рис. 2.7. Схема, соответствующая исходному выражению

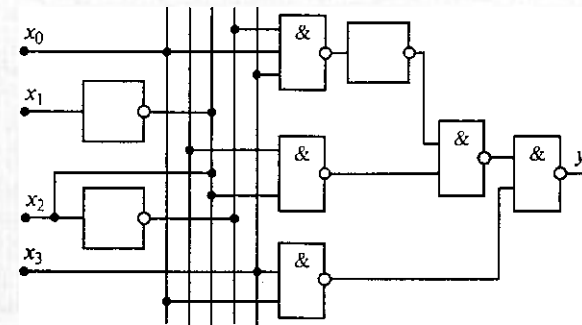


Рис. 2.8. Схема в базисе И-НЕ

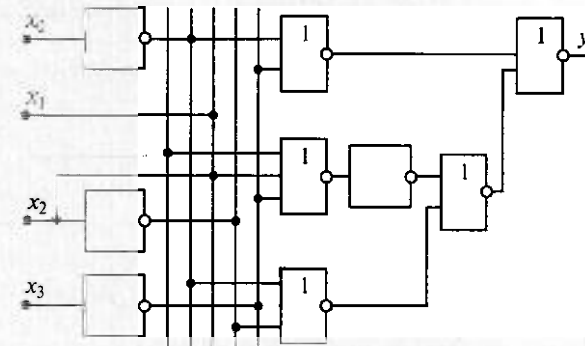


Рис. 2.9. Схема в базисе ИЛИ-НЕ

#### Контрольные вопросы

1. Что представляет собой логическое устройство?
2. Как получить число возможных бинарных комбинаций типа «0» и «1» для логического устройства с числом аргументов  $n = 3$  и  $n = 6$ ?
3. Как получить для логического устройства с одним бинарным выходом общее число бинарных функций (вариантов), если число входных сигналов  $n = 2$ ,  $n = 3$ ?
4. Запишите в двоичном коде следующие числа: 7, 9, 15 и 35. Поясните запись.
5. Сформулируйте правило перевода числа из десятичной системы счисления в двоичную (методом деления).
6. Изложите особенности восьмеричной системы счисления, записав число 152 в этой системе.
7. Покажите на примере числа 79, как перевести его в двоичную систему счисления.
8. Изложите особенности шестнадцатеричной системы счисления, записав число 378 в этой системе.
9. Поясните и сформулируйте количественные меры оценки информации: бит и байт.



10. Изложите особенности логической операции ИЛИ.
11. Сформулируйте особенности логической операции И.
12. Приведите основные показатели логической операции НЕ.
13. Запишите в логической форме функцию  $y = x_1x_2 + x_3x_4$ .
14. Что представляет собой таблица истинности?
15. Изложите особенности логической операции ИЛИ-НЕ.
16. Приведите основные показатели логической операции И-НЕ.
17. Построить логическую схему функции, описываемой выражением

$$y = x_0x_3 + (\bar{x}_0 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2).$$

18. Постройте логическую схему функции, описываемой выражением

$$y = (x_0 | x_3) | ((\bar{x}_0 | \bar{x}_2 | x_3) | (\bar{x}_1 | x_2)).$$

19. Постройте логическую схему функции, описываемой выражением

$$y = (\bar{x}_0 \downarrow \bar{x}_3) \downarrow ((\bar{x}_0 \downarrow x_2 \downarrow \bar{x}_3) \downarrow (x_1 \downarrow \bar{x}_2)).$$

20. Докажите тождество

$$x_0 | x_1 | x_2 = x_0 | (\bar{x}_1 | x_2).$$

21. Докажите тождество

$$x_0 \downarrow x_1 \downarrow x_2 = x_0 \downarrow (\bar{x}_1 \downarrow x_2).$$

22. Приведите логическую функцию к базису 2ИЛИ-НЕ

$$y = (x_1 + x_2)(\bar{x}_1 + x_3)(\bar{x}_2 + \bar{x}_3).$$

## Глава 3

# ТИПОВЫЕ ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ И УСТРОЙСТВА ЭВМ

### 3.1. Классификация элементов и устройств ЭВМ

В структуре ЭВМ выделяют следующие структурные единицы: элементы, узлы, блоки и устройства. Элементы предназначаются для обработки единичных электрических сигналов, соответствующих битам информации. Узлы обеспечивают одновременную обработку группы сигналов — информационных слов. Блоки реализуют некоторую последовательность в обработке информационных слов. Устройства предназначаются для выполнения отдельных машинных операций и их последовательностей.

Различают два типа цифровых устройств: комбинационные (однотактные) и последовательностные (многотактные), которые часто называют конечными автоматами.

*Комбинационными* называют цифровые устройства, в которых значения выходных сигналов определяются заданным в данный момент времени сочетанием входных воздействий. Нетрудно заметить, что в комбинационных логических устройствах отсутствуют запоминающие элементы.

*Последовательностными* называют устройства, в которых выходные сигналы зависят не только от входных воздействий в заданный момент времени, но и от их предыдущих значений. Поэтому последовательностные устройства в отличие от комбинационных содержат запоминающие элементы. Исследования последовательностных устройств — задача, существенно более сложная, чем комбинационных.

### 3.2. Типовые функциональные узлы комбинационных логических устройств

**Мультиплексоры и демultipлексоры.** *Мультиплексором* называется комбинационное логическое устройство, предназначенное для управляемой передачи данных от нескольких источников ин-

формации в один выходной канал. Согласно определению, мультиплексор должен иметь один выход и две группы входов: информационные и адресные. Код, подаваемый на адресные входы, определяет, какой из информационных входов в данный момент подключен к выходному выводу. Поскольку  $n$ -разрядный двоичный код может принимать  $2^n$  значений, то, если число адресных входов мультиплексора равно  $n$ , число его информационных входов должно равняться  $2^n$ .

Таблица истинности, отображающая работу мультиплексора с двумя адресными входами, имеет следующий вид (табл. 3.1).

В данной таблице учтено, что мультиплексор обычно снабжается дополнительным инверсным выходом  $\bar{Q}$  и входом разрешения работы  $E$ .

Если на вход разрешения работы  $E$  подан активный логический сигнал ( $E = 1$ ), то выходной сигнал мультиплексора постоянен и не зависит от его входных сигналов.

Функция алгебры логики, описывающая работу мультиплексора, имеет вид

$$Q = D_0 \bar{A}_1 \bar{A}_0 \bar{E} + D_1 \bar{A}_1 A_0 \bar{E} + D_2 A_1 \bar{A}_0 \bar{E}.$$

Логическая схема мультиплексора, соответствующая этой ФАЛ, и условное графическое обозначение мультиплексора на примере ИС типа 555КП7 показаны на рис. 3.1.

Число информационных входов реально выпускаемых промышленностью микросхем мультиплексоров не превышает трех. Поэтому в случае необходимости иметь большее число входов из имеющихся микросхем строят структуру так называемого мультиплексорного дерева на примере устройства с 16 информационными входами, построенного на основе четырехходовых мультиплексоров. Его логическая схема приведена на рис. 3.2. Устройство содержит мультиплексоры первого и второго уровня. Мультиплексоры первого уровня управляются младшими разрядами адресного слова, мультиплексоры второго уровня — старшими разрядами адресного слова.

Таблица 3.1

Таблица истинности, описывающая работу мультиплексора

$E$	$A_1$	$A_0$	$Q$	$\bar{Q}$
1	X	X	0	1
0	0	0	$D_0$	$\bar{D}_0$
0	0	1	$D_1$	$\bar{D}_1$
0	1	0	$D_2$	$\bar{D}_2$
0	1	1	$D_3$	$\bar{D}_3$

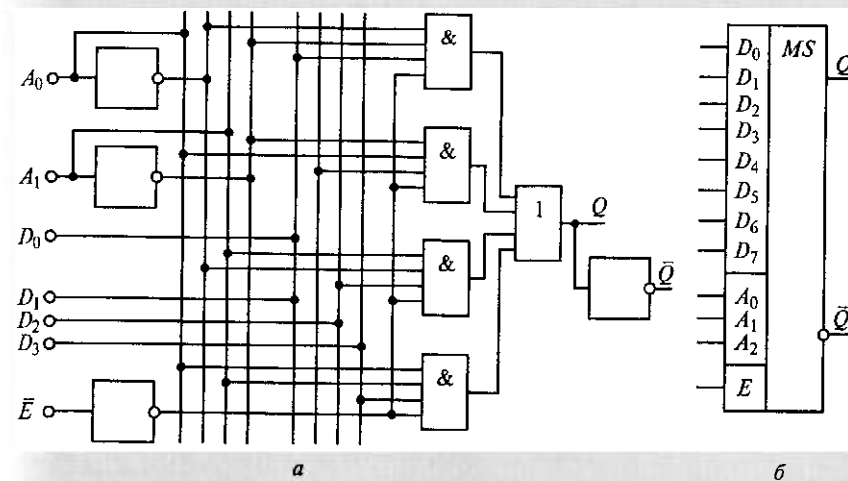


Рис. 3.1. Логическая схема мультиплексора (а) и его условное графическое изображение (б)

Рассмотрим работу схемы на конкретном примере.

#### Пример 3.1

Допустим, задано адресное слово 0110. Входы  $D_2$  младшими разрядами адресного слова 10 подключаются к выходам мультиплексоров первого уровня. На информационных входах мультиплексора второго уровня появятся соответственно сигналы  $x_2$ ,  $x_6$ , и  $x_{10}$ . Из этих сигналов мультиплексор второго уровня по старшим разрядам адресного слова 01 выберет сигнал, присутствующий на его входе  $D_1$ . В результате на выходе мультиплексорного дерева появится входной сигнал  $x_1$ , что и соответствует заданному адресу.

По описанному алгоритму можно строить устройства с любым необходимым числом входов. При этом следует помнить, что мультиплексоры одного уровня должны иметь одинаковое число информационных входов. Число информационных входов мультиплексоров разных уровней может быть различным, что определяется суммарным числом информационных входов и типом используемой элементной базы.

При передаче информации от нескольких источников по общему каналу с разделением по времени нужны не только мультиплексоры, но и устройства обратного назначения, распределяющие информацию, полученную из одного канала между несколькими приемниками. Эту задачу решают демultipлексоры.

Демultipлексором называется комбинационное логическое устройство, предназначенное для управляемой передачи данных от одного источника информации в несколько выходных каналов.

Согласно данному определению, демultipлексор в общем случае имеет один информационный вход,  $n$  адресных входов и  $2^n$

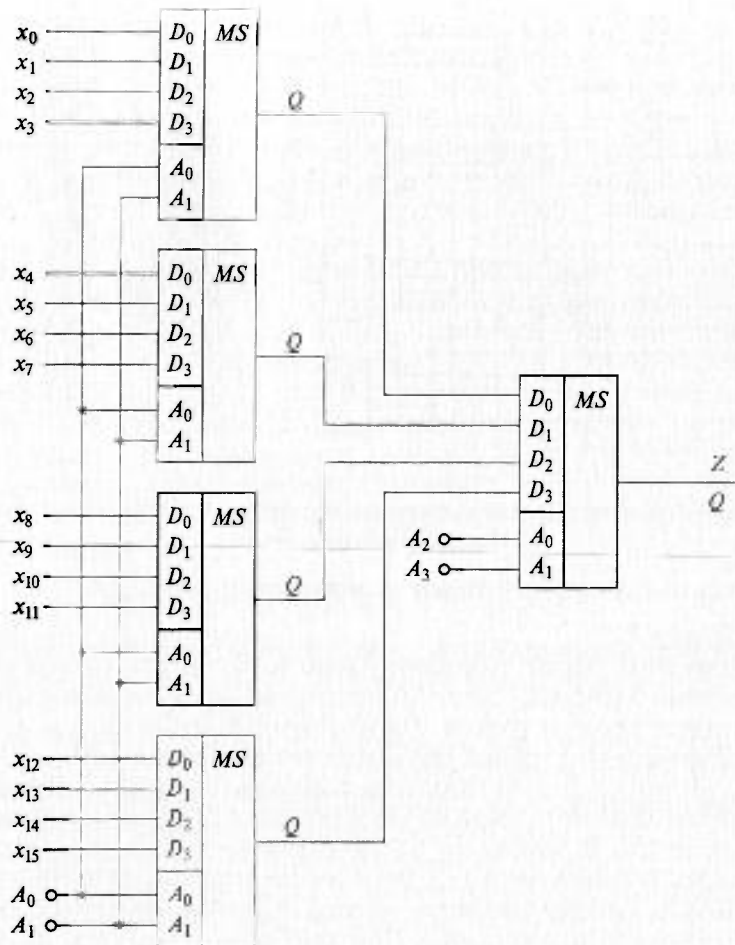


Рис. 3.2. Схема мультиплексорного дерева

выходов. Таблица истинности, описывающая работу демультиплексора, снабженного двумя адресными входами и входом разрешения работы  $E$ , имеет вид (табл. 3.2).

Таблица 3.2

Таблица истинности, описывающая работу демультиплексора

$E$	$A_1$	$A_0$	$Q_0$	$Q_1$	$Q_2$	$Q_3$
1	X	X	0	0	0	0
0	0	0	$D$	0	0	0
0	0	1	0	$D$	0	0
0	1	0	0	0	$D$	0
0	1	1	0	0	0	$D$

Данной таблице соответствует следующая система ФАЛ:

$$\begin{aligned}
 Q_0 &= D\bar{A}_1\bar{A}_0\bar{E} = \bar{D} \downarrow A_1 \downarrow A_0 \downarrow E; \\
 Q_1 &= D\bar{A}_1A_0\bar{E} = \bar{D} \downarrow A_1 \downarrow \bar{A}_0 \downarrow E; \\
 Q_2 &= DA_1\bar{A}_0\bar{E} = \bar{D} \downarrow \bar{A}_1 \downarrow A_0 \downarrow E; \\
 Q_3 &= DA_1A_0\bar{E} = \bar{D} \downarrow \bar{A}_1 \downarrow \bar{A}_0 \downarrow E.
 \end{aligned}
 \tag{3.1}$$

На рис. 3.3, а приведена логическая схема демультиплексора, удовлетворяющая системе ФАЛ (3.1), а на рис. 3.3, б показано его условное графическое изображение.

При необходимости увеличить число выходных выводов на основе рассмотренной схемы можно построить структуру демультиплексорного дерева. Его структура с точностью до зеркального отображения аналогична структуре мультиплексорного дерева (см. рис. 3.2). При этом демультиплексор первого уровня также управляется младшими разрядами адресного слова, а демультиплексоры второго уровня — его старшими разрядами (рис. 3.4).

Следует отметить, что для построения демультиплексорного дерева входящие в его состав демультиплексоры должны быть обязательно снабжены входами разрешения работы (стробирования).

**Преобразователи кодов.** В цифровой технике применяются различные виды кодирования информации. Так, при выполнении операций в ЭВМ обычно применяют несколько разновидностей двоичного кода (прямой, обратный, дополнительный, двоично-десятичный и т. д.). При передаче информации по линиям связи удобнее использовать другие виды кодов, позволяющие, напри-

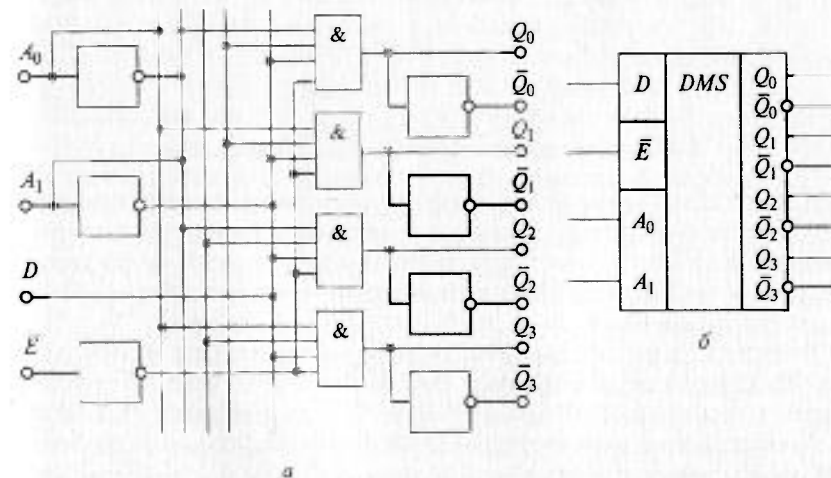


Рис. 3.3. Логическая схема демультиплексора (а) и его условное графическое изображение (б)



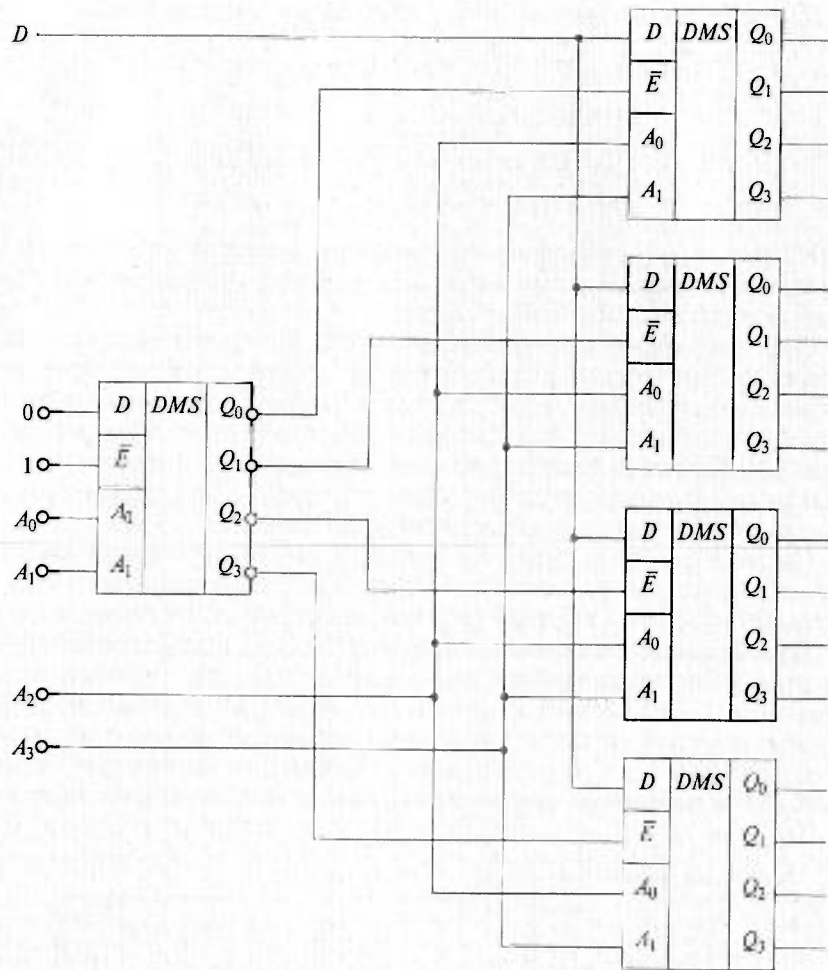


Рис. 3.4. Логическая схема демультимплексорного дерева

мер, уменьшать вероятность появления ошибки или даже исправлять ее в дальнейшем. Примерами являются коды, построенные по принципу 2 из 5 (в которых из пяти символов два всегда имеют единичное значение), коды с проверкой четности или нечетности, коды Хемминга.

В связи с этим всегда стоит задача преобразования информации из одного кода в другой. Эту задачу на аппаратном уровне решают комбинационные устройства — преобразователи кодов.

*Преобразователем кода* называется комбинационное устройство, предназначенное для изменения вида кодирования информации. Как и всякое комбинационное устройство, преобразователь кода характеризуется таблицей истинности, ставящей в соответствие

кодам, подаваемым на вход, коды, снимаемые с выхода устройства. Следует заметить, что в этой таблице в общем случае число разрядов входного и выходного кодов может не совпадать. Главное — она должна давать однозначное соответствие различных кодов. Данная таблица является основанием для синтеза логической структуры конкретного преобразователя кодов. Условное графическое изображение преобразователей кодов на принципиальных электрических схемах приведено на рис. 3.5. В качестве примера преобразователя кодов, выпускаемых в виде ИС, можно привести схемы, обеспечивающие преобразование информации из двоичного в двоично-десятичный код. Частным случаем преобразователей кода являются шифраторы и дешифраторы.

**Шифраторы и дешифраторы.** *Шифратором*, или *кодером* называется комбинационное логическое устройство для преобразования чисел из десятичной системы счисления в двоичную. Входам шифратора последовательно присваиваются значения десятичных чисел, поэтому подача активного логического сигнала на один из входов воспринимается шифратором как подача соответствующего десятичного числа. Этот сигнал преобразуется на выходе шифратора в двоичный код. Согласно сказанному, если шифратор имеет  $n$  выходов, число его входов должно быть не более чем  $2^n$ . Шифратор, имеющий  $2^n$  входов и  $n$  выходов, называется *полным*. Если число входов шифратора меньше  $2^n$ , он называется *неполным*.

Рассмотрим работу шифратора на примере преобразователя десятичных чисел от 0 до 9 в двоично-десятичный код. Таблица истинности, соответствующая этому случаю, имеет вид (табл. 3.3).

Так как число входов данного устройства меньше  $2^n = 16$ , имеем неполный шифратор. Используя таблицу для  $Q_3$ ,  $Q_2$ ,  $Q_1$  и  $Q_0$ , можно записать следующие выражения:

$$\begin{aligned} Q_3 &= x_8 + x_9; \\ Q_2 &= x_4 + x_5 + x_6 + x_7; \\ Q_1 &= x_2 + x_3 + x_6 + x_7; \\ Q_0 &= x_1 + x_3 + x_5 + x_7 + x_9. \end{aligned} \quad (3.2)$$

Полученная система ФАЛ характеризует работу шифратора. Логическая схема устройства, соответствующая системе (3.2), приведена на рис. 3.6.

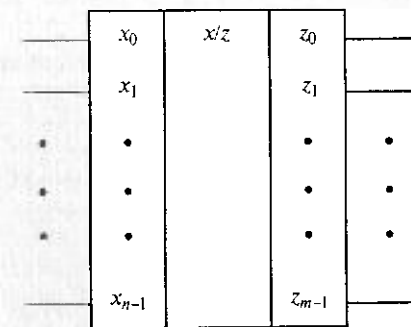


Рис. 3.5. Условное графическое изображение преобразователя кодов

Таблица 3.3

Таблица истинности, описывающая работу шифратора

$x_9$	$x_8$	$x_7$	$x_6$	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

Нетрудно заметить, что в шифраторе рассматриваемого типа сигнал, подаваемый на вход  $x_0$ , не используется. Поэтому отсутствие сигнала на любом из входов  $x_0, \dots, x_9$  трактуется схемой как наличие на входе нулевого сигнала.

Основное применение шифратора в цифровых системах — это введение первичной информации с клавиатуры. При нажатии

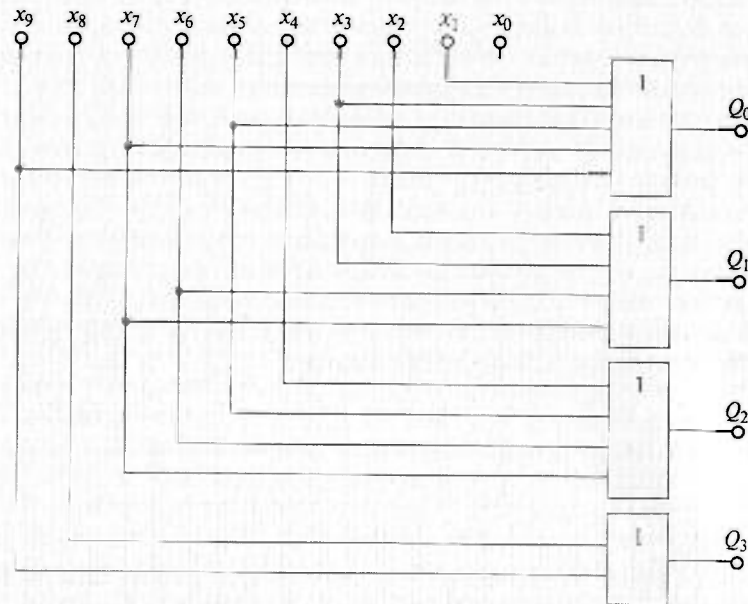


Рис. 3.6. Логическая схема шифратора десятичных чисел

любой клавиши на соответствующий вход шифратора подается сигнал логической «1», который и преобразуется на выходе в двоично-десятичный код. Вариант устройства ввода информации показан на рис. 3.7.

Дешифратором, или декодером называется комбинационное логическое устройство для преобразования чисел из двоичной системы счисления в десятичную. Согласно определению дешифратор относится к классу преобразователей кодов. Здесь также понимается, что каждому входному двоичному числу ставится в соответствие сигнал, формируемый на определенном выходе устройства. Таким образом, дешифратор выполняет операцию, обратную шифратору. Если число адресных входов дешифратора  $n$  связано с числом его выходов  $m$  соотношением  $m = 2^n$ , то дешифратор называют *полным*, если  $m < 2^n$ , дешифратор называют *неполным*.

Поведение дешифратора описывается таблицей истинности, аналогичной таблице истинности шифратора (см. табл. 3.3), только в ней входные и выходные сигналы меняются местами. В соответствии с данной таблицей, так как выходной сигнал равен «1» только на одном единственном наборе входных переменных, т. е. для одной конstituенты единицы, алгоритм работы дешифратора описывается следующей системой уравнений вида:

$$\begin{aligned}
 x_0 &= \bar{Q}_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0; \\
 x_1 &= \bar{Q}_3 \bar{Q}_2 \bar{Q}_1 Q_0; \\
 x_2 &= \bar{Q}_3 \bar{Q}_2 Q_1 \bar{Q}_0; \\
 &\dots
 \end{aligned}$$

и так далее, где  $Q_i$  — значение логической переменной на  $i$ -м выходе устройства.

**Цифровые компараторы.** Цифровым компаратором называется комбинационное логическое устройство, предназначенное для сравнения чисел, представленных в виде двоичных кодов.

Число входов компаратора определяется разрядностью сравниваемых кодов. На выходе компаратора обычно формируется три сигнала:

- $F_e$  — равенство кодов;
- $F_>$  — числовой эквивалент первого кода больше второго;
- $F_<$  — числовой эквивалент первого кода меньше второго.

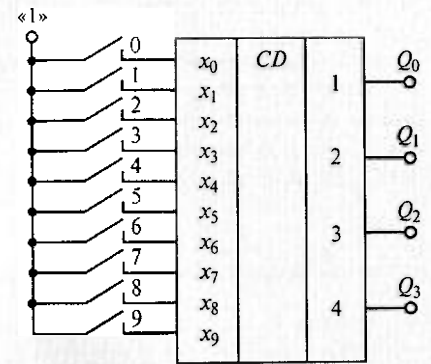


Рис. 3.7. Устройство ввода информации с клавиатуры

Таблица истинности компаратора одноразрядных кодов

$x_0$	$x_1$	$F_+$	$F_>$	$F_<$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Работу компаратора при сравнении двух одноразрядных кодов поясняет таблица истинности (табл. 3.4).

Анализ таблицы истинности показывает, что при любой комбинации входных сигналов на выходе компаратора может быть сформирован только один активный (единичный) логический сигнал. Поэтому при любой разрядности входных кодов достаточно, используя входные сигналы, сформировать только любые два из выходных сигналов. Третий сигнал всегда может быть получен по двум известным.

Система ФАЛ, соответствующая приведенной таблице истинности, имеет следующий вид:

$$F_+ = \bar{x}_1 \bar{x}_0 + x_1 x_0 = \bar{F}_< \bar{F}_>;$$

$$F_< = \bar{x}_1 x_0 = \bar{F}_+ \bar{F}_>;$$

$$F_> = x_1 \bar{x}_0 = \bar{F}_+ F_<.$$

Анализируя приведенные выражения с точки зрения уменьшения аппаратных затрат на реализацию, отметим, что, используя входные переменные, удобнее было бы получить значения  $F_>$  и  $F_<$ , а  $F_+$  — реализовать как их функцию. Однако ввиду того, что выражение для определения  $F_+$  имеет в цифровой технике большое самостоятельное значение, то на нем следует остановиться более подробно.  $F_+$  носит название операции Исключающее ИЛИ-НЕ или инверсия от суммы по модулю два. Пример реализации этой операции с использованием элементов И, ИЛИ и НЕ, а также ее условное графическое обозначение приведены на рис. 3.8. На рис. 3.9 дана структурная логическая схема, соответствующая таблице истинности цифрового компаратора.

**Арифметико-логическое устройство.** Арифметико-логическим устройством называется функционально законченный узел ЭВМ, предназначенный для реализации логических и арифметических операций по обработке информации. Эти операции могут выполняться либо аппаратным способом — с использованием соответствующих электронных устройств, либо программным способом — с применением последовательного исполнения нескольких опе-

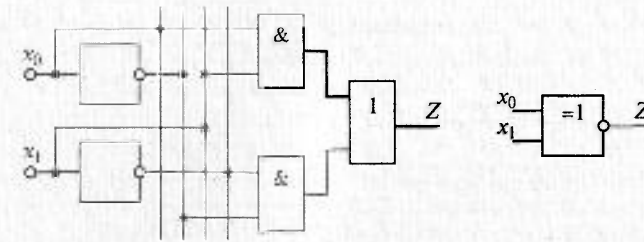


Рис. 3.8. Схемная реализация операции Исключающее ИЛИ-НЕ и условное графическое изображение

раций, выполняемых аппаратным способом. В соответствии со сказанным, АЛУ является одним из основных узлов ЭВМ.

Вне зависимости от того, насколько широк круг операций, реализуемых современными АЛУ, главными среди них остаются операции арифметического сложения и умножения. Важность этих операций подтверждается тем, что при описании характеристик новых машин продолжительность этих операций, как правило, указывается в качестве основных характеристик ЭВМ.

Для выполнения арифметических и логических операций над входными переменными они должны быть введены в АЛУ, поэтому его дополняют вспомогательными устройствами, предназначенными для промежуточного хранения как исходных данных, так и результатов выполнения той или иной операции. Функции этих устройств возлагают на дополнительные регистры.

На рис. 3.10 приведен вариант схемы соединения АЛУ с дополнительными регистрами. По существу эта схема является упрощенной схемой микропроцессора.

Как правило, АЛУ снабжается двумя группами входных и одной группой выходных выводов данных, а также группой выходов, предназначенных для получения вспомогательной информации. Обе группы входных выводов (входных портов) снабжаются буферными регистрами, предназначенными для временного хранения данных. Каждый буферный регистр способен хранить одно слово информации. Разрядность этого слова определяется конкретным типом устройства. Один входной порт АЛУ позволяет принимать данные непосредственно с шины данных, а второй — либо с шины данных, либо из специализированного регистра, называемого *аккумулятором*. Вход этого регистра соединен с выходным портом.

В ряде случаев аккумулятор снабжается вторым входом, подключае-

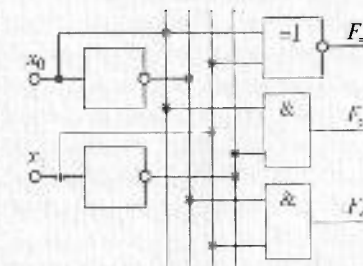


Рис. 3.9. Логическая схема компаратора





Рис. 3.10. Подключение АЛУ к внешним регистрам

мым к шине данных. Поэтому в общем случае в аккумуляторе могут храниться как данные, полученные в результате выполнения предыдущей операции, так и данные, переданные по шине данных. Группа выводов, предназначенных для получения вспомогательной информации о работе АЛУ, подключается к специальному регистру, называемому регистром состояния кода условий или индикатором. В его разрядах хранится служебная информация о результате исполнения последней операции, например, указание о том, что аккумулятор сброшен, в ходе последней операции получен отрицательный результат и т. д.

В зависимости от типа операции АЛУ может оперировать одним или двумя словами данных и, следовательно, пользоваться одним или двумя входными портами. Например, при выполнении операции арифметического сложения используются два порта, а при выполнении операции получения обратного кода (инвертирование кода) нужен только один порт. Результат операции всегда оказывается в аккумуляторе.

Конкретный перечень операций, реализуемых АЛУ, может быть достаточно обширным и различным для устройств разных классов. Однако среди этого разнообразия можно выделить ряд операций, выполняемых АЛУ всех типов. К таким операциям относятся: арифметическое сложение, арифметическое вычитание, логическое умножение, логическое сложение, сумма по модулю два (Исключающее ИЛИ), инверсия, сдвиг вправо, сдвиг влево, приращение положительное (инкремент), приращение отрицательное (декремент).

Перечисленные операции выполняются с использованием только аппаратных средств (схем на ЛЭ), заложенных в АЛУ, и поэтому являются элементарными. Более сложные операции, такие как

арифметические умножение и деление, выполняются, как правило, программно путем комбинаций описанных элементарных операций (микропрограммным способом).

Следует отметить, что по своему построению АЛУ относится к разряду комбинационных устройств, так как не содержит собственных элементов памяти. Поэтому значения его выходных сигналов определяются исключительно комбинацией входных сигналов, а время выполнения конкретной элементарной операции зависит от времени задержки распространения сигнала, т. е. определяется частотными свойствами используемой элементной базы и видом реализуемых ФАЛ.

Анализ работы ЭВМ показал, что до 50 % выполняемых ею операций являются операциями арифметического умножения, а до 45 % — операциями арифметического сложения. Отсюда становится понятным, почему время выполнения операций арифметического сложения и умножения относят к основным параметрам ЭВМ. Первое определяет совершенство применяемой элементной базы, второе — совершенство используемых алгоритмов. Ниже рассмотрим только вопросы, связанные с построением логических схем, используемых при выполнении логических и арифметических операций аппаратным способом.

**Сумматор.** Сумматором называется комбинационное логическое устройство, предназначенное для выполнения операций арифметического сложения чисел, представленных в виде двоичных кодов.

Сумматоры являются одним из основных узлов арифметико-логического устройства. Термин сумматор охватывает широкий спектр устройств, начиная с простейших логических схем до сложнейших цифровых узлов. Общим для всех этих устройств является арифметическое сложение чисел, представленных в двоичной форме.

В каждом из разрядов применяют одноразрядный суммирующий элемент (рис. 3.11) на три входа: два — для слагаемых  $x_1$  и  $x_2$  данного разряда, а третий ( $y_1$ ) — для переноса «1» из соседнего младшего разряда. Устройство снабжают двумя выходами: по одному ( $S$ ) выдается искомая сумма  $x_1 + x_2$ , а по другому ( $p$ ) переносится «1» в соседний старший разряд.

Для сложения двоичных чисел с несколькими разрядами используют двух- и трехразрядные сумматоры с последовательным переносом «1» в старшие разряды. Быстродействие сумматоров последовательного действия лимитируется временем переноса «1» через все элементы устройства. Для улучшения этого показателя применяют сумматоры с параллельным переносом.

**Выполнение операций арифметического умножения.** Традиционно операции арифметичес-

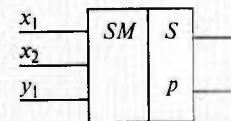


Рис. 3.11. Условное графическое изображение сумматора

кого умножения и деления в ЭВМ выполнялись с использованием последовательностей описанных ранее элементарных функций. Однако в последнее время в связи с успехами технологии были разработаны специализированные ИС, выполняющие эти операции аппаратным способом. Применение таких устройств позволило значительно увеличить быстродействие вычислительных систем.

Логика построения аппаратных умножителей неразрывно связана с традиционным алгоритмом выполнения операции умножения, базирующемся на суммировании частных произведений разрядов сомножителей. Проиллюстрируем сказанное на примере.

**Пример 3.2**

Умножение двухразрядных двоичных кодов  $a_1 a_0$  и  $b_1 b_0$ :

$$\begin{array}{r} \phantom{+} \phantom{+} \phantom{+} \phantom{+} \\ \phantom{+} \phantom{+} \phantom{+} \phantom{+} \phantom{+} \\ \phantom{+} \phantom{+} \phantom{+} \phantom{+} \phantom{+} \phantom{+} \\ \phantom{+} \phantom{+} \phantom{+} \phantom{+} \phantom{+} \phantom{+} \phantom{+} \\ \hline M_3 \phantom{M_2} \phantom{M_1} \phantom{M_0} \end{array}$$

Структурная схема устройства показана на рис. 3.12. Частные произведения разрядов сомножителей формируются логическими элементами 2И, обозначенными DD1...DD4. Суммируя эти произведения сумматорами DD5 и DD6, находят значение кода результата. Приведенная структура носит название *матричного множительного блока*.

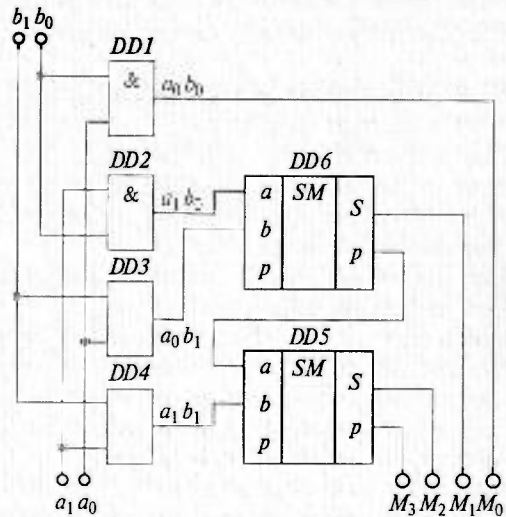


Рис. 3.12. Структурная схема матричного умножителя

**3.3. Цифровые автоматы (триггеры, регистры, счетчики)**

**Цифровые триггеры. Классификация.** В цифровых автоматах в роли элементов памяти используют микроэлектронные триггеры различных модификаций.

*Триггером* называется устройство, способное формировать два устойчивых значения выходного сигнала (логического «0» и логической «1») и скачкообразно изменять эти значения под действием внешнего управляющего сигнала.

Триггеры классифицируются на несколько типов в зависимости от принципа их работы и целевого назначения.

По способу управления триггеры подразделяют на два класса: синхронизируемые и несинхронизируемые (асинхронные). Синхронизируемый триггер снабжен вспомогательным входом синхронизации, который разрешает переключение триггера при наличии на этом входе соответствующего сигнала.

По способу организации логических связей триггеры классифицируют на следующие типы: *RS* — с раздельной установкой состояний «1» и «0»; *T* — со счетным входом; *D* — с приемом информации по одному входу; *JK* (универсальный) — совмещает в себе свойства *D*, *RS* и *T*-триггеров (позволяет раздельную установку состояний «1» и «0», но способен при одновременном воздействии входных сигналов функционировать как *T*-триггер).

**Условное обозначение.** В качестве поясняющего символа для условного обозначения триггера применяют букву *T*, которую помещают в верхней части основного поля графического обозначения (рис. 3.13, а). Выходы триггера обозначают буквой *Q*. При этом выход «0» снабжают индексом отрицания (черточкой сверху) для отличия от выхода «1», (см. рис. 3.13, а) для *RS*-триггеров.

**Принципы функционирования.** Асинхронный *RS*-триггер (см. рис. 3.13, а) имеет два информационных входа, один из которых обозначают буквой *S* (от англ. *set* — устанавливать), а другой — *R* (от англ. *reset* — сбрасывать). Прямой выход триггера, как отмечалось, обозначают буквой *Q*, а инверсный —  $\bar{Q}$ .

Асинхронный *RS*-триггер можно получить на двух логичес-

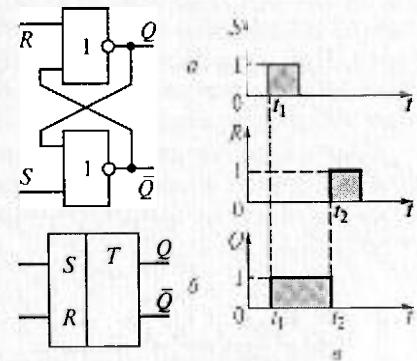


Рис. 3.13. Функциональная схема (а) условное графическое изображение (б), и временные диаграммы процессов (в) асинхронного *RS*-триггера

ких элементах ИЛИ-НЕ, если снабдить их перекрестными положительными обратными связями, как показано на рис. 3.13, б. Временные диаграммы, поясняющие принцип работы этого триггера, приведены на рис. 3.13, в (длительностью переходных процессов при коммутациях пренебрегаем).

При подключении ко входу  $S$  логической «1» (см. рис. 3.13, б), т.е.  $S = 1$  (момент  $t_1$  см. рис. 3.13, в), на инверсном выходе  $\bar{Q}$  получаем «0». Поскольку этот сигнал по цепи обратной связи подается на один из входов верхнего элемента ИЛИ-НЕ, а на другом его входе действует также сигнал «0» со входа  $R$ , имеем  $Q = 1$ . Это состояние триггера может сохраняться сколь угодно долго и не зависит от последующих изменений сигнала на входе  $S$ , если на входе  $R$  поддерживается состояние «0» (состояние  $S = 1$  и  $R = 1$  является для данного триггера запрещенным, так как при этой комбинации входных сигналов выходное состояние триггера будет непредсказуемо). Таким образом,  $RS$ -триггер запоминает информацию о состоянии  $S = 1$  и поддерживает ее до тех пор, пока на входе  $R$  не появится сигнал сброса 1 (момент  $t_2$  см. рис. 3.13, в).

При  $R = 1$  на выходе  $Q$  получим сигнал  $Q = 0$ , а соответствующий канал обратной связи обеспечит  $\bar{Q} = 1$  и т.д.

Синхронный  $RS$ -триггер, обозначаемый также буквами  $RST$ , имеет дополнительный  $C$ -вход (от англ. *clock* — часы), на который подают импульсы синхронизации (рис. 3.14, а). Синхронный триггер получают при подключении ко входу асинхронного  $RS$ -триггера двух дополнительных элементов И-НЕ (рис. 3.14, б).

Как видно, при отсутствии сигнала на  $C$ -входе, т.е.  $C = 0$ , входы  $R$  и  $S$  оказываются отключенными от собственно триггера (очерчен пунктиром), и изменение сигналов на этих входах не может изменить состояние триггера. С момента появления разрешающего сигнала  $C = 1$  входные элементы И-НЕ выполняют функцию инверторов. При этом состояние триггера будет однозначно определяться значениями сигналов на  $S$ - и  $R$ -входах по аналогии с асинхронным  $RS$ -триггером.

Следует подчеркнуть, что описанный триггер может изменять свое состояние в любой момент на интервале действия сигнала  $C = 1$ . Такой триггер называют триггером со статическим входом синхронизации.

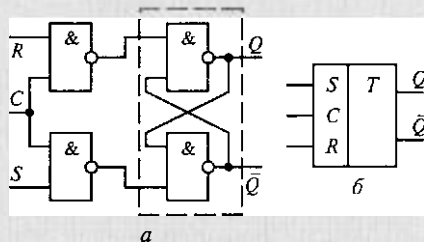


Рис. 3.14. Функциональная схема синхронного  $RS$ -триггера (а) и его условное графическое изображение (б)

Наибольшее практическое распространение получили триггеры с динамическим (импульсным) входом синхронизации. Суть построения такого триггера заключается в обеспечении его переключения лишь на интервале изменения сигнала входа  $C$ , т.е. либо по фронту, либо по срезу импульса синхронизации. Такое решение позволяет значительно повысить надежность и помехозащищенность триггерных устройств, так как сводит к минимуму интервал, на котором возможна перезапись информации. Технически указанный режим работы достигается заменой дополнительных логических элементов (см. рис. 3.14), вспомогательными  $RS$ -триггерами (так называемая схема трех триггеров).

$T$ -триггер (от англ. *tumbler* — опрокидыватель) находит широкое применение в счетчиках импульсов цифровых систем. В этом триггере, условное обозначение которого приведено на рис. 3.15, а, имеется один  $T$ -вход, при каждом воздействии на который (импульсом) происходит очередное переключение триггера из одного состояния в другое. Поэтому такой прибор именуют триггером со счетным входом. Число переключений равно числу поступивших на вход импульсов.

В основу большинства  $T$ -триггеров положена схема двухступенчатого триггера, образованного последовательным соединением двух синхронных  $RST$ -триггеров, охваченных перекрестными обратными связями (рис. 3.15, б). Кроме того,  $C$ -входы этих  $RST$ -триггеров соединены между собой элементом НЕ и объединены в общий для всего  $T$ -триггера вход  $T$ . Таким образом, любое управляющее воздействие на вход  $C_1$  вызывает отключение второго  $RST$ -триггера по входной цепи. Однако за счет соединения триггера  $T_1$  с  $T_2$  цепями обратной связи каждое принудительное переключение  $T_1$  вызывает опрокидывание  $T_2$ . Принцип работы  $T$ -триггера иллюстрируют временные диаграммы процессов на  $T$ -входе и  $Q$ -выходе (рис. 3.15, в).

$D$ -триггер (от англ. *delay* — задержка), условное обозначение которого приведено на рис. 3.16, а, формирует выходной сиг-

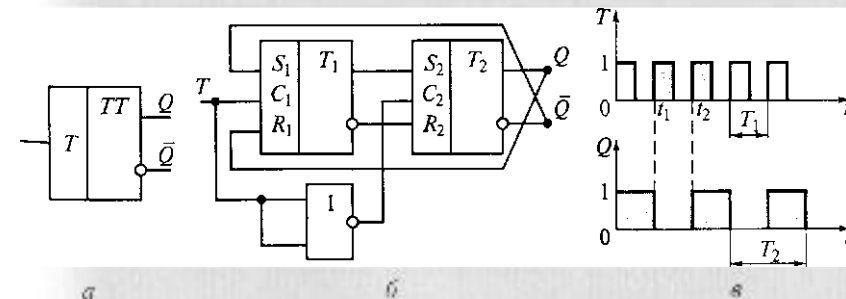


Рис. 3.15. Условное графическое изображение (а), функциональная схема (б), временные диаграммы процессов (в)  $T$ -триггера



нал на  $Q$ -выходе с задержкой относительно управляющего воздействия на  $D$ -входе. Для этого его снабжают  $C$ -входом, возбуждение которого (статическое или импульсное) позволяет переключать триггер в состояние, соответствующее сигналу на  $D$ -входе.  $D$ -триггеры получают на основе  $RS$ -триггеров, используя на входе однотипные логические устройства И-НЕ (рис. 3.16, б). Временные диаграммы процессов представлены на рис. 3.16, в.

Универсальный  $JK$ -триггер, условное обозначение которого приведено на рис. 3.17, а, имеет три входа: два информационных ( $J$  и  $K$ ) и один синхронизирующий ( $C$ ). Он легко реализуется на двух  $RST$ -триггерах с обратными связями, как показано на рис. 3.17, б.

$JK$ -триггер относят к универсальному типу, поскольку при различных вариантах переключения его входных зажимов можно получить схему триггеров всех других типов. В микроэлектронном исполнении  $JK$ -триггер имеет расширенную входную логику, при которой можно использовать такой многофункциональный триггер в различных микросхемах одной серии.

**Регистры.** Устройство, предназначенное для хранения информации, представленной в двоичной форме, называют *регистром*.

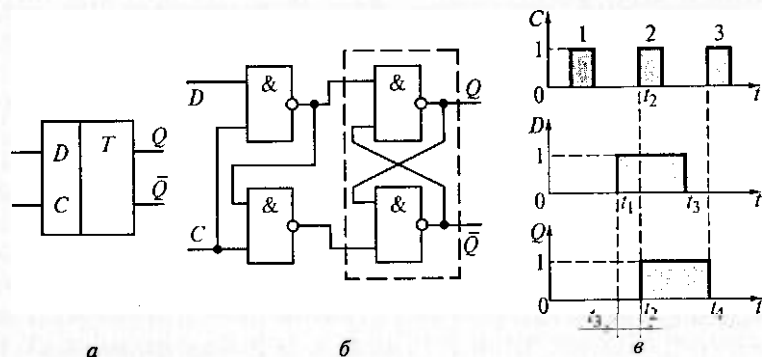


Рис. 3.16. Условное графическое изображение (а), функциональная схема (б) и временные диаграммы процессов (в)  $D$ -триггера

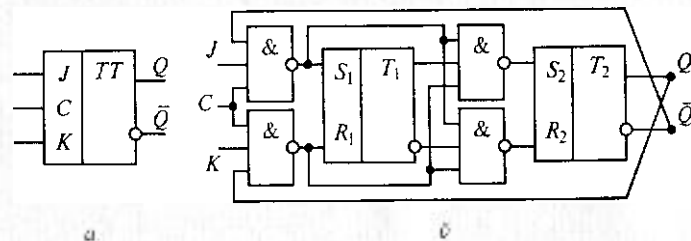


Рис. 3.17. Условное графическое изображение (а) и функциональная схема (б) универсального  $JK$ -триггера

Он состоит из связанных друг с другом триггеров, каждый из которых служит для хранения «0» или «1». Количество триггеров определяет разрядность регистра. Регистры могут выполнять также операции приема, передачи и преобразования информации, обеспечиваемые системой управления.

В зависимости от выполняемых функций регистры подразделяют на параллельные и сдвиговые.

На рис. 3.18, а приведено условное обозначение простейшего двухразрядного параллельного регистра, выполненного на  $RST$ -триггерах, а на рис. 3.18, б показана его функциональная схема. Для записи информации в двоичном коде сначала выполняют операцию обнуления триггеров путем подачи соответствующих сигналов на все  $R$ - и  $S$ -входы. После этого через  $S$ -входы осуществляют либо переключения триггеров (для записи «1»), либо не меняют их состояния (для сохранения «0»). Считывание информации, заключенной в регистре, производится с  $Q$ -выходов триггеров регистра.

Важнейшими показателями регистра являются число разрядов и быстродействие устройства. Низкое быстродействие ограничивает максимальную частоту тактовых импульсов системы управления, которая обеспечивает запись, считывание и простейшую обработку информации. Последняя операция достигается в сдвиговых регистрах путем перемещения информации вправо и влево — в соответствии с операциями деления и умножения двоичных чисел на значения  $2^n$  ( $n = 0, 1, 2, \dots$ ). В регистрах сдвига обычно применяют соединенные последовательно  $D$ -триггеры с тремя входами: информационным ( $D$ -вход), сдвигающим ( $C$ -вход) и установочным ( $R$ -вход). При этом  $C$ - и  $R$ -входы всех триггеров объединяют для одновременного управления (при сдвигах или установке нуля) всем устройством. Регистр сдвига можно построить на  $RS$ -триггерах, если в каждом разряде использовать два триггера, управляемых двумя сдвинутыми во времени тактовыми импульсами. В этом случае появляется возможность поразрядного продвижения записываемой информации от входа к выходу.



Рис. 3.18. Условное графическое изображение (а) и функциональная схема (б) двухразрядного регистра

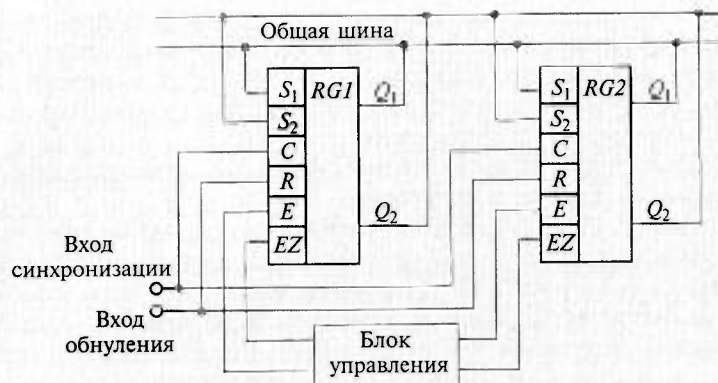


Рис. 3.19. Схема подключения нескольких регистров к общей шине

В сложных цифровых устройствах обмен информацией между отдельными регистрами, как правило, осуществляется по общей шине, соединяющей входы и выходы соответствующих разрядов всех регистров (рис. 3.19). При этом каждый регистр снабжается входом  $E$  разрешения записи и входом  $EZ$ , отключающим выходы его триггеров от соответствующих выходных выводов. Выбор конкретного пути передачи информации в этой схеме производится подачей соответствующих разрешающих сигналов с блока управления. Так, для перезаписи информации из регистра  $RG1$  разрешающие сигналы необходимо подать на вход  $EZ$  регистра  $RG2$  и вход  $E$  регистра  $RG1$ . В этом случае к общей шине будут подключены только выходы  $RG2$  и входы  $RG1$  и по импульсу синхронизации произойдет требуемая перезапись информации.

**Цифровые счетчики импульсов.** *Цифровым счетчиком* называют функциональное устройство на триггерах, обеспечивающее счет поступающих на его вход импульсов. Результат счета, формируемый обычно в двоичном коде, может считываться или храниться в триггере счетчика. При необходимости результат считывают после каждого счетного импульса на входе. Максимальное число, отсчитываемое счетчиком в двоичном коде, составляет  $2^n - 1$ , где  $n$  — число последовательно включенных триггеров. При отсутствии ограничений на число поступающих импульсов счетчик через каждые  $2^n$  импульсов будет возвращаться в исходное нулевое состояние. Счетчики такого типа именуют *пересчетными*.

Цифровые счетчики можно получить, используя совокупность простейших триггерных и логических микросхем. Имеются также многоразрядные универсальные счетчики, выполненные в виде одной микросхемы высокого уровня интеграции (например, микросхемы К155ИЕ7, К564ИЕ14 и др.).

**Классификация счетчиков.** По целевому назначению счетчики классифицируют на следующие типы: суммирующие (производят

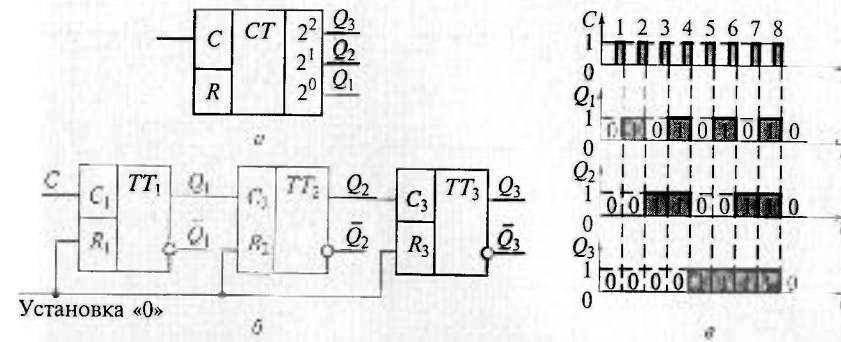


Рис. 3.20. Условное графическое изображение (а), функциональная схема (б) и временная диаграмма работы (в) трехразрядного счетчика

прямой счет импульсов); вычитающие (выполняют обратный счет от наибольшего значения до нуля); реверсивные (применяют как в режимах прямого, так и обратного счета).

**Принцип функционирования счетчика** рассмотрим на примере трехразрядного счетчика, собранного на  $T$ -триггерах. Его условное обозначение и функциональная схема показаны на рис. 3.20, а, б. Для обнуления счетчика (перед началом работы) используют специальную шину Установка «0» (см. рис. 3.20, б), к которой подключены все  $R$ -входы триггеров.

При появлении счетных импульсов на  $C$ -входе счетчика наблюдается последовательное переключение каждого из взаимосвязанных триггеров, как показано на диаграммах рис. 3.20, в. Как видно из диаграммы, период переключения каждого последующего триггера увеличивается в два раза по сравнению с предыду-

Таблица 3.5

Таблица состояний трехразрядного счетчика

Номер импульса	Состояние триггеров			Выходной код	
	$Q_3$	$Q_2$	$Q_1$	двоичный	десятичный
0	0	0	0	0	0
1	0	0	1	01	1
2	0	1	0	10	2
3	0	1	1	11	3
4	1	0	0	100	4
5	1	0	1	101	5
6	1	1	0	110	6
7	1	1	1	111	7
8	0	0	0	0	0

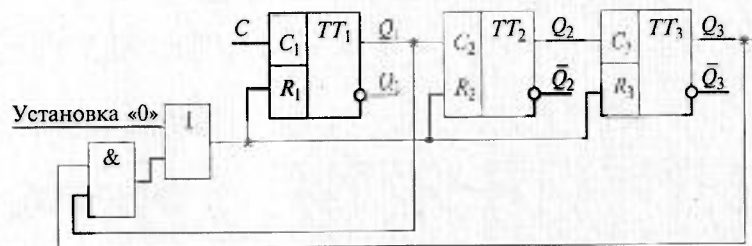


Рис. 3.21. Двоичный счетчик, обеспечивающий счет до 4

шим. Приписывая  $Q$ -выходам переключенных триггеров символ «1» (эти состояния на временной диаграмме показаны серым цветом, можно систематизировать состояния счетчика, как показано в табл. 3.5.

**Анализ работы счетчика.** Как видно из табл. 3.5, переключения триггеров в состояния «1» имеют место при предварительных переключениях предыдущего триггера от «1» к «0». Это означает, что в указанном режиме формируется сигнал переноса, вызывающий опрокидывание следующего триггера.

Из табл. 3.5 следует также, что модуль пересчета, т. е. число состояний счетчика между его обнулениями, трехразрядного счетчика равен  $2^3 = 8$ . При этом вес первого триггера равен  $2^0$ , второго —  $2^1$  и третьего —  $2^2$ . Указанные «веса» показаны в правой (вспомогательной) зоне условного обозначения трехразрядного счетчика (см. рис. 3.20, а).

### Пример 3.3

Спроектировать двоичный счетчик импульсов, осуществляющий счет до 4.

Для решения задачи могут быть использованы три триггера, соединенные по схеме трехразрядного двоичного счетчика с организованными обратными связями, обеспечивающими счет до 4. Каждый пятый входной импульс будет обнулять триггеры, и счет начнется сначала (рис. 3.21).

## 3.4. Запоминающие устройства ЭВМ

**Память ЭВМ** — функциональная часть ЭВМ, предназначенная для записи, хранения и выдачи данных. В соответствии с этим различают три режима работы памяти: записи, хранения и считывания. Запись в запоминающее устройство (ЗУ) или считывание из него информации иначе называются обращением к ЗУ. Быстродействие памяти определяется продолжительностью операции обращения к ЗУ. Время обращения  $t_0$  при записи информации складывается из времени поиска операнда  $t_n$ , стирания ранее за-

писанной информации  $t_{ст}$  (при необходимости) и записи нового числа  $t_{зн}$

$$t_0 = t_n + t_{ст} + t_{зн}.$$

При считывании информации время цикла обращения складывается из времени поиска, времени считывания  $t_{сч}$  и восстановления считанных кодов  $t_{восст}$  (при необходимости, в случае разрушающего считывания)

$$t_0 = t_n + t_{ст} + t_{восст}.$$

Запоминающие устройства разделяют: по использованию — на внешние и внутренние (или оперативные); по назначению — на сверхоперативные, оперативные, постоянные, буферные и внешние; по физическим принципам действия — на полупроводниковые, магнитные и оптические; по способу хранения информации — на статические и динамические; по способу доступа к заданной ячейке (для адресных ЗУ) — с последовательным, циклическим и произвольным доступом; по характеру обращения — с адресным обращением (или адресной выборкой) и с ассоциативным обращением (ассоциативной выборкой).

Для достижения в ЭВМ одновременно и большой информационной емкости и высокого быстродействия используется принцип иерархического построения памяти. Техническая реализация иерархических структур обеспечивает большую емкость памяти и малое время обращения, что позволяет решать на ЭВМ сложные задачи, требующие хранения большого количества данных. При иерархическом принципе построения структуры ЗУ логическая организация потоков информации должна быть такой, чтобы все информационное поле ЭВМ или ВС представлялось в виде внутреннего абстрактного ЗУ с единым адресным пространством. Это абстрактное ЗУ называют виртуальным (кажущимся) ЗУ. Адресацию его ячеек осуществляют посредством абстрактных математических адресов с использованием страничных таблиц.

В иерархической структуре можно выделить следующие уровни памяти (рис. 3.22).

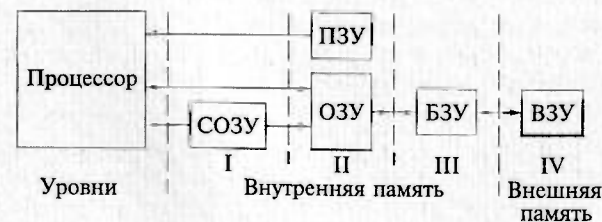


Рис. 3.22. Схема иерархии ЗУ ЭВМ



*Сверхоперативные ЗУ (СОЗУ) или местная память.* Они имеют быстродействие, соизмеримое с быстродействием процессора. Емкость СОЗУ обычно составляет от нескольких десятков до нескольких тысяч слов, а период обращения — десятые или сотые доли микросекунды. Они предназначаются для хранения ряда чисел, используемых некоторой текущей последовательностью команд программы. Сверхоперативные ЗУ применяют в том случае, если быстродействие процессора ограничивается быстродействием ОЗУ.

*Оперативные ЗУ (ОЗУ) или основная память* предназначены для хранения данных и программ текущих вычислений, а также программ, к которым следует быстро перейти, если в ходе вычислительного процесса возникло прерывание. ОЗУ современных ЭВМ имеют емкость от нескольких тысяч до сотен тысяч слов и период обращения от долей до нескольких микросекунд. ОЗУ может быть связано с процессором как непосредственными связями, так и через СОЗУ. В качестве элементов памяти ОЗУ и СОЗУ используются полупроводниковые элементы, тонкие магнитные пленки, ферритовые сердечники и др.

*Постоянные ЗУ (ПЗУ) или односторонние ЗУ* предназначены только для хранения и считывания информации, которая не изменяется в процессе вычислений, например постоянно используемые программы, различные константы, таблицы функций, микропрограммы и т. п. В ПЗУ информация записывается один раз при изготовлении, а при работе только считывается. Такие ПЗУ имеют меньшую аппаратную сложность, чем ОЗУ.

*Буферные ЗУ (БЗУ).* Эти ЗУ используются для промежуточного хранения данных при обмене между устройствами с разным быстродействием, например между ОЗУ и внешним ЗУ. По емкости и быстродействию БЗУ занимают промежуточное место между ОЗУ и внешним ЗУ. Они могут быть построены на полупроводниковых элементах, ферритовых сердечниках и на магнитных дисках.

*Внешние ЗУ (ВЗУ).* Они предназначены для хранения больших массивов информации. Объем данных, которые могут одновременно храниться в таких ЗУ, обычно превышает сотни миллионов слов, однако период обращения к ним составляет от нескольких миллисекунд до нескольких десятков секунд. Данные, хранящиеся во ВЗУ, непосредственно не используются в вычислительном процессе, а передаются из ВЗУ в ОЗУ. В качестве ВЗУ чаще всего используют накопители информации на магнитных лентах (НМЛ), на гибких и жестких магнитных дисках (НГМД, НЖМД), на микросхемах на основе материалов, содержащих цилиндрические магнитные домены (ЦМД), а также на оптических дисках.

Основой полупроводникового запоминающего элемента (ЗЭ) служит бистабильная схема триггера, состоящая из инверторов с перекрестными связями, с цепями управления для записи и считывания информации. Достоинством таких ЗЭ является единство тех-

нологического процесса изготовления как самих ЗЭ, так и элементов электронного обрамления — регистров, дешифраторов и др.

Разнообразие типов ЗЭ объясняет большое количество способов объединения ЗЭ в большие интегральные схемы (БИС), а также способов объединения БИС в накопитель.

Классификацию ЗЭ можно провести по трем типам признаков: физико-технологическим, схемотехническим и системотехническим.

По физико-технологическим признакам ЗЭ подразделяются на выполненные на биполярных транзисторах и на выполненные на МДП-транзисторах. В основе биполярных транзисторов лежат явления, происходящие в полупроводниковом блоке, содержащем 3 области с чередующимися типами проводимости  $n$  и  $p$ , и, следовательно, с двумя  $p-n$ -переходами, отстоящими друг от друга на расстояние, меньшее диффузионной длины неосновных носителей — электронов и «дырок». С использованием биполярной технологии разработана большая группа ЗЭ: на одноэмиттерных и многоэмиттерных транзисторах, тиристорах, инжекционных схемах.

В основе МДП-транзисторов (униполярных) лежит принцип управления проводимостью полупроводникового материала между двумя электродами при помощи электрического поля (полевые транзисторы). МДП-транзистор — активный прибор, в котором ток создается основными носителями под действием продольного электрического поля, а управление величиной тока осуществляется поперечным электрическим полем, приложенным к управляющему электроду. Существуют две разновидности МДП-транзисторов: с каналами  $n$ - и  $p$ -типов. Плотность упаковки достигает  $10^6$  бит на кристалл.

Схемотехнические признаки классификации:

- число транзисторов в ЗЭ;
- количество и функции адресных и разрядных шин выборки (совмещенные или отдельные функции записи, считывания, подачи питающих напряжений);
- характер нагрузочных сопротивлений в триггере (линейная, нелинейная, переключаемая нагрузка);
- тип связи ЗЭ с разрядными шинами (непосредственная, через ключевые элементы).

Системотехнических признаков три: принцип хранения; принцип считывания; форма и вид считанного из ЗЭ сигнала. Принципы хранения — статический, динамический и квазистатический. Принципы считывания — с разрушением (ЗЭ на ферритовых сердечниках с прямоугольной петлей гистерезиса и ЗЭ на конденсаторах) и без разрушения информации. Формы считанного из ЗЭ сигнала — однополярный сигнал, двухполярный и парафазный. И, наконец, виды считанного сигнала — импульсный ток или импульсное напряжение.

**Запоминающие элементы статических ОЗУ.** Микросхемы памяти изготавливаются методами полупроводниковой технологии при использовании в качестве исходных материалов кремния, арсенида галлия, аморфных полупроводников или структур «кремний на сапфире». В настоящее время кремний остается основным материалом для изготовления микросхем памяти.

Особенности различных типов микросхем памяти определяются типом использованных в них запоминающих элементов, дешифраторов, формирователей сигналов записи-считывания и буферных каскадов. В большинстве статических микросхем памяти эти схемы реализованы на базе биполярных элементов ТТЛ, ТТЛШ, ЭСЛ или на базе элементов комплементарных МОП и МДП структур.

На рис. 3.23 приведена принципиальная электрическая схема ЗЭ на биполярных транзисторах. Данный элемент использует технологию ТТЛ и предназначен для применения в ЗУ с двухмерной адресацией. Его основу составляют два инвертора, выполненных на трехэмиттерных транзисторах  $VT1$  и  $VT2$ . Инверторы включены последовательно и охвачены глубокой положительной обратной связью (ПОС). Две пары попарно объединенных эмиттеров транзисторов образуют выходы выборки элемента  $CS1$  и  $CS2$ . Третья пара эмиттеров транзисторов образует прямой  $P_1$  и инверсный  $P_2$  выходы элемента, которые через входные сопротивления усилителя считывания (см. рис. 3.23) подключены к общей шине.

В режиме хранения на один или оба выхода выборки ( $CS1$ ,  $CS2$ ) ЗЭ подается напряжение низкого уровня. При этом триггер, образованный инверторами, находится в одном из устойчивых состояний. Предположим, что транзистор  $VT1$  насыщен, а  $VT2$  заперт. Весь ток насыщенного транзистора  $VT1$  замыкается через один из выводов выборки элемента на общую шину. Поэтому в цепи вывода  $P_1$  ток отсутствует и информация из ЗЭ не попадает на вход усилителя считывания ( $U_{R_{вх\ ус}} = 0$ ).

Для считывания информации на оба входа выборки элемента необходимо подать напряжение высокого уровня. При этом единственным путем протекания тока насыщенного транзистора остается выходной вывод  $P_1$  ЗЭ. Этот ток создает на входном сопротивлении усилителя считывания напряжение  $U_{R_{вх\ ус}}$ , полярность которого соответствует записанной в элемент информации. Следует отметить, что при считывании из элемента информации она не теря-

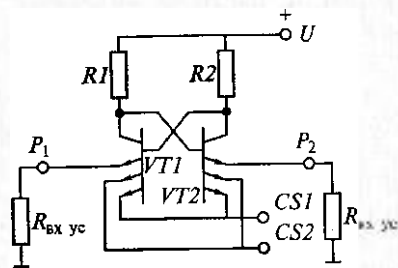


Рис. 3.23. Принципиальная электрическая схема ЗЭ на основе биполярных транзисторов

ется. При подаче на один или оба входа выборки напряжения низкого уровня триггер продолжает оставаться в том же состоянии.

При необходимости записать в элемент новую информацию на него также первоначально подаются сигналы выборки. После чего на внешних шинах устанавливается полярность напряжения, соответствующая новой информации. Для рассматриваемого случая на вывод  $P_1$  подается напряжение высокого уровня, а на  $P_2$  — низкого уровня. При этом, так как все цепи протекания эмиттерного тока транзистора  $VT1$  оказываются разорванными, на его коллекторе формируется напряжение высокого уровня. Это напряжение насыщает транзистор  $VT2$ , который, формируя на своем коллекторе напряжение низкого уровня, подтверждает запертое состояние транзистора  $VT1$ . В ЗЭ записывается новая информация. После снятия с элемента сигналов выборки новая информация будет храниться в триггере до момента очередной перезаписи.

**Запоминающие элементы динамических ОЗУ.** Запоминающие элементы динамических ОЗУ построены на базе МОП-транзисторов. Здесь используется тот факт, что у МОП-транзистора высокое входное сопротивление. Это позволило построить схемы ЗЭ, у которых входной ток близок к нулю. В основу положен принцип накопления и восстановления — регенерации — заряда на затворе МОП транзистора. Основой ЗЭ является паразитная емкость  $C_n$ , накапливающая заряд. Из-за постепенного разряда  $C_n$  за счет токов утечки необходимы схемы восстановления, регенерации заряда. Периодическое, а не постоянное поступление энергии в ЗЭ позволяет сократить потребляемую мощность и упростить схему, что снижает размер площади и увеличивает плотность компоновки на кристалле. На рис. 3.24 приведена схема ЗЭ динамического типа. В режиме записи  $VT1$  и  $VT3$  совмещают функции возбуждения разрядных шин и нагрузки для транзисторов триггера  $VT2$ ,  $VT4$ .

В режиме записи подаются положительный импульс в адресную шину ША и отрицательный — в разрядную шину ШР0 либо ШР1 в зависимости от того, записывается логический «0» или логическая «1». При этом  $VT1$  или  $VT3$  открывается, информация заносится в триггер. Информация хранится в виде зарядов паразитных емкостей  $C_{n1}$  и  $C_{n2}$ , причем «0» хранится неограниченное время, а «1» из-за токов утечки должна периодически восстанавливаться. Для регенерации информации периодически возбуждается шина ША и одновременно на разрядные шины подается одинаковое по знаку напряжение низ-

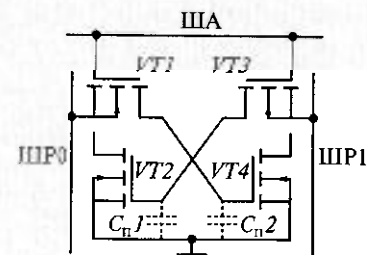


Рис. 3.24. Динамические ЗЭ на МОП-транзисторе

кого уровня в случае  $p$ -канальных транзисторов. В этом состоит одна из отличительных особенностей динамических ОЗУ.

В режиме считывания возбуждают ША, как бы подключая источник питания к триггеру, и ток считывания через  $VT1$  или  $VT3$  поступает в соответствующую ШР — в ШР0, если хранился «0», или в ШР1, если хранилась «1».

**Структуры матриц накопителей информации.** Статические ОЗУ на биполярных транзисторах представляют собой матрицу запоминающих элементов, которая и является основой ОЗУ. На одном кристалле биполярной БИС можно расположить, например, ОЗУ на 4096 бит с временем выборки менее 60 нс. Построение (организация) матрицы определяется способом выборки (опроса) ЗЭ при записи или считывании.

В структурной схеме матрицы с пословной выборкой и одной ступенью дешифрации (рис. 3.25, а) одна строка данных образует слово из  $m$  разрядов. На схеме символами  $A_1, A_2, \dots, A_n$  обозначены адресные, а  $P_1, P_2, \dots, P_m$  — разрядные (информационные) шины. Как видно из схемы, адресные шины связаны с каждым ЗЭ одного слова, в то время как разрядные шины имеют связь с ЗЭ одноименного разряда всех слов. При наличии в адресной шине  $A_i$  сигнала выбора  $i$ -го слова, состояние каждого из ЗЭ в этом слове может быть считано по разрядным шинам  $P_1 - P_m$ . В режиме записи информации по выбранному адресу  $A_i$  на разрядные шины  $P_1, P_2, \dots, P_m$  подаются электрические сигналы «0» и «1», составляющие двоичный код записываемого слова, которые попадут соответственно на каждый из ЗЭ  $i$ -й строки:  $ЗЭ_{i1}, ЗЭ_{i2}, ЗЭ_{i3}, \dots, ЗЭ_{im}$ .

На рис. 3.25, а не показаны устройства управления матрицей (дешифратор с адресными формирователями, усилители считывания и записи), которые для повышения надежности работы ОЗУ изготавливаются на одном кристалле с матрицей.

В схеме двухкоординатной матрицы накопителя информации с двумя ступенями дешифрации (рис. 3.25, б) ЗЭ выбирается с помощью двух адресных шин:  $X_1, X_2, \dots, X_n$  и  $Y_1, Y_2, Y_3, \dots, Y_m$ . Например, при наличии сигнала на адресных шинах  $X_1, Y_1$  будет выбран только ЗЭ<sub>11</sub>. Его состояние можно считывать по общей для всех элементов разрядной шине. Чтобы записать «1» в выбранный ЗЭ, по разрядной шине необходимо подать сигнал, соответствующий уровню логической «1». Эта организация матриц позволяет оперировать  $m \times n$  одноразрядными словами, что позволяет гибко менять разрядность слов при проектировании.

В матрице памяти (см. рис. 3.25, б) из трех координат, которые являются входами для каждого элемента памяти, две ( $X$  и  $Y$ ) используются для выбора ячейки или даже ЗЭ с заданным адресом, а третья координата — для записи информации. Такой способ организации ОЗУ обозначается  $3D$  (от англ. *dimension* — измерение, координата). Для вывода считываемой информации может использоваться та же третья координата.

При организации матрицы памяти по способу  $2D$  (см. рис. 3.25, а) только одна координата используется для выбора ячейки по заданному адресу, а другая — для записи информации в разряды ячейки памяти и ее считывания. Код адреса в таком ЗУ преобразуется в один управляющий сигнал, подаваемый в выбранную ячейку.

Организация матрицы памяти по способу  $2,5D$  является промежуточной по отношению к способам  $3D$  и  $2D$ , что и отражено в названии способа. ЗЭ имеют две координаты. Одна из них, аналогично способу  $2D$ , используется для выбора, но выбирается одновременно  $p$  ячеек. Вторая координата служит не только для записи и считывания информации, но и для выбора одной из  $p$  ячеек аналогично способу  $3D$ . Код адреса в таком ЗУ делится на две части. Одна часть служит для выбора группы из  $p$  ячеек, другая позволяет выбрать одну из этих ячеек.

**Элементная база и организация постоянных запоминающих устройств.** В ПЗУ между входной информацией — адресом — и выходной информацией — считанным словом — существует взаимнооднозначное соответствие. По существу ПЗУ представляют собой конечные автоматы без памяти. ПЗУ используются лишь в режиме считывания информации, при этом записанная информация не разрушается. Хранящаяся в ПЗУ информация не изменяется в процессе решения задачи. Она записывается заблаговременно при изготовлении устройств и может быть произвольной. ПЗУ часто обозначают *ROM (read only memory)*. В ПЗУ универсальных ЭВМ хранятся константы, стандартные подпрограммы, контрольные программы — тесты и т.п. В виде ПЗУ выполняется микропрограммное устройство управления, в котором хранятся микропрограммы.

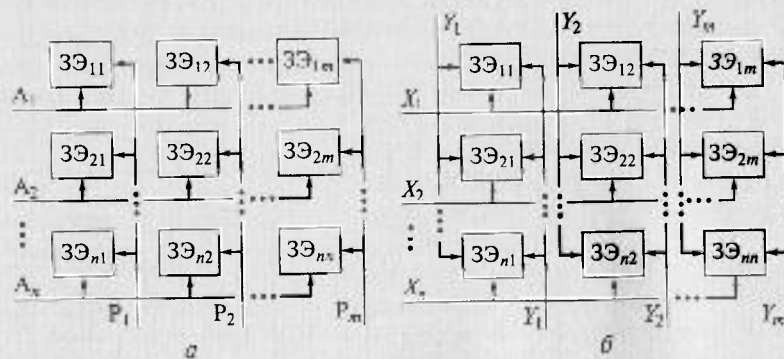


Рис. 3.25. Структурные схемы матриц накопителей информации: а — с пословной выборкой и одной ступенью дешифрации; б — с двухкоординатной выборкой и двумя ступенями дешифрации (с поразрядной выборкой)



В специализированных ЭВМ, работающих в ряде случаев без вмешательства оператора, в ПЗУ хранятся основные программы работы ЭВМ. ПЗУ могут выполнять роль кодирующих и декодирующих устройств, могут использоваться для выполнения арифметических операций табличным способом.

ПЗУ проще, дешевле и надежнее ОЗУ, поскольку в них отсутствуют схемы записи информации, и для хранения информации могут применяться более простые и дешевые элементы. Простота элементов матрицы ПЗУ, а также отсутствие цепей регенерации позволяют получить период обращения к ПЗУ в несколько раз меньший, чем период обращения к ОЗУ. В ПЗУ, предназначенных для хранения констант и команд, период обращения обычно делается равным периоду обращения к ОЗУ.

Программируемые постоянные запоминающие устройства (ППЗУ) отличаются от ПЗУ тем, что пользователь может самостоятельно запрограммировать ПЗУ — ввести в него программу с помощью специального устройства программатора, но только один раз: после введения программы содержимое памяти изменить нельзя.

Репрограммируемое постоянное запоминающее устройство (РПЗУ), называемое также стираемым ПЗУ, допускает неоднократное стирание информации и запись новой с помощью программатора. Это облегчает исправление обнаруженных ошибок и позволяет изменять содержимое памяти.

Структурная схема ПЗУ аналогична структурной схеме ОЗУ. Различие состоит в том, что в ПЗУ отсутствуют схемы, предназначенные для записи информации. Матрица накопителя информации обычно организуется по системе 2D. В узлах матриц располагаются запоминающие элементы, иногда называемые для ПЗУ элементами связи (ЭС). ЭС связывают между собой электрически (или не связывают) адресные и разрядные шины. Наличие ЭС соответствует коду «1», отсутствие ЭС — коду «0».

Полупроводниковые ЭС могут быть диодными и транзисторными. На рис. 3.26 приведены схемы ЭС полупроводниковых ППЗУ и РПЗУ (× — программируемая маскированием перемычка; ~ — плавная перемычка). Наибольшую емкость при наименьшей потребляемой мощности имеют ПЗУ, построенные на *n*-канальных МОП-транзисторах (см. рис. 3.26, б). В масочных ПЗУ (см. рис. 3.26, а... в) информация записывается при изготовлении ПЗУ на заводе с помощью соответствующих фотошаблонов.

На рис. 3.27 приведена схема ПЗУ, состоящего из диодной матрицы и дешифратора адреса. Горизонтальные шины матрицы — адресные, вертикальные — разрядные, с них снимаются восьмиразрядные двоичные числа, записанные в ПЗУ. Если, к примеру, с выхода дешифратора возбуждается адресная шина ША<sub>2</sub>, то логическая «1» проходит с нее через диоды на разрядные шины ШР<sub>2</sub>, ШР<sub>6</sub>, ШР<sub>7</sub> и ШР<sub>8</sub>, сообщая им высокий потенциал, на выход

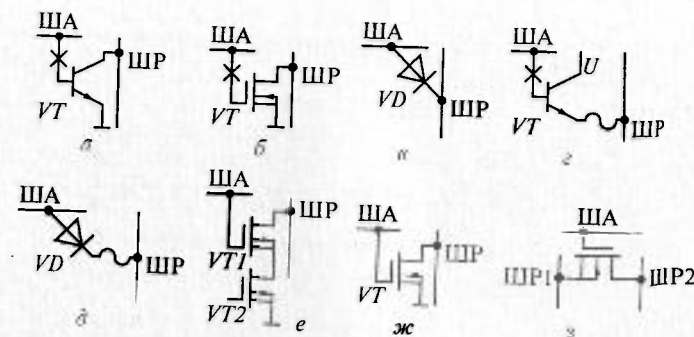


Рис. 3.26. Запоминающие элементы полупроводниковых ППЗУ и РПЗУ: а, б, в — ППЗУ, программируемые маскированием; г, д — ППЗУ, программируемые выжиганием плавких перемычек; е, ж, з — РПЗУ

устанавливается двоичное число 11100010. Аналогично, при возбуждении других адресных шин при поступлении на дешифратор кодов их адреса, на выходах устанавливаются другие двоичные числа из записанных. Диодная матрица (см. рис. 3.27) является совокупностью элементов ИЛИ. Каждый из элементов ИЛИ состоит из диодов, подключенных к одной разрядной шине, а входами элемента являются адресные шины. Элементы ИЛИ, подобные диодным, могут быть выполнены на основе многоэмиттерных транзисторов. В этом случае структура ППЗУ будет выглядеть так, как это представлено на рис. 3.28. При возбуждении одной из шин адреса отпирается транзистор, база которого присоединена к этой шине. Благодаря этому возбуждаются те разрядные шины, к которым присоединены эмиттеры этого транзистора. В интегральном ПЗУ при изготовлении все эмиттеры соединяются с разрядными шинами через плавкие перемычки.

При программировании изготовителем или заказчиком ППЗУ часть плавких перемычек ЭС типа приведенных (см. рис. 3.26, г, д) расплавляется импульсами тока, которые пропускают через определенные эмиттеры согласно карте заказа. В результате нарушают-

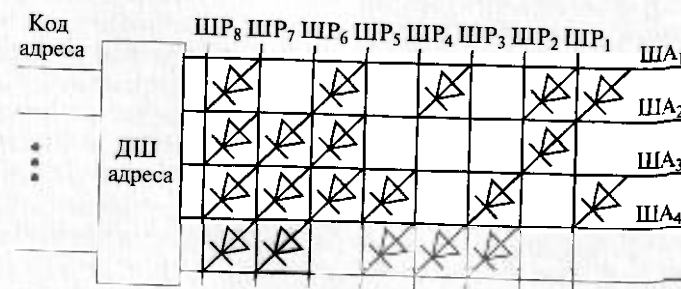


Рис. 3.27. Схема ПЗУ на основе диодной матрицы

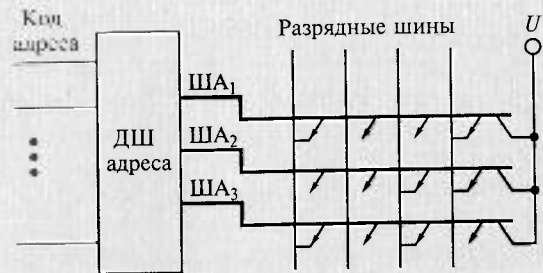


Рис. 3.28. Структура БИС ПЗУ на основе многоэмиттерного транзистора

ся некоторые связи источника питания с разрядными шинами (см. рис. 3.28), что обеспечивает при считывании появление логических «0» в определенных разрядах числа (слова). Занести новую информацию в ПЗУ невозможно, так как нарушенные связи восстановлению не поддаются.

Элементы РПЗУ выполняются двумя способами и подразделяются на ЭС с режимом записи и стирания электрическим сигналом, и ЭС с записью электрическими сигналами и стиранием ультрафиолетовым излучением. Соответственно они выполняются либо на полевых транзисторах со структурой металл, нитрид кремния, окисел, полупроводник (МНОП), либо на МОП-транзисторах с плавающим затвором, нередко называемых МОП-транзисторами с лавинной инжекцией заряда (ЛИЗ). Эти транзисторы под воздействием программирующего напряжения способны запасать электрический заряд под затвором и сохранять его много тысяч часов без подачи напряжения питания. Заряд изменяет пороговое напряжение транзистора: оно становится меньше того значения, которое имеет транзистор под затвором. На этом свойстве основана возможность программирования матрицы РПЗУ. Однако время программирования довольно значительное, что делает практически невозможным использование РПЗУ в качестве ОЗУ.

Для перепрограммирования такого ПЗУ необходимо предварительно стереть имеющуюся информацию. Эту операцию осуществляют по-разному: в РПЗУ на МНОП-транзисторах стирание производит электрический сигнал, который вытесняет накопленный под затвором заряд; в РПЗУ на ЛИЗМОП-транзисторах эту функцию выполняет ультрафиолетовое излучение, которым облучают кристалл через специальное окно в корпусе микросхемы. Запись в РПЗУ не сопровождается никакими необратимыми процессами, поэтому запись информации можно повторять сколько угодно раз. Если диэлектрик выполнен из  $\text{SiO}_2$ , то даже при температуре  $125^\circ\text{C}$  заряд уменьшается на 70% от первоначального значения за 10 лет.

Стирание содержимого РПЗУ осуществляется достаточно просто: поверхность полупроводниковой пластины освещается ульт-

рафиолетовым светом или рентгеновскими лучами. Под их воздействием происходит ионизация и утечка заряда. Время стирания и записи составляет несколько минут. Для БИС с ультрафиолетовым стиранием время стирания длится несколько десятков минут.

Время считывания информации из РПЗУ невелико, но значительно превышает время доступа к оперативной памяти.

### 3.5. Организация безадресной и виртуальной памяти

**Стековая память.** В рассмотренных ранее оперативных ЗУ с адресной организацией (адресных ЗУ) размещение и поиск информации основаны на использовании адреса хранения слова, которым служит номер ячейки памяти. При записи или считывании слова должен указываться адрес, по которому производится запись или чтение. Стековые и ассоциативные ЗУ являются безадресными.

Структуры типа стек используются в целях увеличения быстродействия ЗУ при хранении промежуточной информации и для более рационального использования пространства памяти. В различных областях они имеют следующие синонимы: линейный список, реверсивная память, магазин, линейный список типа *FIFO*, список типа *LIFO* и гнездовая память. Почти все примеры использования стековой организации памяти в ЭВМ относятся к оперативной и сверхоперативной (регистровой) памяти.

В стековой памяти (рис. 3.29, а) ячейки образуют одномерный массив, в котором соседние ячейки связаны друг с другом разрядными цепями передачи слов. Стек заполняется с одной стороны, при этом слова записываются в свободные ячейки с последовательными номерами, начиная с ячейки с адресом 0, а считывание с удалением слова из памяти производится с другой стороны стека и только из нижней ячейки (ячейки 0). При этом все остальные слова в стеке сдвигаются вниз в соседние ячейки с меньшими номерами. Таким образом, порядок считывания подчиняется правилу: первым поступил — первым обслуживается (*FIFO*). Этот стек называется стеком с проталкиванием снизу и больше известен как список очередности. ШИВх — входная информационная шина, ШИВых — выходная информационная шина.

В состав стековой памяти входит счетчик-указатель стека (СчУСт). При записи в стек нового слова содержимое счетчика-указателя увеличивается на 1, при считывании уменьшается на 1. При обнулении счетчика (СчУСт = 0) вырабатывается сигнал УСт = 0, обозначающий, что стек пуст. При СчУСт =  $N - 1$  формируется сигнал УСт =  $N - 1$ , указывающий, что стек заполнен, где  $N$  — размер стека. Стековая память используется в ЭВМ для аппаратной организации различных очередей.

В магазинной памяти (рис. 3.29, б) запись нового слова производится в верхнюю ячейку (ячейку 0), называемую указателем стека или вершиной стека. При этом все ранее записанные слова, включая слово, находившееся в ячейке 0, сдвигаются вниз, в соседние ячейки с большими номерами. Считывание возможно только из верхней (нулевой) ячейки. При этом, если производится считывание с удалением, то все остальные слова в памяти сдвигаются вверх в соседние ячейки с большими номерами. В этой памяти порядок считывания слов соответствует правилу: последним поступил — первым обслуживается (*LIFO*). Вставка и удаление элемента вершины стека называются соответственно операциями проталкивания в стек и выталкивания из стека. В ряде устройств рассматриваемого типа предусматривается также операция простого считывания слова из нулевой ячейки (без его удаления и без сдвига слов в памяти). Память снабжается счетчиком-указателем магазина (*СчУМг*), действие которого аналогично рассмотренному (см. рис. 3.29, а). Магазинная память оказывается весьма эффективной при обработке вложенных структур данных, при проведении вычислений, в частности, она тесно связана с рекурсивными вычислениями. При обнулении  $СчУМг = 0$  вырабатывается сигнал  $УМг = 0$ , обозначающий, что магазин пуст. При  $СчУМг = N - 1$  формируется сигнал  $УМг = N - 1$ .

Если стек организован так, что вставки и удаления элементов можно делать на обоих концах списка, то это — очередь с двумя концами, или двухсторонняя очередь.

Использование ВЗУ не исключает стековой организации памяти и может значительно увеличить эффективность работы вычислительной системы. Например, при работе с накопителями на магнитной ленте (МЛ) при стековой памяти информационные массивы всегда занимают непрерывный отрезок МЛ, что позволяет уменьшить необходимый объем МЛ. Кроме того, блок головок всегда будет находиться в конце информационного участка

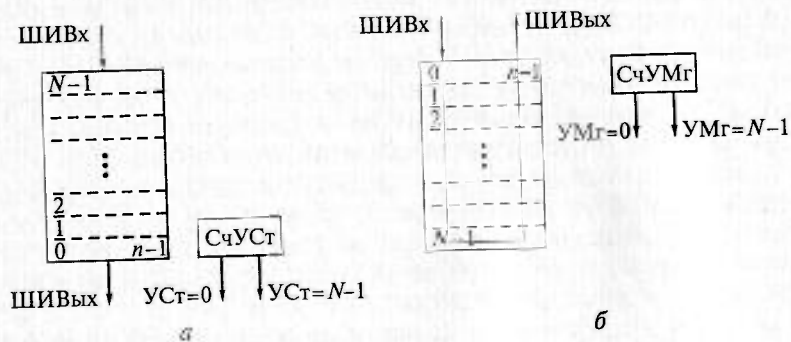


Рис. 3.29. Безадресная память:  
а — стековая (*FIFO*); б — магазинная (*LIFO*)

(стека) и будет готов либо к записи очередного массива, либо к чтению последнего записанного массива, потратив на его поиск минимально возможное время. Почти то же относится и к накопителям на магнитных дисках, так как в этом случае информацией последовательно заполняются цилиндры магнитного диска.

При стековой организации памяти не только рационально используется объем ВЗУ, но и ускоряется вычислительный процесс за счет уменьшения холостых прогонов МЛ и перемещения головок магнитного диска. Особенно это заметно при выполнении вычислений, требующих частого и длительного обращения к ВЗУ.

Еще одно удобство, предоставляемое стековой организацией памяти, проявляется при организации обмена между ОЗУ и ВЗУ (рис. 3.30). В каждый момент времени ЦП обращается к вершине стека, находящейся в ОЗУ. Если при этом стек заполняет почти все ОЗУ, то его «хвост» можно заблаговременно перемещать в ВЗУ; аналогично, при уменьшении объема стека в ОЗУ его можно помещать из ВЗУ. Это можно организовать, не привлекая ЦП, а используя процессор ввода-вывода (ПВВ).

**Ассоциативная память.** Основным фактором, сдерживающим рост производительности ЭВМ традиционной фон-Неймановской структуры, является используемый в них принцип адресации памяти: операнды записываются в память и считываются поодиночке, последовательно, в то время как во многих задачах эти операции гораздо целесообразнее выполнять параллельно, т.е. одновременно над целой совокупностью переменных. Кроме того, с увеличением емкости памяти растет и длина адресов, в результате чего расширяется поле, отведенное в ячейках памяти для записи адресов.

Для обеспечения возможности работы одновременно со многими ячейками памяти и упрощения поиска операндов в 1960-х гг. предлагалось построить весь процесс вычислений на базе адресации по содержанию и ассоциативному признаку. Однако проведенные в течение 30-ти лет работы в данном направлении не привели к созданию подобной вычислительной машины. В настоящее время ассоциативные запоминающие устройства (АЗУ) в основном используются как вспомогательные устройства в некоторых вычислительных системах и позволяют повысить их производительность, обеспечивая процессору более простой доступ к операндам.

В АЗУ поиск нужной информации производится не по адресу, а по ее содержанию, по ассоциативному признаку. При этом по-

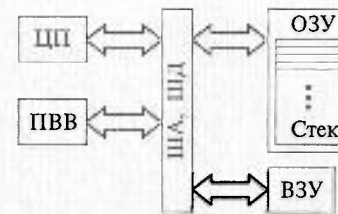


Рис. 3.30. Организация обмена между ВЗУ и ОЗУ посредством стека



иск по ассоциативному признаку или последовательно по отдельным разрядам этого признака может быть осуществлен для всех ячеек запоминающего массива как последовательно, так и параллельно во времени. Во многих случаях ассоциативный поиск позволяет существенно упростить и ускорить обработку данных. Это достигается за счет того, что в памяти этого типа операция считывания совмещена с выполнением ряда логических операций.

Типичная структура АЗУ приведена на рис. 3.31. Запоминающий модуль (ЗМ) содержит  $N(n+1)$ -разрядных ячеек. Для указания занятости ячейки используется служебный  $n$ -й разряд («0» — ячейка свободна, «1» — в ячейке записано слово).

По входной информационной шине данных на регистр ассоциативного признака, иногда называемый регистром аргумента поиска (РгАП), в разряды  $0 \dots (n-1)$  поступает  $n$ -разрядный ассоциативный запрос — слово-признак, характеризующее содержимое требуемой ячейки, а не ее обычный адрес. Затем по этой же шине на регистр маски (РгМаски) поступает код маски поиска, при этом  $n$ -й разряд РгМаски устанавливается в «0».

В общем случае маскирование — логическая операция над байтом, словом или блоком данных, выполняемая с целью модификации или, как в данном случае, с целью выделения их части, а маска — битовая комбинация той же длины, что и выделяемая комбинация. Ассоциативный поиск производится лишь для совокупности разрядов РгАП, которым соответствуют «1» в РгМаски (незамаскированные разряды РгАП). Все  $N$  ЗМ поочередно сравниваются с содержимым РгАП и маской на комбинационной логической схеме (КС). Для слов, в которых цифры в разрядах совпали

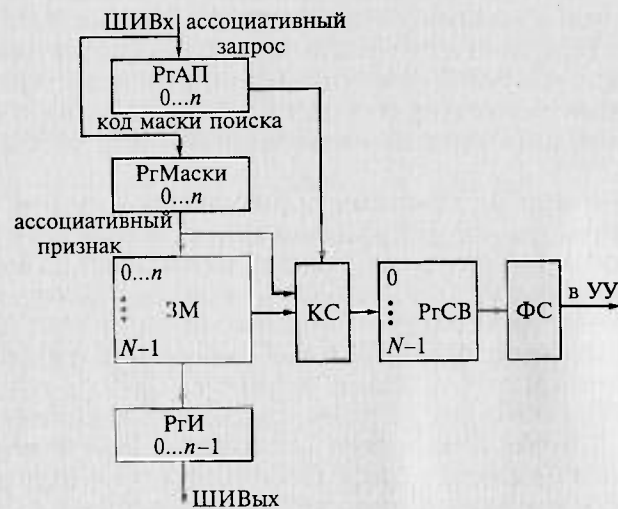


Рис. 3.31. Структурная схема АЗУ

с незамаскированными разрядами РгАП, КС устанавливает «1» в соответствующие разряды  $N$ -разрядного регистра совпадения (РгСв) и «0» в остальные разряды. Таким образом, в РгСв образуется позиционный код, его разряды, содержащие «1», указывают на те ячейки ЗМ, содержимое которых отвечает запрошенному ассоциативному признаку. Позиционный код поступает на вторую комбинационную схему — схему формирования результата проведенного ассоциативного поиска (ФС). Она формирует из слова, образовавшегося в РгСв, одну из трех трехразрядных кодовых комбинаций, соответствующих следующим случаям: отсутствию слов в ЗМ, удовлетворяющих ассоциативному признаку; наличию одного слова; наличию более чем одного такого слова.

Формирование содержимого РгСв и кода на выходе ФС по содержимому РгАП, РгМаски и ЗМ называется операцией контроля ассоциации. Эта операция является составной частью операций считывания и записи, хотя она может иметь и самостоятельное значение. При считывании сначала производится контроль ассоциации по ассоциативному признаку в РгАП. Затем при отсутствии искомой информации считывание отменяется, при наличии одного искомого слова это найденное слово считывается в информационный регистр (РгИ). При наличии более чем одного удовлетворяющего ассоциативному запросу слова в РгИ считывается слово из ячейки, имеющей наименьший номер среди ячеек, отмеченных «1» в РгСв. Из РгИ считанное слово выдается на ШИВых.

При записи сначала отыскивается свободная ячейка. Для этого выполняется операция контроля ассоциации при коде ассоциативного признака  $111 \dots 10$  и коде маски  $000 \dots 01$ . При этом свободные ячейки отмечаются «1» в РгСв. Для записи выбирается свободная ячейка с наименьшим номером. В нее записывается слово, поступившее с ШИВх в РгИ.

#### Пример 3.4

Рассмотрим работу АЗУ на примере продажи микрокалькуляторов в специализированном магазине вычислительной техники. Покупателю предоставляется выбор микрокалькуляторов по четырем признакам:

- 1) область применения — бухгалтерские, инженерные расчеты, словари;
- 2) набор выполняемых операций — количество операций;
- 3) тип источника питания — от сети, аккумуляторной батареи;
- 4) цена.

Информация о калькуляторах хранится в ЗМ (рис. 3.32). Работа по автоматизированному поиску типа калькулятора, удовлетворяющего требованиям покупателя по четырем техническим параметрам в порядке их приоритета, выполняется в следующем порядке.

1. Покупатель вводит информацию о необходимой ему покупке. Это соответствует появлению на входной адресной шине (ШИАВх) и в регистре ассоциативного признака (РГАП) двоичного кода 4-х признаков выбора микрокалькулятора.

2. Покупатель информирует ЭВМ о том, что вначале выбор будет осуществляться только по двум признакам: область применения и источник питания, при этом в РгМаски появляется соответствующий код маски.

3. Производится операция считывания из ЗМ в информационный регистр РгИ всех слов, удовлетворяющих ассоциативному признаку, например поочередное считывание в порядке возрастания цены.

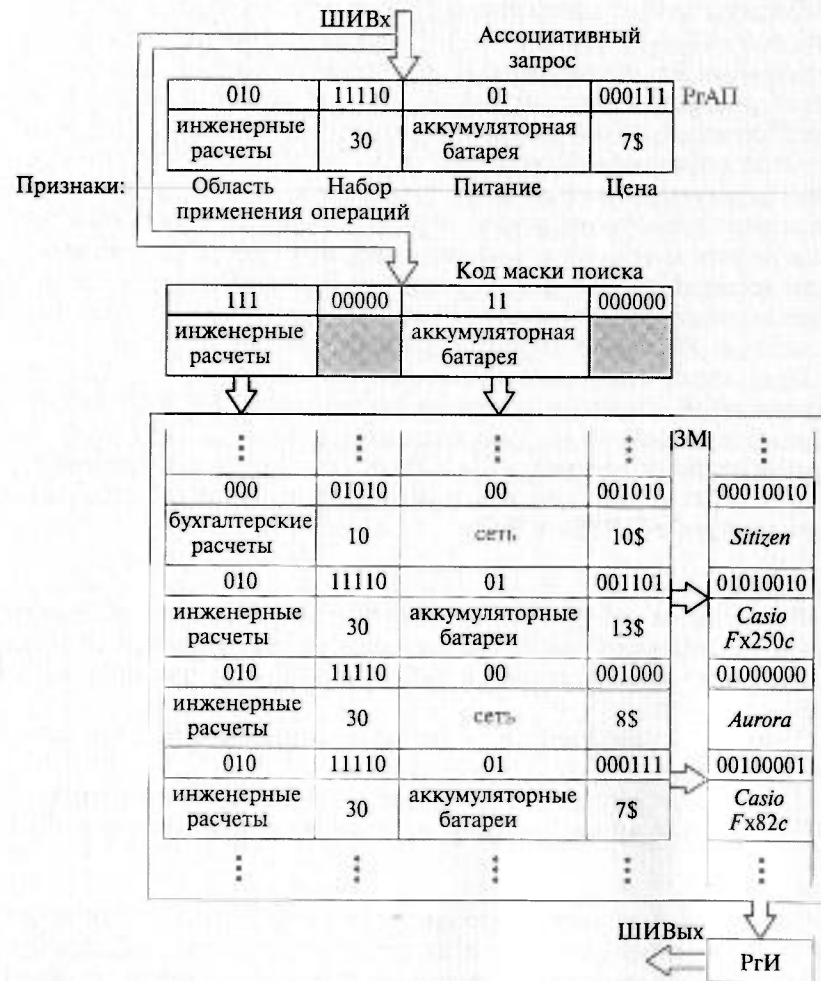


Рис. 3.32. Упрощенная схема ассоциативного поиска

тения цены. В такой последовательности на экран дисплея будет выдаваться информация о тех типах калькуляторов, которые отвечают введенным признакам запроса.

Для реализации АЗУ необходимы запоминающие элементы, допускающие считывание без разрушения записанной в них информации (неразрушающее считывание). Это связано с тем, что при ассоциативном поиске считывание производится по всему ЗМ для всех незамаскированных разрядов и негде сохранять временно разрушаемую считыванием информацию. Такими запоминающими элементами могут быть триггеры на МДП-транзисторах и на bipolarных транзисторах.

Посредством операции контроля ассоциации можно, не считывая слова из памяти, определить по содержимому регистра совпадения, сколько в памяти слов, удовлетворяющих ассоциативному признаку, например, реализовывать запросы типа: сколько студентов в группе имеют отличную оценку по данной дисциплине. При использовании соответствующих комбинационных схем в АЗУ могут выполняться достаточно сложные логические операции, такие как поиск большего (меньшего) числа, поиск слов, заключенных в определенных границах, поиск максимального (минимального) числа и др.

АЗУ могут решать задачи, связанные с обработкой больших массивов информации: накопления, отбора, сортировки и статистического анализа экономической и измерительной информации; динамического программирования; распознавания образов; преобразования кодов; контроля и коррекции ошибок. АЗУ входят в состав специализированных ЭВМ, предназначенных для поиска и упорядочения вывода информации; в универсальных ЭВМ в качестве сверхоперативного ЗУ (СОЗУ), позволяющих значительно ускорить процесс обработки информации. АЗУ может применяться для хранения страничной таблицы виртуальной памяти.

**Виртуальная память со страничной организацией.** Физическая память современных ЭВМ и систем имеет иерархическую многоуровневую структуру. Чем выше уровень, тем больше быстродействие соответствующей памяти, но меньше ее емкость. К верхнему уровню относятся ЗУ, с которыми процессор непосредственно взаимодействует в процессе выполнения программы. Это — оперативная память, относящаяся к наиболее дефицитным ресурсам в ВС, которыми надо пользоваться экономно и эффективно. Быстродействие ОЗУ оказывается недостаточным для обеспечения требований, предъявляемых к скорости работы ЭВМ средней и высокой производительности. Поэтому в состав процессора включают буферную сверхоперативную память небольшого объема и повышенного быстродействия, т. е. СОЗУ, хранящую команды и данные, относящиеся к обрабатываемому участку программы. Проблема усложняется при переходе к системам с распределени-

ем времени (мультипрограммным системам), так как в них ОЗУ одновременно используется несколькими программами. В таких системах необходимо исключить несанкционированное воздействие одних программ на другие, т. е. предотвратить или ограничить обращения одной программы к областям памяти, используемым другими программами. Это достигается посредством механизма защиты памяти.

Эффективное распределение ресурса памяти между программами не может быть статическим, т. е. невозможно провести распределение ресурса предварительно, до пуска программы, так как в процессе обработки данных по программам потребности в ресурсах памяти отдельных программ изменяются, что заранее не может быть учтено. Необходимо распределять память между программами непосредственно в ходе вычислительного процесса, т. е. осуществлять динамическое распределение памяти. При этом должна обеспечиваться независимость работы нескольких программистов над своими программами, подлежащими мультипрограммной обработке. Динамическое распределение памяти не должно приводить к дроблению ее свободного пространства, называемому фрагментацией памяти, затрудняющему его использование. Это достигается на основе одноуровневой виртуальной памяти, допускающей адресацию на все адресное пространство, размер которого определяется полем адресного кода команды.

Гибкое динамическое распределение памяти мультипрограммной ВС, устранение фрагментации памяти и удобство для работы программистов достигаются при организации памяти ЭВМ в виртуальную память.

В виртуальной памяти все рабочее адресное пространство ЭВМ разбито на две части, соответствующие аппаратуре внутренней памяти и более дешевой внешней памяти, а именно на быстродействующую память и медленную внешнюю память. В ЭВМ на базе процессора Intel 80286 адресное пространство составляет 16 Мбайт, а в случае процессора Intel 80386 это пространство составляет уже 4 Гбайт.

Термин «виртуальная память» был введен Деннингом в 1970 г. Необходимость создания виртуальной памяти была обусловлена двумя причинами: необходимостью запуска задач, требующих для своей обработки большого объема оперативной памяти, иногда превышающего размер физической памяти. Без механизма реализации виртуальной памяти решение таких задач не возможно. Второй причиной явилась необходимость надежной изоляции адресных пространств выполняемых в режиме мультипрограммирования задач в целях защиты самой операционной системы и удешевления разработки программного обеспечения. Позже в подтверждение необходимости создания виртуальной памяти было показано, что если при постановке задачи заранее оговаривает-

ся адресное пространство, в котором должно выполняться задание, то это вызывает резкое повышение стоимости программирования. Эти проблемы могли бы решаться применением ОЗУ чрезвычайно больших размеров, но такой подход экономически не выгоден.

Другой подход заключается в использовании операционной системы, которая создает иллюзию чрезвычайно большой оперативной памяти. Поскольку это только иллюзия, такая память называется «виртуальной», кажущейся.

Существуют два основных способа управления виртуальной памятью: управление страничной памятью по запросам и управление сегментной памятью. При страничной организации все адресное пространство делится на страницы, т. е. на установленные «порции» информации для обмена данными между памятью и устройствами передачи данных в системе со страничным обменом. Количество слов или байтов в странице жестко регламентировано для конкретной ВС и почти всегда кратно двум. Обычно страница содержит 4 Кбайт данных. Термин «страница» является синонимом термина страничный «блок», но второй чаще используется, когда речь идет о копии страницы, которая имеется в буфере устройства передачи данных.

При страничной организации памяти по запросам коэффициент использования оперативной памяти может превысить 100 %. Это означает, что сумма адресных пространств заданий, выполняемых в режиме мультипрограммной обработки, может превысить объем физической памяти. Тогда отказываются от требования, чтобы все адресное пространство задания находилось в основной памяти, вместо этого в оперативную память загружается только часть адресного пространства задачи, а оставшуюся часть загружают во внешнюю память. Это возможно осуществить благодаря следующим обстоятельствам:

- многие программы во время прогона взаимно исключают друг друга, например подпрограммы ввода исходной информации, подпрограммы фаз компиляции, подпрограммы выдачи результатов. Даже если в конкретном прогоне участвуют все эти подпрограммы, они могли бы и не находиться в оперативной памяти одновременно;

- некоторые функции программ могут быть взаимоисключающими, например сортировка платежных ведомостей либо по имени, либо по номеру служащих;

- под многие таблицы отводится фиксированное количество адресного пространства, хотя в действительности используется только небольшая часть таблицы. Например, если таблица имен ассемблера рассчитана на 1000 имен, а при конкретном прогоне программы их только 100, то 90 % таблицы не используется в этом прогоне;



- программы обработки ошибок, написанные пользователем, используются только при ошибочных ситуациях.

При страничной организации виртуальной памяти на всех этапах подготовки программы, включая загрузку в ОЗУ, программа представляется в виртуальных (говорят также математических) адресах и лишь при самом исполнении машинной команды производится преобразование, перекодирование виртуальных адресов в адреса действующей памяти, в так называемые физические адреса. Это преобразование составляет содержание динамического распределения памяти.

Преобразование виртуальных адресов в физические упрощается и устраняется фрагментация памяти, если физическую и виртуальную памяти разбить на страницы, содержащие одно и то же количество байт, поэтому каждая физическая страница способна хранить одну из виртуальных страниц. Порядок нумерации байт в обеих страницах сохраняется одним и тем же.

В мультипрограммной системе страничная организация памяти дает определенные преимущества:

- при загрузке новой программы в ОЗУ она может быть автоматически направлена в любые свободные в данный момент физические страницы независимо от того, расположены они подряд или нет, при этом не требуется перемещения информации в остальной части памяти, важно только, чтобы суммарный объем свободной части памяти был достаточен для размещения новой программы, что и приводит к дефрагментации памяти;

- сокращается объем передачи информации между ВЗУ и ОЗУ, так как страница программы не должна загружаться до тех пор, пока она действительно не понадобится, при этом уменьшаются требования к емкости ОЗУ, так как в нем должны содержаться только те участки различных программ, которые действительно работают (или будут работать) в течение ближайшего периода

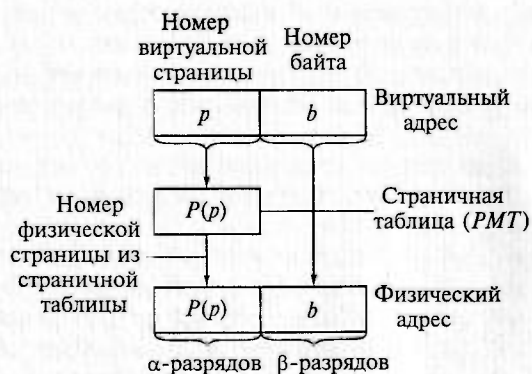


Рис. 3.33. Схема перевода виртуального адреса в физический

времени, неактивные же части программы продолжают оставаться в ВЗУ.

Основной недостаток разбиения на страницы, заключающийся в неэффективном использовании объема памяти, проявляется тогда, когда требуются участки очень маленького объема, поскольку наименьшей единицей распределения памяти является страница.

Правила перевода адресов виртуальных страниц в физические задаются в виде страничной таблицы, показанной на рис. 3.33. Страничная таблица, обозначаемая *PMT* (*page memory table*), указывает соответствие номеров виртуальных страниц  $p$  и физических страниц  $P(p)$ ;  $b$  — номер байта. Страничная таблица формируется супервизором — управляющей программой, называемой также программой-монитором или диспетчером, представляющей собой часть операционной системы, постоянно находящуюся в ОЗУ и непосредственно имеющую дело с аппаратной частью ЭВМ или ВС. Формирование *PMT* производится в процессе распределения памяти, и содержание таблицы перерабатывается супервизором каждый раз, когда в распределении памяти проводится изменение.

На рис. 3.34 приведена схема устанавливаемого с помощью *PMT* соответствия между виртуальными и физическими адресами при страничной организации памяти; на рис. 3.35 показан пример реализации такой схемы для случая 32-разрядной ЭВМ. Здесь показано распределение памяти только для одной программы. В вычислительной системе с разделением времени обрабатывается сразу несколько программ (рис. 3.36). Кодирование поля состояния страничной таблицы таково:  $Y$  — данная страница находится в опера-

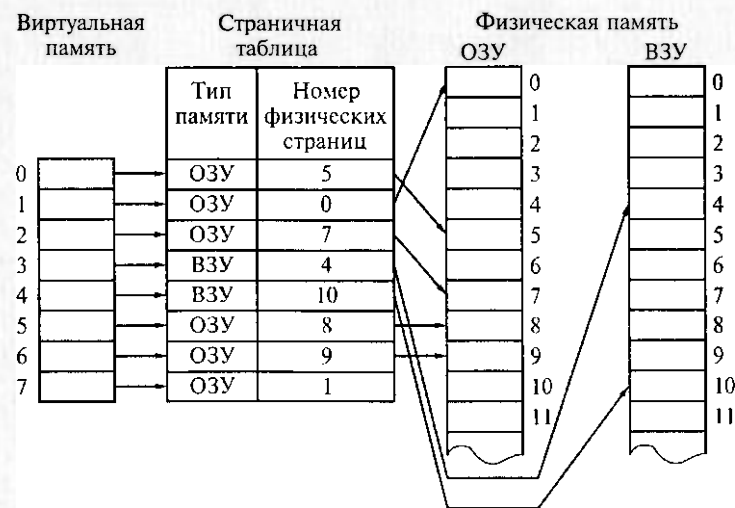


Рис. 3.34. Страничная организация виртуальной памяти

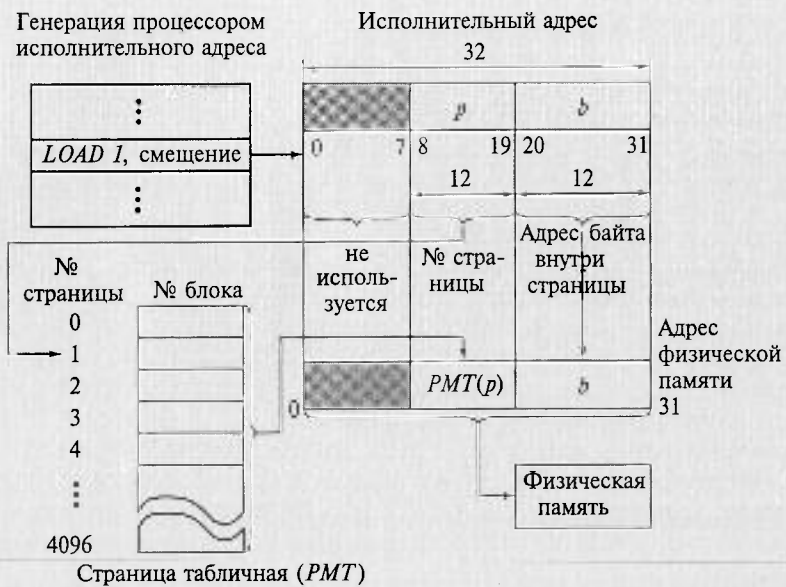


Рис. 3.35. Пример схемы вычисления адреса физической памяти по исполнительному адресу

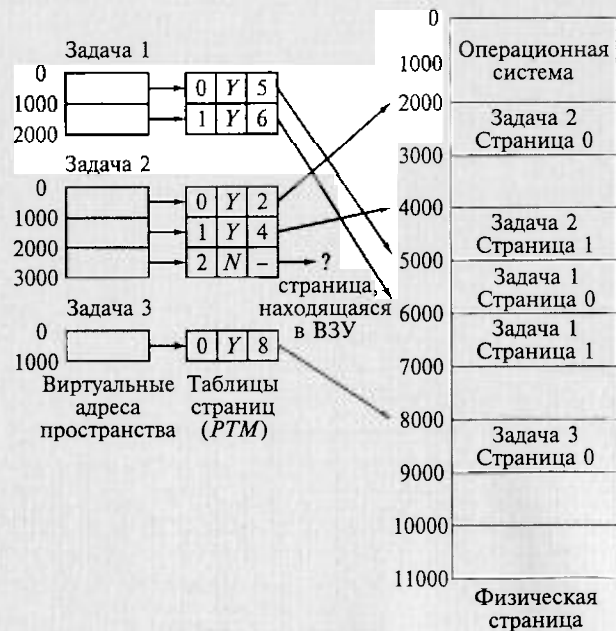


Рис. 3.36. Структура виртуальной памяти при страничном распределении по запросам

тивной памяти,  $N$  — страница отсутствует в ОЗУ.  $PMT$  во избежание наложения программ друг на друга должна учитывать также принадлежность страницы различным пользователям, а номер физической страницы  $P$  может быть представлен как функция двух переменных: номера виртуальной страницы  $p$  и номера программы.

Страницы, к которым редко обращались в процессе вычислений, так называемые неактивные страницы, должны быть удалены из ОЗУ и записаны во внешнее ЗУ, например, на магнитный диск. Стоимость хранения на магнитносителе на один или два порядка дешевле, чем в полупроводниковой памяти. Однако размер объема виртуальной памяти ограничен размером памяти ВЗУ и, кроме того, скорость доступа в ВЗУ относительно низка.

При страничной организации памяти необходимо выполнять следующие функции.

1. Отслеживание состояния памяти с помощью таблиц:

- таблиц страниц, по одной на каждую задачу;
- таблиц блоков памяти, одной на систему;
- карт файлов — по одной на каждое адресное пространство.

2. Определение, кто и когда получает доступ к памяти. Это частично определяется программой — планировщиком заданий, отвечающей за управление распределением совместного ресурса ЭВМ между коллективом пользователей; в динамике это определяется также прерыванием по запросу страниц.

3. Распределение. При возникновении необходимости в выделении блока программы должен быть найден свободный блок памяти и должно быть изменено состояние этого блока в таблицах.

4. Освобождение. Если при необходимости выделения памяти свободный блок не найден, должен быть освобожден и перераспределен один из уже выделенных блоков памяти. При завершении задания все используемые заданием блоки освобождаются.

Операция преобразования адресов может быть реализована и аппаратно. Если  $PMT$  хранится в ОЗУ, то выборка каждого слова требует двух машинных циклов обращения к памяти вместо одного: лишний цикл необходим для обращения к  $PMT$ . Для сокращения потерь времени при этом  $PMT$  хранят в СОЗУ небольшой емкости. Поскольку сравнительно небольшое число физических страниц памяти активно используется в течение определенного промежутка времени, целесообразно иметь небольшое ассоциативное запоминающее устройство, содержащее данные только о недавно использовавшихся физических страницах. Вся остальная информация по преобразованию адресов может храниться в обычной памяти или в СОЗУ в виде  $PMT$ , построенной по виртуальным страницам (рис. 3.37). Наиболее простой случай — когда информация о запрашиваемой странице находится в АЗУ. В результате ассоциативного поиска находится регистр (на рисунке это третий регистр), хранящий нужную информацию, и из него номер физической



Рис. 3.37. Пример организации работы ассоциативной памяти страничных таблиц

страницы передается в адресный регистр ОЗУ, а его разряд использования устанавливается в «1».

Случаи, когда в АЗУ нет информации о странице (неактивная страница) и в АЗУ имеется или отсутствует свободный регистр, показаны на рис. 3.38 и 3.39.

Другой способ хранения неактивных страниц — существование этих страниц не в ВЗУ, а в каком-нибудь параллельном адресном пространстве, зарезервированном в полупроводниковой памяти. Этот способ обладает гораздо более высокой скоростью переключения страниц. В персональных компьютерах используется хранение неактивных страниц в ОЗУ. На рис. 3.40 показана спецификация отображаемой памяти EMS (Expanded Memory Specification), иногда обозначаемой LIM, т.е. точное описание действий, которые необходимо выполнять для отображения неактивных страниц в полупроводниковой памяти. Спецификация EMS базируется на переключении блоков дополнительной памяти в зарезервированную область адресного пространства. EMS-метод позволяет увеличивать объем оперативной памяти при ограничении адресного пространства со стороны процессора, при этом задействуются все линии шины адреса.

Отметим достоинства способа динамического распределения памяти страницами по запросам. За счет того, что адресные пространства заданий не ограничиваются объемом физической памяти, можно получить большой объем виртуальной памяти. Благодаря



Рис. 3.38. Функционирование АЗУ PMT при отсутствии в нем информации о странице и наличии свободного регистра



Рис. 3.39. Функционирование АЗУ PMT при отсутствии в нем информации о странице и отсутствии свободного регистра



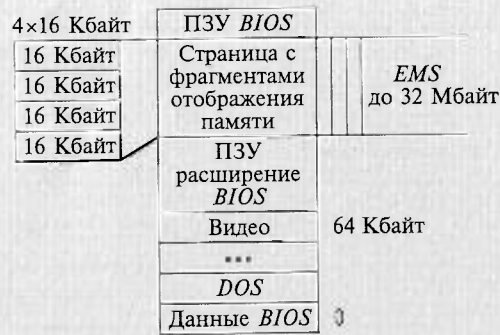


Рис. 3.40. Спецификация отображаемой памяти EMS (LIM)

этому достигается лучшая совместимость больших и малых вычислительных машин при их объединении в сети. Как показали исследования, не менее 25 % адресных пространств заданий не загружается при конкретном прогоне. При распределении страниц по запросам те части адресных пространств, которые не используются или используются редко, вообще не будут загружены во внутреннюю память, чем и достигается более эффективное ее использование. Кроме того, поскольку сумма всех адресных пространств заданий может превышать объем физической памяти за счет использования ВЗУ, достигается практически неограниченное мультипрограммирование, позволяющее одновременно работать за терминалами универсальной ЭВМ высокой производительности до 280 пользователей.

К недостаткам распределения страницами по запросам относятся следующие. Объем таблиц и накладные расходы процессора, вызываемые необходимостью обработки страничных прерываний, возрастают по сравнению со схемой простого страничного распределения памяти, когда сумма всех адресных пространств заданий не превышает объема физической памяти.

В связи с отсутствием явных ограничений, налагаемых на размер адресного пространства заданий, необходимого при заданном уровне мультипрограммирования, требуются дополнительные средства, исключающие возможность «пробуксовки» системы, когда часть времени ЦП тратится на обработку страничного прерывания, в то время как на выполнение заданий пользователя тратится небольшая часть времени.

#### Структура виртуальной памяти при сегментном распределении.

Сегментное распределение обеспечивает большее удобство при программировании и повышение эффективности использования памяти особенно в случае программ, состоящих из нескольких массивов — подпрограмм, одной или нескольких секций данных (X, A, B на рис. 3.41).

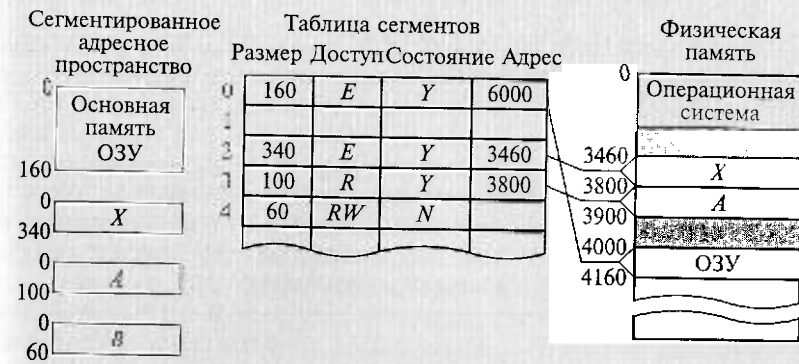


Рис. 3.41. Отображение сегментного адресного пространства

От программиста не требуется усилий по объединению различных частей программы в единый массив. Виртуальная память каждой программы подразделяется на части, именуемые сегментами, с независимой адресацией байт внутри каждой части.

Сегмент определяется как логическая группа информации, такая как программа, массив или область данных (см. рис. 3.41). Здесь в поле доступа заносится: E — выполнение разрешено; R — чтение разрешено; W — запись разрешена. Состояние может иметь два значения Y (Yes) или N (No).

Для поддержки сегментного распределения необходимы следующие таблицы:

- сегментов, по одной на адресное пространство;
- перераспределения области, одна на систему;
- активных ссылок, по одной на адресное пространство;
- активных сегментов, одна на систему.

Последние два типа таблиц (см. рис. 3.41) не показаны. На рис. 3.42 приведено формирование сегментного адреса для 64-разрядного слова ЭВМ средней производительности.

При сегментной организации виртуальной памяти удается устранить фрагментацию путем сдвига сегментов в памяти и объединения в одну свободную область фрагментов свободной основной памяти. Появляется также возможность динамического расширения сегментов и автоматического контроля их границ, а также динамического связывания (установления связей между сегментами программы) и загрузки. Сегментное распределение так-

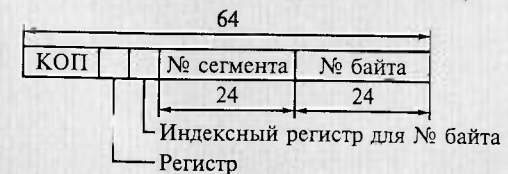


Рис. 3.42. Формирование сегментного адреса

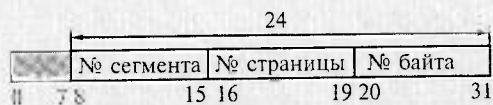


Рис. 3.43. Формирование исполнительного 24-битового адреса

же облегчает совместное использование сегментов области данных и процедур и дает возможность контролируемого доступа, но приводит к значительным накладным расходам, вызываемым переконфигурацией памяти для обеспечения динамического расширения сегментов и устранения фрагментации.

Вызывает трудности и управление сегментами переменной длины во внешней памяти. Необходимы затраты и на сокращение «пробуксовки» сегментов. Хотя страничная и сегментная организация памяти в логическом отношении тесно связаны друг с другом и похожи по реализации, достигаемые с их помощью цели различны.

Пользователь может не знать о существовании в машине страничной организации памяти. Целью сегментной организации является создание удобств для пользователя, однако он должен не только знать о существовании сегментов, но и отмечать каждую независимую часть программы присвоением соответствующего номера сегмента.

Сегментно-страничная память объединяет в себе оба рассмотренных способа организации виртуальной памяти. Сегмент представляется не как нечто единое и непрерывное целое, а как сово-

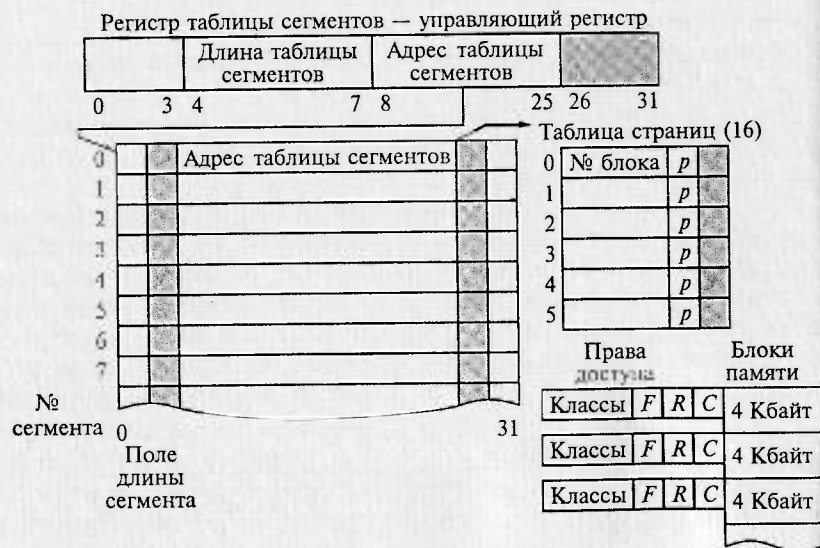


Рис. 3.44. Аппаратная реализация сегментно-страничной памяти

купность страниц. В этом случае разрешаются проблемы переконфигурации, управления внешней памятью и устраняются ограничения на размер сегмента. Возникает определенная иерархия в организации программ, состоящая из четырех ступеней: программа, сегмент, страница, байт.

Этой иерархии программ соответствует иерархия таблиц, служащих для перевода виртуальных адресов в физические. На рис. 3.43 показано формирование исполнительного адреса, а на рис. 3.44 — схема аппаратной реализации сегментно-страничной памяти.

### Контрольные вопросы

1. Укажите особенности устройств комбинационной логики и конечных автоматов в отношении формирования выходного сигнала.
2. Чем отличается реализация логических операций в комбинационных устройствах и цифровых автоматах?
3. Поясните особенности функционирования цифрового сумматора.
4. Изложите принципы работы шифратора и дешифратора.
5. Перечислите разновидности цифровых триггеров.
6. Изложите принципы работы асинхронных и синхронных RS-триггеров.
7. Поясните принципы работы T- и D-триггеров.
8. Изложите особенности работы универсального JK-триггера.
9. Расскажите о принципах функционирования регистров памяти цифровых устройств.
10. Поясните принцип функционирования трехразрядного счетчика импульсов.
11. Спроектируйте двоичный счетчик, осуществляющий счет до 5.
12. Поясните принципы организации ОЗУ.
13. Поясните принципы организации ПЗУ.
14. Назовите два типа безадресных ЗУ. В чем их достоинства и в каких случаях они применяются?
15. Назовите отличия в организации стеков типа LIFO и FIFO.
16. По какому признаку производится поиск информации в АЗУ?
17. Перечислите, какие операции, помимо ассоциативного поиска, могут выполняться в АЗУ.
18. В чем заключается принцип динамического распределения (перераспределения) памяти и в каких ВС он применяется?
19. На какие две части разбито все адресное пространство виртуальной памяти?
20. Какие две цели достигаются при виртуальной организации памяти ВС? Назовите два основных способа управления виртуальной памятью.
21. Как поступают при организации виртуальной памяти, когда потребность программ в оперативной памяти превышает 100% ее объема?
22. Перечислите достоинства и недостатки способа динамического распределения памяти страницами по запросам.
23. В чем состоит различие целей, достигаемых при страничной и сегментной организации памяти?

## СТРУКТУРА И ФУНКЦИОНИРОВАНИЕ ПРОЦЕССОРА

### 4.1. Классификация и типовая структура микропроцессоров

Все микропроцессоры можно разделить на три группы:

- микропроцессоры с полным набором команд типа — CISC (Complete Instruction Set Computing);
- микропроцессоры с сокращенным набором команд — RISC (Reduced Instruction Set Computing);
- микропроцессоры с минимальным набором команд — MISC (Minimum Instruction Set Computing), находящиеся в стадии разработки.

Микропроцессоры типа CISC используются в большинстве современных IBM PC — совместимых персональных компьютеров (табл. 4.1).

Микропроцессоры типа RISC содержат набор только простых, чаще всего встречающихся в программах команд. При необходимости выполнения более сложных команд в микропроцессоре производится их автоматическая сборка из простых команд. У RISC-процессоров получается регулярная структура простых команд, поэтому комбинация из нескольких команд выполняется быстрее, чем одна, эквивалентная этой комбинации сложная команда. Такие микропроцессоры работают быстрее, команда выполняется за один машинный цикл, тогда как в других машинах — за несколько машинных циклов. Большинство операций в RISC-процессорах — типа «регистр—регистр», а обращения к основной памяти происходят только для выполнения простых операций загрузки в регистр и занесения в память. Уменьшается число логических вентилях и объем микропрограммного ПЗУ, необходимых для декодирования и реализации сложных команд, поэтому размеры МП с RISC-архитектурой меньше и соответственно стоимость его тоже ниже. Так, RISC-процессору, состоящему из 50 000 транзисторов, соответствует 32-разрядный МП традиционной архитектуры на 150 000 транзисторов. Цикл разработки RISC-процессора короче, чем у МП традиционной архитектуры.

Характеристики CISC микропроцессоров фирмы Intel

Модель МП	Разрядность, бит		Тактовая частота, МГц	Адресное пространство, байт	Число команд	Число элементов	Год выпуска
	данных	адреса					
4004	4	4	4,77	$4 \times 10^3$	45	2300	1971
8080	8	8	4,77	$64 \times 10^3$	—	10 000	1974
8086	16	16	4,77 и 8	$10^6$	134	70 000	1982
80186	16	20	8 и 10	$10^6$	—	140 000	1984
80286	16	24	10... 33	$4 \times 10^6$ виртуальное $10^9$	—	—	—
80386	32	32	25... 50	$16 \times 10^6$ виртуальное $4 \times 10^9$	240	275 000	1987
80486	32	32	33... 100	$16 \times 10^6$ виртуальное $4 \times 10^9$	240	$1,2 \times 10^6$	1989
Pentium	64	32	50... 150	$4 \times 10^9$	240	$3,1 \times 10^6$	1993
Pentium Pro	64	32	66... 200	$4 \times 10^9$	240	$5,5 \times 10^6$	1995
Pentium II	64	32	233... 450	$4 \times 10^9$	—	$7,5 \times 10^6$	1997
Pentium III	64	32	550... 1200	$4 \times 10^9$	—	—	1999

**Устройство и принцип функционирования микропроцессора.** Работу микропроцессора рассмотрим на примере типового МП-устройства (рис. 4.1), обеспечивающего цифровую обработку многоуровневой двоичной информации. Для удобства анализа примем за основу простейшую 8-разрядную конструкцию. Каждый МП, реализующий программный способ обработки цифровой информации, должен располагать заданным составом команд, которые он может выполнять. Эти команды хранятся в специальной области памяти в виде цифровых кодов. Считывание и выполнение команд осуществляется в МП по тактовым импульсам высокочастотного генератора сигналов. Информация внутри МП пересылается по специальным многопроводным шинам (магистралям), которые имеют связь с внешними устройствами через выводы корпуса МП. Таких шин в МП три: шина данных, адреса и управления.

Для функционирования МП управляющая программа должна задавать очередную команду указанием ее адреса в памяти. Тогда



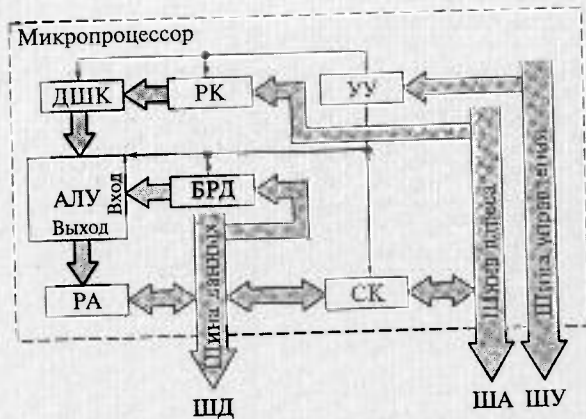


Рис. 4.1. Функциональная схема 8-разрядного микропроцессора

код этой команды, извлекаемый из памяти, по шине адреса пересылается в регистр команд (РК) и записывается в нем. Далее дешифратор команд (ДШК) его расшифровывает и пересылает в АЛУ для организации соответствующей операции над операндами (сложение, вычитание, сдвиг, сравнение и т.п.). Подлежащие обработке данные поступают на вход АЛУ через буферный регистр данных (БРД), обеспечивающий временное хранение введенной информации.

Полученный результат через выход АЛУ поступает в устройство оперативного хранения, представленное регистром-аккумулятором (РА). Из этого регистра данные по ШД поступают к выходу МП. Заметим, что в реальных микропроцессорах РА является элементом блока регистров, называемых регистрами общего назначения (РОН). Назначение РОН — создание внутренней сверхоперативной памяти МП, в которой можно временно хранить операнды и результат, участвующие непосредственно в вычислении.

Взаимосвязанная работа всех устройств МП обеспечивается УУ, снабженным автономной шиной управления. Устройство управления совместно с тактовым генератором обеспечивает требуемую последовательность работы элементов МП на всех этапах обработки информации (извлечение команды из памяти, ее дешифровка, пересылка данных в АЛУ, реализация текущей операции и т.п.). Особая роль в этом процессе отводится счетчику команд (СК), соединенному с шиной данных и обеспечивающему совместно с УУ контроль за исполнением текущей команды. Одновременно СК организует периодическую подготовку МП к выполнению каждой последующей команды. Для этого СК имеет непосредственную связь с шиной адреса и, получив адрес очередной команды, фиксирует начало процесса ее выполнения. С этого момента СК изменяет свое значение, формируя адрес следующей команды, и т.д.

Реальные МП содержат и другие элементы, например регистр состояния и стековую память — для обеспечения программных переходов и т.д. Как видно, в микропроцессорах стремятся использовать повышенное число регистров, что способствует улучшению быстродействия МП и расширению областей их применения.

#### 4.2. Микропроцессоры с «жестким» и программируемым принципами управления

**Устройство управления с программируемой логикой.** Микропрограммные УУ, называемые в технической литературе как УУ с программируемой логикой, с хранимой в памяти логикой, появились позже, чем УУ с «жесткой» логикой. Но, поскольку микропрограммное УУ обладает наглядностью, облегчающей изучение процесса функционирования ЭВМ, рассмотрим его в первую очередь.

Схему микропрограммного управления предложил в 1851 г. Уилкс. Она изображена на рис. 4.2. Управляющие сигналы  $V_i$ , соответствующие каждой своей микрооперации, по одному или по несколько будут поступать в собственно *операционный блок* (АЛУ, ПЗУ, ОЗУ) с управляющей матрицы С. Матрица С — управляющая и матрица S — память микропрограмм представляют собой ПЗУ. В простейшем случае ПЗУ реализуется как горизонтальные и вертикальные шины, связанные между собой в нужных точках через диоды или перемычки, прожигаемые при программировании в процессе изготовления микропроцессорной БИС. Чтобы не загромождать рисунок, элементы связи между горизонтальными и вертикальными шинами ПЗУ — диоды или перемычки — показаны условно в виде жирных точек. Вертикальные выходные шины матрицы S являются линиями управляющей шины операционного блока. Каждая линия соответствует одному из управляющих сигналов микроопераций  $V_1, V_2, \dots, V_i, \dots, V_m$ . Появление сигналов на шине инициирует выполнение соответствующей микрооперации в АЛУ.

Вертикальные шины матрицы S устанавливают определенный код в регистре номера (адреса) микрокоманды (РгНМК). В момент поступления синхросигнала дешифратор (Дш) в соответствии с одноразрядным кодом на РгНМК возбуждает одну из  $2^1$  горизонтальных шин, это шина номер 3 (см. рис. 4.2, она показана жирной чертой). Когда горизонтальная шина возбуждена, она, в свою очередь, возбуждает отмеченную точкой (или отмеченные точками) вертикальную (вертикальные) шину управления матрицы С, задавая таким образом последовательность управляющих сигналов, соответствующих набору микроопераций, выполняемых



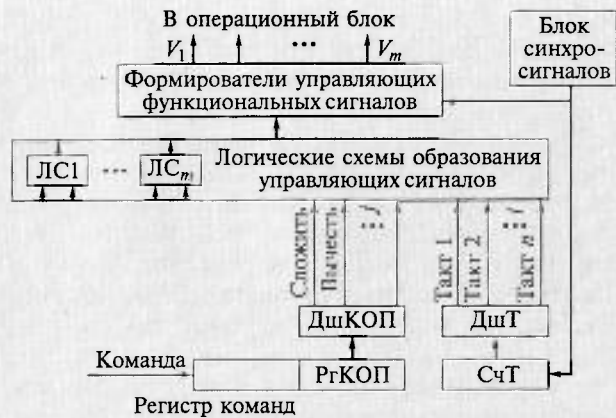


Рис. 4.3. Структура УУ с «жесткой» логикой

На СчТ поступают сигналы от блока синхросигналов, и счетчик с каждым сигналом меняет свое состояние. Состояния счетчика представляют номера тактов, изменяющиеся от 1 до  $n$ . ДшТ формирует на  $i$ -м выходе единичный сигнал при  $i$ -м состоянии счетчика, т. е. во время  $i$ -го такта.

ДшКОП вырабатывает единичный сигнал на  $j$ -м выходе, если должна исполняться  $j$ -я команда. Если на ЛС поступают две логические «1» с ДшКОП и ДшТ, образуется управляющий сигнал.

ЛС образования управляющих сигналов для каждой команды возбуждают формирователи управляющих функциональных сигналов для выполнения требуемых в данном такте микроопераций,  $m$  — число логических схем (см. рис. 4.3).

Однокристалльные наборы МП серий K536, K580, K581, K586, K588, K1801 и K1810 имеют УУ с жесткой логикой управления операционным блоком.

### 4.3. Рабочий цикл процессора. Микропрограммная интерпретация команд центрального процессора

Один из многих параметров, характеризующих ВС, рабочий цикл процессора иначе называется циклом центрального процессора либо машинным циклом. Это время, требуемое для выбора из памяти и выполнения одной простой машинной команды, например, сложения и вычитания. На рис. 4.4, а оно обозначено  $T_{\text{ц}}$ . По команде программы в АЛУ центрального процессора выполняются арифметические и логические действия над операндами операции. Выполнение операции производится по частям, называемым микрооперациями, — элементарным операциям, выполняемым за один машинный такт  $T_{\text{т}}$ . Таким образом, каждой микрооперации соответствует выполняемое в АЛУ элементарное действие передачи информации между регистрами общего назначения (РОН) или ее преобразования. Это действие инициируется (вызывается) поступлением управляющего сигнала  $V$  на внутреннюю управляющую шину. Для реализации команды необходимо на соответствующие линии ШУ подать распределенную во времени последовательность управляющих сигналов  $V_i, V_{i+1}, V_{i+2}, \dots$ . Их вырабатывает блок УУ.

Последовательность микрокоманд, необходимых для выполнения операции, называется микропрограммой. Микропрограммная интерпретация микрокоманд иллюстрируется примером (рис. 4.4, б).

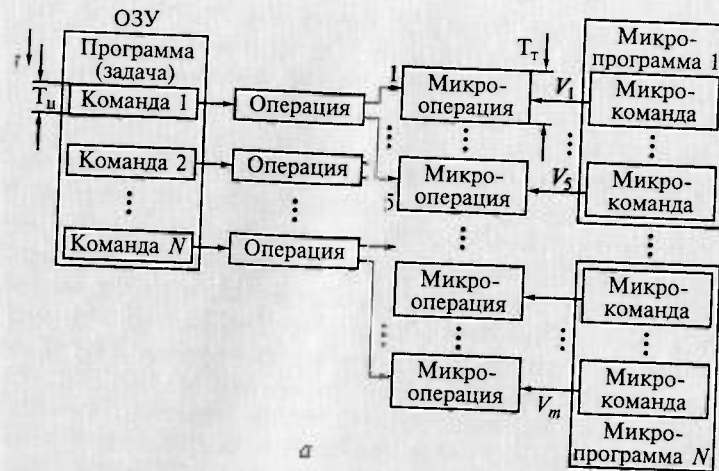
#### Пример 4.1

Выполнение команды операции сложения. К управляющим шинам АЛУ необходимо подвести следующую последовательность управляющих сигналов:

Номера тактов  $T_{\text{т}}$       Микрооперации и названия управляющих сигналов

- 1
- 2

Прием Рг 1  
Прием Рг 2



Сложить	Рг 1	Рг 2	Рг Сумм
Код операции	Адрес 1-го операнда	Адрес 2-го операнда	Адрес результата
Операционная часть		Адресная часть	

Рис. 4.4. Микропрограммная интерпретация команд и микрокоманд: а — распределение во времени выполнения команд и микрокоманд; б — структура трехадресной команды



3	Прием Rг Сумм или Прием Rг Сумм + 1 Смещения
4	Прием Rг Аккумулятор
5	Прием шина данных

Таким образом, УУ вырабатывает управляющие сигналы для осуществления выборки кода очередной команды из памяти, дешифрации кода команды, нормирования адресов операндов, выборки операндов из памяти, передачи их в АЛУ, выполнения в АЛУ операции, предусмотренной кодом команды, передачи результата в память. Последовательности управляющих сигналов  $V$  вызывают выполнение в процессоре микроопераций. В процессорах малых и средних ЭВМ часто применяют микропрограммное УУ. В больших же машинах применяется управление с «жесткой» логикой, т.е. аппаратно реализованной, что приводит к повышению быстродействия ЭВМ. В МП наборах разных типов применяются оба способа выработки управляющих сигналов.

**Слово состояния процессора.** Для нормального функционирования МПС недостаточно управляющих сигналов  $V_i$ , генерируемых управляющим устройством МП или поступающих из ОЗУ и УВВ, таких как Сброс, Синхронизация, Готовность или Ожидание, Запрос прямого доступа к памяти (ПДП) и Разрешение ПДП, Управление прерыванием, Управление шиной — Чтение, Запись и др. МПС в каждом машинном цикле должна получать более полную информацию о состоянии МП. В условиях «узкого» интерфейса, когда внешних выводов для индикации внутреннего состояния (слова состояния СС) МП недостаточно, эта задача решается с использованием мультиплексирования шины данных и представления внутреннего состояния МП на внешнем по отношению к МП регистре слова состояния (RгСС). Слово состояния процессора полностью описывает состояние процессора в каждый момент и указывает, какие классы операций разрешаются и какие запрещаются, а также состояние всех прерываний. Это слово содержит адрес команды, выполняемой в текущий момент, а в ряде случаев еще и адрес слова, в котором хранится очередная подлежащая выполнению команда.

Процессор в первом такте каждого машинного цикла генерирует на шине данных слово состояния, которое содержит информацию о процессах, происходящих в МП. На рис. 4.5 приведена схема присоединения RгСС к шине данных МП. Сигнал синхронизации Синхро вырабатывается в МП в начале каждого машинного цикла и используется в качестве сигнала, идентифицирующего информацию, представленную в шине данных, как слово состояния. Слово состояния загружается во внешний регистр RгСС под воздействием сигнала Синхро. Сигналы, которые показаны на выходе в данном случае 8-разрядного RгСС ( $D_7 \dots D_0$ ), исполь-

зуются в качестве управления периферийными устройствами МПС и ОЗУ. Если при реализации любой команды первый машинный цикл есть цикл выборки команды, то машинные циклы во время выполнения собственно команды могут следовать в достаточно произвольном порядке, что определяется кодом команды. Всего, например, МП К580 имеет 10 типов машинного цикла и соответственно 10 кодов слова состояния, идентифицирующих эти циклы (см. рис. 4.5). Каждый разряд слова состояния заводится на соответствующие управляющие входы адаптеров или схем сопряжения с УВВ, определяя тем самым их режим функционирования в соответствии с данным текущим состоянием МП.

В условиях узкого интерфейса управление МПС осуществляется генерацией управляющих воздействий на двух уровнях:

- 1) на уровне управляющих сигналов  $V_i$  микрокоманд по шине управления ШУ собственно МП в каждом такте работы УУ  $T_T$ ;
- 2) на уровне команд путем генерации слова состояния в каждом машинном цикле  $T_{Ц}$ .

Выходы RгСС и управляющие линии корпуса МП БИС образуют шину управления ШУ МПС. Для МП К580, например, 12 линий

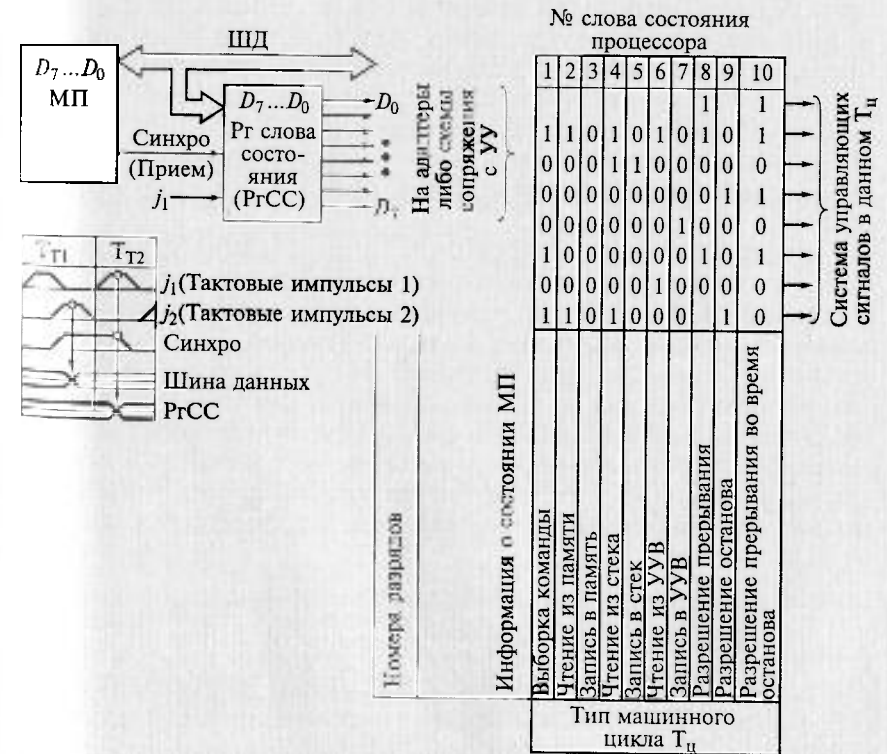


Рис. 4.5. Слово состояния процессора типа К580

шины системного управления обеспечивают возможность работы МПС со сложным многофункциональным периферийным оборудованием. При этом использование временного мультиплексирования ШД для вывода на внешний регистр состояния сигналов управления МПС снижает общую производительность системы.

#### 4.4. Микроконтроллеры. Особенности организации однокристалльных и секционных микропроцессоров

**Микроконтроллеры.** Для дешифрации слова состояния МП и генерации ряда управляющих сигналов может быть применен микроконтроллер, состоящий из минимального количества компонентов микропроцессорного набора (рис. 4.6).

*Микроконтроллер* — управляющее устройство, выполненное на одном или нескольких кристаллах, выполняющее функции логического анализа и управления. За счет исключения арифметических операций аппаратная сложность микроконтроллеров по сравнению с МП ниже, а функции логического управления более развиты. Микроконтроллеры применяются не только в составе микроЭВМ как устройства управления, но и в системах передачи данных, системах управления технологическими процессами. Микроконтроллер имеет сравнительно небольшую разрядность слова и богатый набор команд манипулирования отдельными битами, но не способен реализовывать некоторые арифметические и строковые операции, характерные для универсальных микропроцессоров.

Программируемые микроконтроллеры лишены недостатков схем управления с жесткой логикой, с жесткими электрическими связями и обеспечивают изменение программы функционирования с помощью встроенного или автономного устройства программирования, что предопределило их использование и для создания современных автоматизированных систем управления (АСУ) технологическими процессами. При этом отпадает необходимость в отключении технологического оборудования и исключаются трудоемкие и дорогостоящие процессы, связанные с перепроектированием систем управления. Программируемые кон-



Рис. 4.6. Структурная схема микроконтроллера

троллеры кроме функций управления могут выполнять и функции регулирования.

Микропроцессорные контроллеры обладают высокой степенью автономности и могут продолжать работу при отказе центральной ЭВМ и линий связи, что повышает живучесть системы управления, значительно снижает общую ее стоимость, габариты и потребляемую мощность.

**Особенности организации однокристалльных и секционных микропроцессоров.** При проектировании микропроцессоров решают задачу определения наборов команд, выполняемых программным или аппаратным способом на основе заданной системы микрокоманд. Аппаратная реализация сложных команд в УУ с «жесткой» логикой дает возможность увеличить быстродействие микропроцессора, но требует значительных аппаратных ресурсов кристалла БИС МП. Программная реализация сложных команд в УУ с программируемой логикой позволяет обеспечивать программирование сложных задач, изменять количество и особенности исполнения сложных команд. Однако скорость исполнения программных команд ниже скорости исполнения аппаратно реализованных команд.

Практически во всех современных МП системах используются сложные развитые системы команд. Их ядро, состоящее из набора универсальных команд, реализуется аппаратным способом в центральном МП. Кроме того, специализированные части наборов системы команд реализуются вспомогательными или периферийными микропроцессорами МПС. Эти специальные арифметические или логические микропроцессоры расширяют возможности обработки данных и позволяют ускорить выполнение определенных команд и тем самым сократить время исполнения программ.

По числу БИС в микропроцессорном комплекте различают МП однокристалльные, многокристалльные и многокристалльные секционные. Однокристалльные МП получены в результате реализации всех аппаратных средств процессора в виде одной БИС или сверхбольших ИС (СБИС). По мере увеличения степени интеграции элементов в кристалле и числа выводов корпуса параметры однокристалльных МП улучшаются. Однако возможности однокристалльных МП ограничены аппаратными ресурсами кристалла и корпуса. Поэтому более широко распространены многокристалльные МП, а также многокристалльные секционные МП.

Для получения многокристалльного МП необходимо провести разбиение его логической структуры на функционально законченные части: операционную, управляющую и интерфейсную (рис. 4.7). Сложность операционной и управляющей частей МП определяется их разрядностью, системой команд и требованиями к системе прерываний; сложность интерфейсной части — разрядностью и возможностями подключения других устройств МПС — памяти, внешних устройств, датчиков и исполнительных механизмов и др.

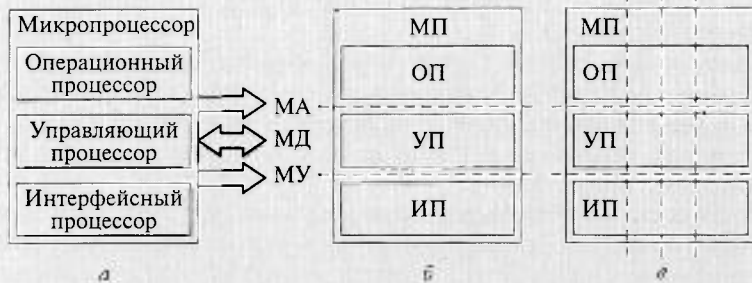


Рис. 4.7. Разбиение функциональной структуры МП:

*а* — однокристалльный микропроцессор; *б* — трехкристалльный микропроцессор; *в* — микропроцессор на базе комплекта секционных БИС

Интерфейс МП содержит несколько десятков шин информационных магистралей данных (МД), адресов (МА) и управления (МУ). При этом МПС и приобретает магистрално-модульную структуру, когда к единой внутренней информационной магистрали подключаются все основные функциональные блоки. Функциональная законченность БИС многокристалльного МП означает, что его части выполняют заранее определенные функции и могут работать автономно, а для построения развитого процессора не требуется организации большого количества новых связей и каких-либо других электронных БИС.

Операционный процессор (ОП) служит для обработки данных, управляющий процессор (УП) выполняет функции выборки, декодирования и вычисления адресов, а также генерирует последовательности микрокоманд. Автономность работы и большое быстродействие БИС УП позволяет выбирать команды из памяти с большей скоростью, чем скорость их исполнений БИС ОП. При этом в УП образуется очередь еще не исполненных команд, а также заранее подготавливаются те данные, которые потребуются ОП в следующих циклах работы. Такая опережающая выборка команд экономит время ОП на ожидание операндов, необходимых для выполнения команд программ. Интерфейсный процессор (ИП) позволяет подключить память и периферийные устройства к МП, он, по существу, является сложным контроллером для УВВ. БИС ИП выполняет также функции канала прямого доступа к памяти.

Многокристалльные секционные МП получают в том случае, когда в виде БИС реализуются части — секции логической структуры процессора при функциональном разбиении ее вертикальными плоскостями (см. рис. 4.7, *в*). Для построения многоразрядных процессоров при параллельном включении секции БИС в них добавляются средства «стыковки».

Для создания высокопроизводительных многоразрядных МП требуется столь много аппаратных средств, не реализуемых в дос-

Таблица 4.2

Характеристики МП комплектов БИС

Серия МП БИС или МП	Базовая технология	Разрядность МП	Число БИС/ИС в МП наборе		Тактовая частота, МГц
			общее	в базовом комплекте	
Однокристалльные МП					
K536	<i>p</i> -МДП	8	12	2	—
K580	<i>n</i> -МДП	8	3	1	2,5
K186	<i>n</i> -МДП	16	4	1	—
K1801	<i>n</i> -МДП	16	2	1	5
K588	КМДП	16	3	2	1
Многокристалльные МП					
K584	биполярные	4 · <i>m</i>	3	2	0,5
K1800	ЭСЛ	4 · <i>m</i>	8	2	36
K589	ТТЛШД	8 · <i>m</i>	8	2	—
K586	КМДП	4 · <i>m</i>	4	2	—
K1802	ТТЛШ	8 · <i>m</i>	11	2	—
K1883	<i>n</i> -МДП	8 · <i>m</i>	—	—	—

Примечание: *m* — число кристаллов, *m* = 1, 2, 4, ...

тупных БИС, что может возникнуть необходимость еще и в функциональном разбиении структуры МП горизонтальными плоскостями. В результате функционально и конструктивно законченные части процессора реализуются в виде БИС каждая. Все они образуют комплект секционных БИС МП. Таким образом, микропроцессорная секция — это БИС, предназначенная для обработки нескольких разрядов данных или выполнения определенных управляющих операций. Секционность БИС МП позволяет «наращивать» разрядность обрабатываемых данных или усложнять УУ при «параллельном» включении большого числа БИС.

Кроме того, выбираемые из памяти команды распознаются и выполняются каждой частью МП автономно, и поэтому может быть обеспечен режим одновременной работы всех БИС МП, т. е. конвейерный поточный режим выполнения последовательности команд программы. Такой режим работы значительно повышает производительность МПС.

За 15 лет с момента создания однокристалльные МП развились от простых специализированных 4-разрядных до 16-разрядных МП, сравнимых по параметрам с процессорами средних мини-ЭВМ. Трехкристалльные МП имеют разрядность до 32 бит и параметры,



сравнимые с параметрами старших моделей рядов мини-ЭВМ и средних ЭВМ общего применения. Многокристальные секционные МП имеют разрядность от 2...4 до 8...16 бит и позволяют создавать разнообразные высокопроизводительные процессоры ЭВМ. Так, германская фирма Parsytec производит суперЭВМ Super Cluster (SC), выполненную в виде наращиваемой ВС, содержащей от 64 до 1000 секционных процессоров (транспьютеров) типа T800. ЭВМ SC имеет 64...16 384 транспьютерных узлов на базе транспьютера типа T9000. Ее производительность достигает 400 опер/с. Название «SC» происходит от GigaCluster.

Однокристалльные и трехкристалльные БИС МП, как правило, изготавливают на основе микроэлектронных технологий униполярных полупроводниковых приборов, а многокристалльные секционные БИС МП — на основе технологии биполярных полупроводниковых приборов.

Использование многокристалльных МП БИС, имеющих функциональную законченность при малой физической разрядности обрабатываемых данных и монтируемых в корпус с большим числом выводов, позволяет организовать разветвление связи в процессоре, а также осуществить конвейерные принципы обработки информации и повысить производительность ЭВМ и ВС.

В табл. 4.2 приведены характеристики некоторых из выпускаемых отечественной промышленностью МП комплектов БИС.

#### Контрольные вопросы

1. Сформулируйте принцип микропрограммного управления.
2. Какая важная часть ЦП предназначена для управления всеми событиями внутри системы?
3. Каково назначение устройства управления МП?
4. Назовите два типа УУ и в чем заключаются общее и различие в их реализации?
5. Для чего и как формируется слово состояния процессора?
6. В чем заключаются особенности реализации и применения однокристалльных и секционных МП наборов?
7. Назовите способы организации связи между МП и УВВ.
8. Для чего предназначаются буферные регистры?
9. Как осуществляется запись числа в ОЗУ? Как его считать из ОЗУ?
10. Какие устройства участвуют в управлении работой всей микроЭВМ?
11. К скольким ячейкам памяти и портам можно получить доступ посредством 16-ти линий адресной шины?
12. Назовите по крайней мере три типа выходов МП.
13. Охарактеризуйте кратко назначение режимов Запись, Чтение.

## Глава 5

# ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА ЭВМ, МЕТОДЫ И СРЕДСТВА СОПРЯЖЕНИЯ, СЕТИ ЭВМ

### 5.1. Классификация периферийных устройств

Внешние устройства предназначены для обеспечения нормального функционирования ЭВС и для коммуникации центральных устройств с внешними источниками и потребителями информации.

К внешним относятся вспомогательные устройства, такие как устройства электропитания и аппаратура интерфейса питания, стабилизаторы напряжения, устройства защиты от перегрузок, устройства кондиционирования и вентиляции, счетчики времени и электронные часы, а также сервисная аппаратура для автономной проверки работоспособности плат и блоков.

В разработке и производстве вспомогательной аппаратуры на современном этапе достигнуты определенные успехи. Что же касается устройств ввода-вывода, стоимость которых в современных вычислительных системах достигает порой половины стоимости всей ВС, то, несмотря на имеющиеся достижения, эти устройства еще не совершенны. Основная проблема заключается в большом отличии по быстродействию процессора и устройств ввода-вывода. Существуют два аспекта этой проблемы: ограниченное быстродействие периферийного оборудования и ограничение скорости, с которой человек (либо внешняя среда) способен воспринимать и генерировать информацию.

Вторая проблема — более низкие, чем у центральных устройств, показатели ожидаемой надежности функционирования ввиду наличия у большинства периферийных устройств электромеханических блоков.

Третью проблему породила ориентация применения ЭВМ, особенно персональных, на компьютерно «неграмотного» пользователя, потребовавшая обеспечения максимального удобства общения с ЭВМ.

Рассмотренные в предыдущих главах внешние запоминающие устройства разных типов так же, как и устройства ввода-вывода

информации, которым посвящена данная часть учебника, зачастую относят к периферийным устройствам, так что выделение ВЗУ в самостоятельную группу устройств ЭВС довольно условно, хотя оно и оправдывается более тесной их взаимосвязью с внутренней памятью, особенно при динамическом распределении и организации виртуальной памяти.

Основное назначение периферийных (внешних) устройств ЭВС — организация входных и выходных потоков управляющей информации, данных для обработки и результатов вычислений. Таким образом, *периферийное устройство* (ПУ) — это любое отличное от центрального процессора оборудование, обеспечивающее коммуникацию вычислительной системы с внешними источниками и потребителями информации. Для этого ПУ обеспечивают согласование информационных и физических характеристик внешних объектов и сигналов, используемых в ЭВС. Эта связь поддерживается посредством преобразования в УВв разнообразных форм представления информации (символ, изображение, цвет, звук, непрерывно изменяющиеся параметры объекта наблюдения или управления) в цифровые машинные коды и, наоборот, преобразование с помощью УВыв результатов обработки из машинных кодов в форму, удобную для восприятия человеком или пригодную для воздействия на исполнительные органы объекта управления.

Как назначение, так и разновидность ПУ в вычислительной системе зависят от самой ВС и от характера решаемых задач.

По назначению ПУ могут быть разбиты на три группы: регистрирующие, оперативные и автоматические. *Регистрирующие УВыв* — устройства, использующие промежуточные носители (например, магнитоносители) для длительного хранения информации в виде, пригодном для последующего использования в ЭВС или в виде, удобном для использования человеком (графики, таблицы, печатный текст, чертежи).

*Оперативные УВв* — устройства непосредственного, оперативного взаимодействия оператора с ЭВМ, предназначенные для организации диалога между ЭВМ и человеком в процессе отладки программ и решения задач. Это оперативные УВв (клавиатура, световое перо, дигитайзеры, микрофон) и средства отображения (СО) результатов — цифровые индикаторы, экраны, звуковые сигнализаторы.

*Автоматические УВв* — устройства связи с объектом, предназначенные для ввода в ЭВС данных непосредственно с объектов автоматизации и выдачи управляющих воздействий на объекты. Это аналого-цифровые преобразователи (АЦП) и цифроаналоговые преобразователи (ЦАП). Сюда же можно отнести и читающие автоматы — сканеры.

Для большинства типов УВв существуют соответствующие УВыв, оперирующие с одинаковыми по форме сигналами и но-

сителями: ленточный перфоратор — ленточный перфосчитыватель; графопостроитель (плоттер) — графический оцифровщик (дигитайзер).

Иногда УВв и УВыв объединены функционально и конструктивно: экран на электронно-лучевой трубке (ЭЛТ) и световое перо — графический дисплей; электронный блокнот (ноутбук); накопитель на магнитном диске (НМД). Такие периферийные устройства называются совмещенными.

Большинство УВв информации содержат электромеханические узлы, скорость работы которых значительно ниже скорости ЭВС. Поэтому большие ЭВМ снабжаются большим числом УВв. Каждое устройство при этом может работать параллельно с другими и независимо от них, что повышает производительность ЭВМ.

За основу классификации УВв и УВыв возьмем один из основных параметров — быстродействие, оцениваемое количеством байтов информации, передаваемым в секунду. По этому признаку, а также по функциональному назначению УВв можно разделить на три группы: низкоскоростные (порядка 10 байт/с), среднескоростные (от 100 до 1000 байт/с) и высокоскоростные (от  $10^4$  до  $10^6$  байт/с). На рис. 5.1 и 5.2 приведены классификации устройств ввода и вывода соответственно. Двойной рамкой обведены специализированные и перспективные УВв. Штриховкой показаны необязательные элементы.

АЦП и ЦАП могут принадлежать к любой из трех групп УВв в зависимости от скорости протекания процессов, за которыми ведется наблюдение или управление.

*Модемы* — устройства, модулирующие и демодулирующие сигналы, передаваемые с помощью средств связи в случае использования каналов связи для коммутации ЭВС в вычислительных сетях, занимают промежуточное значение между низкоскоростными и среднескоростными УВв.

Устройства подготовки данных (УПД) предназначены для автономного ручного нанесения информации на машинные носители разных типов: перфоленты, магнитные носители; сюда же могут быть отнесены и устройства записи на оптические WORM диски.

В последующих главах и подразделах основное внимание будет уделено сложным по своему устройству и функционированию УВв из получившей широкое распространение серийной периферии (сканеры, плоттеры, дигитайзеры, дисплеи), а также перспективным устройствам, находящимся на стадии разработки и совершенствования. Более простые электромеханические УВв, такие как наборные поля и клавиатура, принтеры, перфораторы и перфосчитыватели, самописцы и устройства считывания графиков, а также средства коллективного отображения информации — табло, экраны, мнемосхемы рассматриваться не будут, хотя они широко

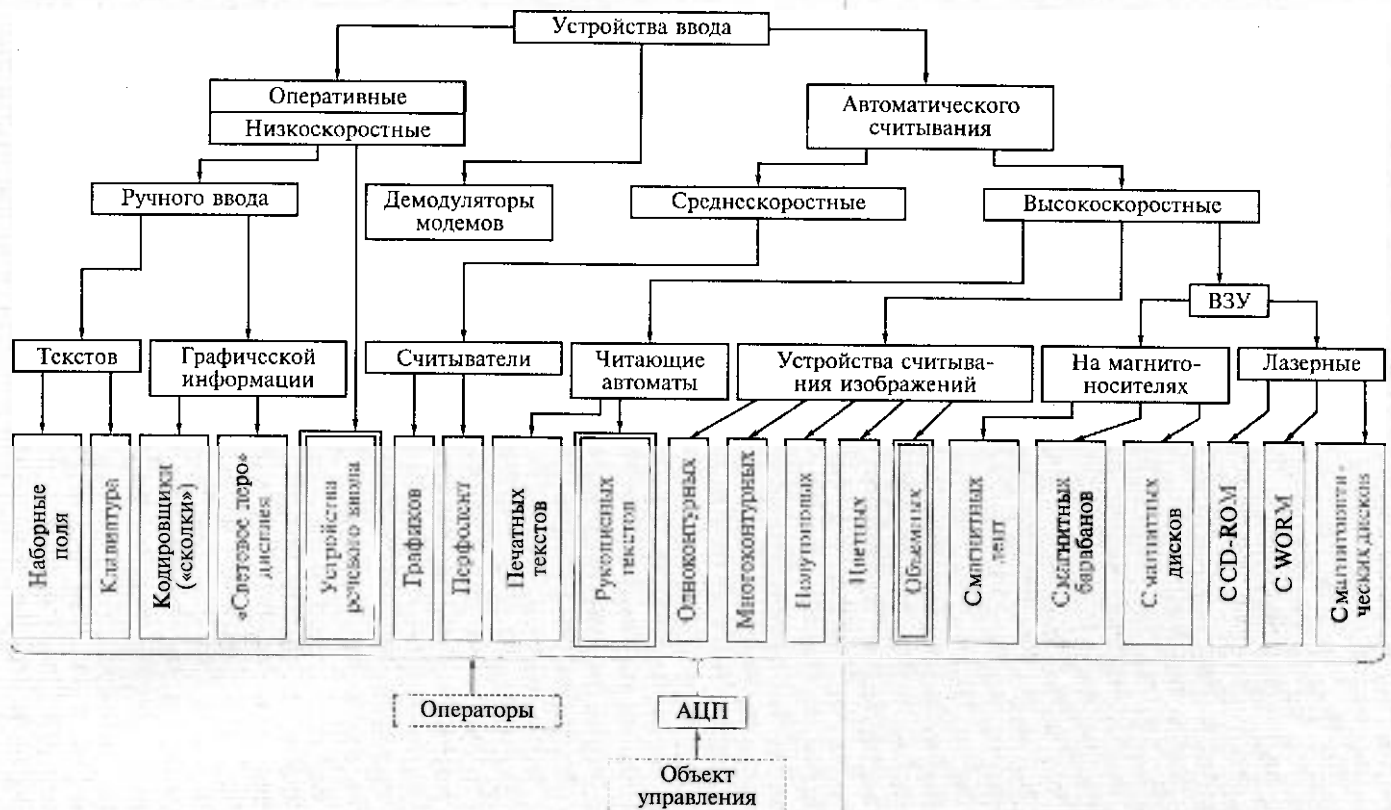


Рис. 5.1. Классификация устройств ввода

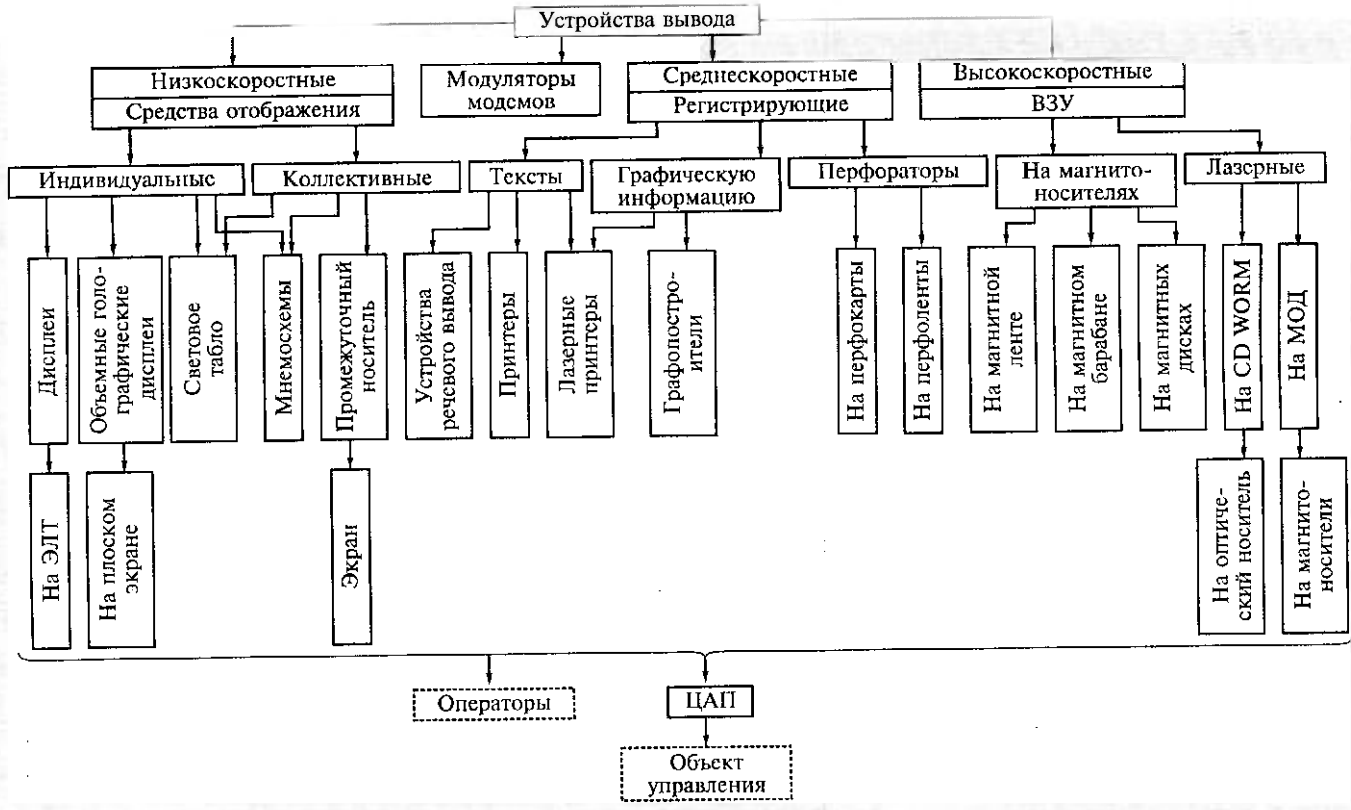


Рис. 5.2. Классификация устройств вывода



распространены. Все эти устройства, так же, как и подробно рассматриваемые далее, выполняют преобразование формы вводимой от источника на обработку в центральные устройства информации в машинные коды и обратное преобразование машинных кодов результатов обработки в требуемую потребителем информации форму.

## 5.2. Внешние запоминающие устройства на магнитных, оптических, магнитооптических носителях

**ВЗУ на магнитных носителях.** ВЗУ на магнитных носителях разделяются на два типа: накопители на гибких магнитных дисках (НГМД) и накопители на жестких магнитных дисках (НЖМД).

Поверхность форматированного диска как гибкого, так и жесткого, разделена на дорожки, которые в свою очередь делятся на секторы. Таким образом, диск имеет четыре измерения:

- 1) сторона диска или номер пластины диска, вертикальное измерение;
- 2) дорожка или цилиндр, радиальное измерение;
- 3) сектор внутри дорожки, измерение по окружности;
- 4) количество данных в секторе.

Основной операционной единицей диска является сектор. Все операции «запись/воспроизведение» используют порции данных, кратные сектору. Программа должна находить и идентифицировать каждый сектор, пользуясь всеми измерениями.

Для работы с новыми дискетами или жесткими дисками необходимо выполнить их форматирование, заключающееся в раз-

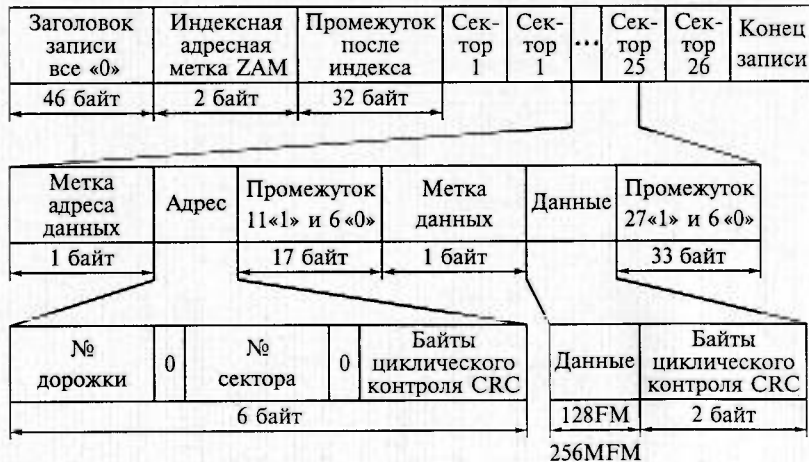


Рис. 5.3. Структура формата записи дорожки НГМД стандарта IBM 3740

биении рабочей поверхности диска на дорожки, секторы и записи служебной информации.

Структура формата записи зависит от типа контроллера — управляющего устройства НМД, и, как правило, включает в себя байты синхронизации, указывающие на начало каждого сектора, идентификационные заголовки, состоящие из номера сектора и цилиндра, а также байты циклического контроля, предназначенные для обнаружения ошибок в информации при считывании данных, и служебной информации. На рис. 5.3 показан стандартный формат (IBM формат 3740), принятый для накопителей на гибких дисках в IBM PC. Начало дорожки определяется индексным импульсом. Заголовок записи, следующий за индексом и состоящий из последовательности «0», используется для синхронизации генератора тактовых импульсов чтения, которая обеспечивается схемой фазовой автоподстройки частоты (ФАПЧ).

После заголовка записи следует адресная индексная метка. Первый байт метки содержит набор битов с отсутствующими тремя первыми битами. Эти пропущенные биты обнаруживаются в процессе считывания и идентифицируют следующий байт как тип метки. Тип метки определяется комбинацией битов (5...7) во втором байте метки, приведенной ниже:

Разряды 5 6 7	Тип метки
0 0 0	Не заполнено
0 1 1	Данные
1 0 0	Индекс
1 1 0	Адрес

В случае распознавания индексной адресной метки IAM контроллер оповещается о том, что после 32 байтов, заполненных «0», появится адресная метка первого информационного сектора. Затем следует пустой промежуток в 17 байт, который позволяет блоку считывания/записи переключаться в случае необходимости в режим записи. Следующая за промежутком в 17 байт метка данных сообщает, имеется ли в поле данных полезная информация. Поле данных всегда заполнено байтами данных, импульсы, считанные при прохождении этого поля под головкой считывания/записи, поддерживают синхронизацию генератора тактовых импульсов, используемого в схеме разделения данных. При записи контроллер указывает в метке данных какими байтами данных, полезными или бесполезными, заполнено это поле. После чтения метки данных осуществляется чтение или запись блока данных. Для проверки контрольных сумм используется символ циклического контроля CRC в 2 байта, который формируется контроллером при записи данных. Блок данных заканчивается пустым промежутком данных, необходимым для того, чтобы схема успела перестроиться для распознавания следующего адреса.

**Накопители на гибких магнитных дисках.** Важнейшим достоинством НГМД является высокий уровень стандартизации аппаратных и программных средств, что позволяет использовать их как универсальное средство обмена информацией между различными ЭВМ. Невысокая стоимость и достаточная надежность НГМД способствуют их широкому распространению.

Независимо от типа НГМД состоит из следующих основных компонентов: корпуса, механизма фиксации и привода дискеты, блока магнитных головок (МГ), датчиков контроля, платы электроники. В качестве привода для вращения диска используется шпиндельный двигатель. Скорость вращения должна быть фиксированной, 300 или 360 об/мин, что необходимо для обеспечения взаимозаменяемости дискет, записанных на разных устройствах.

Блок МГ включает в себя каретку, на которой закреплены магнитные головки, шаговый двигатель, вращательное движение которого преобразуется в поступательное движение каретки, и механизма прижима головок. Каретка с МГ при помощи шагового двигателя перемещается по поверхности диска в радиальном направлении фиксированными шагами. Величина шага зависит от используемой на носителе радиальной плотности записи. Так, например, в НГМД с плотностью 96 дорожек на дюйм (96 tpi, где tpi от англ. *track per inch*) величина шага соответствует повороту ротора шагового двигателя на 1,8°. Перемещение блока МГ из одного крайнего положения в другое осуществляется в пределах одного оборота шагового двигателя. Полное время движения головок — позиционирования — составляет приблизительно 200 мс. МГ находятся в постоянном контакте с поверхностью дискеты, что, хотя и приводит к дополнительному износу МГ и дискеты, однако, позволяет упростить конструкцию головок. В паузах между операциями записи/чтения МГ приподнимается над поверхностью дискеты.

Головки расположены с каждой стороны диска, смонтированы в общем блоке и поддерживаются шарнирными подвесными механизмами, причем головка записи/чтения расположена между двумя стирающими головками (рис. 5.4). При записи используется

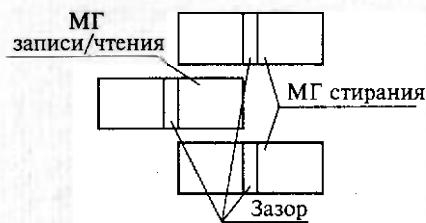


Рис. 5.4. Блок магнитных головок НГМД

туннельное стирание двумя МГ стирания: они урезают по ширине дорожку, образуемую МГ записи. Таким образом, между записанными дорожками создаются размагниченные участки, что повышает надежность последующего чтения.

Для управления режимами работы НГМД в нем используется ряд датчиков. Для опреде-

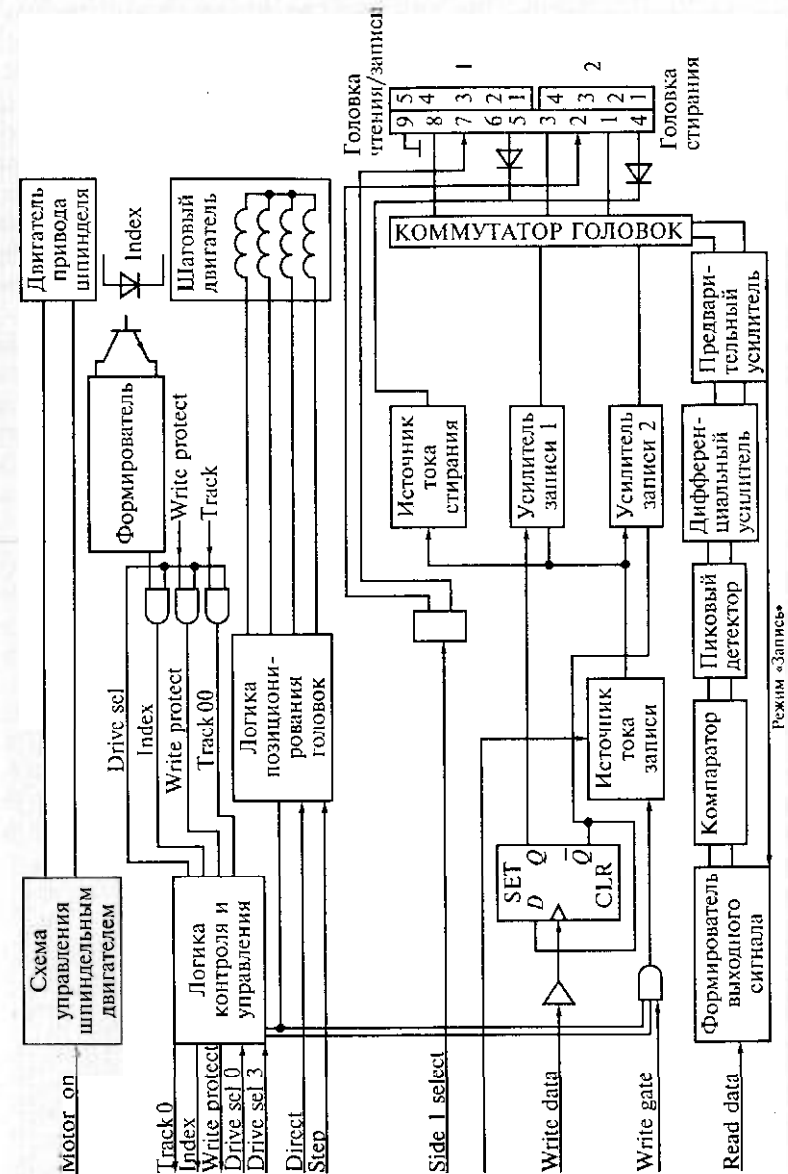


Рис. 5.5. Структурная схема НГМД

ления начала дорожки при форматировании дискеты используется датчик индекса, представляющий собой датчик Холла в 3-дюймовых дисководов. Положение нулевой дорожки в зависимости от модели определяется либо при помощи микропереключателя, либо с использованием фотооптического преобразователя.

Плата электроники содержит электронную схему управления дисководом, включающую в себя функциональные узлы и блоки, приведенные на рис. 5.5. Управление работой НГМД осуществляется сигналами, поступающими от контроллера НГМД по интерфейсу.

В настоящее время в ПК типа IBM PC и совместимых с ними используется свыше 300 различных типов НГМД. Они различаются диаметром дискеты, высотой, плотностью записи и другими параметрами. Количество секторов и дорожек и их размер определяются, как правило, программно, в процессе форматирования. Именно поэтому гибкие диски называют дисками с программной разметкой секторов Soft sector.

Дискетоводы 1,44 Мбайт достигают своей емкости благодаря использованию носителя высокой плотности, обеспечивающего размещение 80 цилиндров с 18-ю секторами на дорожку. Такая линейная плотность записи даже выше, чем у некоторых жестких дисков, имеющих 17 секторов на дорожку.

Функции сопряжения НГМД с центральным процессором выполняет контроллер, который освобождает ЦП от рутинных обязанностей низкоуровневого управления приводом. При этом роль ЦП сводится к общему управлению контроллером с помощью команд и слов состояния процессора. Такая иерархия управления упрощает программирование ввода-вывода и увеличивает общую производительность системы.

**Накопители на жестких магнитных дисках типа винчестер и их контроллеры.** Подсистема дисковой памяти на НЖМД состоит так же, как и память на НГМД, из контроллера, собственно накопителя и соединительных кабелей интерфейса. Контроллер вырабатывает команды для накопителя и обеспечивает обмен данными между ним и компьютером.

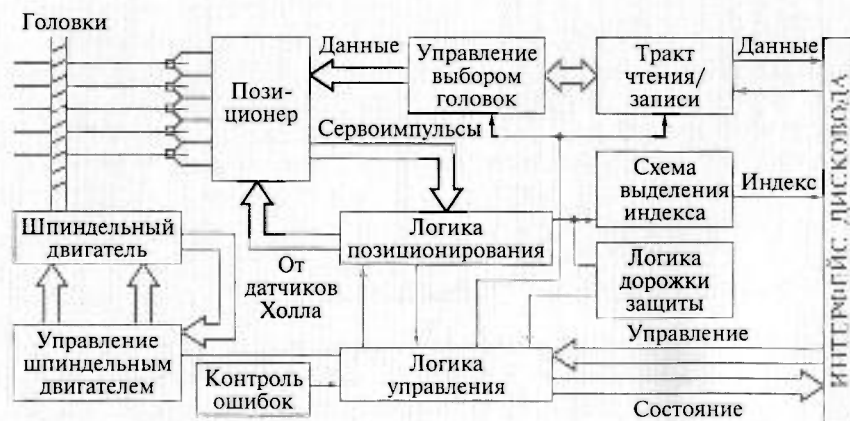


Рис. 5.6. Структурная схема НЖМД винчестер

Большинство компьютеров комплектуются комбинированными контроллерами НЖМД/НГМД и всеми необходимыми кабелями.

Структурная схема НЖМД для персональных компьютеров приведена на рис. 5.6. В настоящее время существует широкая номенклатура НЖМД, отличающихся используемым интерфейсом, физическими размерами, быстродействием и надежностью. Первоначально такие накопители включали два диска емкостью по 30 Мбайт, составлявших единый блок. Результирующая емкость получаемого НМД обозначалась цифрами 30/30, подобно калибру охотничьего ружья «винчестер». Отсюда и возникло название НЖМД винчестер.

Емкость современных НЖМД составляет от 20 Мбайт до нескольких Гбайт.

Показатель быстродействия накопителя — скорость передачи данных  $V$  бит/с (Бод). Она определяется методом кодирования и используемым интерфейсом. Максимальная скорость считывания данных из накопителя

$$V_{\max} = m \omega N n,$$

где  $m = 8$  бит;  $\omega$  — скорость вращения диска (число оборотов в минуту);  $N$  — число секторов на дорожке;  $n$  — количество байтов в секторе.

В большинстве накопителей, используемых в персональных ЭВМ (ПЭВМ), скорость вращения (3600 об/мин) и количество байтов (512 байт) в секторе одинаковы. Следовательно, максимальная скорость передачи данных НЖМД определяется в основном количеством секторов на дорожку, поскольку все другие факторы фиксированы. Так, например, любой накопитель с 17 секторами на дорожку будет иметь строго определенную скорость передачи данных 4 177 920 бит/с.

Однако не все контроллеры могут работать с такой скоростью, так как контроллеру и процессору требуется время на запоминание считанной информации, и они не успевают за непрерывно вращающимся диском и передвигающимися секторами. Для преодоления этого недостатка в НЖМД применяется чередование нумерации секторов (INTERLIVE). Например, при чередовании 1:3 сектора на диске нумеруются в следующем порядке: 1, 7, 13, 2, 8, 14, 3 и т. д. Таким образом, фактор чередования показывает, за какое количество оборотов может быть считана информация с дорожки.

В настоящее время наибольшее распространение получили дискетоводы 3,5 дюймовые, как и дискетоводы НГМД. Благодаря более высокой скорости вращения дисков и надежной защите от пыли плотность записи на жестких дисках в 20...30 раз превышает плотность записи на гибких магнитных дисках.

В пакете дисков одна из поверхностей освобождается от использования для запоминания данных, и на нее в заводских усло-



виях записывается специальная информация. Эта поверхность, называемая сервоповерхностью (DSS, от англ. *dedicated servo surface*), содержит индексные метки, указывающие позиции дорожек. Этот метод записи называется «выделенная запись сервосигналов». Иногда сервоинформацию записывают на поверхности вместе с данными между дорожками. Специальные датчики на головке чтения/записи следят за серводанными, и сервосистема подводит головки к середине дорожки в случае их отклонения. Такой метод называется «встроенная запись сервосигналов».

Электронные схемы управления шпиндельным двигателем и механизм позиционирования, тракты записи/чтения, схемы формирования сигналов от датчиков, а также схемы сопряжения с интерфейсом размещены на интерфейсной печатной плате. Надежность функционирования накопителей, характеризующаяся средним временем наработки между отказами  $T_n$ , составляет 20 000... 50 000 ч.

В настоящее время для обеспечения информационной совместимости при обмене информацией между НЖМД и ПЭВМ используются интерфейсы четырех типов:

- 1) ST 506/412;
- 2) усовершенствованный интерфейс малых устройств ESDI;
- 3) интерфейс малых вычислительных машин SCSI;
- 4) интерфейс встроенных дисковых накопителей IDE.

Стандарты DIN (Deutsche Industrie Norm), ECMA (European Computer Manufactures Association), ISO (International Standard Organisation) определяют следующие элементы интерфейса: принципы физического подключения устройств; временные соотношения и протокол обмена по шине; набор команд подключаемых устройств. (Протокол обмена определяет структуру, которую должно иметь сообщение для сопряжения периферийного оборудования с центральными устройствами BC.)

В основу интерфейса ST 506/412 легли два других: SA 450, применяемый для 5,25-дюймовых НГМД, и S1000 — для 8-дюймовых НЖМД. Этот интерфейс рассчитан на максимальную скорость передачи 5 Мбит/с. Для увеличения скорости передачи и размещения большего объема информации на диске используется методика сжатия информации RLL (Run Length Limited). Количество секторов на дорожке при RLL-методе увеличивается до 25...26, т.е. информационная емкость накопителя увеличивается на 50 %, скорость же передачи повышается до 7,5 Мбит/с.

В течение длительного времени интерфейс вполне соответствовал своему назначению. Жесткий диск совместно с этим интерфейсом осуществлял передачу данных быстрее, чем компьютер мог их обработать. С момента появления процессоров серии 386 с тактовой частотой 25 МГц интерфейс ST 506/412 стал узким местом в системе.

Более производительный и надежный интерфейс ESDI был создан как усовершенствованная версия интерфейса ST 506. Здесь применен новый метод записи NRZ и более эффективный способ декодирования данных, что позволило вдвое увеличить плотность записи — 36 секторов на дорожку. Самое существенное отличие ESDI состоит в том, что функции разделителя данных были перенесены из контроллера в накопитель. ESDI обеспечивает скорость передачи данных НЖМД до 24 Мбит/с.

Оба рассмотренных интерфейса передают информацию последовательно, побитно. Интерфейс малых компьютерных систем SCSI представляет собой параллельный интерфейс, разрешающий побайтную пересылку данных, что позволяет повысить скорость обмена данными до 40 Мбит/с. Интерфейс позволяет подключить к одной шине до 8-ми различных периферийных устройств, которые могут взаимодействовать друг с другом. Для сопряжения шины контроллера и шины SCSI используется HOST-адаптер — главный адаптер шины SCSI.

Адаптеры SCSI берут на себя функции управления накопителем, а осуществляют лишь передачу накопителю запросов на чтение/запись. Установленные в накопитель контроллеры интерпретируют запросы SCSI в команды накопителя. Высокопроизводительные контроллеры, устанавливаемые в накопители, могут выполнять оптимальное чередование и допускают использование сверхоперативной (КЭШ) памяти. Интерфейс стандарта SCSI выполняет каждую команду в два этапа: на первом этапе HOST-адаптер преобразует команды операционной системы в команды формата SCSI, на втором этапе контроллер НЖМД осуществляет преобразование команд шины SCSI в сигналы управления накопителем.

С появлением малогабаритных ПЭВМ большое значение стали иметь габариты дисководов и контроллера, и в 1988 г. был разработан интерфейс IDE (Intelligent Drive Electronics) как недорогая альтернатива интерфейсам ESDI и SCSI. Его отличительная особенность — реализация функций контроллера в накопителе, благодаря чему повысилась общая производительность системы. Большинство IDE дисков форматируется на 34 и более секторов и имеет чередование 1:1.

Контроллеры НЖМД в отличие от контроллеров НГМД имеют более сложную структуру, что объясняется расширением и усложнением функций, возложенных на контроллеры НЖМД. Для разгрузки ПЭВМ от рутинных операций в контроллере предусмотрено микропрограммное управление накопителем, прямой доступ к памяти ПЭВМ, буферизация считанных и записываемых данных, процедура повышения достоверности считываемой информации (повторение операции чтения или коррекция считанных данных).

**Оптические и магнитооптические ВЗУ.** Современные электро-механические магнитные ВЗУ достигли высоких технико-экономических показателей, однако дальнейшее их совершенствование, диктуемое непрерывным развитием ЭВС, становится все труднее, так как рост основных характеристик электро-механических ВЗУ достиг предела. В связи с этим ведутся интенсивные поиски и совершенствование методов и средств записи и хранения информации. Среди новых методов выделяются оптический и магнито-оптический.

Оптические ВЗУ имеют очень высокую плотность записи информации, на несколько порядков большую плотности магнитных ВЗУ, так как для регистрации одного бита достаточно участка на носителе с размером порядка длины волны излучаемого лазером света (0,5 мкм). Этот тип ВЗУ имеет также высокое быстродействие и надежность, что делает его наиболее реальным претендентом для полной замены традиционных магнитных электро-механических устройств памяти объемом от 10 Гбайт и выше. Однако в оптических накопителях трудно решается проблема перезаписи информации. Кроме того, стоимость на бит хранимой информации пока еще велика по сравнению с НМД.

Как запись на оптический носитель — оптический диск, так и воспроизведение с него, осуществляются лазерным лучом. Лазеры (от англ. *light amplification by stimulated emission of radiation*) способны генерировать и усиливать электромагнитные колебания в диапазонах 0,4 мкм...0,78 мкм (инфракрасная часть оптического спектра, это лазеры), 0,78...0,38 мкм (волны видимого света) и 0,38 мкм ... 2 нм (ультрафиолетовая часть спектра).

**Оптические диски.** В 1980-х гг. на смену виниловым музыкальным дискам пришли оптические компакт-диски (CD), примерно в это же время появились первые персональные компьютеры фирмы IBM. Компакт-диски явились результатом сотрудничества двух фирм — японской Sony и голландской Philips. CD был рассчитан на 74 мин звучания, что соответствует 640 Мбайт при диаметре 120 мм; это эквивалентно 444-м дискетам высокой плотности записи. Разработанные для высококачественной звуковоспроизводящей аппаратуры, CD прочно вошли и на рынок компьютерных устройств благодаря малым размерам, большой емкости, высокой надежности и долговечности. Недостатком CD-ROM (compact disk read only memory) является то, что на них невозможно записывать информацию непосредственно с ЭВМ. Проблема была решена позже, созданием нереверсивных WORM (write once read many) оптических и реверсивных, стираемых магнитооптических дисков.

Цифровой оптический диск состоит из рабочего (реги-стрирующего, информационного) слоя, на который наносится информационная сигналограмма в виде определенных чередований его состояний, и основы, на которой находится этот рабочий слой.

Изменение состояния рабочего слоя зависит от изменения его оптических характеристик.

Структура и состав рабочего слоя определяются технологией записи информации и влияют на разрешающую способность CD. Для CD с однократной записью (CD-ROM и WORM) наиболее часто применяются три типа технологии записи:

- 1) удаление участка рабочего слоя при нагреве его лазерным лучом (абляция);
- 2) образование вспучивания рабочего слоя путем создания под ним газового пузырька;
- 3) изменение фазового состояния участка бита информации, приводящее к изменениям оптических констант: коэффициента преломления либо коэффициента поглощения.

Последний тип технологии позволяет многократно перезаписывать информацию и наряду с магнитооптическим методом используется для создания реверсивных (стираемых) дисков. Известны также разработки по использованию цвета рабочего слоя.

Если основа прозрачна и достаточно однородна, оптическая запись и считывание информации могут вестись через основу. При этом основа используется в качестве защитного слоя. При использовании прозрачных основ возможна реализация записи, считывания на основе модуляции коэффициента отражения или пропускания. Первый путь проще конструктивно и поэтому более распространен.

Информация на CD расположена по дорожке, представляющей собой пунктир из меток записи. Метки являются областями, имеющими контраст с окружающей их зеркальной поверхностью диска. Оптическая головка накопителя преобразует изменения отражения луча в электрический сигнал.

Другим достоинством накопителей на оптических дисках является отсутствие механического контакта оптической головки с поверхностью диска.

По формату дорожек диски делятся на диски с концентрическими и спиралевидными дорожками. Диски с концентрическими дорожками менее дорогие, но время доступа велико, так как головка чтения/записи должна перемещаться с одной дорожки на другую. Спиралевидные дорожки не разрывают длинные файлы, но механизм позиционирования головки сложнее и дороже.

Технология изготовления CD-ROM напоминает технологию изготовления грампластинок: вначале создается диск-оригинал, состоящий из массивной стеклянной основы с нанесенным на нее регистрирующим слоем, запись информации на который ведется посредством прецизионной и сложной аппаратуры в помещении с высокой степенью чистоты воздуха. Полученный путем лазерной записи на рабочем слое диска-оригинала рельеф переносится на металлическую матрицу, которая затем непосредствен-



но используется при массовом изготовлении CD с пластмассовой основой для создания информационной рельефной сигналограммы на его рабочей поверхности. При этом не всегда достигается хорошее качество.

Для самого распространенного и наиболее простого способа записи, каким является абляционный способ, рабочий слой CD может быть выполнен так, что в результате образования отверстия в нем при записи возможно как увеличение, так и уменьшение коэффициента отражения в этом отверстии по сравнению с остальной площадью диска. Для создания канавок на основах из неорганического стекла применяют либо непосредственное травление канавок в стекле после нанесения маски на основу, либо травление канавок в защитном слое, нанесенном на его основу. Последнее выполняется посредством лазерного луча и последующего реактивного ионного травления этого слоя.

Поскольку линейная скорость носителя не одинакова для разных дорожек, при записи происходят искажения.

В конструкциях дисководов для борьбы с искажениями диск при чтении вращается с переменной угловой скоростью. Таким образом, чтение внутренних секторов (дорожек) осуществляется при увеличении, а наружных — при уменьшении числа оборотов. Этим обуславливается довольно низкая скорость доступа по сравнению, например, с винчестерами.

Преимуществами CD-ROM являются: сходство со звуковыми CD; упрощение распространения больших объемов информации, стоящего дешевле, чем распространение посредством печатной машины; это более надежное средство для хранения информации, чем остальные, вследствие неуязвимости по отношению к электромагнитным и механическим повреждениям; CD снабжены легкой тонкой упаковкой и действительно компактны, что облегчает их транспортировку и распространение.

К недостаткам CD-ROM относятся: невозможность обновления информации; высокая стоимость изготовления оригинальных дисков; длительность процесса изготовления оригинала и копирования, занимающего несколько дней; большое время доступа для большинства типов CD-ROM, в среднем достигающее 400 мкс, тогда как у винчестера время доступа равно от 8 до 30 мкс.

WORM-диски, известные как диски с записью, хранят данные для компьютеров, они идеальны для архивов и тиражирования.

Компакт-диск, записанный самостоятельно, работает так же, как и серийные отштампованные диски, поэтому он может быть прочитан на любом дисковом CD-ROM. Конструкция оптической головки, предназначенной для записи и считывания WORM-дисков, приведена на рис. 5.7.

Преимуществами WORM-дисков являются возможность записи больших объемов информации и их неуязвимость по отноше-

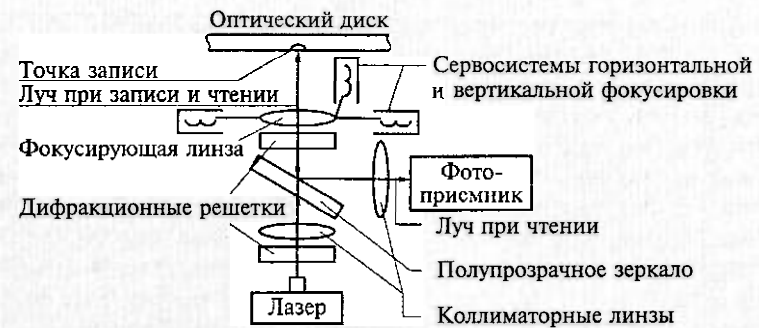


Рис. 5.7. Оптическая головка для WORM-дисков и дисков с перезаписью

нию к порче, краткосрочному огню и разрушению. Диски выдерживают длительное хранение, от 10 до 100 лет. Запись и воспроизведение производятся в реальном масштабе времени, причем чтение возможно непосредственно после записи. К недостаткам относятся высокая стоимость накопителей и самих дисков.

**Магнитооптические диски.** Для изготовления стираемых дисков, называемых также реверсивными или дисками с перезаписью, в качестве рабочего слоя применяются реверсивные светочувствительные материалы в виде пленки, такие, как субокисел теллура и его соединения с селеном, индием и сурьмой. Для компенсации отражения и улучшения параметров светочувствительности пленок используют многослойные системы.

Запись информации на диск осуществляется с помощью магнитного поля и луча лазера. Конструкция магнитной головки схожа с приведенной конструкцией для WORM-дисков (см. рис. 5.7). Магнитный слой устроен таким образом, что при нормальной температуре интенсивность магнитного поля, называемого полем смещения, недостаточна для переориентации содержащихся в материале рабочего слоя магнитных доменов. При локальном нагреве точки записи лазерным лучом поле смещения способно изменить намагниченность этой области, при этом меняется вектор напряженности магнитного поля материала, заранее намагниченного до насыщения в определенном направлении. Характеристики материала таковы, что воздействие одного из этих факторов для перемагничивания участка в точке записи недостаточно. Нагрев же участка до некоторой критической температуры создает условия для возможности его перемагничивания под воздействием внешнего сравнительно слабого магнитного поля.

При считывании используется лазерный луч меньшей, чем при записи, интенсивности. Чтение данных основано на использовании эффектов Керра или Фарадея. Для этого участки записанной сигналограммы в процессе ее движения под лазерной оптической головкой последовательно освещают пучком линейно поляризо-



ванного света и анализируют с помощью отраженного пучка света или же пучка света, прошедшего через рабочий слой. Если применяется эффект Керра, то при воспроизведении сопоставляются ориентации плоскости линейной поляризации падающего и отраженного пучков света. Если же используется эффект Фарадея, то в связи с прозрачностью диска световой пучок проходит через диск и сопоставляются ориентации плоскостей линейной поляризации входного и выходного световых пучков.

Емкость магнитооптических (МО) кассет достигла 1,3 Гбайт (производители IBM, Sony, Hewlett-Packard и др.) и постоянно увеличивается, приближаясь к 30 Гбайт на одной стороне 133 мм диска. Плотность записи стремится к 2...2,5 Мбит/мм<sup>2</sup>, скорость передачи данных за счет буферизации повышается до 200 Мбит/с, время доступа в пределах одного магнитооптического диска (МОД) снизится до 0,01 с. Ожидается реализация поперечной плотности оптической записи, достигающая 1000 дорожек/мм, даже при сохранении длины волны излучения лазера и традиционных принципов работы.

### 5.3. Устройство ввода данных — клавиатура

*Клавиатура* — это одно из основных устройств ввода информации в ЭВМ, позволяющее вводить различные виды информации. Вид вводимой информации определяется программой, интерпретирующей нажатые или отпущенные клавиши. С помощью клавиатуры можно вводить любые символы — от букв и цифр до иероглифов и знаков музыкальной нотации. Клавиатура позволяет управлять курсором на экране дисплея — устанавливать его в нужную точку экрана, перемещать по экрану, «прокручивать» экран в режиме скроллинга, отправлять содержимое экрана на принтер, производить выбор при наличии альтернативных вариантов и т. д.

В последнее время наблюдаются тенденции отказа от клавиатуры в пользу альтернативных устройств: мыши, речевого ввода, сканеров. Но полностью эти устройства клавиатуру не заменяют.

Стандартная клавиатура IBM PC имеет несколько групп клавиш.

1. Алфавитно-цифровые и знаковые клавиши (с латинскими и русскими буквами, цифрами, знаками пунктуации, математическими знаками).

2. Специальные клавиши: <Esc>, <Tab>, <Enter>, <BackSpace>.

3. Функциональные клавиши: <F1>...<F10>.

4. Служебные клавиши для управления перемещением курсора (стрелки: <Up>, <Down>, <Left>, <Right>, клавиши <Home>, <End>, <Pg Up>, <Pg Dn>).

5. Служебные клавиши для управления редактированием <Ins>, <Del>.

6. Служебные клавиши для смены регистров и модификации кодов других клавиш <Alt>, <Ctrl>, <Shift>.

7. Служебные клавиши для фиксации регистров <CapsLock>, <ScrollLock>, <NumLock>.

8. Вспомогательные клавиши <PrtSc>, <Break>.

Если клавиша первой, четвертой, а иногда и пятой группы оказывается нажатой дольше, чем 0,5 с, начинает генерироваться последовательность ее основных кодов с частотой 10 раз в секунду (в IBM PC XT), что имитирует серию очень быстрых нажатий этой клавиши.

Общее число клавиш в основной модификации клавиатуры — 83, в расширенной клавиатуре — до 101. Количество различных сигналов от клавиатуры значительно превышает это число по следующим причинам:

1) при нажатии и освобождении клавиши в ЭВМ передаются разные кодовые комбинации: при нажатии — порядковый номер нажатой клавиши на клавиатуре (ее скан-код), а при освобождении — скан-код, увеличенный на 80h в шестнадцатеричной системе счисления;

2) заглавные и строчные буквы первой группы клавиш (алфавитно-цифровых и знаковых) набираются на разных регистрах. Оперативное переключение регистров производится клавишей <Shift>. Если при нажатой (и удерживаемой в нажатом состоянии) клавише <Shift> нажать любую алфавитную клавишу, то в ЭВМ будет отправлен код заглавной буквы, соответствующий нажатой клавише;

3) после однократного нажатия клавиши <CapsLock> (зажигается лампочка на клавиатуре рядом с клавишей) изменяется порядок работы клавиши <Shift>: без нажатия на нее будут набираться заглавные буквы, а при нажатии (совместном) — строчные. После повторного нажатия на <CapsLock> порядок работы клавиши <Shift> восстанавливается, а лампочка гаснет. Такой режим (переключательный) работы клавиши называется триггерным режимом, или flip-flop;

4) аналогично клавише <Shift> действуют <Alt> и <Ctrl> — при одновременном нажатии с ними любой другой клавиши, в ЭВМ передается не скан-код, а расширенный код (2 байта). Иногда таким же образом используется клавиша <Esc>;

5) клавиша <NumLock> является триггерным переключателем дополнительной цифровой клавиатуры: при негорящей лампочке она работает как клавиатура для управления курсором; при зажатой — как цифровая;

6) для переключения регистров (или даже групп регистров) иногда используются другие комбинации клавиш: например, про-

граммы-русификаторы клавиатуры переключают РУС/ЛАТ с помощью правой клавиши <Shift> или при одновременном нажатии двух клавиш <Shift> (правой и левой) и т.д. Эти комбинации клавиш обладают триггерным эффектом.

Сигналы, поступающие от клавиатуры, проходят трехуровневую обработку: на физическом, логическом и функциональном уровнях.

Физический уровень имеет дело с сигналами, поступающими в вычислительную машину при нажатии и отпускании клавиш.

На логическом уровне, реализуемом BIOS через прерывание 9, скан-код транслируется в специальный 2-байтовый код. Младший байт для клавиш группы 1 содержит ASCII-код, соответствующий изображенному на клавише знаку. Этот байт называют главным. Старший байт (вспомогательный) содержит исходный скан-код нажатой клавиши.

На функциональном уровне отдельным клавишам программным путем приписываются определенные функции. Такое «программирование» клавиш осуществляется с помощью драйвера-программы, обслуживающей клавиатуру в операционной системе.

На IBM PC AT используется клавиатура с большим количеством клавиш. На этих машинах есть возможность управлять некоторыми функциями клавиатуры, например, изменять время ожидания автоповтора, частоту автоповтора, зажигать и гасить светодиоды на панели управления клавиатурой.

Устройство клавиатуры не является простым: в клавиатуре используется свой микропроцессор, работающий по прошитой в ПЗУ программе. Контроллер клавиатуры постоянно опрашивает клавиши, определяет, какие из них нажаты, проводит контроль на «дребезг» и выдает код нажатой или отпущенной клавиши в системный блок ЭВМ.

Выпускаемые разными производителями клавиатуры различаются также по расстоянию между клавишами, числу специальных клавиш, способу переключения на цифровой регистр для быстрого ввода числовых данных, углу наклона, форме и текстуре поверхности клавиш, усилию нажима и величине хода клавиш, расположению часто используемых клавиш и др.

#### 5.4. Устройства ввода изображений — дигитайзеры, сканеры

**Дигитайзеры.** Устройства полуавтоматического ввода графической информации — дигитайзеры. При проведении конструкторских работ в различных областях широко применяются автоматизированные рабочие места проектировщиков (АРМ), представляющие собой диалоговый графический комплекс проектирования индивидуального или коллективного пользования, предназначен-

ный для автоматизации операций по подготовке, преобразованию и редактированию текстовой и графической информации и операций по взаимодействию пользователя с системой автоматизированного проектирования (САПР) в процессе разработки изделия. АРМ имеют модификации в зависимости от области применения, например для разработки радиоэлектронной аппаратуры предназначены АРМ-Р, в машиностроении — АРМ-М.

Математическое обеспечение САПР проблемно-ориентировано, разрабатывается и совершенствуется в сотрудничестве с заказчиком; аппаратное обеспечение в основном схоже. В состав аппаратных средств непременно входит устройство полуавтоматического ручного ввода графических документов, таких как электрические схемы и фотомаски в электронной промышленности, архитектурные и строительные чертежи, геодезические планы, схемы коммуникаций городского хозяйства, конструкторские машиностроительные чертежи, выкройки в швейной промышленности. Работа пользователя не требует никаких особых знаний: языком системы является графический язык, подобный языку чертежников и проектировщиков со всеми знакомыми им функциями, такими как чертежная доска, карандаш, линейка, циркуль, ластик.

В устройствах полуавтоматического ввода графической информации описание исходных данных в виде чертежей или эскизов, выполненных на бумаге, кальке или другом диэлектрическом носителе, производится на графическом языке, который позволяет, в отличие от устройств автоматического ввода, избежать ввода каждой точки изображения, требующего ОЗУ большой памяти. Графические данные задаются в виде признаков основных элементов чертежа (отрезков прямой, дуг, окружностей), их координат и размеров, а также в виде типовых элементов чертежей определенного вида (например, элементов электрических схем) с указанием координат их базовых точек, размеров и других признаков.

При полуавтоматическом вводе описание изображений осуществляется человеком, который одновременно со считыванием координат элементов изображения указывает признаки, характеризующие эти элементы. При использовании определенного графического языка получается полное описание графического изображения в форме цифровых кодов. Такие устройства получили название оцифровщиков или дигитайзеров, в первое время их называли также сколками. Дигитайзеры наряду с графическими видеотерминалами находят применение в интерактивных системах машинной графики при реализации диалога человека с ЭВМ.

Дигитайзеры различаются физическими принципами считывания графической информации, технической реализацией и параметрами, такими как размер поля чертежа, точность ввода, емкость БЗУ, скорость передачи считанных данных.

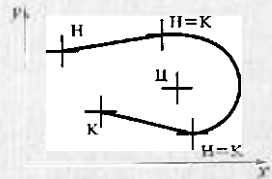


Рис. 5.8. Опорные точки при оцифровке чертежа

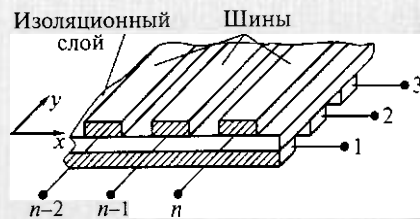


Рис. 5.9. Конструкция поля ввода

Основной составной частью дигитайзеров является устройство, осуществляющее преобразование координат, считываемых съемником с графического документа, расположенного на рабочем поле планшета, в цифровые коды. Считывание координат точки кодируемого изображения выполняется путем выбора этой точки с помощью съемника, наводимого на нее оператором, и посылки в устройство сигнала, осуществляющего запуск цикла считывания.

При оцифровке чертежа применяется линейно-круговая аппроксимация — чертеж представляется в виде отрезков линий, дуг и окружностей. Кодируются (рис. 5.8) только опорные точки — координаты по  $x$  и по  $y$  начал ( $н$ ) и концов ( $к$ ) линий и дуг, а также радиусы  $R$  и координаты центров ( $ц$ ) окружностей и дуг.

В основу работы дигитайзера заложен принцип развертывающего преобразования. На рис. 5.9 показано устройство поля ввода планшета дигитайзера, содержащего ортогональные изолированные друг от друга шины, образующие сетку и дискретизирующие плоскость планшета с шагом порядка 0,5 мм. Шины возбуждаются пос-

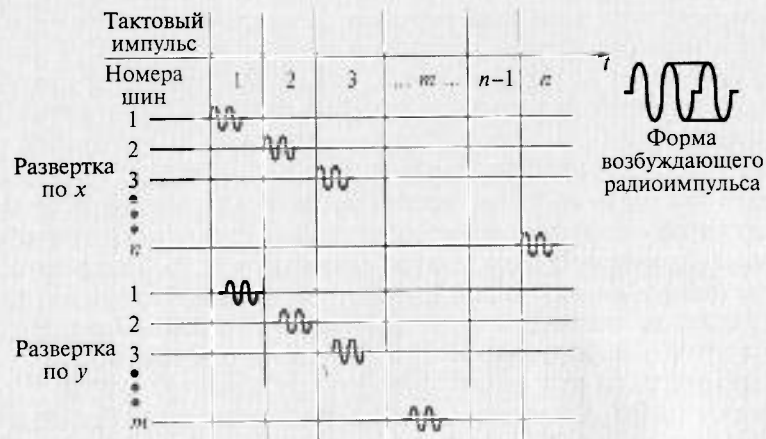


Рис. 5.10. Временные диаграммы напряжений возбуждения шин поля ввода планшета дигитайзера размером  $m \times n$  ( $m < n$ )

ледовательно во времени электрическими радиоимпульсами — импульсно-модулированным синусоидальным напряжением или током высокой частоты (рис. 5.10). В зависимости от характера связи съемника с планшетом (емкостная или индуктивная) шины возбуждаются импульсами напряжения или тока. Положение каждой шины заранее известно и характеризуется определенной координатой  $x$  или  $y$ .

Съемник представляет собой подвижный индуктивный, либо емкостной датчик, содержащий чувствительный элемент — катушку индуктивности или металлический зонд, оформленный в виде пера (карандаша) и связанный кабелем с электронными блоками. Другие названия съемника — штифт, тензор.

На рис. 5.11 приведена укрупненная структурная схема дигитайзера. В процессе работы по кодированию чертежа управляющее устройство подает импульсы на счетчик дискретов по  $x$  ( $Счх$ ), выходы которого через ДШ, входящий в состав узла возбуждения координатных шин  $x$ , соединены с соответствующими шинами

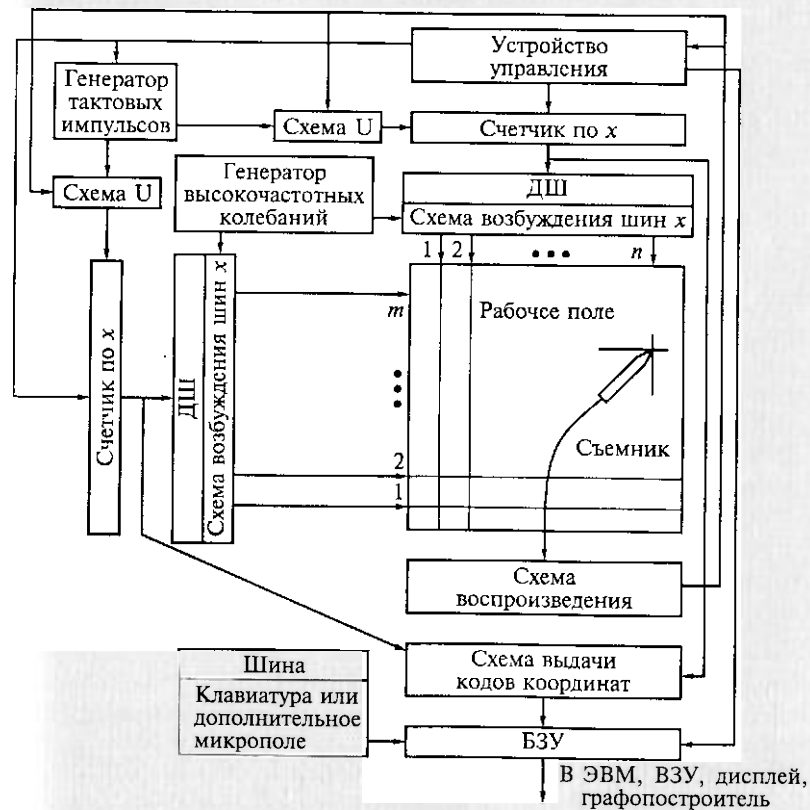


Рис. 5.11. Структурная схема дискретного электрического дигитайзера



планшета. В результате шины возбуждаются последовательно одна за другой со скоростью работы счетчика. По окончании возбуждения шин  $x$  к содержимому счетчика по  $y$  (Счу) добавляется единица, и процесс возбуждения шин  $x$  повторяется. Шины  $y$  также возбуждаются последовательно со скоростью поступления сигналов на вход Счу. Индуктивная связь съемника с планшетом допускает считывание информации с бумажных и других диэлектрических носителей толщиной до 0,5 мм. Скорость оцифровки доходит до 100 точек/с.

В момент возбуждения проходящей под прижатым к планшету съемником шины в последнем вырабатывается сигнал, который усиливается и нормализуется в схеме воспроизведения и останавливает работу счетчиков при наличии поданной оператором команды на выдачу координат. Такая команда подается с замыкаемого оператором микропереключателя, расположенного на съемнике. По этой команде устройство управления сначала блокирует, останавливает работу одного из счетчиков и через схему выдачи кодов координат передает код координаты из этого счетчика в БЗУ, а затем аналогично передает и код другой координаты.

К снятым координатам чертежа может быть добавлена алфавитно-цифровая информация. Для ввода символов имеется клавиатура либо еще одно поле ввода — наборное поле, с помощью которого по указанию пером можно вводить в БЗУ коды типов линий и значения их ширины, коды символов. Координаты положения символов на поле чертежа задаются оператором так же, как и координаты начал и концов линий и центров дуг и окружностей, с помощью пера.

Выдаваемая информация сформирована в кадры, имеющие признаки «начало», «конец», «линия», «окружность», «символ». На рис. 5.12 показан формат кадра. Номера кадров подсчитываются автоматически. Разрядность полей координат зависит от точности ввода (шага координатной сетки) и от размеров чертежа.

Для облегчения работы чертежника-проектировщика, без изменения его рабочих методов за кульманом, выпускаются электромеханические дигитайзеры с крупноформатным полем ввода, от  $600 \times 900$  до  $1000 \times 1500$  мм (рис. 5.13). Съемник индуктивного типа может представлять из себя либо перо, либо витки катушки, расположенные вокруг измерительной линзы с перекрестием, называемой тензором. Съемник перемещается по линейке датчика ли-

Номер кадра	Признак	Начало либо код символа, элемента		Конец либо код символа, элемента		Центр окружности, дуги либо код символа, элемента	
		$x_n$	$y_n$	$x_k$	$y_k$	$x_c$	$y_c$

Рис. 5.12. Пример формата кадра при линейно-круговой аппроксимации

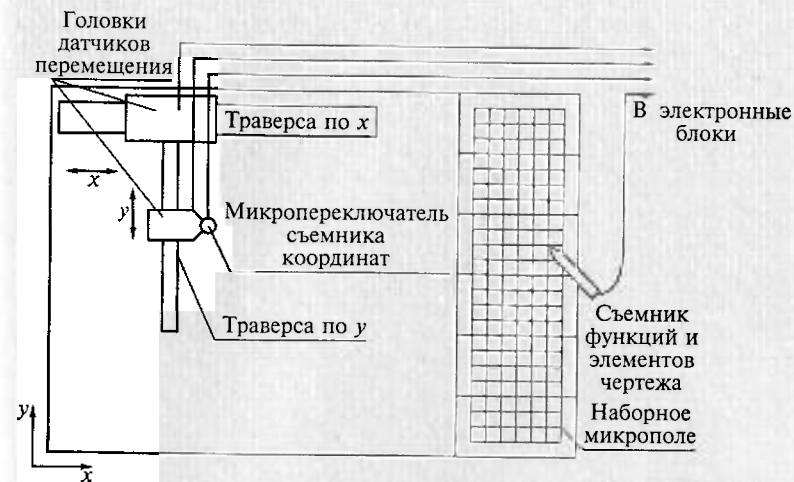


Рис. 5.13. Кинематическая схема электромеханического дигитайзера с датчиками линейных перемещений и наборным микрополем

нейного перемещения по координате  $y$ . По направляющей траверсе перемещается головка второго датчика линейных перемещений — по  $x$ . С помощью тензора указываются места на планшете, куда нужно поместить повторяющиеся детали, символы, убрать (стереть) участок чертежа, изменить масштаб, для чего предназначено дополнительное наборное поле с обозначением этих операций.

Накопленная в БЗУ информация может передаваться непосредственно в ЭВМ на обработку либо записываться в ВЗУ для хранения в архиве. Современные АРМ для удобства оператора и визуальной проверки вводимой информации дополнительно оснащаются графическими дисплеями. Для воспроизведения графической информации АРМ дополняются планшетными или рулонными графопостроителями (плоттеры), печатающими устройствами (принтеры).

**Сканеры.** Сканер — это устройство ввода в ЭВМ информации непосредственно с бумажного документа. Можно вводить тексты, схемы, рисунки, графики, фотографии и другую графическую информацию.

Сканеры являются важнейшим звеном электронных систем обработки документов и необходимым элементом любого «электронного стола». Записывая результаты своей деятельности в файлы и ввода информацию с бумажных документов в ПК с помощью сканера с системой автоматического распознавания образов, можно сделать реальный шаг к созданию систем безбумажного делопроизводства. Сканеры весьма разнообразны, их можно классифицировать по целому ряду признаков. Сканеры бывают черно-белые и цветные.

Черно-белые сканеры могут считывать штриховые изображения и полутоновые. Штриховые изображения не передают полутонов или, иначе, уровней серого. Полутоновые позволяют распознать и передать 16, 64 или 256 уровней серого.

Цветные сканеры работают и с черно-белыми, и с цветными оригиналами. В первом случае они могут использоваться для считывания и штриховых, и полутоновых изображений.

В цветных сканерах используется цветовая модель RGB: сканируемое изображение освещается через вращающийся RGB-светофильтр или от последовательно зажигаемых трех цветных ламп; сигнал, соответствующий каждому основному цвету, обрабатывается отдельно. Число передаваемых цветов колеблется от 256 до 65536 (стандарт High Color) и даже до 16,7 млн (стандарт True Color).

Разрешающая способность сканеров составляет от 75 до 1600 dpi (dot per inch). Конструктивно сканеры бывают ручные и настольные. Настольные, в свою очередь, делятся на планшетные, роликовые и проекционные.

Ручные сканеры конструктивно самые простые: они вручную перемещаются по изображению. С их помощью за один проход вводится лишь небольшое количество строчек изображения (их захват 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 5...50 мм/с (зависит от разрешающей способности).

Планшетные сканеры самые распространенные; в них сканирующая головка перемещается относительно оригинала автоматически; они позволяют сканировать и листовые, и сброшюрованные (книги) документы. Скорость сканирования 2...10 с на страницу (формат A4).

Роликовые сканеры наиболее автоматизированы; в них оригинал автоматически перемещается относительно сканирующей головки, часто имеется автоматическая подача документов, но сканируемые документы только листовые.

Проекционные сканеры внешне напоминают фотоувеличитель — внизу лежит сканируемый документ, а наверху находится сканирующая головка. Сканер оптическим образом сканирует информационный документ и вводит полученную информацию в виде файла в память компьютера.

Файл, создаваемый сканером в памяти машины, называется *битовой картой*. Существуют два формата представления графической информации в файлах компьютера: растровый формат и векторный.

В растровом формате графическое изображение запоминается в файле в виде мозаичного набора множества точек (нулей и единиц), соответствующих пикселям отображения этого изображе-

ния на экране дисплея. Редактировать этот файл средствами стандартных текстовых и графических процессоров не представляется возможным, ибо эти процессоры не работают с мозаичным представлением информации.

Следует также иметь в виду, что битовая карта требует большого объема памяти для своего хранения. Так, битовая карта с 1 листа документа формата A4 (210 × 297 мм) с разрешением 10 точек/мм и без передачи полутонов (штриховое изображение) занимает около 1 Мбайта памяти, она же при воспроизведении 16 оттенков серого — 4 Мбайта, при воспроизведении цветного качественного изображения (стандарт High Color — 65 536 цветов) — 16 Мбайт. Иными словами, при использовании стандарта True Color и разрешающей способности 50 точек/мм для хранения даже одной битовой карты может не хватить емкости НЖМД. Сокращение объема памяти, необходимой для хранения битовых карт, осуществляется различными способами сжатия информации, например TIFF (Tag Image File Format), CTIFF (Compressed TIFF), JPEG, PCX, GIF (Graphics Interchange Format — формат графического обмена) и др. (файлы с битовыми картами имеют соответствующие указанным аббревиатурам расширения).

Наиболее предпочтительным является использование сканера совместно с программами систем распознавания образов, например типа OCR (Optical Character Recognition). Система OCR распознает считанные сканером с документа битовые (мозаичные) контуры символов (букв и цифр) и кодирует их ASCII-кодами, переводя в удобный для текстовых редакторов векторный формат.

Некоторые системы OCR предварительно нужно обучить распознаванию — ввести в память сканера шаблоны и прототипы распознаваемых символов и соответствующие им коды. Сложности возникают при распознавании букв, совпадающих по начертанию в разных алфавитах (например, в латинском (английском) и в русском — кириллица), и разных гарнитур (способов начертания) шрифтов. Но большинство систем не требуют обучения: в их памяти уже заранее помещены распознаваемые символы.

В последние годы появились интеллектуальные программы распознавания образов типа Omnifont, которые опознают символы не по точкам, а по характерной для каждого из них индивидуальной топологии. При наличии системы распознавания образов текст записывается в память ПК уже не в виде битовой карты, а в виде кодов, и его можно редактировать обычными текстовыми редакторами.

Сканер подключается к параллельному порту ПК. Для работы со сканером ПК должен иметь специальный драйвер, желательно драйвер, соответствующий стандарту TWAIN. В последнем случае возможна работа с большим числом TWAIN-совместимых скане-



ров и обработка файлов поддерживающими стандарт TWAIN-программами, например распространенными графическими редакторами Corel Draw, Adobe PhotoShop, PhotoFinish. Большинство драйверов ориентированы на работу с локальным компьютерным интерфейсом SCSI.

### 5.5. Устройства ввода-вывода речевой информации

**Модель речи.** УВВ речевой информации относятся к совмещенным периферийным устройствам, наиболее адаптированным к эргономическим параметрам оператора. Общение с ЭВМ по речевому каналу дает пользователю преимущества, которыми не обладают другие оперативные ПУ:

- общение ведется в двух направлениях, что обеспечивается универсальностью канала;
- общение проходит на естественном языке благодаря удобному интеллектуальному интерфейсу, поэтому не требуется дополнительное обучение оператора;
- речевой канал для пользователя является самым быстродействующим и экономичным с точки зрения энергетических затрат, позволяет освободить зрительный и тактильный аппараты и повысить степень сосредоточенности на основной выполняемой работе;
- УВВ речи легко совмещаются с другими ПУ и не ограничивают свободы действий человека;
- аппаратные средства при современном уровне электронной техники могут быть выполнены малогабаритными, механически прочными и дешевыми.

Устное общение человека с ЭВМ удобно во многих областях человеческой деятельности, особенно там, где оператор перегружен визуальной информацией, например, при управлении сложным технологическим процессом или сложным техническим объектом (летательным аппаратом, системой вооружения), при работе в условиях слабой освещенности. Особенно это полезно при передаче сообщений об экстремальных ситуациях. Ни мигающая красная лампочка, ни громкий зуммер не способны передать информацию о характере и источнике экстремальной ситуации.

Работа оператора в диалоговом режиме при вводе и отладке программ, написанных на алгоритмических языках, при составлении программ для станков с числовым программным управлением также представляется удобной при голосовом программировании.

«Говорящие компьютеры» могли бы с успехом применяться и в непромышленной сфере, например, при обучении родному или иностранному языку, при обучении незрячих и слабослыша-



Рис. 5.14. Три этапа процесса ввода речевого сообщения

щих людей. К далекому будущему следует отнести машины-переводчики с голоса в реальном времени.

В общем случае системы речевого общения строятся на базе специализированных речевых процессоров двух основных типов: анализаторов (УВв) и синтезаторов (УВыв).

Процесс ввода речи, как процесс распознавания слуховых образов, состоит из трех этапов: анализа, идентификации и собственно ввода в ЭВМ (рис. 5.14). Распознавание речи с 1960-х гг. привлекает внимание ученых, к настоящему времени созданы УВв речи в дискретной языковой форме («диктовка»), однако реальные результаты в распознавании слитной речи существенно скромнее. Основные трудности представляет индивидуальность голоса и слитность речи, что усложняет анализ и идентификацию единиц речи — звуков, фонем, слов.

Человеческая речь представляет собой последовательность звуковых колебаний в диапазоне от 50 до 5000 Гц (рис. 5.15, а). Средняя скорость произнесения равна примерно 200 слов/мин. На рис. 5.16 показана фонограмма речевого сообщения «Как наш Пушкин хорош!». Из рисунка видно, что найти границы звуков и слогов не легко, так как речь непрерывна. В результате проведения этапа

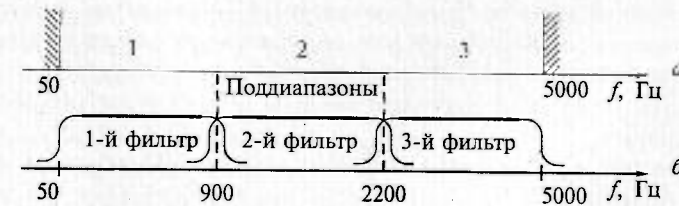


Рис. 5.15. Частотный спектр речи (а) и амплитудно-частотные характеристики полосовых фильтров (б)

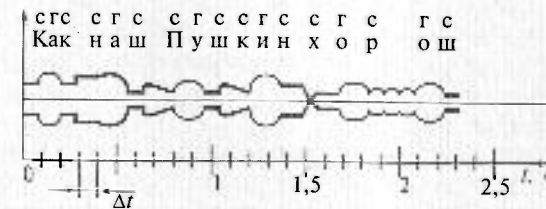


Рис. 5.16. Фонограмма речи



анализа речь должна быть разложена на элементы — фонемы. Фонема, с одной стороны, это элементарная смысловоразличительная единица письменной речи, с другой стороны, это абстрактное обозначение конкретного звука устной речи. Фонема описывается совокупностью различительных признаков, таких как гласная, согласная, глухая (с — согласный; г — гласный звуки), звонкая, шипящая, взрывная, сонорная и др. Например, слова «там» и «дам» отличаются одной фонемой [т] и [д], различительными признаками фонем [т] и [д] является глухость — звонкость. Русский язык содержит 42 фонемы, из них 6 гласных и 36 согласных. В английском языке 20 гласных, из них 5 дифтонгов, и 24 согласных. Во французском языке — 16 гласных и 20 согласных фонем. В потоке речи в зависимости от конкретных условий характеристики фонем изменяются, что приводит к появлению оттенков фонем — аллофонов. Общее число аллофонов гласных равно 480, согласных — 8880.

Акустические характеристики фонем тесно связаны с артикуляционными особенностями их образования — местом и способом. Речевой аппарат человека представляется в виде двух параллельных каналов — ротового и носового, образующих единую акустическую систему, возбуждаемую периодическими колебаниями голосовых связок либо турбулентным шумом. Распространение акустических волн в такой системе описывается уравнением Вебстера:

$$S(x) \frac{\partial}{\partial x} \left( S(x) \frac{\partial P}{\partial x} \right) = \frac{1}{c^2} \frac{\partial^2 P}{\partial t^2},$$

где  $S(x)$  — функция площади сечения речевого тракта вдоль оси  $x$  распространения волн;  $P$  — давление;  $c$  — скорость звука;  $t$  — время. Решение этого уравнения для речевого тракта различных конфигураций, соответствующих различным звукам речи, является основным предметом исследования акустической теории речеобразования.

Интонация и ударение в слитной речи реализуются тремя характеристиками:

- 1) мелодика — изменение частоты основного тона голоса;
- 2) ритмика — текущее изменение длительности звуков и пауз;
- 3) энергетика — текущее изменение интенсивности звука.

Минимальным элементом, из совокупности которых складывается интонация фразы и текста в целом, является акцентная группа — одно или несколько полнозначных слов, объединенные общим просодическим контуром, привязанным к единому групповому ударению.

Существуют несколько методов анализа речи. Первым был применен метод предварительной визуализации речи. При этом анализируются оптические изображения губ оператора. Этот метод построен на опыте языка общения глухонемых и тяготеет к бионике.

Второй метод — метод анализа колебаний голосовых связок, снимаемых с помощью лорингофона. Он, как и первый метод, тяготеет к бионике и пригоден к работе в условиях сильных звуковых помех, например, в кабине летательного аппарата, вблизи прокатного стана. Третий метод анализа — анализ спектральных характеристик речи — энергетических, частотных, временных и амплитудных спектров. Этот метод рассмотрим подробнее в применении к распознаванию отдельных слов, например, команд управления.

Этап анализа распадается на ряд операций. Вначале производится сегментация речи — машинное разбиение речи на фонемы. Для этого речевой сигнал разбивается по времени на 10-миллисекундные кванты  $\Delta t$  (см. рис. 5.16). В каждом кванте производится оценка шести спектральных параметров. 1-, 2-, 3-й параметры — это максимальные значения амплитуд колебаний в трех поддиапазонах: 50...900 Гц, 900...2200 Гц и 2200...5000 Гц. 4-, 5- и 6-й параметры — число переходов через нуль сигналов этих частот, т. е. косвенно определенные значения частот тех колебаний, которые имеют максимальную амплитуду в каждом из трех поддиапазонов.

**Структурная схема анализатора речи.** Анализаторы подразделяются на два основных класса: анализаторы сигналов и анализаторы сообщений. В анализаторах сигналов достигается сжатие (компрессия) информационного потока сигналов с микрофона ( $10^5$  бит/с) за счет учета акустических и статистических характеристик речевого сигнала без обращения к его смысловой функции.

В анализаторах речевых сообщений осуществляется компрессия информационного потока за счет введения операции распознавания смысловых элементов речи (фразы, слова, морфемы, фонемы). Анализаторы речевых сообщений подразделяются на две группы: с ограниченным словарем и универсальные.

Анализаторы ограниченного словаря ориентированы на распознавание заданного конкретной задачей числа речевых команд (обычно 100), т. е. на идентификацию одной из произносимых речевых команд словаря в виде номера команды. На этапе распознавания производится сопоставление эталонов команд с произносимой командой и выбора наиболее схожего эталона.

Универсальные анализаторы ориентированы на текущее распознавание полного набора смысловых элементов речи (фонем или морфем), из которых может быть составлено и в дальнейшем распознано любое слово или слитно произнесенное речевое сообщение. Распознавание осуществляется лингвистическим процессором по правилам, заложенным в базе знаний.

В общем случае системы речевого общения строятся на базе специализированных речевых процессоров. Анализатор реализуется аппаратно и представляет собой специализированное устройство, включающее в себя радиоэлектронные и вычислительные блоки, зачастую называемое предпроцессором. Предпроцессор —

программно-управляемое аналого-цифровое устройство, которое осуществляет спектральный анализ речевого сигнала с последующим преобразованием данных в цифровую форму.

Для получения значений шести спектральных параметров звука (при анализе по методу спектральных характеристик речи) электрический сигнал, полученный с микрофона, пропускается через три полосовых фильтра (рис. 5.17) с полосами пропускания, равными поддиапазонам речевого спектра (см. рис. 5.15, б). В каждом канале трех поддиапазонов пиковый детектор выделяет максимальное значение амплитуд сигналов за время кванта (рис. 5.18); аналого-цифровой преобразователь выдает в двоичном коде значение величины выделенной амплитуды. В процессе работы, особенно в сложных условиях полета, уровень речевого сигнала пилота, поступающего на вход анализатора, может изменяться в несколько раз. Для обеспечения стабильной работы в схему анализатора введены усилители, охваченные глубокой обратной связью, которые осуществляют автоматическую регулировку усиления амплитуды сигнала. В сложных УВв речи служебного назначения число разрядов АЦП обычно берут равным 8-ми, что соответствует 256-ти дискретным уровням сигнала, т.е. дискреты квантования по уровню отличаются друг от друга менее, чем на 0,5%. Так получаются 1-, 2- и 3-й параметры, описывающие звук.

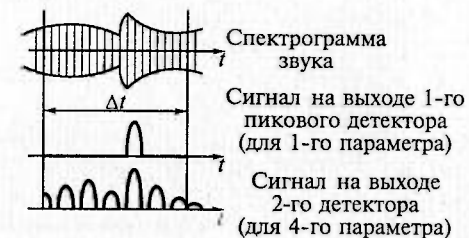
На выходе порогового устройства получают полувольт гармонических составляющих спектра сигнала в данном поддиапазоне (см. рис. 5.18). Число полувольт (число переходов через нуль) подсчитывается счетчиком и выдается в двоичном коде, это 4-, 5- и 6-й параметры.

Затем программно производится объединение или разбиение квантов речи в зависимости от того, установившийся сегмент речи



Рис. 5.17. Структурная схема анализатора речи по методу спектральных характеристик

Рис. 5.18. Сигналы детекторов анализатора речи



или переходной, параметры соседних квантов которого резко меняются. Для этого необходимо измерять сходство между параметрами двух соседних квантов, а затем и сегментов. При большом сходстве кванты объединяются, если же изменение параметров слишком велико, сегменты разбиваются. Таким образом определяются границы фонем.

Число частотных фильтров в значительной мере определяет достоверность распознавания речевой команды, слова, поэтому оно зависит от объема словарного запаса, имеющегося в ПЗУ или в памяти ЭВМ.

Данные с выхода анализатора обычно снимают с частотой ниже минимально возможной частоты основного тона. Этому условию соответствует принятая в УВв речи периодичность опроса данных, которая составляет порядка 100 мкс.

Вторая операция этапа анализа речевого сообщения — машинное описание речи с помощью тех же спектральных характеристик. В зависимости от словаря распознаваемых слов и от задач ввода — слитная речь или дискретная («диктовка»), можно описывать фонемы, слоги, слова. Соответственно и эталоны должны быть описаны такими же спектральными характеристиками. Емкость памяти словаря эталонов зависит от числа распознаваемых фонем и слов.

Эталоны слов в виде их описаний на уровне фонем помешаются в словарь, хранящийся в памяти ЭВМ либо в ПЗУ эталонов. Затем полученное на этапе анализа описание сравнивается по типу ассоциативного поиска методом перебора или на матричном процессоре с описаниями всех эталонов. Вводимой фонеме (слову) приписывается имя того эталона, код описания которого наиболее близок к коду описания распознаваемого слухового образа.

На третьем этапе ввода в ЭВМ поступает последовательность цифровых кодов с существенно меньшим, чем на входе УВв речи, информационным потоком, порядка  $10 \dots 10^4$  бит/с.

**Устройства вывода речевой информации — синтезаторы.** Задача вывода речевой информации сводится к преобразованию машинных кодов, поступающих от ЭВМ, ПЗУ или линии связи, в колебания звуковых частот, составляющих речевой сигнал. Устройства вывода речевых сообщений при любой реализации аппаратно и программно проще, чем устройства ввода, и не требуют больших системных ресурсов. Особенно экономится память при синтезе по фонетическим правилам.



Синтезаторы подразделяются на классы и группы по тем же признакам, что и анализаторы. Классам анализаторов речевых сигналов и сообщений соответствуют такие же классы синтезаторов: речевых сигналов и речевых сообщений. Синтезаторы речевых сообщений, как и анализаторы речевых сообщений (рис. 5.19), делятся на две группы: синтезаторы ограниченного словаря — компиляторы (см. рис. 5.19, а) и универсальные синтезаторы (см. рис. 5.19, б).

В компиляторах любое сложное речевое сообщение может быть получено путем компиляции (простого соединения) элементов речи. Этот метод называется методом синтеза по образцам (по образцам). Элементы речи заранее начитываются диктором, соответствующие им сигналы компрессируются, кодируются и записываются в ПЗУ (см. рис. 5.19, а). Закодированные таким образом речевые элементы в процессе вывода считываются в нужной последовательности, и одновременно восстанавливается речевой сигнал.

На рис. 5.20 приведена структурная схема компилятора. Электрические сигналы с генератора основного тона (при формировании звонких звуков) и генератора шума (при формировании глухих звуков) через переключатель поступают на многозвенный управляющий цифровой фильтр. Фильтр и ЦАП управляются от ЗУ кадров, куда заранее из памяти описаний образцов заносится выбранное устройством управления описание сообщения, представляющее собой последовательность кадров. Каждый кадр содержит параметры громкости, частоты основного тона, длительности синтезируемого звука и ряд других. Кадры опрашиваются с интервалом квантования по времени, равным 20 мс. Если основной тон модулировать по частоте в соответствии с таблицей частот музыкальных нот можно синтезировать пение.



Рис. 5.19. Структуры устройств вывода речевых сообщений:

а — синтезатор ограниченного словаря — компилятор; б — универсальный синтезатор

При использовании в качестве элементов речи отдельных слогов, подставляемых в определенное место стандартной фразы, компиляционный метод позволяет получить удовлетворительный по качеству синтез речи. Попытки добиться высококачественного синтеза фонем или аллофонов не привели пока к положительным результатам, поскольку все эти элементы тесно связаны между собой внутри фразы. В слитной речи не существует аналогов этих элементов, произнесенных изолированно, и, наоборот, речь из изолированно произнесенных элементов звучит ненатурально. В связи с этим подготовка словаря в компиляторах представляет собой самостоятельную и сложную проблему.

При разработке универсальных синтезаторов речевых сообщений стремятся получить функциональную модель речеобразования, адекватную реально существующим языковым и акустическим явлениям. Речь формируется по правилам фонетики языка, поэтому метод получил также названия синтеза по правилам и фонетический метод. На входе такой модели — орфографический или фонемный текст произвольного содержания, на выходе — звучащая речь. По своему существу такие синтезаторы являются кибернетической функциональной моделью чтения текста человеком. В базе знаний синтезатора хранится не только информация об элементарных единицах речи (эталон фонем, аллофонов и интоном), но и алгоритмические правила их модификации в зависимости от конкретного контекста звуковой реализации. Процесс применения этих правил к эталонам фонем и интоном для входного синтезируемого текста реализуется лингвистическим процессором. Универсальные синтезаторы отличаются малым расходом памяти на один элемент речи и неограниченностью словаря синтезируемых сообщений.

Одним из основных параметров, характеризующих синтезаторы речи, является емкость памяти, необходимая для хранения объема речи, произносимого за 1 с.



Рис. 5.20. Структурная схема компилятора



При формировании по образцам словаря компилируемых слов объем описаний образцов, хранимых в ПЗУ, ограничен, при формировании по правилам словарь практически не ограничен.

### 5.6. Устройства отображения информации — дисплеи

**Классификация дисплеев.** Средства отображения (визуализации) текстовой и графической информации, называемые также дисплеями, получили наибольшее распространение как оперативные устройства вывода индивидуального пользования. Дополненные устройствами ручного ввода информации — клавиатурой, мышью, световым пером, дигитайзером, они являются совмещенными устройствами ввода-вывода информации, обеспечивающими работу пользователя в интерактивном режиме.

Основу любого дисплея составляет индикатор — устройство, осуществляющее непосредственное преобразование электрических сигналов в наглядное изображение текстовой либо графической информации (рис. 5.21). В роли индикатора могут выступать разнообразные приборы, предназначенные для получения видимых изображений (традиционные ЭЛТ и другие электронно-лучевые приборы, плоские панели различных видов) и имеющие дополнительные особые свойства, например, воспроизведение цветного изображения или хранение визуальной информации при выключенном питании. Работу всех устройств дисплея организует УУ, в котором необходимые процедуры по управлению отображением и редактированием могут быть реализованы либо аппаратными средствами, либо сочетанием аппаратных и программных средств. В последнем случае в качестве УУ используется дисплейный процессор.

Коды отображаемых знаков и элементов чертежа через блок согласования поступают от ЭВМ или с клавиатуры либо светового пера и хранятся в БЗУ. Эти коды преобразуются специальным образом в сигналы, с помощью которых на индикаторе формиру-



Рис. 5.21. Обобщенная структурная схема дисплея

ется изображение знака или элемента чертежа. Функциональная клавиатура позволяет управлять положением указателя (маркера) на экране индикатора, устанавливать режим работы, стирать изображение или его часть и т. п.

Область применения текстовых дисплеев чрезвычайно широка и охватывает все сферы использования проблемно-ориентированных вычислительных комплексов. Это наиболее распространенный, массовый и доступный вид устройства отображения информации при ее выводе и вводе. Дисплеи применяются для отображения информации, получаемой удаленным абонентским пунктом; для передачи скорректированной или вновь введенной оператором информации по каналам связи с ЭВМ; центральные дисплеи контролируют функционирование системы и располагаются в непосредственной близости от ЭВМ.

Графические дисплеи предназначены для активного взаимодействия человека с ЭВМ в диалоговом режиме на графическом языке (чертежи, схемы, графики и т. п.) в различных системах автоматизации проектирования, автоматизации научных исследований, моделирования, автоматизированные системы управления (АСУ) полетами и наблюдения за воздушной обстановкой, автоматизации технологических процессов, робототехнике, процессе обучения и др.

Все графические дисплеи способны отображать и текстовую информацию, а некоторые текстовые дисплеи содержат аппаратуру и программы построения простейших графических изображений. Это является одной из причин, по которым практически очень сложно провести четкую классификацию дисплеев по какому-то одному признаку ввиду многообразия свойств и функциональных возможностей устройств отображения информации. В основу классификации, приведенной на рис. 5.22, положен принцип объеди-



Рис. 5.22. Классификация дисплеев

нения дисплеев по виду отображаемой информации, типу индикатора, способам построения изображения, размерности воспроизводимой информации и, наконец, по уровню «интеллектуальности».

Уровень интеллектуальности определяется набором системных функций, которые реализуются в самом дисплее. В простом дисплее аппаратно реализуется ограниченный набор функций по отображению, хранению, редактированию и вводу-выводу данных. Такие дисплеи выполняются в качестве специализированных УВВ различного применения.

Микропрограммируемый дисплей позволяет изменять и расширять набор функций, а также производить предварительную несложную обработку данных. Интеллектуальный дисплей, как правило, содержит процессор, ОЗУ, ПЗУ, ВЗУ, УВВ и позволяет реализовать широкий набор функций по обработке и отображению данных. Примером таких дисплеев служат современные графические дисплеи.

**Способы формирования изображения на экране телевизионного дисплея.** Основой телевизионного дисплея является приемная ЭЛТ — кинескоп. Первый кинескоп был изобретен в 1907 г. русским ученым Борисом Розингом и в 1920-е гг. усовершенствован другим нашим соотечественником Владимиром Зворыкиным. Он же сформулировал принципы цветного кинескопа. Устройство кинескопа, в части создания электронного пучка (луча) и обеспечения его сканирования по экрану, весьма схоже с устройством передающей ЭЛТ. На рис. 5.23 показано устройство кинескопа. Электронная пушка, называемая также электронным прожектором, состоит из электрода накала (подогревателя), катода, испускающего электроны при его нагреве до температуры порядка 1100 К, и фокусирующей системы. В состав электронной пушки входит еще и управляющий электрод (модулятор) в виде сетки, расположенной перпендикулярно к продольной оси кинескопа.

Под действием электростатического ускоряющего поля, создаваемого положительным напряжением между катодом и анодом,

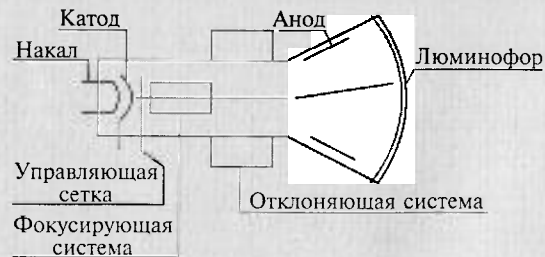


Рис. 5.23. Устройство монохромного кинескопа

электроны разгоняются и устремляются в сторону анода, образуя электронный луч, достигающий экрана.

Экран изнутри покрыт люминофором — химическим соединением, испускающим полезное излучение под воздействием электрических полей или лучистой энергии. Наиболее важной для индикации является фосфоресценция, т.е. испускание света после возбуждения. Механизм генерирования свечения заключается в передаче энергии от быстрых падающих электронов электронам, связанным с кристаллом люминофора. В результате этого последние переходят в возбужденное состояние. При их возвращении в нормальное состояние избыточная энергия выделяется в виде лучистой энергии. Люминофор в точке бомбардировки его электронным лучом начинает светиться. Минимальный размер пятна на экране составляет 0,25...0,4 мм и определяет разрешающую способность ЭЛТ. Другая важная характеристика люминофора — послесвечение. Яркость пятна быстро падает после прекращения бомбардировки люминофора потоком электронов, так, через 0,2 с после выключения электронного луча яркость уменьшается более чем наполовину. Для получения устойчивого немерцающего изображения необходимо многократно, с частотой 30...60 раз в секунду повторять изображение, т.е. восстанавливать электронным лучом видимые точки. Процесс восстановления изображения называется регенерацией.

На расположенную на пути электронного луча сетку подается положительное управляющее напряжение, от величины которого зависит, какая часть электронов луча сможет пройти через нее и достичь люминофора. От этого будет зависеть интенсивность свечения пятна луча в точке экрана — яркость. Управляя величиной подаваемого на модулятор напряжения яркости, осуществляют модуляцию луча и, таким образом, управляют яркостью светящегося пятна луча на экране.

В случае монохромной ЭЛТ люминофор нанесен равномерно на внутреннюю часть экрана. В случае цветной ЭЛТ изнутри на экран нанесены три типа люминофора, способные под воздействием электронной бомбардировки светиться каждый своим цветом: окись иттрия, активированная европием, — красным; смесь сернистых цинка и кадмия, активированная серебром, — синим; сернистый цинк, активированный серебром, — зеленым. Именно эти три спектральных составляющих видимого света выбраны в качестве основных потому, что смешением указанных цветов могут быть получены почти все встречающиеся в природе цветовые тона и большое число градаций их насыщенности. Такой способ образования цветов и их оттенков, используемый в телевидении, называется слагательным или аддитивным, в отличие от вычитательного или субтрактивного способа смешения, применяемого в полиграфии, живописи и цветном кино, при котором в качестве основных цветов используют голубой, желтый и пурпурный.

Люминофоры трех цветов нанесены на экран в виде сотен тысяч (до полутора миллионов в зависимости от размера экрана) размещенных в строго определенном порядке мельчайших точек, объединенных в триады. Внутри цветной ЭЛТ находятся три электронных пушки, по одной на каждую составляющую цвета, причем модуляция каждого из трех электронных лучей производится отдельно. Созданные одновременно три изображения - зеленое, синее и красное смешиваются и преобразуются в зрительном аппарате наблюдателя в многоцветное изображение.

Само изображение создается путем сканирования электронного луча по экрану, достигаемого с помощью отклоняющей системы, подобной отклоняющей системе передающей ЭЛТ. В зависимости от формы напряжений или токов, подаваемых в горизонтальную и вертикальную отклоняющие системы с генераторов развертки, луч ЭЛТ сканирует либо по строкам, образуя растр, подобный телевизионному, либо непрерывно движется по экрану, вычерчивая линии изображения.

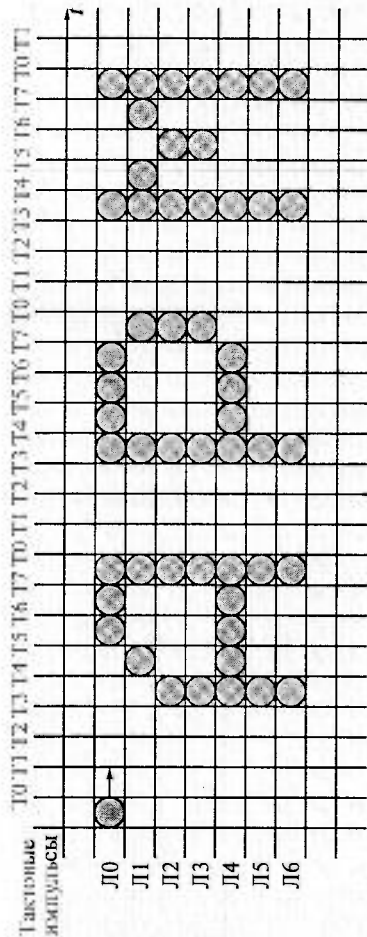


Рис. 5.24. Растровый способ формирования знаков

В первом случае, при растровой развертке, изображение формируется путем управления интенсивностью свечения точек строк. Изображение, как мозаика, складывается из точек, разноцветных в случае цветной ЭЛТ. Такой способ создания изображения называется растровым. Перемещение луча по экрану не зависит от изображаемых знаков и осуществляется всегда по одной и той же невидимой сетке, образованной строками (линиями).

Знак изображается в пределах предусмотренного знакоместа. На изображение строки текста затрачивается, например, 16 телевизионных линий, из них 7 линий отводятся для собственно знаков (рис. 5.24), а следующие 9 — для промежутка между строками текста. Луч движется слева направо по всей рабочей части экрана сначала по линии Л0, затем по линии Л1 и т.д.

Схема синхронизации устройства управления дисплеем вырабатывает сигналы линий Л0, Л1, ..., Л6, запускающие генератор строчной развертки, а также сигналы Т0, Т1, ..., Т7 тактов формирования элементов знаков. Такты Т0, Т1, Т2 формируют промежуток между знаками, в это время луч гасится. Остальные такты, с 3-го по 7-й, участвуют в изображении знака. При движении пятна луча по линии Л0 модуляция луча формирует только верхние элементы всех знаков данной текстовой строки, при движении по второй линии — вторые элементы и т.д.

Таким образом, для формирования на линии элементов знаков нужно иметь сведения: какой знак должен изображаться (код знака), на какой линии находится пятно луча (сигналы Л0... Л7) и на каком столбце знакоместа должно находиться пятно (сигналы Т3, Т4, ..., Т7). Эти сведения в виде микропрограмм формирования знаков путем модуляции луча хранятся в ПЗУ знакогенератора. В процессе отображения текста ПЗУ выдает сигналы модуляции, соответствующие знаку и положению луча на сетке раstra.

Второй способ формирования знаков — функциональный. Перемещение луча осуществляется двумя электромагнитными отклоняющими системами. Одна из них — знаковая система (ЗС) перемещает пятно в пределах знакоместа, другая — координатная система (КС) — после завершения изображения некоторого знака перемещает погашенное пятно в исходную точку следующего соседнего знакоместа. При этом способе устройство управления должно обеспечивать перемещение луча с постоянной скоростью по контуру изображаемого знака (рис. 5.25) и его модуляцию в нужные моменты времени. Здесь I...IV столбцы знакоместа используются для изображения знака, V столбец — для формирования промежутка между соседними знаками. На рисунке цифрами отмечена траектория луча при изображении буквы Б. Микропрограммы формирования знаков также хранятся в ПЗУ знакогенератора.

**Устройство плоских экранов.** Помимо ЭЛТ, совместно с ЭВМ могут использоваться и другие типы устройств отображения информации, позволяющие уменьшить габариты экрана и даже сделать его плоским. К ним принадлежат плоские экраны, выполненные в виде тонкой плоской панели. Такие устройства отображения разделяются на светоизлучающие и светоклапанные, или пассивные. К первым относятся плазменные, светодиодные и электролюминесцентные индикаторы. К светоизлучающим индикаторам относятся и ЭЛТ. Вторая категория включает в себя индикаторы на жидких кристаллах (ИЖК). Пассивные индикаторы потребляют значительно меньшую

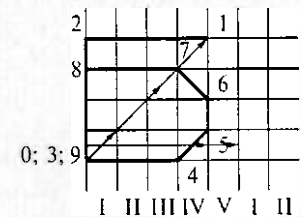


Рис. 5.25. Функциональный способ формирования знаков



мощность, чем светоизлучающие, что является их преимуществом, особенно при изготовлении портативной аппаратуры, например, ноутбуков.

Плазменные индикаторы представляют собой заманчивую альтернативу ЭЛТ в тех ситуациях, когда достаточно воспроизводить небольшие объемы информации (до 240 знаков). Индикаторы этого типа имеют малые габаритные размеры, массу и потребляемую мощность, имеют высокую яркость и контрастность, отличаются простотой формирования изображения, отсутствием мерцаний и искажений. Конструкция плазменного дисплея (плазменной панели) приведена на рис. 5.26.

Основу плазменной панели составляют две собранные с зазором стеклянные пластины, на внутренних сторонах которых расположены вертикальные и горизонтальные полупрозрачные электроды. Пространство между панелями заполнено смесью инертных газов и запаяно. Электроды выведены наружу и подключены к формирователям сигналов записи. Заданная на экране точка зажигается при одновременной подаче на соответствующую пару электродов импульсов записи. Свечение возникает за счет газового разряда и может поддерживаться, а изображение запоминаться, если на соответствующие электроды после окончания импульса записи подавать переменное напряжение высокой частоты. Уменьшение напряжения ведет к исчезновению свечения. Плазменная панель позволяет получать изображения растрового типа. Скорость записи и стирания изображения около 20 мкс на один символ, она может быть повышена, если осуществлять параллельную запись или стирание нескольких или всех точек, расположенных в ряд по горизонтальной шине. Наиболее часто плазменный индикатор komponуется из разрядов сгруппированных ячеек (размером 5 × 7 или 7 × 9 ячеек), удобных для индикации символов. Изготавливаются и матрицы с равномерным распределением элементов по всему полю индикации, которые применяются как экраны, способные отображать графическую информацию.

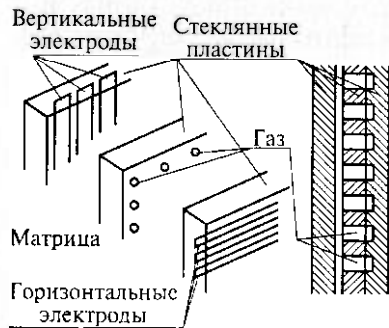


Рис. 5.26. Конструкция плазменной панели

Смесь газов на основе неона создает свечение красного или оранжевого цвета. В цветных плазменных панелях используют различные люминофоры. Из плоских плазменных панелей можно собирать экраны большей размерности. Плазменные индикаторы больших размеров оказываются, однако, дороже ЭЛТ. К недостаткам плазменных панелей можно отнести низкую разрешающую способ-

ность адресации (порядка 0,5 мм) и малую скорость записи и стирания информации.

Светодиодные индикаторы обычно используются, если число воспроизводимых знаков не превышает 40, например, в калькуляторах, цифровых часах, весах и других приборах. Светодиодный индикатор состоит из матрицы светодиодов — полупроводниковых диодов, излучающих свет при подаче прямого смещения, то есть постоянного напряжения, необходимого для поддержания электрического тока в диоде. Светодиоды выполнены из соединений галлия, мышьяка и фосфора; из сульфидов кадмия и цинка; из теллурида цинка; из арсенида галлия и др.

Электролюминесцентные (ЭЛ) индикаторы вначале применялись для экранов портативных компьютеров — ноутбуков. Большинство форм люминесценции в области индикации создается люминофорами. Электролюминесценция — свечение люминофора при приложении к нему электрического поля. Типичная конструкция электролюминесцентных панелей (ЭЛП) приведена на рис. 5.27.

Это мозаичный экран, изображение на котором формируется из точек. Подложка выполнена из стекла. ЭЛ пленка, состоящая из легированного марганцем сульфида цинка, заключена между двумя прозрачными изолирующими слоями. На эти слои нанесены тонкопленочные вертикальные и горизонтальные электроды (столбцовые и строчные). Электролюминесценция возникает на участках ЭЛ слоя в узлах матрицы, под действием электрического поля. Поле образуется импульсами противоположной полярности, подаваемыми на систему строчных и столбцовых электродов. Строчные электроды возбуждаются последовательно, независимо от формируемого изображения; на столбцовые электроды сигнал подается, если в соответствующем узле матрицы на экране должна быть получена яркая точка. Люминофор обладает послесвечением, поэтому изображение на таком экране не требует регенерации. Изображение настолько контрастно, что его можно видеть и при ярком солнечном свете. Стирается изображение отключением напряжения на всех столбцовых электродах.

Жидкокристаллические индикаторы выполнены на основе особой группы веществ, относящихся к классу полимеров и их сис-



Рис. 5.27. Конструкция электролюминесцентной панели

тем, представляющих собой промежуточную фазу между твердым и жидким состояниями. Эти вещества состоят из сигарообразных органических молекул, обладающих текучестью подобно жидкостям, но имеют молекулярный порядок твердых веществ.

В жидких кристаллах (ЖК) наблюдается ряд электрооптических эффектов, из которых наибольшее практическое применение находит динамическое рассеяние света: при помещении ЖК в электрическое поле напряженностью не менее 200 В/см в них возникают турбулентные области, на которых происходит рассеяние падающего света. При снятии поля турбулентность исчезает, расположение молекул упорядочивается и ЖК снова становятся прозрачными. Чтобы управлять прозрачностью с помощью более низких напряжений, необходимы довольно тонкие, микронные слои ЖК (обычно порядка 100 мкм). В пленке толщиной 20 мкм при напряженности электрического поля 0,3 В/мкм можно получить степень рассеяния, отвечающую контрастности 1:7.

ЖК могут работать на просвет и на отражение. В обоих случаях ЖК представляют собой диэлектрический слой, заключенный между двумя металлическими обкладками, роль которых играют вертикальные и горизонтальные электроды. При работе на просвет электроды полупрозрачны. При работе на отражение внешние электроды полупрозрачны, задние — отражающие.

При работе на постоянном напряжении происходит электролиз, приводящий к разрушению вещества. Для уменьшения этого эффекта питание осуществляют переменным током частотой 20... 100 Гц. При этом потребляемая мощность составляет 10 мкВт/см<sup>2</sup> (индикаторы на светодиодах потребляют на 2... 3 порядка больше).

С помощью ЖК получены и цветные индикаторы. В этом случае панель выполняется трехслойной, как и в цветной ЭЛТ, по красному, синему и зеленому цвету. Для создания видимого изображения необходим внешний источник света, а само устройство выполняет функцию модуляции проходящего или отраженного света. ИЖК характеризуются высоким контрастом при ярком внешнем освещении, возможностью применения метода проекции, совместимостью с технологией интегральных микросхем, простотой конструкции. Однако ИЖК имеют такие недостатки, как ограниченный рабочий диапазон температур (от -30 до +130 °С), довольно большая инерционность и ограниченный угол наблюдения.

На ИЖК строятся матричные экраны для портативных компьютеров. Ноутбуки обеспечиваются цветными экранами двух типов: с двойным сканированием (DSTN) и с активной матрицей (TFT). Экраны типа DSTN дают менее четкое изображение, экраны типа TFT по качеству изображения почти не уступают экранам на ЭЛТ. Максимальный размер экранов ноутбуков — 11,3... 12,1 дюйма, но есть модели мониторов с диагональю экрана 13,3 дюйма. Жидкокристаллические дисплеи ноутбуков дороже, чем монито-

ры на ЭЛТ настольных персональных компьютеров, но безопаснее для здоровья, потому что не дают сильного электромагнитного облучения, а всего лишь отражают спектральные составляющие освещающего их света.

## 5.7. Устройства вывода информации — принтеры и графопостроители

**Принтеры.** *Принтер* — это внешнее устройство ЭВМ, предназначенное для вывода информации на твердый носитель (бумагу) в символьном или графическом виде.

Классификация принтеров может быть проведена по следующим критериям: по способу вывода, по принципу формирования изображения, по способу регистрации и по принципу управления процессом печати.

По способу вывода изображения принтеры делятся на две группы: символьные и графические. Символьные принтеры могут выводить информацию в виде отдельных символов по мере их поступления в ПУ. При этом за один цикл печати формируется один знак (посимвольные печатающие устройства). В построчных ПУ вывод на печать осуществляется только после заполнения БЗУ, которое по емкости равно одной строке. Постраничные ПУ за один цикл печати формируют и распечатывают целую страницу.

Графические ПУ выводят информацию не целыми символами, а отдельными точками или линиями. Количество точек на единицу длины определяет разрешающую способность принтера, которая имеет разную величину в зависимости от направления: по горизонтали и по вертикали. В принтерах этого типа каждая точка имеет свои координаты, которые являются адресом этой точки.

По принципу формирования выводимого изображения ПУ делятся на три вида: литерные, матричные и координатные (векторные).

Литерные устройства выводят информацию в виде символов, каждый из которых является графическим примитивом данного устройства. Литеры сформированы при изготовлении принтера, нанесены на специальные рычаги или литерные колеса-шрифтоносители и при эксплуатации принтера без замены шрифтоносителя не изменяются.

Матричные устройства выводят информацию в виде символов, сформированных из отдельных точек, объединенных в символьную матрицу. Печатающая головка матричного принтера имеет вертикальный ряд иголок, каждая из которых может сделать оттиск самого маленького элемента изображения — пиксела (точки). Печать символа происходит при перемещении головки по горизонтали. Если подлежащий печати символ имеет размеры боль-



шие, чем может обеспечить печатающая головка, такой символ печатается за несколько проходов, после каждого из которых осуществляется перемещение по вертикали (относительно печатающей головки) носителя изображения (например, бумаги).

Одной из наиболее существенных характеристик матричного принтера является количество игловок, с помощью которых формируется изображение. В печатающей головке принтера могут находиться 9, 18 или 24 иглолки, которые располагаются вертикально в 1...2 ряда. От количества игловок, их расположения и размера зависят качество и скорость печати. Качество печати регулируется переключением режима: *Draft* (черновая печать за один проход), *LQ* (чистовая печать), *NLQ* (получистовая печать) и определяет скорость печати (количество знаков в секунду) и разрешающую способность (количество точек, печатаемых на одном дюйме). Обычно матричные принтеры имеют диаметр иглолки около 0,2 мм, скорость печати от 180 до 400 символов в секунду (в режиме *Draft*), разрешение 360 × 360 точек на дюйм.

Координатные устройства — плоттеры, графопостроители — выводят информацию как текстовую, так и графическую либо в виде отдельно адресуемых точек, либо сформированную из различных линий — так называемое «штриховое» изображение. При решении экономических задач координатные печатающие устройства используются редко.

По способу регистрации изображения ПУ делятся на ударные и безударные.

ПУ ударного действия формируют изображение на бумаге, сжимая с помощью удара на короткий промежуток времени рельефное изображение символа или его части, красящей ленты и бумаги. Иногда краска наносится на поверхность литеры, красящая лента в этом случае отсутствует.

Существуют принтеры, использующие ударочувствительную бумагу, цвет которой изменяется за счет механического воздействия на нее без дополнительного нанесения краски.

ПУ безударного действия характеризуются тем, что изображение на бумагу наносится через промежуточный носитель, чувствительный к электрическому воздействию, электростатическому полю, магнитному полю и др. Обычно промежуточный носитель исполняется в виде барабана. Изображение на него наносится лазерным лучом, с помощью магнитных головок и др. Затем изображение на промежуточном носителе проявляется — на поверхность барабана наносится смесь сухого красителя с порошком, «прилипающим» к зафиксированному на барабане изображению (например, если изображение наносилось на барабан магнитным полем, в качестве порошка используются мелкие металлические опилки). После этого к барабану «прикатывается» чистый лист бумаги, на который переносится краситель с бараба-

на. Лист с накатанным на него красителем подвергается термообработке — нагревается до расплавления красителя, который в жидком виде проникает в поры бумаги и хорошо закрепляется на ней. После расплавления красителя отдельные точки сливаются в единое целое, поэтому качество изображения получается высоким. Разрешающая способность таких принтеров очень высока. Например, лазерные принтеры *LaserJet III* и *LaserJet IV* обеспечивают 300...600 точек на дюйм. Скорость печати у лазерных принтеров измеряется количеством страниц в минуту и составляет 4...12 страниц/мин при монохромной печати и 2...6 страниц/мин при цветной печати.

К ПУ безударного действия также относятся термические принтеры, использующие термочувствительную бумагу, которая изменяет свой цвет под действием тепловых лучей, и струйные принтеры, у которых жидкий краситель (чернила) находится в печатающей головке. Головка имеет отверстия, через которые краситель не может вылиться из-за сил поверхностного натяжения. Внутри головки находится терморезистор, который при подаче на него импульса тока разогревает краситель, увеличивая его испарение. Пары красителя проникают через отверстие в головке и попадают на бумагу в виде капли. Благодаря тому, что головка может работать с несколькими красителями, выпускаются и цветные струйные принтеры. Длительностью нагрева терморезистора можно регулировать количество выбрасываемых чернил, а следовательно, размеры и яркость точки. Разрешающая способность струйных принтеров составляет от 360 до 720 точек на дюйм. Скорость печати 4...10 страниц в минуту. Печатающая головка струйного принтера содержит от 48 до 416 отверстий (сопел).

Несмотря на большое разнообразие типов принтеров, различия принципов управления печатью касаются в основном способов знакогенерации. Матричный принтер, а также большинство принтеров безударного действия формируют изображение из отдельных точек, образующих символьную матрицу. Обычно кодовые комбинации, характеризующие форму символов на матрице, образует матричный шрифт (фонт), который заносится в запоминающее устройство знакогенератора. Каждый шрифт (фонт) представляет собой комплект букв, цифр и специальных символов, оформленных в соответствии с едиными требованиями.

Альтернативой матричной является векторная знакогенерация. Векторные шрифты строятся на базе математического описания формы символа. Для векторной знакогенерации характерна легкость изменения формы, размеров, наклона шрифта, поэтому они называются свободно масштабируемыми шрифтами. Генерация шрифтов и управление процессом вывода изображения производятся для векторных шрифтов на специальном языке (PCL, PostScript).



При использовании векторных шрифтов математическое описание формы каждого символа, с учетом его размеров и стиля, преобразуется перед печатью в матричную форму, в соответствии с конкретными размерами матрицы принтера. Поэтому форма шрифта, выводимого на различные устройства, остается примерно постоянной, не зависящей от расстояний между точками и размеров символов. Для такого преобразования в состав ПУ включаются вычислительные устройства — ускорители, в качестве которых нашли применение матричные процессоры, транспьютеры. Это накладывает серьезный отпечаток на архитектуру системы управления принтером.

**Графопостроители.** Полученный в результате проектирования комплект технической документации на изготовление какого-либо изделия включает в себя как текстовые, так и графические документы. Текстовую документацию получают с помощью устройств вывода — ПУ (принтеров), преобразующих закодированную информацию, выходящую с процессора, в удобную для чтения форму на бумажном носителе. Графические документы — чертежи, на поле которых имеются соответствующие тексты, для объектов проектирования различных классов составляют 80...90 % от общего объема документации. Для регистрации графической информации используются чертежно-графические автоматы — графопостроители (плоттеры).

Графопостроители преобразуют выводимые из ЭВМ данные из цифровой формы в форму рисунка, чертежа или графика на бумаге, кальке, полиэфирном листе или подобном им носителе информации. Современные плоттеры предназначены для построения машиностроительных и строительных чертежей, функциональных и принципиальных электрических схем, чертежей печатного монтажа, метеорологических карт, регистрации результатов моделирования.

Существуют многообразные типы плоттеров, отвечающих различным требованиям по размерам, точности, скорости вычерчивания, количеству воспроизводимых цветов и другим показателям. Графопостроители систематизируют по двум основным признакам: по способу формирования изображения на носителе и по способу управления (рис. 5.28).

Графопостроитель, как правило, включает в себя собственно построитель — электромеханическую часть, перемещающую каретку с пером, а иногда и носитель, блок преобразования и подготовки данных и блок управления. Конструкции построителей электромеханических плоттеров показаны на рис. 5.29. Регистрирующим органом планшетных и барабанных (рулонных) плоттеров служит пишущий узел — каретка с закрепленным шариковым стержнем, карандашом, пером с чернилами, резцом и др. Узел записи может иметь одно или несколько (до восьми, как в



Рис. 5.28. Классификация графопостроителей

плоттере типа A0 Union итальянского производства), перьев различных цветов и толщин. В более быстродействующих растровых плоттерах узел записи представляет собой «гребенку» металлических электродов и контрэлектродов, образующих растр.

Разновидностью электромеханических чертежных автоматов являются координатографы, отличающиеся повышенной точностью и способом нанесения изображения. Изображение наносится путем вырезания, гравирования, экспонирования, а также с помощью фрезерных и режущих головок, фото головок с набором диафрагменных дисков и других специальных инструментов. Применяются координатографы, например, для изготовления высокоточных фотошаблонов. Координатографы, как правило, работают в автономном режиме с управлением с магнитносителей, подготовленных на ЭВМ.

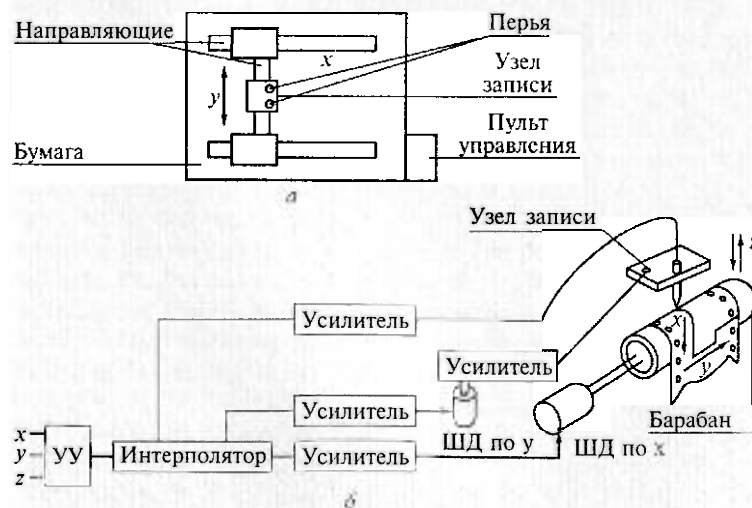


Рис. 5.29. Конструкция электромеханических графопостроителей: а — планшетный; б — барабанный

В основе работы графопостроителя лежит принцип преобразования команд на входе в пропорциональное перемещение пишущего узла. В планшетном графопостроителе пишущий узел приводится в движение от шаговых электродвигателей (ШД) и механизмов, преобразующих вращательное движение валов ШД в поступательное движение пишущего узла. Это преобразование движения осуществляется блочно-троссовой системой, ленточными барабанами, ходовыми винтами и зубчатыми рейками.

Управление шаговым двигателем значительно проще, чем двигателем постоянного или переменного тока, и заключается в преобразовании цифрового кода в унитарный (число-импульсный) код. Угол поворота вала ШД пропорционален числу импульсов, поданных на его обмотки. Частота вращения вала, а отсюда и скорость вычерчивания, определяется частотой следования импульсов. Преобразование цифрового кода в унитарный выполняет интерполятор. При одновременном или попеременном вращении валов обоих двигателей пишущий узел обеспечивает вычерчивание горизонтальных или наклонных линий. Число возможных направлений, по которым может перемещаться пишущий узел, для разных устройств различно и меняется от 4 (перемещение только по координатам  $x$  и  $y$ ) до 32-х (с шагом в  $11^{\circ}25'$ ). Минимальная величина перемещения пишущего узла называется единичным шагом и составляет для разных типов плоттеров 0,1; 0,05; 0,025; 0,01; 0,005 мм. Скорость вычерчивания линий достигает 2300 мм/с (скоростной графопостроитель типа AccuPlot 700 производства Gerber Garment Technology, Inc.).

В планшетных плоттерах носитель неподвижен, в барабанных носитель имеет краевую перфорацию и перемещается ведущим барабаном в прямом и обратном направлениях. При одновременных перемещениях узла записи и бумаги, оба движения складываются, образуя требуемую траекторию.

Плоттеры могут работать непосредственно с ЭВМ, для этого они подключаются к каналу стандартного сопряжения, а также автономно, используя в качестве промежуточного носителя магнитный носитель. Способ автономного управления применяется наиболее часто в САПР, так как скорость записи информации на магнитный носитель намного выше, чем скорость вычерчивания, а, следовательно, при этом способе процессор ЭВМ занят при выводе информации меньше время, что повышает эффективность использования ресурсов ВС.

Блок преобразования и подготовки данных принимает и хранит в БЗУ порцию данных, подлежащих выводу. Как и в графических дисплеях, для вычерчивания элементов чертежа применяется аппроксимация. Для реализации процедур упрощенной аппроксимации — интерполяции произвольных линий с помощью единичных шагов пишущего узла используется линейно-круговой

интерполятор — специальный процессор, решающий задачу представления линий отрезками единичных приращений. Интерполятор входит в число устройств плоттера. В случае, если при выводе графической информации встречаются сложные кривые, выше 3-го порядка, то соответствующие вычисления проводятся в процессоре ЭВМ. В блок преобразования входит также схема прерываний, обеспечивающая построение штриховых, штрихпунктирных и других необходимых типов линий.

Наибольшая трудность заключается в нанесении алфавитно-цифровых и специальных символов. Символы в поле чертежа пишутся перьями, входящими в состав пишущего узла, или с помощью специальных буквопечатающих головок. Нанесение необходимых символов в поле чертежа в точке с заданными координатами обеспечивает генератор символов хранящий в ПЗУ микропрограммы описания каждого символа и вырабатывающий сигналы управления приводом с целью перемещения пишущего узла. Размеры наносимых символов могут изменяться путем масштабирования. Расположение символов может быть горизонтальным, вертикальным или под другим фиксированным углом наклона. Скорость вычерчивания символов определяется быстродействием исполнительных органов плоттера, например ШД. Более высокой скоростью обладают плоттеры с буквопечатающей головкой.

Блок управления на основании данных, полученных от интерполятора и генератора символов, вырабатывает сигналы управления электродвигателями и пишущим узлом в процессе построения элементов чертежа. Работа блока управления совместно с интерполятором и генератором символов позволяет значительно уменьшить объем информации, поступающей из ЭВМ, а следовательно, упростить программное обеспечение ЭВМ, улучшить использование пропускной способности канала, уменьшить объем промежуточных носителей.

## 5.8. Функции и типы интерфейсов. Способы организации связи между МП и устройством ввода-вывода

**Принципы организации интерфейсов.** Под стандартным интерфейсом понимается совокупность унифицированных аппаратных, программных и конструктивных средств, необходимых для реализации взаимодействия различных функциональных элементов в автоматических системах сбора и обработки информации при условиях, предписанных стандартом и направленных на обеспечение информационной, электрической и конструктивной совместимости указанных элементов (ГОСТ 23633 — 79, ISO, IEEE). Иногда вместо термина «интерфейс» используется понятие «стык» — место соединения устройств передачи сигналов данных, входя-



щих в системы передачи данных. Структурная схема интерфейса показана на рис. 5.30, где ИБ — интерфейсный блок.

Интерфейс предназначен для унификации внутрисистемных и межсистемных связей и устройств сопряжения с целью эффективной реализации существующих и перспективных элементов ЭВС.

Одна из основных функций интерфейса — это обеспечение информационной совместимости между элементами системы, которая заключается в согласованности взаимодействий функциональных элементов в соответствии с совокупностью логических условий. Логические условия определяют структуру и состав унифицированного набора шин, набор процедур по реализации взаимодействия и последовательность их выполнения для различных режимов функционирования, способ кодирования и форматы данных, команд, адресной информации и информации состояния, временные соотношения между управляющими сигналами, ограничения на их форму и взаимодействие.

Условия информационной совместимости определяют объем и сложность схмотехнического оборудования и программного обеспечения, а также основные технико-экономические показатели — пропускную способность, надежность работы интерфейса и объем аппаратных затрат на устройства сопряжения.

Составными физическими элементами связей интерфейса являются электрические цепи, называемые линиями интерфейса. Часть линий, сгруппированных по функциональному назначению, называется шиной, а вся совокупность линий — магистралью. В системе шин интерфейсов условно можно выделить две магистрали: информационную и управления (каналом); по информационной передаются коды данных, адресов, команд и состояний устройств. Аналогичные наименования присваиваются соответствующим шинам интерфейса. К наиболее распространенным командам относятся ЧТЕНИЕ, ЗАПИСЬ, КОНЕЦ ПЕРЕДАЧИ, ЗАПУСК. Коды состояния описывают состояния сопрягаемых устройств или формируются в ответ на действия команд. Широко используются такие коды состояний, как ЗАНЯТОСТЬ УСТРОЙСТВА, НАЛИЧИЕ ОШИБКИ, ГОТОВНОСТЬ УСТРОЙСТВА (к приему или передаче информации) и др.



Рис. 5.30. Структурная схема интерфейса

Магистраль управления информационным каналом делится на ряд шин. Шина управления обменом состоит из линий синхронизации передачи информации. В зависимости от принятого принципа обмена (асинхронного, синхронного) число линий может изменяться от одной до трех.

Асинхронная передача происходит при условии подтверждения принимающим устройством готовности к приему и завершается подтверждением о приеме данных. При синхронной передаче темп выдачи и приема данных задается регулярной последовательностью сигналов. Линии шины управления обменом выполняются, как правило, двунаправленными.

Шина передачи управления выполняет операции приоритетного занятия (захвата) магистрали информационного канала. Состав и конфигурация линий этой шины зависят от структуры управления интерфейсом. Различают децентрализованную и централизованную структуры управления. В интерфейсах, предназначенных для объединения только двух устройств (соединение типа «точка — точка»), эта шина отсутствует.

Шина прерывания применяется в основном в машинных интерфейсах мини- и микроЭВМ и программно-модульных систем. Основная ее функция — идентификация устройства, запрашивающего сеанс обмена информацией. Идентификация состоит в определении контроллером (процессором) исходной информации о запрашиваемом устройстве. В качестве информации об устройстве используется адрес источника прерывания текущей программы либо адрес программы обслуживания прерывания (вектор прерывания).

Шина специальных управляющих сигналов включает в себя линии, предназначенные для обеспечения работоспособности и повышения надежности устройства интерфейса. К этим линиям относятся линии питания, контроля источника питания, тактовых импульсов, защиты памяти, общего сброса, контроля информации и т. п.

В соответствии с ГОСТ 26.016—81 структуры связей интерфейсов подразделяют на магистральную, радиальную, цепочечную и смешанную (комбинированную).

Для обеспечения информационной совместимости интерфейс реализует ряд функций:

- 1) селекцию (выбор) информационного канала;
- 2) синхронизацию обмена информацией;
- 3) координацию взаимодействия;
- 4) буферное хранение информации;
- 5) преобразование формы представления информации.

Первые три функции выполняет канал управления, четвертую и пятую — информационный канал. Рассмотрим особенности реализации указанных функций в типовых интерфейсах, используемых в ВС и локальных сетях массового применения.



Управление операциями селекции выполняется централизованно и децентрализованно. При централизованном управлении возможны несколько вариантов реализации селекции. На рис. 5.31, а приведена схема временной селекции магистрали на основе генератора временных интервалов контроллера. Магистраль предоставляется каждому устройству через равные промежутки времени, определяемые скоростью работы генератора тактов, а моменты занятия магистрали определяются счетчикам, синхронно работающими в каждом из подключаемых устройств. Такое решение обеспечивает правило приоритетного обслуживания «первым пришел — последним обслуживается».

На рис. 5.31, б приведена схема пространственной селекции на основе последовательного адресного сканирования источников

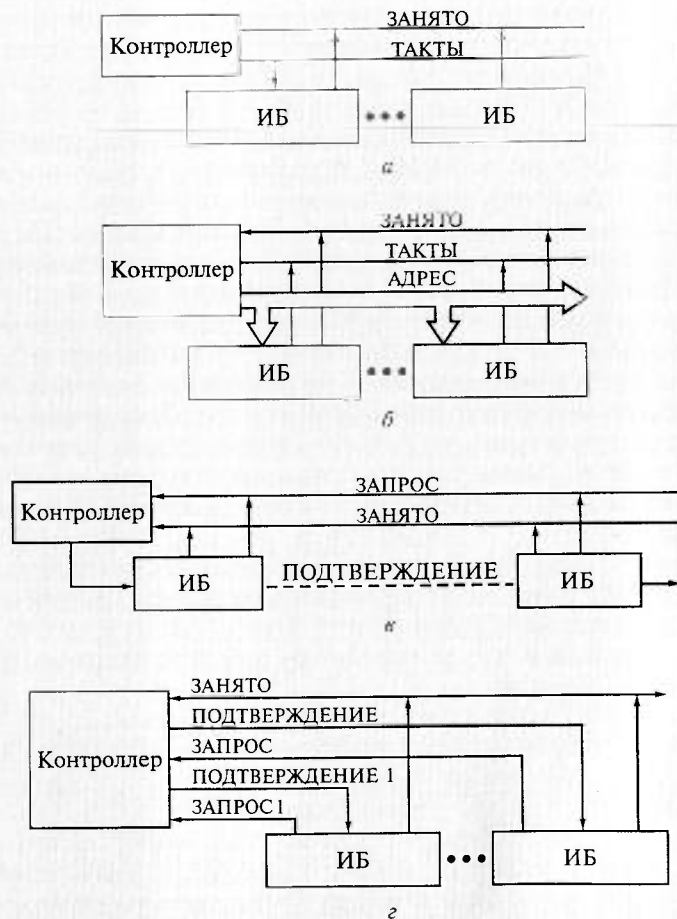


Рис. 5.31. Схемы селекции магистрали централизованной структуры

запроса. Выбор источника запроса начинается по общему сигналу запроса и выполняется последовательно кодовой адресацией всех подключаемых устройств в соответствии с принятым правилом обслуживания. При обнаружении источника запроса устанавливается сигнал ЗАНЯТО и дальнейшая выдача контроллером адресов прекращается. По окончании обслуживания данного запроса возобновляется поиск следующего источника. Достоинство этого варианта — гибкость в реализации правил обслуживания, недостаток — низкое быстродействие. Этот вариант широко применяется в стандартных интерфейсах (ГОСТ 26.003 — 80).

Схема последовательной (цепочечной) селекции показана на рис. 5.31, в. Такая селекция наиболее распространена в машинных интерфейсах как наиболее простая и достаточно быстродействующая. Поиск источника запроса начинается по сигналу ЗАПРОС. Идентификация наиболее приоритетного устройства выполняется сигналом ПОДТВЕРЖДЕНИЕ, который последовательно проходит через все устройства. Приоритетным в данном случае будет устройство, наиболее близко расположенное к контроллеру. При поступлении сигнала ПОДТВЕРЖДЕНИЕ в устройство — источник запроса, дальнейшее его прохождение блокируется и устройством выставляется сигнал ЗАНЯТО.

Отличие схемы селекции по выделенным линиям (рис. 5.31, г) заключается в том, что общие линии ЗАПРОС и ПОДТВЕРЖДЕНИЕ заменяются системой радиальных линий. Максимальное время занятия информационной магистрали для этой схемы будет меньше, чем для цепочечной структуры, так как сигналы по шинам запроса и подтверждения могут передаваться параллельно. Этот вариант обладает также гибкостью установления дисциплины обслуживания, поскольку контроллер с помощью масок может устанавливать произвольный приоритет и порядок опроса. Однако это достигается за счет существенного увеличения числа линий и усложнения аппаратуры.

При децентрализованном управлении также имеются варианты реализации селекции (рис. 5.32). В схемах децентрализованной пространственной селекции наличествуют замкнутые линии запроса и подтверждения. Вариант (см. рис. 5.32, а) отличается исключением линии ЗАНЯТО и замыканием общей линии ЗАПРОС с линией ПОДТВЕРЖДЕНИЕ

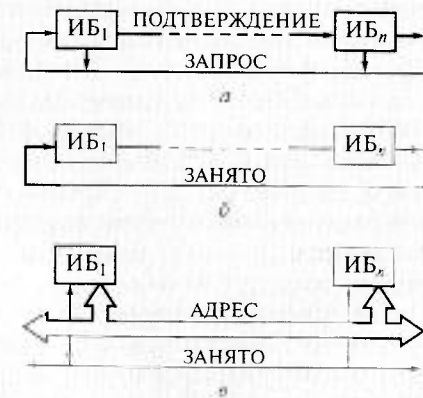


Рис. 5.32. Схемы селекции магистрали децентрализованной структуры

НИЕ. Необходимым условием установления запроса любым устройством является отсутствие входного сигнала подтверждения. При выдаче запроса этот сигнал «дизъюнктивно» формируется на линии и трансформируется в сигнал ПОДТВЕРЖДЕНИЕ, который будет проходить до устройства, выставившего запрос и находящегося наиболее близко по отношению к участку замыкания.

Вариант децентрализованной кольцевой структуры показан на рис. 5.32, б. Здесь используется одна линия, определяющая состояние занятости информационного канала по циркуляции в линии маркерного импульса или серии импульсов. Устройство, запрашивающее шину, не пропускает маркер к следующему устройству, и, таким образом, циркуляция импульсов прекращается. Эта структура широко распространена в интерфейсах локальных сетей. Достоинство кольцевой структуры — использование малого количества оборудования и линий связи, основной недостаток — низкая помехоустойчивость.

Способ параллельного адресного сравнения или децентрализованного кодового управления (ДКУ) является одним из перспективных способов селекции для магистральных систем сопряжения (см. рис. 5.32, в). Сущность алгоритма ДКУ заключается в параллельном выделении приоритетного кода запроса с помощью поразрядного сравнения кодов приоритета в асинхронном режиме одновременно во всех устройствах интерфейса, выставивших запросы.

Процессы передачи массива могут быть детерминированными и стохастическими. К детерминированным относятся процессы передачи массива слов фиксированной длины (от одного до нескольких тысяч слов за сеанс связи), к стохастическим — переменной длины. При детерминированных процессах используется синхронный способ сигнализации окончания процесса взаимодействия, при стохастических процессах — асинхронный. Синхронный способ используется редко, причем в основном в интерфейсах, где фиксированная длина массива слов изменяется от 1 до 256 слов. Основное преимущество синхронного способа сигнализации — отсутствие в системе шин линии окончания сеанса связи.

Асинхронный способ сигнализации при передаче массива слов наиболее распространен. Сигналы синхронизации могут выдаваться в произвольный момент времени отправителем (передатчиком) по информационной шине или специально выделенным линиям управляющего канала.

**Классификация интерфейсов.** Интерфейсы можно разделить на следующие основные классы (рис. 5.33):

- машинные или системные;
- периферийного оборудования;
- мультиплексорных систем;
- распределенных вычислительных систем;
- локальных сетей.

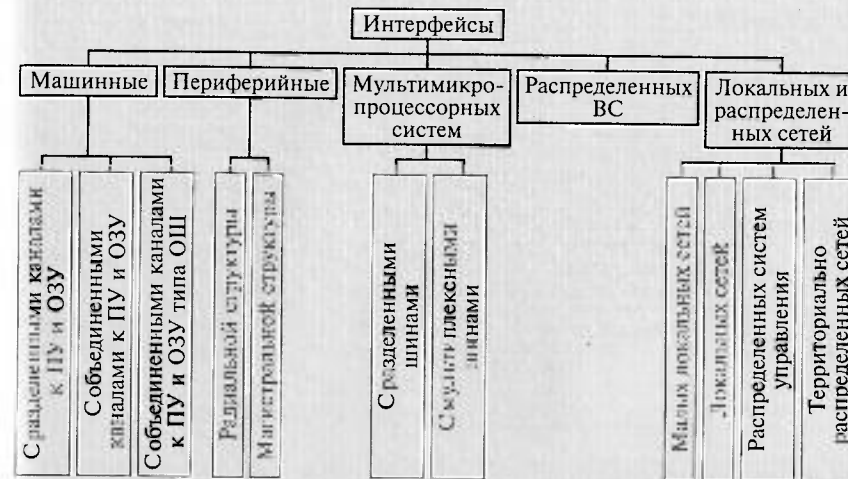


Рис. 5.33. Классификация интерфейсов

Машинные интерфейсы предназначены для организации связи между составными компонентами ЭВМ и систем. В свою очередь они разделяются на три группы. Первая — интерфейсы ввода-вывода в (из) ЭВМ с разделенными информационными каналами к УВВ и к ОЗУ. Вторая группа — интерфейсы ЭВМ с объединенным информационным каналом к УВВ и ОЗУ типа «общая шина» (ОШ). Некоторые классы персональных ЭВМ, мини- и микроЭВМ имеют такую архитектуру. К третьей группе относятся интерфейсы одноплатных ЭВМ с объединенным информационным каналом к УВВ и ОЗУ типа ОШ, ориентированные на внутрисистемное применение. Такие интерфейсы предназначены для организации сопряжения между составными компонентами микропроцессорных комплектов БИС и составных функциональных узлов СБИС мини-ЭВМ (ОШ) и микроЭВМ (Q-шина или GBUS). На рис. 5.34 приведена структура интерфейса типа «общая шина», а в табл. 5.1 — система линий интерфейса «общая шина». ОШ широко используется как внешняя магистраль микроЭВМ, ее прототипом явилась магистраль UNIBUS фирмы DEC.

Интерфейсы периферийного оборудования выполняют функции сопряжения процессоров, контроллеров с УВВ, измерительными приборами, исполнительными механизмами, аппаратурой передачи данных и ВЗУ. Широкая номенклатура этих интерфейсов позволяет использовать разнообразную периферию. По функциональному назначению интерфейсы делятся на радиальной и магистральной структуры. Интерфейсы радиальной структуры обеспечивают схему сопряжения «точка — точка» и составляют малые интерфейсы, используемые для сопряжения исполнительных механизмов ввода-вывода с контроллерами. К этим ин-



Рис. 5.34. Структура интерфейса «Общая шина»

терфейсам относятся системы сопряжения с параллельной передачей информации, предназначенные для подключения стандартной периферии, системы сопряжения для подключения устройств, размещенных на большом удалении друг от друга.

На рис. 5.35 приведена схема машинных интерфейсов. Стандартный интерфейс физически представляет собой многоконтактное разъемное кабельное соединение с четко определенными функциями и параметрами сигналов, передаваемых по каждому проводу (шине). Интерфейс оперативной памяти и интерфейс процессора-канала являются внутренними, стандартизируются в рамках одного семейства ЭВМ и обладают наибольшим быстродействием. Через них информация передается параллельно словами или словами двойной длины, иногда полусловами. Через интерфейс ввода-вывода информация чаще всего передается байтами. Стандартизация этих трех интерфейсов дает возможность расширения вычислительного комплекса, в том числе его модернизации.

Интерфейсы магистральной структуры, обеспечивающие схему «многоточечного» подключения, используются как самостоятельно, так и в качестве системотехнического дополнения, расширяющего функциональные возможности ЭВМ на уровне связи с объектом управления. К ним относятся магистральные интерфейсы программно-модульных систем типа КАМАК. Они обеспечивают сопряжение программируемых контроллеров и ЭВМ с широким спектром цифровых измерительных приборов, преобразователей информации, генераторов, датчиков, пультов оператора.

Интерфейсы мультимикропроцессорных систем представляют собой магистральные системы сопряжения, ориентированные на объединение нескольких процессоров, модулей ОЗУ, контроллеров ВЗУ.

Таблица 5.1

Система линий интерфейса «Общая шина»

Название линии	Обозначение	Назначение
Адрес	A(17—00)	Для выборки исполнителя
Данные	D(15—00)	Для передачи данных, команд и состояния ПУ
Контрольный разряд младшего байта данных	KPM	—
Контрольный разряд старшего байта данных	KPC	—
Управление	УПР(1—0)	Для передачи кода операции
Синхронизация задатчика	СХЗ	Для указания, что адрес и управляющая информация выставлены на общую шину
Синхронизация исполнителя	СХИ	Для указания, что исполнитель принял или выдал данные на общую шину
Запрос передачи	ЗП(7—4)	Для передачи от ПУ запроса соответствующего уровня приоритета
Разрешение передачи	РП(7—4)	Для передачи арбитрам ответа ПУ на ЗП
Запрос прямого доступа к памяти	ЗПД	Для передачи от ПУ запроса наивысшего приоритета
Разрешение прямого доступа	РПД	Ответ схемы арбитра на сигнал ЗПД
Подтверждение выборки	ПВБ	Ответ ПУ на сигналы РП, РПД
Прерывание	ПРЕР	Для указания процессору, что вектор прерывания выставлен на общую шину
Занято	ЗАН	Для указания, что общая шина занята текущим задатчиком
Подготовка	ПОДГ	Для установки процессором регистров ПУ в исходное состояние

Интерфейсы распределенных ВС предназначены для интеграции средств обработки информации, размещенных на значительном расстоянии. К ним можно отнести внутриблочные и процессорно-независимые системы сопряжения. Отличим их от интер-



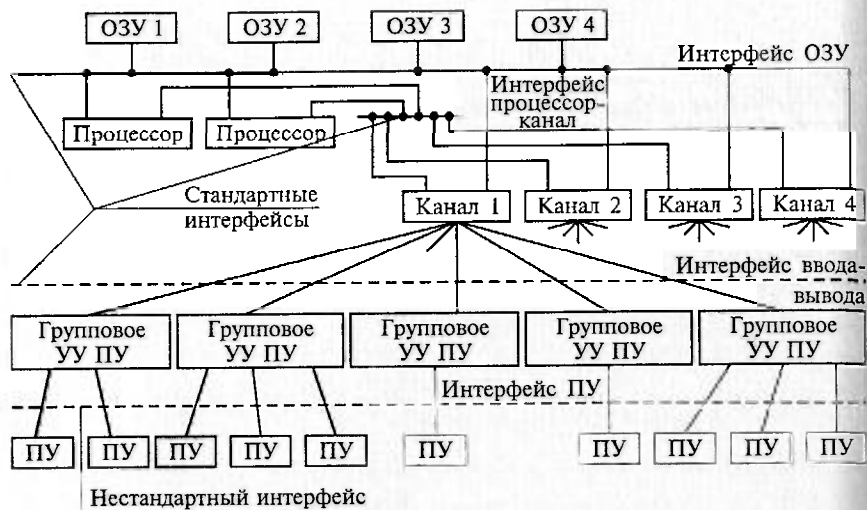


Рис. 5.35. Интерфейсы ЭВМ

фейсов типа ОШ является техническая реализация функций селекции и координации, что позволяет подключать к ним один или несколько процессоров как обычные УВВ. Этот класс интерфейсов отличают высокая пропускная способность и минимальное время доступа процессора к общему ОЗУ.

Интерфейсы распределенных ВС и локальных систем предназначены для интеграции средств обработки информации, размещенных на значительном расстоянии. Они ориентированы на использование в системах различного функционального назначения. Обычно это системы сопряжения с бит-последовательной передачей информации магистральной или кольцевой структуры. Интерфейсы этого класса в зависимости от назначения разделяют на группы интерфейсов малых локальных и локальных сетей (с длиной магистрали от десятков метров до нескольких километров); распределенных систем управления; территориально и географически распределенных сетей ЭВМ (с длиной линии более 10 км).

На рис. 5.36 условно показаны межуровневые интерфейсные услуги ВС семиуровневой архитектуры (ГОСТ 23633—79). Две ВС объединяются между собой системой передачи данных.

1. Физический уровень — это средства связи (канал связи, линия, кабель и др.).

2. Уровень звена данных включает в себя функциональные и процедурные средства передачи, протоколы обмена сигналами.

3. На сетевом уровне выполняются функции маршрутизации, адресации, организации виртуальных соединений, адресации пакетов.

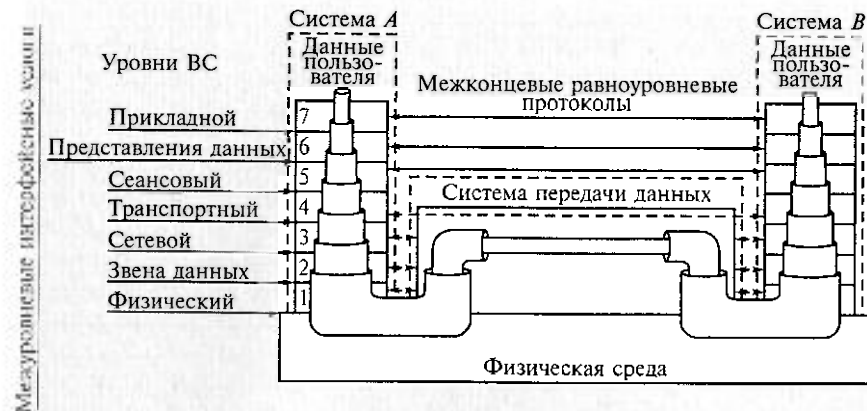


Рис. 5.36. Семиуровневая архитектура ВС

4. На транспортном уровне — функции адресации конечных абонентов, установки соответствия между адресами и сетевыми именами абонентов, доставки данных от системы-источника к системе-адресату.

5. Сеансовый уровень содержит механизмы организации структуры взаимодействий между прикладными процессами.

6. На уровне представления данных выполняются функции преобразования синтаксиса и форматов данных, кодов изображений, графических и алфавитных данных, организации файлов, форматирование и компоновка данных.

7. Прикладной уровень содержит прикладные процессы, обеспечивающие обработку информации. Назначение этого уровня — смысловое содержание (семантика) процессов.

По конструктивному исполнению интерфейсы можно разделить на четыре категории:

1) межблочные, обеспечивающие взаимодействие компонентов на уровне прибора, автономного устройства, блока, стойки, шкафа;

2) внутриблочные, обеспечивающие взаимодействие на уровне плат, субблоков;

3) внутриплатные, обеспечивающие взаимосвязь между интегральными схемами (средняя интегральная схема, большая интегральная схема, сверхбольшая интегральная схема) на печатной плате;

4) внутрикорпусные, обеспечивающие взаимодействие компонентов внутри сверхбольшой интегральной схемы.

Межблочное сопряжение выполняется с использованием следующих конструктивных средств: коаксиального и оптоволоконного кабеля, многожильного плоского кабеля, многожильного кабеля на основе витой пары проводов.

Внутриблочное сопряжение печатных плат, субблоков выполняется печатным способом или накруткой витой парой проводов внутри блока, стойки, шкафа. Ряд интерфейсов может быть выполнен комбинацией внутриблочного и межблочного исполнений.

Внутриплатное сопряжение реализуется печатным способом, внутрикорпусное — методами микроэлектронной технологии.

**Интерфейсы системной шины.** Системная магистраль (СМ) является узким местом ЭВМ, так как все устройства, подключенные к ней, конкурируют за возможность передавать свои данные по ее шинам. СМ — это среда передачи сигналов управления, адресов, данных, к которой параллельно и одновременно может подключаться несколько компонентов вычислительной системы. Физически системная магистраль представляет собой параллельные проводники на материнской плате, которые называются линиями. Но это еще и алгоритмы, по которым передаются сигналы, правила интерпретации сигналов, дисциплины обслуживания запросов, специальные микросхемы, обеспечивающие эту работу. Весь этот комплекс образует понятие *интерфейс системной магистрали* или *стандарт обмена*.

Исторически все интерфейсы СМ ведут свою родословную от стандарта IBM MULTIBUS, для которого был разработан комплект микросхем (chipset). Этот стандарт мог обслуживать передачу 8- и 16-битовых данных, работать в мультипроцессорном режиме с несколькими ведущими устройствами. Понятие «ведущее/ведомое устройство» могло динамически переопределяться с помощью сигналов управления. Для этого стандарта характерно наличие следующих линий: 20 линий адресов, 16 линий данных, 50 управляющих и служебных линий.

В 1987 г. для IBMPS-2 был разработан стандарт Микроканал — MCA (Micro Channel Architecture). В нем 24-разрядная шина адреса. Шина данных увеличена до 32 бит. Отказались от перемычек и переключателей, определяющих конфигурацию технических средств, и ввели CMOS-память (Complementary Metal Oxide Semiconductor), позволяющую хранить эту информацию и при отключении питания. Все оборудование, подключаемое к системной магистрали, содержит специальные регистры POS (Programmable Option Select), позволяющие конфигурировать систему программным путем. При тактовой частоте 10 МГц скорость передачи данных составляла 20 Мбайт/с.

Для IBM PC XT был разработан стандарт ISA (Industry Standard Architecture), который имеет две модификации — для XT и AT. В ISA XT шина данных — 8 бит, шина адресов — 20 бит, шина управления — 8 линий. В ISA AT шина данных увеличена до 16 бит. Встречаются и 32-битовые ISA, но это — нестандартизированное расширение. Тактовая частота для работы СМ в стандарте ISA

составляет 8 МГц. Производительность ISA XT — 4 Мбайт/с, ISA AT — от 8 до 16 Мбайт/с.

Стандарт EISA (Extended ISA) — это жестко стандартизованное расширение ISA до 32 бит. Конструктивно совместима с ISA-адаптерами внешних устройств. Предназначена для многозадачных систем, файл-серверов и систем, в которых требуется высокоэффективное расширение ввода-вывода. При тактовой частоте 8,33 МГц скорость передачи данных составляла 33 Мбайт/с.

Стандарт VESA (VESA Lokal Bus или VLB) разработан Ассоциацией стандартов видеоданных (Video Electronics Standard Association) как расширение стандарта ISA для обмена видеоданными с адаптером SVGA. Обмен данными по этому стандарту ведется под управлением микросхем, расположенных на карте, устанавливаемой в специальный слот (разъем) расширения VLB и соединяемой с СМ через стандартный слот расширения. В отличие от стандартных слотов расширения слот VLB связан с микропроцессором напрямую, минуя системную магистраль. Карта VLB, работая совместно с системной магистралью, реализующей стандарт ISA, обеспечивает 32-разрядную передачу данных с тактовой частотой микропроцессора (но не более 40...50 МГц). В стандартные слоты материнской платы с интерфейсом VLB устанавливаются карты расширения с интерфейсом ISA. Производительность стандарта VLB достигает 132 Мбайт/с.

Стандарт PCI (Peripheral Component Interconnect) разработан фирмой Intel для ЭВМ с МП Pentium. Это не развитие предыдущих стандартов, а совершенно новая разработка. Системная магистраль в соответствии с этим стандартом работает синхронно с тактом МП и осуществляет связь между локальной шиной МП и интерфейсом ISA, EISA или MCA. Но поскольку для этого интерфейса используются микросхемы, выпускаемые другими фирмами (Saturn — для 486, Mercury, Neptune, Triton — для Pentium), скорость работы СМ реально составляет 30...40 Мбайт/с при теоретически возможной 132/264 Мбайт/с. Стандарт PCI разрабатывался как процессорно-независимый интерфейс. Помимо Pentium, с этим интерфейсом могут работать и МП других фирм (Alpha корпорации DEC, MIPS R4400 и Power PC фирм Motorola, Apple и IBM). Стандарт PCI позволяет реализовать дополнительные функции: автоматическую конфигурацию периферийных устройств (которая позволяет пользователю устанавливать дополнительные платы, не задумываясь над распределением прерываний, каналов прямого доступа к памяти (ПДП) и адресного пространства); работу при пониженном напряжении питания; возможность работы с 64-разрядным интерфейсом. «Слоевая» структура интерфейса PCI снижает электрическую нагрузку на МП и позволяет одновременно управлять шестью периферийными устройствами, подключенными к СМ. Стандарт PCI позволяет использовать «мосты» (Bridges)



для организации связи с другими стандартами (например, PCI to ISA Bridge).

Стандарт USB (Universal Serial Bus) — универсальный последовательный интерфейс, обеспечивающий обмен со скоростью 12 Мбайт/с и подключение до 127 устройств.

Стандарт PCMCIA (Personal Computer Memory Card International Association) — интерфейс блокнотных ПЭВМ для подключения расширителей памяти, модемов, контроллеров дисков и стримеров, сетевых адаптеров и др.

**Способы организации связи между МП и устройством ввода-вывода.** Различают три способа организации связи между МП и УВВ: программно-управляемая передача данных; использование прерываний; прямой доступ к памяти ПДП.

Под программно-управляемой передачей данных подразумевается обмен данными между МП и УВВ. Отличительной особенностью этого способа обмена является использование одной или нескольких специальных команд, инициирующих этот обмен.

При оценке программно-управляемой передачи данных необходимо учитывать следующие факторы:

- большинство электромеханических УВВ имеют на несколько порядков меньшее быстродействие по сравнению с МП, поэтому при рассматриваемом способе обмена время ожидания готовности УВВ может быть очень большим;
- программно-управляемая передача данных не может быть инициирована по запросу от УВВ.

Эти факторы вызвали появление различных систем прерывания.

Система прерываний позволяет УВВ инициировать обмен данными с МП. Для этого МП должен иметь вход, периодически опрашиваемый МП (обычно после выполнения каждой команды). Сигнал прерывания на этом входе заставляет МП отложить исполнение основной программы и перейти в режим обслуживания прерывания. Организация системы прерываний требует решения следующих задач:

- должно быть идентифицировано устройство, от которого поступил запрос на прерывание;
- должно быть сохранено состояние активных регистров МП, так как оно может использоваться после возобновления выполнения основной программы;
- после окончания обслуживания прерывания состояние этих регистров должно быть восстановлено.

Различают два способа реализации перечисленных функций: программный и аппаратный.

Программная реализация системы прерываний отличается простотой и характеризуется минимальным числом сигналов для идентификации источника прерываний. Однако ее существенным не-

достатком является большее время реакции на запросы — промежуток времени между моментами поступления запроса и началом его обслуживания.

Значительное уменьшение времени реакции системы достигается при аппаратной реализации учета приоритетов, сохранения содержимого регистров, перехода к нужной прерывающей программе и маскирования входов запросов на прерывание от отдельных УВВ.

Канал прямого доступа в память ПДП обычно используется для подключения к МП быстродействующих УВВ. Обмен данными через канал ПДП обеспечивает максимальную скорость ввода и вывода информации.

Работа канала ПДП основывается на непосредственной передаче данных между памятью и УВВ без участия МП. Такой обмен осуществляется при откладывании основной программы на время обращения УВВ к памяти. Поскольку память отключена от МП и подключена к УВВ только на время записи в ЗУ или чтения из ЗУ, можно говорить, что обмен происходит в режиме захвата цикла памяти.

Обмен информацией по каналу ПДП может происходить, если устройство подключено к линии запроса шин. При таком способе периферийное устройство выставляет периодический сигнал — ЗАПРОС МАГИСТРАЛИ. После этого процессор приостанавливает выполнение программы, переходит в состояние «захват» и подает сигнал ОСВОБОЖДЕНИЕ МАГИСТРАЛИ. Дальнейший обмен происходит под управлением периферийного устройства или второго модуля процессора, подключенного к магистрали; при этом магистраль может быть общей для нескольких процессоров.

Для синхронизации процесса передачи по магистрали от одного модуля к другому предусмотрено приоритетное синхронизирующее устройство. Все ведущие модули или внешние устройства, подключенные через канал прямого доступа, распределены по уровням модульного приоритета через зону нестандартных соединений унифицированной магистрали.

В целях минимизации внешних соединений внешние устройства, ведущие обмен информацией с процессором или памятью, подключаются к модулю отдельными разъемами. Такое подключение позволяет, кроме этого, значительно увеличить количество цепей внешних соединений, поскольку возможности коммутации со стороны унифицированных магистралей ограничены.

Передача данных с помощью канала ПДП по сравнению с использованием системы прерываний не дает потерь времени на вход в прерывающую программу и выход из нее, на сохранение, а затем и восстановление состояния регистров МП, что достигается за счет существенного усложнения интерфейса канала ПДП.



## 5.9. Автоматические устройства ввода-вывода аналоговой информации в ЭВМ

**Назначение УВВ аналоговой информации в ЭВМ.** Рост промышленного производства разнообразных средств вычислительной техники привел к интенсивному внедрению ЭВМ в процессы измерений и управления. Использование малых ЭВМ в качестве элемента измерительной системы позволяет повысить точность измерений, увеличить их производительность, автоматизировать процесс получения, накопления и обработки информации.

Использование ЭВМ в контуре систем управления позволяет реализовывать более сложные и эффективные по сравнению с аналоговыми системами управления алгоритмы управления с легко изменяемой структурой, осуществлять изменение режима и параметров управления по заданной программе в реальном времени.

Цифровые системы управления имеют целый ряд преимуществ по сравнению с аналоговыми системами. Основными из них являются:

- 1) возможность модернизации и внедрения более прогрессивных и эффективных алгоритмов управления, например, адаптивных. Внедрение, исследование и анализ новых алгоритмов управления в цифровой системе осуществляются значительно проще;

- 2) возможность изменения структуры и параметров алгоритмов управления, которое значительно проще осуществлять программным путем, чем изменениями механических или электрических параметров;

- 3) возможность простого программного изменения режима и параметров управления в реальном времени, которого невозможно достичь в аналоговых системах управления;

- 4) стабильность и повторяемость характеристик цифровых систем управления в течение неограниченного промежутка времени, отсутствие дрейфов и интерференций отдельных частей;

- 5) возможность совмещения функций управления с функциями получения (измерения) и обработки информации. Физические величины, используемые цифровой системой для управления в ряде приложений являются одновременно и информативными величинами, определяющими действующие в контролируемой системе процессы.

На рис. 5.37 представлена обобщенная функциональная схема цифровой системы управления, включающая объект управления или контролируемый процесс и цифровой регулятор. Контролируемые физические величины или параметры процесса (световой поток, температура, давление, электрическое или магнитное поле и т.д.) измеряются системой датчиков, преобразующих физические величины в электрические сигналы. В качестве датчиков могут использоваться фотоэлектрические и пьезоэлектрические прибо-

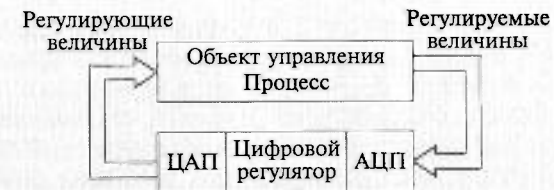


Рис. 5.37. Функциональная схема цифровой системы управления

ры, термопары, индуктивные и емкостные сенсоры и т.д. Электрические сигналы, снимаемые с выхода датчика, несут в себе информацию о характеристиках регистрируемого физического воздействия, причем эта информация может заключаться в различных параметрах сигналов: величине непрерывно изменяющегося электрического напряжения или тока, амплитуде импульсов. Для того чтобы цифровой регулятор мог воспринимать эту информацию, ее надо преобразовать в цифровой код. Поэтому неизменным элементом цифровой системы управления являются аналого-цифровые преобразователи (АЦП).

Цифровой регулятор в соответствии с определенным алгоритмом управления и по определенной программе вырабатывает управляющие объектом или процессом сигналы, которые должны определенным образом через всевозможные исполнительные устройства (электромагниты, электромагнитные муфты, электроприводы, двигатели и т.д.) воздействовать на объект управления. Исполнительные устройства обычно управляются аналоговыми электрическими сигналами (током, напряжением), для чего в системе управления применяются цифроаналоговые преобразователи (ЦАП), преобразующие цифровые управляющие сигналы регулятора в используемые исполнительными устройствами аналоговые эквиваленты. Следует отметить, что структура (см. рис. 5.37) является общей для большинства цифровых систем управления другого назначения.

Функции цифрового регулятора может выполнять ЭВМ или специализированный контроллер. Если же стоит задача совмещения функций управления объектом или процессом и функций получения и обработки информации о данном объекте, то в системе со специализированным контроллером должна быть введена дополнительная ЭВМ, занимающаяся только получением и обработкой информации, так как контроллеры обладают недостаточной мощностью (быстродействием и особенно памятью) для выполнения обеих функций. Существует другая концепция использования в качестве цифрового регулятора и системы сбора и обработки информации одной достаточно мощной ЭВМ, тем более что для сбора информации ЭВМ должна содержать АЦП, аналогичные используемым для управления.

В качестве управляющей ЭВМ можно использовать IBM-PC совместимый персональный компьютер. Как зарубежный, так и отечественный рынок широко насыщен компьютерами данного класса, которые обладают достаточно высокими техническими характеристиками для решения поставленной задачи.

Таким образом, задача разработки цифровой системы управления сводится в первую очередь к разработке принципов построения и программирования универсальной системы ввода-вывода аналоговой информации в ЭВМ и из нее. Система ввода-вывода может быть реализована в виде универсальной платы стандартного подключения, размещаемой в одном из свободных слотов внутри корпуса компьютера и связанной с компьютером через общую шину. Универсальность системы и платы ввода-вывода заключается в том, что она может размещаться в любом IBM-PC совместимом компьютере и использоваться для любых задач цифрового управления или сбора и обработки информации, для которых достаточны реализованные в системе разрядность и быстродействие ввода-вывода информации. Существенным преимуществом данной концепции является возможность использования как ассемблера, так и языков высокого уровня для программирования и отладки программ, реализующих цифровые алгоритмы управления объектами или процессами, и программ управления системой ввода-вывода. Компьютер, используемый для задач управления и получения и обработки информации в течение ограниченных периодов времени, может быть использован для выполнения обычных задач расчетов, обработки текстов и т.д. в течение всего оставшегося времени.

Данная концепция построения цифровых систем управления и получения и обработки информации на основе применения массовых компьютеров и универсальных плат ввода-вывода информации в компьютер широко применяется за рубежом.

**Принципы построения ЦАП и АЦП.** Рассмотрим вкратце принципы построения ЦАП и АЦП как необходимых элементов систем ввода-вывода аналоговой информации в ЭВМ. Более детальная информация может быть найдена в специальной литературе.

ЦАП и АЦП характеризуются: погрешностью, быстродействием и динамическим диапазоном.

Погрешность состоит из методической и инструментальной составляющих. Методическая погрешность определяется абсолютной погрешностью квантования аналоговой величины по уровню:

$$\Delta = x - N\Delta x,$$

где  $N$  — числовое выражение величины  $x$ ;  $\Delta x$  — шаг квантования (очевидно,  $|\Delta| \leq 0,5 \Delta x$ ).

Таким образом, шаг квантования определяет методическую погрешность преобразования по уровню. Инструментальная по-

грешность определяется нестабильностью параметров элементов схемы преобразователя, и неточностью его настройки.

Быстродействие — определяется временем преобразования: для ЦАП — интервалом между моментами поступления входного кода и установления выходного сигнала (с заданной точностью), для АЦП — интервалом от момента пуска преобразователя до момента получения кода на выходе.

Динамический диапазон — допустимый диапазон изменения входного напряжения для АЦП и выходного напряжения для ЦАП.

Основу ЦАП составляет матрица резисторов, подключенных ко входу операционного усилителя ключами, которые управляются двоичным кодом (например, параллельным кодом регистра или счетчика). На рис. 5.38 в качестве примера приведена схема четырехразрядного ЦАП. Каждому двоичному разряду соответствует свой разрядный ток  $I_0 \dots I_3$ . Величины токов задаются матрицей резисторов, сопротивления которых удваиваются при переходе от старшего разряда к младшему, реализуя количественное соответствие цифровых и аналоговых сигналов. Разрядные токи преобразуются усилителем в выходные напряжения ЦАП.  $R_0, R_1$  — сопротивления резисторов в прямой и обратной цепи операционного усилителя.

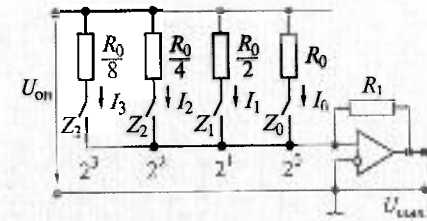


Рис. 5.38. Устройство четырехразрядного ЦАП

Коэффициенты передачи  $K = -\frac{U_{\text{вых}}}{U_{\text{оп}}}$  по входам  $2^0, 2^1, 2^2, 2^3$

равны соответственно:

$$K_0 = \frac{R_1}{R_0} Z_0; K_1 = \frac{2R_1}{R_0} Z_1; K_2 = \frac{4R_1}{R_0} Z_2; K_3 = \frac{8R_1}{R_0} Z_3,$$

где  $U_{\text{вых}}$  — выходное напряжение;  $U_{\text{оп}}$  — опорное напряжение;  $Z_0 \dots Z_3$  — числа, принимающие значение «0» или «1» в зависимости от положения соответствующих ключей.

Выходное напряжение ЦАП определяется суммой:

$$U_{\text{вых}} = -U_{\text{оп}} (K_0 + K_1 + K_2 + K_3) = -U_{\text{оп}} \frac{R_1}{R_0} (Z_0 + 2Z_1 + 4Z_2 + 8Z_3).$$

Таким образом четырехразрядный двоичный код преобразуется в уровень  $U_{\text{вых}}$  в диапазоне от 0 до  $15\Delta U$ , где  $\Delta U = U_{\text{оп}} \frac{R_1}{R_0}$  — шаг

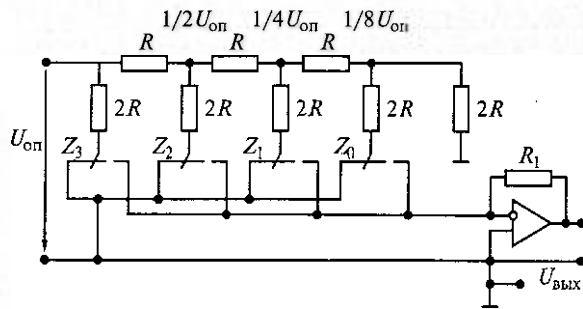


Рис. 5.39. Схема ЦАП с матрицей  $R-2R$

квантования. Для уменьшения погрешности квантования необходимо увеличивать число двоичных разрядов ЦАП.

Недостатком схемы на рис. 5.39 являются:

- жесткие требования к точности и стабильности резисторов;
- изменение нагрузки источника  $U_{оп}$  в зависимости от положения ключей, что требует стабилизации напряжения  $U_{оп}$ .

От перечисленных недостатков свободна схема (см. рис. 5.39), где используется матрица резисторов  $R-2R$ , имеющая постоянное входное сопротивление со стороны источника опорного напряжения.

Выходное напряжение ЦАП определяется выражением

$$U_{вых} = -U_{оп} \frac{R_1}{16R} (Z_0 + 2Z_1 + 4Z_2 + 8Z_3).$$

Для переключения резисторов применяют транзисторные ключи, например, на полевых транзисторах (рис. 5.40).

Рассмотренные принципы организации ЦАП используются при построении АЦП. На рис. 5.41 представлена функциональная схема АЦП последовательного счета.

Генератор импульсов (ГИ) вырабатывает последовательность импульсов, которая с помощью счетчика (Сч) преобразуется в двоичный код. Этот код управляет ключами ЦАП. Выходное напряжение ЦАП поступает на один из входов компаратора (К), где сравнивается с входным напряжением  $U_{вх}$ , поданным на другой вход компаратора. При равенстве напряжений  $U_{вх}$  и  $U_{цап}$  в момент времени  $t_1$  (рис. 5.42) компаратор выдает сигнал, останавливающий работу генератора импульсов. При

Рис. 5.40. Схема транзисторных ключей

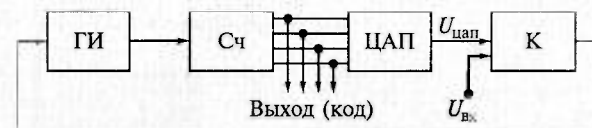


Рис. 5.41. Функциональная схема АЦП последовательного счета

этом на выходе Сч фиксируется двоичный код, соответствующий напряжению  $U_{вх}$ .

На рис. 5.43 приведена функциональная схема АЦП поразрядного уравнивания.

Поступающие на вход регистра импульсы последовательно переводят разряды регистра в состояние «1» начиная со старшего, при этом остальные разряды находятся в состоянии «0». Если старший разряд находится в состоянии «1», ЦАП вырабатывает соответствующее напряжение, которое сравнивается с входным.

Если  $U_{цап} > U_{вх}$ , то по команде компаратора старший разряд регистра сбросится в «0». Если  $U_{цап} < U_{вх}$ , то в старшем разряде останется «1». Затем «1» появляется в следующем разряде по старшинству и снова производится сравнение. Цикл повторяется пока не произойдет сравнение в младшем разряде. Генератор останавливается, и на выходе регистра появляется выходной двоичный сигнал, соответствующий входному аналоговому сигналу (рис. 5.44).

**Принципы построения и программирования системы ввода-вывода аналоговой информации в ЭВМ.** На рис. 5.45 представлена функциональная схема системы ввода-вывода аналоговой информации в компьютер. Система в минимальной конфигурации должна содержать генератор импульсов синхронизации, схему дешифрации адреса, блок АЦП, блок ЦАП и несколько внутренних регистров. Генератор вырабатывает сигнал синхронизации определенной частоты, необходимый для функционирования системы и синхронизации всех происходящих в системе операций. В состав генератора может входить делитель частоты, осуществляющий ее деление в необходимое число раз.

Схема дешифрации адреса служит для декодирования адресных сигналов общей шины компьютера и выработки сигналов выбора внутренних регистров системы ввода-вывода, к которым центральный процессор ЦП компьютера обращается в данный момент. Блок АЦП производит преобразование аналогового сигнала в цифровой эквивалент и в многоканальных системах содержит мультиплексор входного канала. Кроме того, в состав блока входит устройство вы-

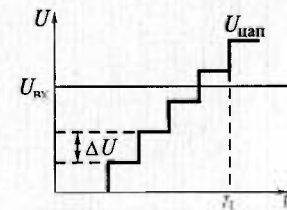


Рис. 5.42. Временная диаграмма работы АЦП последовательного счета



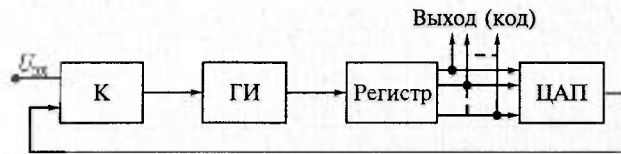


Рис. 5.43. Функциональная схема АЦП поразрядного уравнивания

борки-хранения, осуществляющее отключение АЦП от аналогового входа во время выполнения преобразования. Блок ЦАП производит преобразование цифрового сигнала в аналоговый эквивалент. Внутренние регистры системы осуществляют функции интерфейса с компьютером, т. е. формируют необходимые в системе сигналы управления, а также хранят преобразованную и передаваемую информацию. Рассмотрим детально принципы построения и программирования каждого из входящих в состав системы ввода-вывода устройств.

**Генератор сигнала синхронизации.** Генератор сигнала синхронизации, входящий в состав системы ввода-вывода, позволяет сделать ее универсальной. Система ввода-вывода может быть использована в любом компьютере, независимо от его собственной рабочей частоты. Существует большое количество всевозможных схем цифровых генераторов сигналов синхронизации, которые могут быть найдены в литературных источниках. На рис. 5.46 приведена схема простого генератора, стабилизируемого кварцевым резонатором, включающая также простой делитель частоты на четырехразрядном двоичном счетчике. Схема генератора позволяет устойчиво получать частоты от 30 кГц до 1 МГц. Выбор частоты определяется кварцевым резонатором и номиналами резисторов и конденсаторов. Делитель позволяет разделить исходную частоту на 2, 4, 8 и 16. Необходимый коэффициент деления выбирается установкой соответствующей перемычки на выходе делителя. При необходимости получения других коэффициентов деления, они могут быть реализованы последовательным включением необходимого

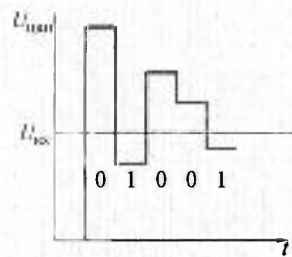


Рис. 5.44. Временная диаграмма работы АЦП поразрядного уравнивания

числа счетчиков либо применением программируемых делителей частоты.

**Схема дешифратора адреса.** Система ввода-вывода представляет собой нестандартное ВУ компьютера. Оно содержит определенное количество внутренних регистров с соответствующими адресами в адресном пространстве компьютера. Обращаясь по этим адресам, ЦП компьютера осуществляет программное управление системой ввода-вывода.



Рис. 5.45. Функциональная схема системы ввода-вывода

При обращении к ВУ процессор устанавливает на двадцати адресных линиях общей шины  $A00...A19$  его адрес и сигнал синхронизации адреса ALE на соответствующей управляющей линии. Схема дешифратора адреса должна иметь возможность воспринимать этот адрес, дешифровать его и формировать сигналы выбора соответствующих внутренних регистров системы ввода-вывода. На рис. 5.47 представлена одна из возможных схем дешифратора адреса, позволяющая формировать 16 сигналов выбора внутренних регистров. Она содержит регистр хранения адреса и собственно дешифратор. Выставленный на адресных линиях адрес записывается по сигналу ALE в регистр хранения адреса, образованный триггерами  $T1...T9$ . При адресации системы ввода-вывода не используется нулевая линия адреса  $A00$ , так как в системе предполагается обмен информацией с компьютером целыми словами (16 бит). В качестве базового адреса системы ввода-вывода может использоваться любой свободный адрес в диапазоне адресов

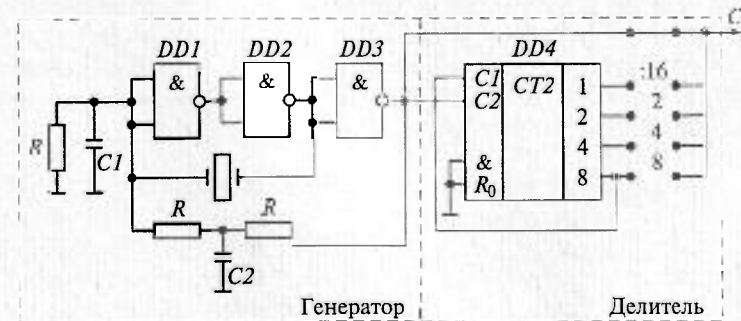


Рис. 5.46. Схема генератора и делителя частоты

000h...3Ffh (0...1023) адресного пространства компьютера, ответственных для внешних устройств.

Вообще говоря, в IBM-совместимых компьютерах для внешних устройств отводится диапазон адресов 0000h...FFFFh (0...65535), но все более старшие, чем 3Ffh, адреса в компьютерах стандартной конфигурации, как правило, не используются. Поэтому при адресации системы ввода-вывода также не используются разряды A10...A19, так как обращения по более старшим, чем 3Ffh, адресам ВУ в компьютерах стандартной конфигурации не происходит. При обращении по адресу любого ВУ на линиях A10...A19 устанавливается логический ноль. Следует также отметить, что в IBM компьютерах для выполнения операций обмена данными между ЦП и ВУ используются два специальных управляющих сигнала IOR, (чтение из ВУ) и IOW (запись во ВУ), которые отличны от аналогичных управляющих сигналов для обмена данными с памятью, что исключает любую нежелательную интерференцию между ВУ устройствами и памятью.

Для установки базового адреса в схеме имеются переключатели. Если все переключатели установлены в верхнее положение, то элемент 8И-НЕ сработает (выдаст на выходе логический ноль) при установке на шине адреса любого из 16 адресов 3F0h...3Ffh (поскольку адресные линии A01...A04 не подсоединены к элементу

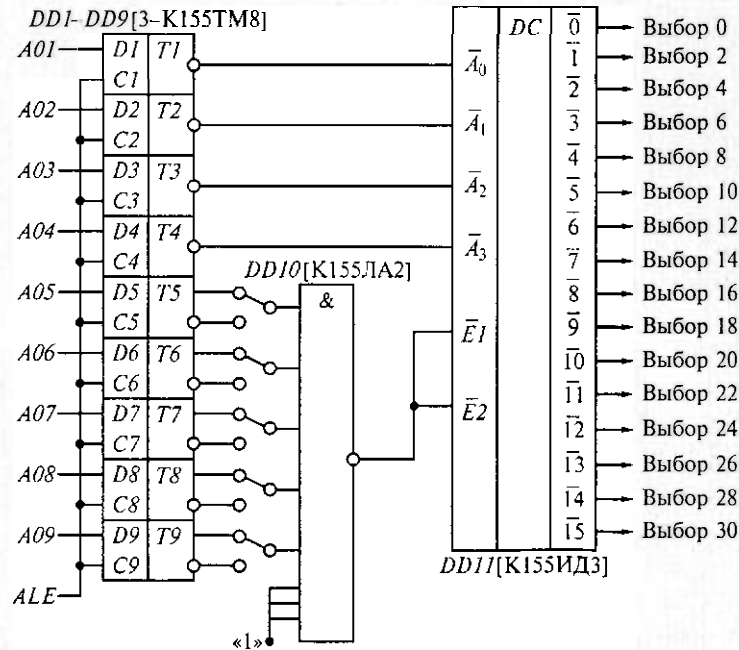


Рис. 5.47. Схема дешифратора адреса

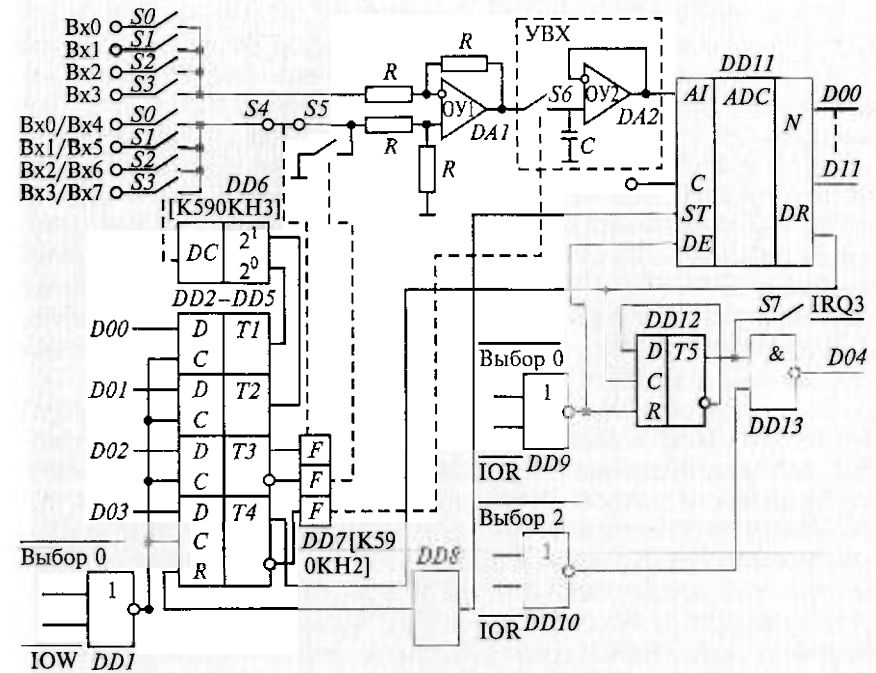


Рис. 5.48. Схема блока АЦП

8И-НЕ). Первый из этих адресов называется базовым и имеет индекс 0. Остальные адреса системы ввода-вывода имеют индексы 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30. Эти 16 имеющихся адресов могут использоваться для обращения по крайней мере к 16 регистрам системы ввода-вывода.

Собственно дешифратор адреса преобразует четырехразрядный двоичный код A01...A04 на входе в сигнал логического нуля на одном из 16 выходов, который и является сигналом выбора (адресации одного из 16 внутренних регистров системы ввода-вывода).

**Блок АЦП.** Блок АЦП содержит внутренние регистры, мультиплексор аналоговых входов, устройство выборки-хранения, собственно АЦП (рис. 5.48). К числу внутренних регистров блока, кроме регистра данных АЦП, отнесем также регистры управления и состояния (см. рис. 5.45), так как это необходимо для объяснения работы схемы.

В рассматриваемом, достаточно общем, варианте блок АЦП содержит три внутренних регистра. Перечислим их.

1. Регистр управления. Расположен по адресу: «Базовый адрес + 0». Регистр доступен при записи и содержит четыре разряда, реализованные на триггерах T1...T4 (DD2...DD5). Входы регистра соединены с разрядами D00...D03 шины данных компьютера. Биты D00...D01 определяют в двоичном коде адрес одного из четырех

каналов, которые возможно опрашивать в системе. Если  $D00 = 0$ ,  $D01 = 0$ , то опрашивается нулевой канал и т. д. В случае бóльшего наличного числа опрашиваемых каналов число необходимых для выбора опрашиваемого канала разрядов регистра управления пропорционально увеличивается. Бит  $D02$  определяет дифференциальный или недифференциальный ввод сигнала по выбранному каналу опроса ( $D02 = 1$  — дифференциальный,  $D02 = 0$  — недифференциальный). Бит  $D03$  служит для управления режимом работы устройства выборки-хранения. Регистр управления выбирается сигналом «Выбор 0», формируемым схемой дешифратора адреса при обращении по базовому адресу системы ввода-вывода. Запись информации в регистр осуществляется при совместном действии сигналов «Выбор 0» и  $\overline{IOW}$  (запись во ВУ).

2. Регистр данных АЦП расположен внутри микросхемы АЦП по адресу: «Базовый адрес + 0». Он доступен при чтении и содержит двенадцать разрядов  $D00...D11$  данных цифрового эквивалента аналогового сигнала. Регистр обладает тремя состояниями, что позволяет осуществлять связь системы ввода-вывода с компьютерной шиной. Регистр данных выбирается сигналом «Выбор 0», формируемым схемой дешифратора адреса при обращении по базовому адресу системы ввода-вывода. Чтение данных из регистра осуществляется совместным действием сигналов «Выбор 0» и  $\overline{IOR}$  (чтение данных из ВУ).

3. Регистр состояния доступен только для чтения и расположен по адресу: «Базовый адрес + 2». Регистр содержит только один разряд, устанавливаемый сигналом  $DR$  готовности данных после завершения цикла преобразования АЦП. Регистр состояния выбирается сигналом «Выбор 2», формируемым схемой дешифратора адреса при обращении по адресу «Базовый адрес + 2» системы ввода-вывода и читается совместным действием сигналов «Выбор 2» и  $\overline{IOR}$ .

Мультиплексор аналоговых сигналов осуществляет подключение ко входу АЦП канала, номер которого определяется двоичным кодом, записанным в разряды  $D00...D01$  регистра управления. Реализуется мультиплексор каналов на аналоговых ключах  $S0...S4$  (например, микросхема  $K590KH3$ ), управляемых двоичным кодом. Кроме того, мультиплексор также реализует дифференциальный и недифференциальный ввод сигнала в АЦП. Режим ввода определяется разрядом  $D02$  регистра управления и реализуется на аналоговых ключах  $S4, S5$  (например, микросхема  $K590KH2$ ). Кроме аналоговых ключей, в состав мультиплексора входит операционный усилитель ( $OU1$ ), осуществляющий необходимое согласование и усиление сигнала. Четырехходовый мультиплексор взят в схеме в качестве примера. Мультиплексор на бóльшее число входов может быть построен аналогично. Характеристики выпускаемых отечественной промышленностью аналоговых ключей приводятся в справочнике.

Устройство выборки-хранения (УВХ), состоящее в простейшем случае из аналогового ключа  $S6$ , конденсатора  $C$  и повторителя напряжения  $OU2$ , может работать в двух режимах — выборки и хранения. Пока триггер  $T4$  регистра управления сброшен (находится в нулевом состоянии), ключ  $S6$  замкнут, а УВХ находится в режиме выборки и отслеживает поступающий на его вход аналоговый сигнал. После установки  $T4$  в единичное состояние  $S6$  замыкается, а УВХ переходит в режим хранения. При этом вход АЦП отключается от аналогового входа, что предотвращает влияние быстрых изменений во входном сигнале на результаты преобразования. При этом АЦП осуществляет преобразование сигнала, который запомнил конденсатор УВХ перед размыканием ключа  $S6$ . Более детальная информация по схемотехнике и характеристикам УВХ представлена в справочниках.

АЦП осуществляет непосредственное преобразование аналогового сигнала в цифровой эквивалент. Отечественная и зарубежная промышленность выпускают большое разнообразие микросхем АЦП, отличающихся по принципу действия, исполнению и характеристикам, но имеющих подобное управление. Не привязываясь к конкретному АЦП, охарактеризуем некоторый абстрактный АЦП, имеющий исполнение и управление, присущие большинству АЦП. На вход  $AI$  АЦП подается преобразуемый сигнал с УВХ. На вход  $C$  подается сигнал синхронизации с генератора. Подача импульса логической единицы на вход  $ST$  начинает преобразование. Появление сигнала логической единицы на выходе  $DR$  указывает, что преобразование завершено и цифровой эквивалент преобразуемого сигнала записан во внутренний регистр данных АЦП. Подача логической единицы на вход  $DE$  выводит выходные внутреннего регистра данных АЦП из высокоимпедансного состояния, что позволяет считать выходной код в шину данных компьютера. АЦП также содержит вход  $V/U$ , позволяющий выбрать биполярный или однополярный ввод аналогового сигнала в АЦП.

Предположим, что базовый адрес системы ввода-вывода  $3F0h$ . Если мы хотим ввести в компьютер сигнал, подсоединенный к нулевому дифференциальному каналу мультиплексора, чтобы сформировать требуемое подключение канала ко входу, необходимо по заданному адресу регистра управления «Базовый адрес + 0» сначала записать двоичное число «100». Запись числа по заданному адресу осуществляется программно. При этом процессор выставляет на адресных линиях адрес регистра управления «111110000», который по сигналу  $ALE$  записывается в регистр адреса. Схема дешифратора адреса формирует сигнал «Выбор 0». Сигнал записи данных во ВУ сопровождается установкой на микропроцессорной шине отрицательного сигнала  $\overline{IOW}$ . Сигнал  $\overline{IOW}$  совместно с сигналом «Выбор 0» осуществляет запись в регистр управ-



ления выставленного на линиях  $D0...D2$  шины данных компьютера двоичного числа «100». При этом выбирается дифференциальное подключение нулевого канала ко входу АЦП ( $S0$  — замкнут,  $S4$  — замкнут,  $S5$  — разомкнут). УВХ находится в режиме выборки входного сигнала ( $S6$  — замкнут, так как при записи «100» в  $T4$  записался «0»). Затем следует по тому же адресу записать двоичное число «1100», что при неизменности подключения приведет к переводу УВХ в режим хранения ( $S6$  — разомкнут). Установка триггера  $T4$  регистра управления в единичное состояние приведет к запуску ждущего одновибратора  $DD8$ , который сформирует на своем выходе положительный импульс определенной длительности, необходимый для запуска АЦП. АЦП начинает преобразование, по окончании которого на выходе  $DR$  устанавливается единичный сигнал, который сбрасывает триггер  $T4$  и устанавливает триггер  $T5$  регистра состояний. В результате УВХ опять отслеживает входной сигнал по тому же каналу, а АЦП готов к новому преобразованию. Кроме того, единичный сигнал триггера  $T5$  сигнализирует, что результаты преобразования могут быть считаны в компьютер. Управляющая работой системы ввода-вывода программа периодически обращается по адресу: «Базовый адрес + 2». Если при чтении по этому адресу на четвертой линии шины адреса  $D04$  будет прочитана единица, то компьютер получает подтверждение того, что АЦП выполнил преобразование и данные могут быть считаны. После этого процессор осуществляет операцию чтения данных из внутреннего регистра АЦП, которая сопровождается формированием сигнала «Выбор 0» и сигнала  $\overline{IOR}$ . При этом данные из внутреннего регистра АЦП пересылаются в процессор компьютера, а также производится обнуление триггера  $T5$  регистра состояний. Следующий цикл преобразования и чтения данных осуществляется аналогично.

Следует отметить, что при работе системы ввода-вывода возможно использовать режим прерываний, который может быть достигнут путем формирования сигнала требования прерываний в системе к компьютеру. На схеме (см. рис. 5.48) режим требования прерываний формируется с выхода триггера  $T5$  регистра состояний, который через переключатель  $S7$  соединен с линией требования прерываний  $IRQ3$  общей шины компьютера. Режим прерываний позволяет процессору компьютера не находиться в режиме ожидания во время выполнения преобразования АЦП, а выполнять другие задачи (расчеты).

Рассмотренная схема блока АЦП может быть упрощена, так как существуют АЦП, содержащие УВХ и мультиплексор в одном корпусе.

Ниже представлена подпрограмма на языке Ассемблер, позволяющая запускать АЦП и читать его регистр данных.

```

CONTRDAT = 03F0h      ; Определяем регистр управления/данных АЦП
ADCRD = 03F2h        ; Определяем регистр состояния АЦП
BIT2 = 100B          ; Определяем константы, необходимые для запуска АЦП
BIT3 = 1000B         ; Определяем переменную Val, которой будет передаваться преобразованное значение из регистра данных АЦП
Val DW?              ; Выбор дифференциального подключения нулевого канала
; мультиплексора
mov dx, CONTRDAT     ; Выбираем регистр CONTRDAT АЦП
mov ax, BIT2          ; Выбираем число 100B
out dx, ax            ; и записываем его в регистр CONTRDAT
; Перевод УВХ в режим хранения и запуск АЦП
mov dx, CONTRDAT     ; Выбираем регистр CONTRDAT АЦП
sub ax, ax            ; Обнуляем регистр ax
or ax, BIT2           ; Записываем в регистр ax число 100B
or ax, BIT3           ; и записываем его в регистр CONTRDAT
out dx, ax            ; и записываем его в регистр CONTRDAT
; Ожидание конца преобразования АЦП
L: mov dx, ADCRD      ; Выбор регистра состояния
   in ax, dx          ; Чтение текущего состояния ADCRD
   and ax, 010h       ; Если текущее состояние единичное, то идти к метке L
   cmp ax, 010h       ; В противном случае преобразование АЦП закончено и можно читать данные из АЦП
   je L               ; Чтение данных из регистра данных АЦП
mov dx, CONTRDAT     ; Выбираем регистр CONTRDAT
in ax, dx            ; Читаем данные из АЦП
mov Val, ax          ; Присваиваем значение преобразованной величины переменной Val

```

**Блок ЦАП.** В состав блока ЦАП входит регистр данных ЦАП с адресом: «Базовый адрес + 4» и собственно ЦАП (рис. 5.49). При выводе данных из компьютера в ЦАП управляющая работой системы ввода-вывода программа воздействует на процессор, кото-

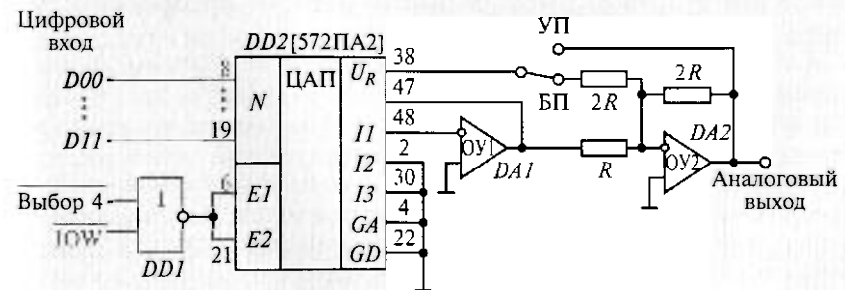


Рис. 5.49. Схема блока ЦАП

рый выставляет на адресных линиях адрес «111110100». При этом схемой дешифратора адреса формируется сигнал «Выбор 4». По выбранному адресу производится запись выводимого числа, которое записывается во внутренний регистр ЦАП. Затем ЦАП преобразует записанное число в аналоговый ток, который с помощью операционных усилителей преобразуется в аналоговое напряжение. Фиксированное напряжение сохраняется на выходе ЦАП до момента записи во внутренний регистр ЦАП нового числа. В блоке ЦАП предусмотрена возможность вывода биполярного (БП) или униполярного (УП) напряжения.

Ниже представлена подпрограмма на языке Ассемблер записи преобразуемой величины в ЦАП.

```
DACDAT = 03F4h      ; Определяем регистр состояния ЦАП
Val DW ?            ; Определяем переменную Val, которая
                    ; содержит преобразуемое значение
; Запись числа в регистр данных ЦАП
mov dx, DACDAT      ; Выбираем регистр данных ЦАП
mov ax, Val          ; Заносим выводимую переменную в ax
out dx, ax           ; и записываем ее в регистр данных ЦАП
```

Таким образом, мы кратко рассмотрели принципы построения и программирования универсальной системы ввода-вывода аналоговой информации в ЭВМ и из нее, позволяющей создавать компьютерные информационно-измерительные системы, а также цифровые системы управления разнообразными объектами или процессами.

Система ввода-вывода может быть реализована в виде универсальной платы стандартного подключения, размещаемой в одном из свободных слотов внутри корпуса компьютера и связанной с компьютером через общую шину. Универсальность системы и платы ввода-вывода заключается в том, что она может размещаться в любом IBM PC-совместимом компьютере и использоваться для любых задач цифрового управления или сбора и обработки информации, для которых достаточны реализованные в системе разрядность и быстродействие ввода-вывода информации.

Существенным преимуществом данной концепции является возможность использования как языка Ассемблера, так и языков высокого уровня для программирования и отладки программ, реализующих цифровые алгоритмы управления объектами или процессами, и программ управления системой ввода-вывода. Компьютер, используемый для задач управления и получения и обработки информации в течение ограниченных периодов времени, может быть использован для выполнения обычных задач расчетов, обработки текстов и т.д. в течение всего оставшегося времени.

## 5.10. Каналы ввода-вывода и аппаратура сопряжения

**Методы взаимодействия ЭВМ с периферийными устройствами. Система прерываний.** В данном разделе остановимся более детально на методах взаимодействия ЭВМ с периферийными устройствами и на организации системы прерываний.

Взаимодействие между ЭВМ и периферийным устройством начинается с момента, когда находящийся в памяти отправителя сообщения — ЭВМ или соответствующего ПУ, массив данных готов для передачи, и заканчивается тогда, когда данные записаны в память адресата — получателя сообщения (ПУ или ЭВМ).

В первых вычислительных машинах обмен информацией с ПУ осуществлялся через процессор, поскольку время обмена было относительно небольшим по сравнению с временем решения задачи. Такая организация обмена не требует выработки предварительных команд, здесь используется жесткая пространственно-временная организация связи между абонентами.

Позднее благодаря введению буферной памяти стал возможен внутренний обмен данными. Для организации обмена в памяти выделялся один дополнительный бит, называемый флагом. После записи данных из ПУ в буферную память флаг устанавливался в состояние «1». ЭВМ анализировала состояние флага, и если оно было равно «1», то данные из памяти ПУ считывались и устанавливался флаг состояния «0». Вторичное считывание из памяти исключалось, так как флаг находился в состоянии «0». При выводе данных из ЭВМ флаг устанавливался в состояние «1» и машина передавала данные в буферную память ПУ. Флаг устанавливался вновь в состояние «0» после завершения операции считывания из ЭВМ в буферную память ПУ.

ВС, в которых операции обмена с несколькими ПУ чередуются с обработкой информации нескольких программ, должны иметь возможность практически в любой момент времени вести обмен информацией сразу с несколькими ПУ и при этом совмещать его с обработкой информации. Такая ВС называется системой с разделением времени или мультипрограммной системой — время разделено между многими одновременно выполняемыми программами. Для возможности организации такого обмена используется система прерываний программ. Система прерываний обеспечивает временную приостановку выполнения текущей программы в ответ на различные внутренние и внешние ситуации, определяющие необходимость запуска других более важных программ.

Необходимость прерывания программы может быть вызвана внутренней причиной (переполнение разрядной сетки, попытка деления на нуль, нарушение правил адресации, возникновение сбоя в аппаратуре, завершение операции ввода-вывода) или вне-



Рис. 5.50. Упрощенная временная диаграмма процесса прерывания

шним воздействием: аварийная ситуация, обнаружение аппаратурой контроля ошибки в передаваемой информации и др. Каждое событие, требующее прерывания, сопровождается сигналом, оповещающим о его возникновении и называемым запросом прерывания (ЗП). Устройства, выставляющие оповещающий сигнал ЗП, называются активными.

На рис. 5.50 приведена временная диаграмма процесса прерывания. Для оценки эффективности системы прерывания используется показатель — время обслуживания запроса:

$$T_{\text{обсл}} = t_{\text{зап}} + t_{\text{в}},$$

где  $t_{\text{зап}}$  — время запоминания состояния прерываемой программы;  $t_{\text{в}}$  — время восстановления этой прерванной программы.

Время реакции на ЗП  $t_p$  — время между появлением запроса и началом выполнения прерывающей программы. Оно гораздо больше, чем время запоминания. Для одного и того же запроса время реакции зависит от того, сколько программ со старшим приоритетом ждет обслуживания. Поэтому время реакции определяют для запроса с наивысшим приоритетом. Если время реакции настолько велико, что запрос окажется необслуженным к моменту прихода нового запроса от того же источника, то возникает насыщение системы прерывания, предыдущий запрос будет утрачен.

Программы могут быть неравнозначны по приоритету — по рангу решаемой задачи, т. е. по степени срочности ее выполнения. Иначе говоря, приоритет определяет, имеет ли право данная пре-



Рис. 5.51. Прерывание в системах с различной глубиной прерывания

рывающая программа прерывать обрабатываемую в данный момент программу. Число программ, которые могут прерывать друг друга называется глубиной прерывания. На рис. 5.51 приведена временная диаграмма процесса прерывания в ВС с глубиной прерывания, равной 1 и 3. Программа 3 с высшим приоритетом начинает выполняться сразу же после поступления от нее ЗП, прервав выполнение других программ, с более низким приоритетом, что ускоряет ее выполнение.

При одновременном появлении нескольких ЗП воспринимается запрос, соответствующий уровню с меньшим номером; таким образом, приоритет является жестким и определяется порядком присоединения линий передачи запросов ко входам системы прерывания. Устройство (программа) с высшим приоритетом «захватывает» магистраль.

Схемы, позволяющие определить номер уровня со старшим приоритетом, называются схемами арбитра. На рис. 5.52 приведена схема подключения арбитра к магистралям интерфейса типа ОШ — Малтибус (Multibus) для однокристальных микропроцессорных наборов типа K580, K1810. Активные устройства, от которых может поступить ЗП, расположены в виде матрицы и разбиты на 8 групп (в соответствии с 8 линиями связи интерфейса, по которым передается однобайтовое слово). Приоритет отдается устройствам верхней группы. BR $i$  — управляющие линии для передачи  $i = 0 \dots 7$  сигналов ЗП в арбитра. BG $i$  — управляющие линии для передачи сигналов разрешений от арбитра к активным устройствам. SACK — оповещающая линия передачи ответа о подключении к магистрали первоочередного претендента, того, у которого самый высокий из активных устройств приоритет. По получении сигнала SACK арбитра прекращает выдачу сигнала разрешения. BBSY — линия передачи сигнала, оповещающего все



Рис. 5.52. Подключение схемы арбитра к магистрали типа ОШ (Multibus, Unibus)



устройства о занятости (захвате) магистрали. По установлению режима захвата система переходит к прерыванию, затем к записи либо чтению. После ее освобождения очередник занимает линию и сам формирует сигнал BBSY. R — вход сброса арбитра. Все ЗП складываются по схеме ИЛИ на линиях запроса BRi. Разрешение выдается арбитром группе с наименьшим номером. В группе разрешение получает устройство, которое ближе к арбитру. Пока идет обмен, арбитр готовит нового следующего очередника.

Схема арбитра здесь имеет жесткую логическую структуру. Каналы с высокими приоритетами оттесняют низкоприоритетные, в результате чего они могут простаивать длительное время и даже полностью лишиться права доступа к магистрали. Для повышения гибкости работы арбитра необходимо периодически перестраивать приоритетную структуру, что достигается применением программного управления приоритетами.

На рис. 5.53 приведена схема арбитра, реализующая 128 различных приоритетных соотношений. Операционная система каждую миллисекунду меняет по определенному алгоритму приоритеты между 8 каналами связи с ПУ в зависимости от кода  $Q_1...Q_7$  на управляющих входах арбитров Ар (табл. 5.2).

Схема механизма прерывания показана на рис. 5.54. Предположим, что в момент выполнения команды A, основной прерываемой программы, происходит какое-нибудь событие, например, завершение обмена с одним из ПУ. В данный момент основную программу следует прервать, чтобы дать возможность прерывающей программе X обработать переданные из ПУ новые данные. С этой целью вырабатывается сигнал ЗП, по которому команда A прерывающей программы X заносится в счетчик команд (СчК), а команда A + 1 — в специальный регистр адреса возвращения (РгАВ). По окончании работы программы X управление возвращается к команде A + 1 основной прерванной программы, путем переноса команды A + 1 из РгАВ в СчК. Основная программа про-

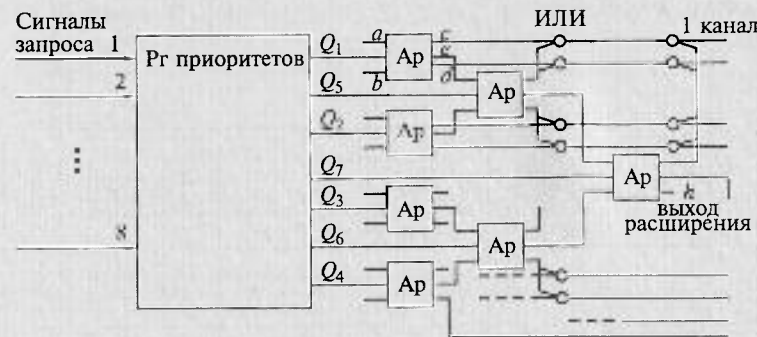


Рис. 5.53. Схема программного управления приоритетами

Таблица истинности схемы арбитра

a	b	$Q_i$	c	d	e
0	0	x	0	0	0
0	1	x	0	1	1
1	0	x	1	0	1
1	1	0	0	1	1
1	1	1	1	0	1

должается с команды A + 1, следующей сразу же за той, на которой произошло прерывание.

Количество причин прерывания может достигать от 16 в персональных компьютерах до нескольких десятков и сотен в ВС и суперЭВМ. В таких случаях систему прерываний выполняют многоуровневой (рис. 5.55). Запросы (1) от всех источников прерывания поступают на регистр запросов прерывания (РгЗП), устанавливая соответствующие ему разряды в состоянии «1», указывающее на факт наличия запроса прерывания определенного источника (причины). Запросы 1-го...k-го уровней прерывания (ЗПУ) формируются элементами ИЛИ, объединяющими разряды РгЗП, относящиеся к соответствующим уровням. Еще одна схема ИЛИ формирует общий сигнал прерывания (ОСП), поступающий в устройство управления процессора. Значение сигнала ОСП определяется выражением

$$ОСП = \bigvee_{i=1}^k ЗПУ_i = \bigvee_{i=1}^k (\bigvee_j I_{ji})_i.$$

Информация о действительной причине прерывания, породившей запрос данного уровня ЗПУ<sub>i</sub>, содержится в «коде прерывания», который отражает состояние разрядов РгЗП, относящихся

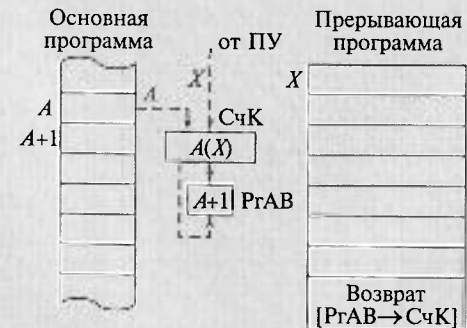


Рис. 5.54. Схема механизма прерывания

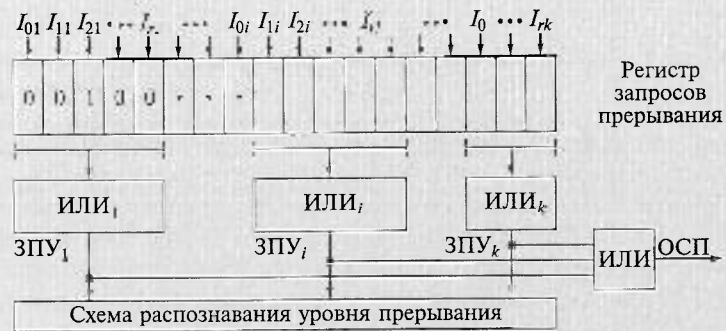


Рис. 5.55. Многоуровневая система прерывания

к данному уровню прерывания. Обычно в многоуровневых системах прерывания распознавание активного уровня прерывания производится аппаратными средствами, а конкретная причина прерывания определяется прерывающей программой по коду прерывания. После принятия запроса прерывания на исполнение и передачи управления прерывающей программе соответствующий триггер РгЗП сбрасывается.

Многоуровневое построение позволяет уменьшить объем аппаратуры, но приводит к замедлению работы системы прерывания. Одноуровневая система, в которой понятия запроса (причины) и уровня прерывания объединяются, позволяет достичь большего быстродействия в работе системы прерывания.

**Организация обмена массивами данных.** Обмен между центральными и периферийными устройствами производится за одну операцию данными, объединенными в массивы, называемые также блоками или кадрами. Обмен инициируется командой, исполняемой ЭВМ, затем протекает автономно, без участия центрального процессора, причем обмен ведется с заранее выделенной областью памяти. На рис. 5.56 показана схема механизма передачи  $N$  слов единым массивом.

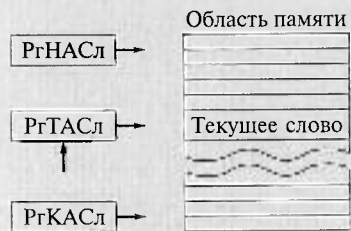


Рис. 5.56. Схема механизма передачи  $N$  слов единым массивом

слов; при этом предполагается, что слова по одному заносятся в память или считываются из нее. Имеется три адреса, которые нужно помнить: начальный, текущий и конечный. Они хранятся в трех специальных регистрах, называемых регистрами начального РгНАСл, конечного РгКАСл и текущего РгТАСл адреса слов. Вначале текущий адрес слова совпадает с начальным адресом. Затем, по мере того как очередное слово заносится в память (или считывается из нее), те-

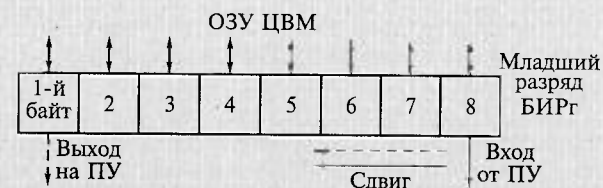


Рис. 5.57. Работа буферного информационного регистра в режиме ввода (сплошные стрелки) и вывода (пунктирные)

кущий адрес слова увеличивается на единицу, так что всякое очередное слово будет заноситься в следующий адрес (или же считываться из него). Перед каждым изменением текущего адреса оно сравнивается с конечным адресом слова. Если текущий адрес совпадает с конечным, то обмен массивами прекращается и вырабатывается сигнал прерывания, по которому ЭВМ переходит к исполнению соответствующей программы. Для выработки текущих адресов РгТАСл должен выполнять функции счетчика.

Рассмотрим прием и передачу информации в виде восьмибайтового слова (рис. 5.57). В режиме вывода слово из ОЗУ ЭВМ записывается за один такт параллельным кодом в буферный информационный регистр (БИРг). Цепи сдвига регистра обеспечивают сдвиг кода влево на 8 разрядов. На выходной шине (Выход) появляется восьмиразрядный код, который поступает на вход ПУ либо фиксируется на носителе информации ВЗУ. Восьмикратная передача байтов обеспечивает передачу всего слова из БИРг в ПУ.

В режиме ввода первый байт слова принимается от ПУ по входным шинам и записывается в 8 младших разрядов регистра. Затем эти разряды сдвигаются влево и от ПУ принимается следующий байт. После приема всего слова оно параллельным кодом передается в ОЗУ. Поскольку обмен между ОЗУ и ПУ осуществляется массивами — страницами или зонами, то емкость БЗУ выбирается равной информационной емкости массива. Выигрыш во времени  $\Delta t$  при такой организации обмена:

$$\Delta t = \Delta t_2 - \Delta t_1,$$

где  $\Delta t_2$  — время передачи массива из БЗУ в ПУ;  $\Delta t_1$  — время передачи массива из ОЗУ в БЗУ.

При наличии лишь одного блока (секции) ОЗУ время  $\Delta t$  для выполнения вычислений процессором не используется. Если же использовать несколько секций ОЗУ и соответствующим образом организовать потоки информации, то передачу массивов можно проводить между БЗУ и теми секциями ОЗУ, которые не участвуют в вычислениях в данный промежуток времени (рис. 5.58).

Обмен массивами освобождает процессор от операции обмена, оставляя за ним только ввод заказов на обмен и передачу

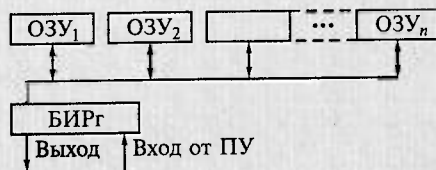


Рис. 5.58. Организация обмена между ПУ и ОЗУ ЭВМ

управления соответствующей программе после окончания обмена. ОЗУ используется в процессе обмена только при вводе (выводе) данных, причем эти обращения чередуются с обращениями процессора по своим программам.

Однако обмен массивами имеет недостаток, заключающийся в необходимости дополнительного оборудования, обеспечивающего собственно обмен и управление им. Аппаратура передачи данных, представляющая собой совокупность каналов обмена и устройств сопряжения с ПУ, на время обмена выделяет каждому ПУ подканал. В нем осуществляется хранение управляющей информации (адреса размещения данных обмена, набора управляющих признаков, определяющих режимы обмена) об обмене с одним из ПУ, подлежащих обслуживанию. Подканалы могут быть разделенными и неразделенными. Разделенный подканал используется для связи с несколькими ПУ с разделением во времени. Неразделенный подканал осуществляет обмен с одним определенным ПУ.

Каналы, образуемые совокупностью подканалов, осуществляют прием и расшифровку запросов на обмен, хранение запросов на обмен до их завершения, введение управляющих слов в подканалы, логическое подключение ПУ к конкретным подканалам. Каналы обмена включают в себя наборы линий связи, по которым осуществляется передача информации и сигналов управления.

Каналы ввода-вывода позволяют вычислительной системе выполнять одновременно ввод, вывод и обработку информации, освобождая центральный процессор от непосредственного управления вводом-выводом. Конструктивно каналы могут быть либо независимыми устройствами, имеющими в своем составе все необходимые логические устройства и память, либо могут быть частью процессора и использоваться для выполнения своих функций оборудование центрального процессора. В обоих случаях функции канала остаются одинаковыми. В основном они заключаются в выборе заданного УВВ, управлении передачей информации между УВВ и ОЗУ, распределении данных при обмене, промежуточном хранении информации ввода-вывода, преобразовании форматов вводимых и выводимых данных, контроле правильности выполнения ввода-вывода.

По характеру обслуживания ПУ различают мультиплексный и селекторный каналы.

**Мультиплексный канал.** Мультиплексный канал (МК) позволяет одновременно обслуживать несколько параллельно работающих ПУ. Каждое из них подключается к каналу на короткие промежутки времени, называемые сеансами связи, только после того, как ПУ подготовлено к приему или выдаче очередного массива информации, накопленного в БЗУ. Для организации такого режима работы ВС используется система прерываний. В МК сеансы связи с различными ПУ чередуются между собой в зависимости от их готовности к обмену (о чем свидетельствует сигнал «Готов») либо выполняются в зависимости от их относительных приоритетов. МК предназначен для работы с медленно действующими ПУ — принтерами, плоттерами, оперативными УВВ.

Средства МК, обеспечивающие обмен с одним ПУ, образуют подканал. Общее количество подканалов определяет максимальное число ПУ, которое может одновременно обслуживаться данным каналом. При работе с разделением времени число ПУ, запросы которых могут быть обслужены и не потеряны, т. е. система прерывания не войдет в насыщение, будет связано с техническими параметрами ЭВМ следующим образом:

$$N = kT_{\text{обсл}} C_{\text{БЗУ}} V / nW_{\text{max}},$$

где  $N$  — число ПУ;  $k$  — коэффициент размерности, равный  $1/\text{с} \cdot \text{бит}^2$ ;  $T_{\text{обсл}}$  — время обслуживания одного запроса, т. е. время передачи единицы информации, например, байта;  $C_{\text{БЗУ}}$  — информационная емкость БЗУ для накопления передаваемого массива;  $V$  — быстродействие центрального процессора;  $n$  — глубина прерывания;  $W$  — пропускная способность канала связи. Параметр  $N$  иногда называется нагрузочной способностью канала.

Пропускная способность канала  $W_{\text{max}}$ , т. е. максимально допустимая скорость передачи информации как для МК, так и для селекторного канала, вычисляется по формуле Шеннона:

$$W_{\text{max}} = \Delta F \ln(1 + p_c/p_n),$$

где  $\Delta F$  — ширина полосы пропускания частот канала,  $p_c$  и  $p_n$  — средняя мощность полезного сигнала и сигналов помех соответственно. За среднюю мощность сигналов принято считать значение их вольт-секундной площади  $S_c$  и  $S_n$ .

#### Пример 5.1

Рассчитать пропускную способность линии связи внутренней шины с шириной полосы пропускания  $\Delta F = 10$  МГц, информационные сигналы прямоугольной формы имеют среднюю вольт-секундную площадь полезного сигнала  $S_c = 3,6$  В·нс и среднюю вольт-секундную площадь сигналов помех  $S_n = 2,1$  В·нс:

$$W_{\text{max}} = 10 \ln(1 + 3,6/2,1) = 10.$$



Это означает, что частота передаваемых информационных сигналов не должна превышать 10 МГц.

**Пример 5.2**

Рассчитать максимально допустимое количество ПУ в ВС с тем, чтобы не происходило насыщения системы прерываний в ВС со следующими значениями параметров:  $T_{\text{обсл}} = 3 \text{ с}$ ,  $C_{\text{БЗУ}} = 64 \cdot 10^2 \text{ бит}$ ,  $V = 10^4 \text{ бит/с}$ ,  $n = 10$ ,  $W_{\text{max}} = 100 \text{ МГц}$ . После расчета получим, что система прерываний не войдет в насыщение при подключении к каналу 192 ПУ.

В состав МК входят следующие основные узлы (рис. 5.59).

1. Набор регистров, где во время сеанса связи с ПУ находятся текущие параметры активного подканала (для которого проводится данный сеанс связи): РгКОУ содержит код и указатели операции ввода-вывода; РгТАД содержит текущий адрес данных; счетчик текущих данных (СчТД) указывает количество байтов, которое осталось ввести в ОЗУ или вывести из него в данной операции; РгАУСл определяет адрес очередного управляющего слова в цепи управляющих слов.

2. Память подканалов, представляющая собой внутреннюю память канала, предназначена для хранения текущих параметров операций ввода-вывода, относящихся к пассивным подканалам.

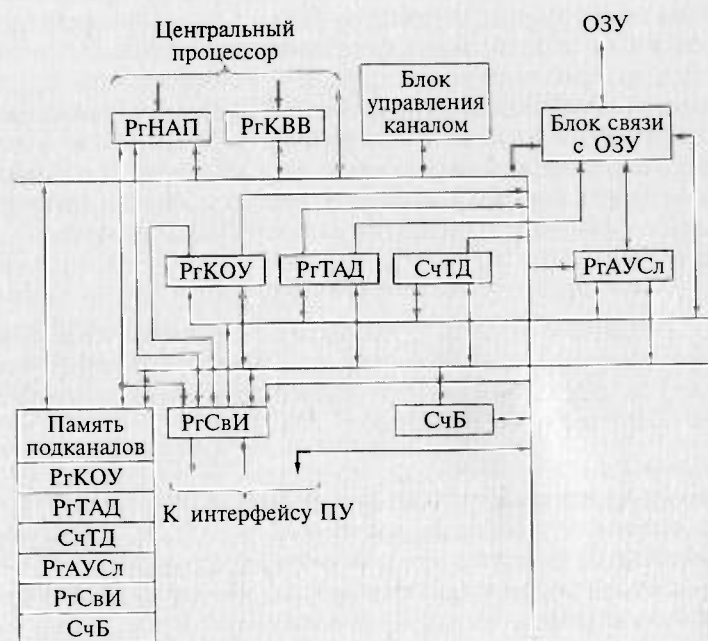


Рис. 5.59. Упрощенная схема мультиплексного канала

Каждому подканалу отведен в памяти участок, в котором хранится содержимое всех четырех вышеперечисленных регистров РгКОУ, РгТАД, СчТД, РгАУСл, соответствующее последнему сеансу связи для данного подканала. Кроме того, в памяти подканалов предусматривается место для хранения других параметров операции, о которых будет сказано позже. Подканалы в памяти располагаются упорядоченно по возрастанию номера подканала, который, таким образом, может использоваться для адресации ячеек памяти подканалов. При формировании адресов ячеек памяти подканалов используется содержимое регистра номера активного подканала (РгНАП).

3. Регистр связи с интерфейсом (РгСвИ), куда поступает информация, получаемая из ПУ при вводе, и откуда в ПУ выводится выводимая информация. Обращение к ОЗУ осуществляется через блок связи с ОЗУ (БСОЗУ). Этот блок в соответствии с алгоритмом выполняемой операции выбирает один из регистров канала в качестве источника адреса, другой — в качестве источника или приемника информации и вырабатывает все сигналы, необходимые для обмена информацией с ОЗУ. Обмен информацией канала с ОЗУ производится словами, обмен с ПУ производится обычно более мелкими единицами информации, например байтами. Поэтому при вводе канал формирует слова из поступающих в канал байтов, а при выводе — разворачивает слова в последовательность байтов. Для определения конца преобразования слов используется счетчик байтов (СчБ), указывающий номер последнего обработанного байта в текущем слове данных. Содержимое СчБ и слово данных, участвующее в обмене, запоминаются в соответствующем участке памяти подканалов.

4. Регистр команд ввода-вывода (РгКВВ) хранит код команды, поступающей в канал из процессора, когда процессор в соответствии с программой, называемой планировщиком, запускает новую операцию ввода-вывода.

**Селекторный канал.** С помощью селекторного канала (СК) к центральному устройству подключаются быстродействующие периферийные устройства, способные ожидать в очереди начала обмена. В такой вычислительной системе информация обрабатывается в пакетном режиме, один пакет программы последовательно за другим. (Слово «пакет» сохранилось в терминологии с тех пор, когда программа поступала на вычислительный центр от программиста в виде пакета перфокарт.) Пока не будет выполнена текущая программа, к выполнению другой не приступают. Селекторный канал обеспечивает, таким образом, обмен информацией в монопольном режиме, когда все средства канала на время обмена монополизуются одним ПУ. В каждый момент времени обмен ведется только с одним внешним устройством. Началу новой процедуры обмена должно предшествовать завершение ранее

начатой процедуры. СК представляет собой, по существу, один разделенный подканал мультиплексного канала. В случае подключения к СК, ПУ после начала обмена остается связанным с каналом до окончания обмена информацией.

На рис. 5.60 приведена структурная схема селекторного канала. СК содержит набор регистров, большинство из которых по своим функциям аналогичны соответствующим регистрам вышеописанного МК. Это регистры RгКВВ, RгКОУ, RгТАД, СчТД, RгАУСл, RгСвИ, СчБ. Регистр номера ПУ (RгНПУ), заполняемый процессором при начальной выборке, указывает, с каким из устройств СК проводится текущая операция. Регистр данных (RгД) и регистр предварительного управляющего слова (RгПУСл) служат для уменьшения задержек в работе селекторного канала при обращениях к ОЗУ.

Процедуры работы СК во многом похожи на процедуры МК. Основное отличие состоит в том, что текущие параметры операции в СК в течение всей операции содержатся и модифицируются в регистрах. Кроме того, за счет введения дополнительных буферных регистров RгД и RгПУСл в СК обеспечивается возможность совмещения во времени обмена информацией с ПУ и с ОЗУ.

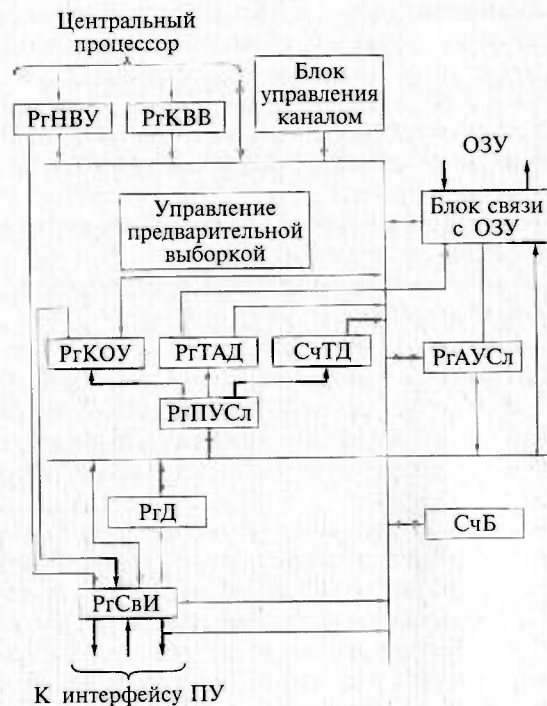


Рис. 5.60. Упрощенная схема селекторного канала

В состав аппаратуры ВС на базе ЭВМ средней производительности входят каналы двух основных типов: один мультиплексный и селекторные (от двух до шести в зависимости от модели). К каждому каналу может быть подключено до 256 УВВ (рис. 5.61). МК может работать как в мультиплексном, так и в монопольном режиме, СК всегда работает в монопольном режиме. В машинах начиная с четвертого поколения функции каналов передаются сравнительно быстродействующей ЭВМ, имеющей память с быстродействием существенно более высоким, чем быстродействие памяти ПУ.

**Устройства сопряжения — мультиплексоры передачи данных.** Связующим звеном в системах автоматизированной обработки данных между ЭВМ и абонентами (объектами управления в АСУ) являются устройства сопряжения, которые часто называют мультиплексорами передачи данных (МПД) или концентраторами, и включают в состав ЭВМ. МПД выполняет две основные функции: передатчика и приемника, поэтому еще одно из его названий — приемопередатчик. Функцию передатчика в каждый момент времени может выполнять одно или несколько устройств, генерирующих сообщения.

МПД выполняет следующие функции по организации сопряжения ЭВМ с ПУ:

- установление связи с ЭВМ и выдача соответствующих сигналов о готовности ЭВМ к работе;
- установление связи с ПУ и выдача соответствующих сигналов о готовности ПУ к работе;
- организация обмена (обслуживания) ПУ;
- проведение дополнительных подготовительных операций исходя из специфики ПУ (выдержка необходимых временных интервалов, выяснение пароля и т. п.);
- прием (выдача) информации (сообщения) от ПУ и формирование сообщения о приеме (выдаче);

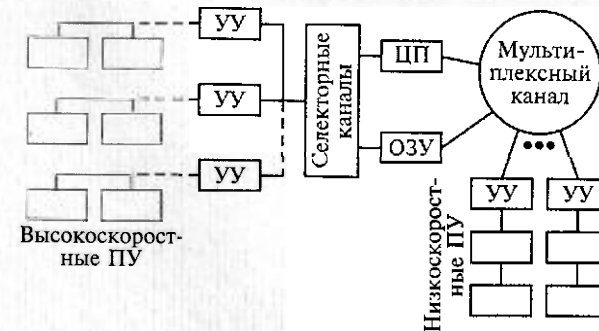


Рис. 5.61. Место ПУ в структуре ЭВМ средней производительности

- собственно передача сообщения от ЭВМ;
- запоминание или выдача информации в (от) ЭВМ, частичная (предварительная) обработка передаваемой информации;
- анализ состояния передачи;
- формирование сообщения ПУ;
- отключение от ЭВМ.

Зачастую ПУ, которые необходимо подключить к ЭВМ, имеют различные временные диаграммы обмена и параметры сигналов в линиях связи, разные составы управляющих и информационных сигналов и др. Это также одна из причин создания аппаратуры устройств сопряжения канала обмена с ПУ. Работы по дальнейшему совершенствованию устройств сопряжения продолжают вестись с целью расширения возможностей ЭВМ, поэтому аппаратура сопряжения чрезвычайно разнообразна по своему составу. На рис. 5.62 показана схема устройства сопряжения и место, которое оно занимает в системе обмена. Мультиплексор организует обмен информацией канала с несколькими ПУ в мультиплексном режиме. Он работает в режиме разделения времени, осуществляя пословную передачу информации между каналом обмена и одним из  $n$  трансляторов, являющихся, в свою очередь, согласующими блоками между мультиплексором и ПУ. Транслятор принимает или передает очередное информационное слово и выдает сигнал готовности к обмену с ЭВМ. Мультиплексор и трансляторы выполняются в виде взаимозаменяемых модулей, что позволяет

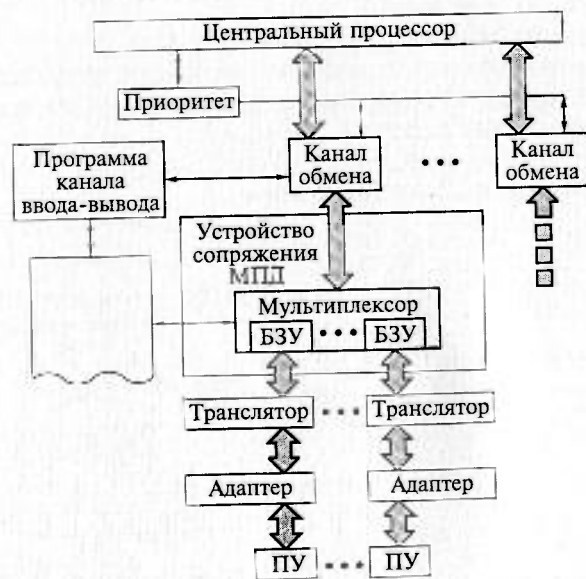


Рис. 5.62. Организация сопряжения ЦП с ПУ

с помощью устройства сопряжения к различным типам ЭВМ подключать различные ПУ.

Схема арбитража приоритета анализирует готовность ПУ и выбирает ПУ, имеющее старший приоритет. Выбранный арбитром номер транслятора передается в канал обмена, который извлекает из памяти подканала текущие параметры процесса обмена, заносит их в соответствующие регистры и готовит свои цели для передачи очередного слова информации. На этом цикл передачи заканчивается и аппаратура становится готовой к следующему циклу передачи.

Архитектура мультиплексора зависит от принципов реализации связи с ЭВМ. Рассмотрим управление вводом-выводом при реализации связи ЭВМ — мультиплексор. Система ввода-вывода представляет собой совокупность аппаратных и программных средств. Для обеспечения эффективной работы системы ввода-вывода программа обработки данных, выполняемая в это время ЦП, должна иметь возможность получать от мультиплексора требуемую информацию сразу же, как только в ней возникает необходимость, и выводить результаты с той же скоростью, с какой она их получает. В задачу мультиплексора входят подсоединение буферной памяти к абонентам и наблюдение за тем, чтобы буфер наполнялся по мере его освобождения и оказывался пустым по мере того, как ЦП собирается его наполнить. Информация в БЗУ должна храниться блоками, объем блока определяется свойствами абонента.

Подсоединение МПД к каналу обмена ЭВМ осуществляется в свою очередь, через блок сопряжения с каналом (БСК) (рис. 5.63). Обмен производится по протоколу обмена — строго заданной процедуре, совокупности правил, регламентирующих способ выполнения определенного класса функций. От канала отходят: шина данных, по которой осуществляется обмен данными между ЭВМ и МПД; шина управления, передающая управляющие сигналы сопряжения МПД с каналом; шина выборки подканала, с которым устанавливается связь с МПД; шина выборки абонента, выбирающая соответствующего абонента. Шины выборки позволяют устанавливать двустороннюю связь между каналом ЭВМ и ПУ.



Рис. 5.63. Организация подсоединения МПД к каналу ЭВМ



Данные ПУ поступают на регистр данных, узел управления выдает соответствующий сигнал в канал, после чего осуществляется передача данных в (из) ЭВМ.

При подсоединении к абонентам МПД выполняет следующие операции: устанавливает связь с абонентом и выдает соответствующие сигналы о готовности; проводит дополнительные подготовительные операции исходя из специфики абонента (выдержка необходимых временных интервалов, выявление пароля, приоритета и т.п.); принимает (выдает) информацию (сообщение) от абонента и формирует сообщение о приеме (выдаче); передает сообщение в ЭВМ; запоминает или передает информацию в (от) ЭВМ; анализирует состояние передачи; формирует сообщение абоненту; отключает от абонента.

Все перечисленные операции сопряжения МПД с абонентом являются общими, однако их реализация для каждого абонента носит сугубо специфичный характер. Рассмотрим в качестве примера структурную схему адаптера (рис. 5.64). Узел прерывания принимает управляющие сигналы из МПД и каналов связи с абонентом и вырабатывает управляющие сигналы для узлов адаптера. Узел синхронизации вырабатывает синхросигналы для реализации операции по передаче информации. Узел сравнения сравнивает управляющие сигналы и слова передаваемых сообщений. Приведенная схема отражает общие принципы организации сопряжения абонента с ЭВМ и может быть реализована в различных вариантах с перераспределением функций отдельных узлов.

При организации сопряжения МПД с абонентами через стандартные каналы связи реализуются три метода синхронизации (доступа):

- 1) стартстопный, применяемый при обмене по телеграфным каналам связи; при этом синхронизация передатчика и приемника устанавливается заново для каждого символа;
- 2) синхронный поблочный, обеспечивающий передачу информации по телефонным каналам связи без ожидания ответа о при-

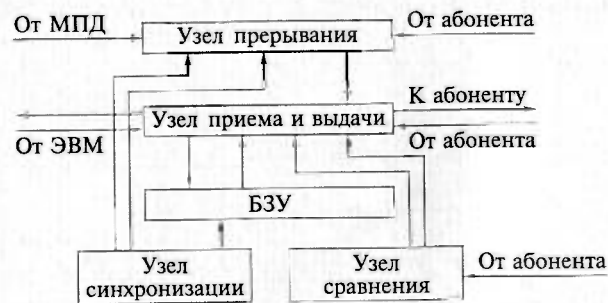


Рис. 5.64. Структурная схема адаптера



Рис. 5.65. Классификация ВС

еме с использованием узкого канала связи. Этот метод использует отдельный канал синхронизации, выделенный из канала для передачи данных;

3) синхронный позначный, обеспечивающий передачу данных по телефонным каналам связи с восстановлением синхронизации по специальным знакам.

Практическим примером организации унифицированных связей и устройств сопряжения составных элементов ЭВМ и систем могут служить интерфейсы.

На приведенной на рис. 5.65 укрупненной классификации ВС удобно просмотреть необходимые для создания ВС типы МПД, интерфейсов, каналов, устройств сопряжения.

## 5.11. Сети ЭВМ, назначение, характеристики

**Основы сетевых технологий.** Сеть — это соединение между двумя или более компьютерами, позволяющее им разделять ресурсы. Хотя большинство сетей значительно сложнее, все они базируются на принципе предоставления ресурсов в совместное использование. Когда один компьютер соединяется с другим, проходит большая и сложная работа многих программных и аппаратных компонентов сети. Существует множество типов физических соединений и соответствующего программного обеспечения.

Существуют три типа сетей: Local Area Network, LAN (локальная сеть); Wide Area Network, WAN (глобальная сеть) и Metropolitan Area Network, MAN (городская сеть). Локальная сеть представляет собой набор соединенных в сеть компьютеров, расположенных в пределах небольшого физического региона, например одного здания. Глобальные сети могут соединять сети по всему миру; для межсетевых соединений обычно используются сторонние средства коммуникаций. Региональные сети используют технологии глобальных сетей для объединения локальных сетей в конкретном географическом регионе, например в городе.

В дополнение к этим различиям, локальные, глобальные и региональные сети могут быть одноранговыми сетями, сетями с выделенным сервером или смешанными сетями (в которых используются как одноранговые технологии, так и технологии с выделенным сервером).

**Одноранговые сети.** Компьютеры в одноранговых сетях могут выступать как в роли клиентов, так и в роли серверов. Так как все компьютеры в этом типе сетей равноправны, одноранговые сети не имеют централизованного управления разделением ресурсов. Любой из компьютеров может разделять свои ресурсы с любым компьютером в той же сети. Одноранговые взаимоотношения также означают, что ни один компьютер не имеет ни высшего приоритета на доступ, ни повышенной ответственности за предоставление ресурсов в совместное пользование.

Каждый пользователь в одноранговой сети является одновременно сетевым администратором. Это означает, что каждый пользователь в сети управляет доступом к ресурсам, расположенным на его компьютере. Он может дать всем остальным неограниченный доступ к локальным ресурсам, дать ограниченный доступ, а может не дать вообще никакого доступа другим пользователям. Каждый пользователь также решает, дать другим пользователям доступ просто по их запросу или защитить эти ресурсы паролем.

Благодаря гибкости и свободе в отношении сетевых ресурсов организационный хаос в одноранговой сети является нормой. По этим причинам основной проблемой в одноранговых сетях является безопасность. Вы можете установить одноранговую сеть, объединив ресурсы в рабочие группы, но у вас не будет средств обеспечения безопасности в масштабе сети. При установке только те пользователи в рабочей группе, которые знают пароль, могут получить доступ к ресурсам.

Этот тип сети может быть работоспособным в маленьких сетях, но требует, чтобы пользователи знали и помнили различные пароли для каждого разделенного ресурса в сети. С ростом количества пользователей и ресурсов одноранговая сеть становится не работоспособной. Это происходит не потому что сеть не может

функционировать правильно, а потому что пользователи не в состоянии справиться со сложностью сети.

К тому же большинство одноранговых сетей состоит из набора типичных персональных компьютеров, связанных общим сетевым носителем. Эти типы компьютеров не были разработаны для работы в качестве сетевых серверов, поэтому производительность сети может упасть, когда много пользователей попытаются одновременно получить доступ к ресурсам какого-то одного компьютера. К тому же пользователь, к чьей машине происходит доступ по сети, сталкивается с падением производительности в то время, когда компьютер выполняет затребованные сетевые службы. Например, если к компьютеру пользователя подключен принтер, к которому осуществляется доступ по сети, компьютер будет замедлять свою работу каждый раз, когда пользователи посылают задание на этот принтер. Это может раздражать того, кто работает на данной машине.

В одноранговой сети также трудно организовывать хранение и учет данных. Когда каждый сетевой компьютер может служить сервером, пользователям трудно отслеживать, на какой машине лежит интересующая их информация. Если 10 пользователей отвечают за набор документов, каждому из них может потребоваться искать конкретный файл на всех 10 машинах. Децентрализованная природа такого типа сети делает поиск ресурсов чрезвычайно сложным с ростом числа узлов, на которых должна происходить проверка. Децентрализация также затрудняет процедуру резервного копирования данных — вместо копирования централизованного хранилища данных вам нужно осуществлять резервное копирование на каждом сетевом компьютере, чтобы защитить разделенные данные.

На первый взгляд может показаться, что все эти проблемы и увеличивающаяся сложность делают одноранговые сети не стоящими рассмотрения. Однако одноранговые сети имеют серьезные преимущества перед сетями с выделенным сервером, особенно для маленьких организаций и сетей. Одноранговые сети являются наиболее легким и дешевым типом сетей для установки. Большинство одноранговых сетей требует наличия на компьютерах, кроме сетевой карты и сетевого носителя (кабеля), только операционной системы. Как только компьютеры соединены, пользователи немедленно могут начинать предоставление ресурсов и информации в совместное пользование.

Некоторые преимущества одноранговых сетей:

- они легки в установке и настройке;
- отдельные машины не зависят от выделенного сервера;
- пользователи в состоянии контролировать свои собственные ресурсы;
- этот тип сетей недорог в приобретении и эксплуатации;



- не нужно никакого оборудования или программного обеспечения, кроме операционной системы;
- нет необходимости иметь отдельного человека в качестве выделенного администратора сети;
- этот тип хорошо подходит для сетей с количеством пользователей, не превышающим десяти.

Недостатки одноранговых сетей:

- пользователи должны помнить столько паролей, сколько имеется разделенных ресурсов;
- нужно производить резервное копирование отдельно на каждом компьютере, чтобы защитить все совместные данные;
- когда кто-нибудь получает доступ к разделенному ресурсу, на компьютере, где этот ресурс расположен, ощущается падение производительности;
- не существует централизованной организационной схемы для поиска и управления доступом к данным.

**Сети с выделенным сервером.** Сети с выделенным сервером опираются на специализированные компьютеры, называемые серверами, представляющими собой централизованные хранилища сетевых ресурсов и объединяющими централизованное обеспечение безопасности и управления доступом. Сервер представляет собой машину, чьей единственной функцией является реакция на клиентские запросы. Серверы редко управляются кем-то непосредственно — и то обычно только для того, чтобы установить, настроить или обслуживать его. В общем, сервер представляет собой сочетание специализированного программного обеспечения и оборудования, которое предоставляет службы в сети для остальных компьютеров (рабочих станций) или других процессов.

Имеется несколько причин для реализации сети с выделенным сервером, включающих централизованное управление сетевыми ресурсами путем использования сетевой безопасности и управление посредством установки и настройки сервера. С точки зрения оборудования, серверные компьютеры обычно имеют более быстрый центральный процессор, больше памяти, большие жесткие диски и дополнительные периферийные устройства, например накопители на магнитной ленте и приводы компакт-дисков, по сравнению с клиентскими машинами. Серверы также ориентированы на то, чтобы обрабатывать многочисленные запросы на разделяемые ресурсы быстро и эффективно. Серверы обычно выделены для обслуживания сетевых запросов клиентов. В дополнение, физическая безопасность — доступ к самой машине — является ключевым компонентом сетевой безопасности. Поэтому важно, чтобы серверы располагались в специальном помещении с контролируемым доступом, отделенном от помещений с общим доступом.

Сети с выделенным сервером также предоставляют централизованную проверку учетных записей пользователей и паролей.

Например, Windows NT использует доменную концепцию для управления пользователями, группами и машинами и для контроля над доступом к сетевым ресурсам. Прежде чем пользователь сможет получить доступ к сетевым ресурсам, он должен сообщить свое регистрационное имя и пароль контроллеру домена — серверу, который проверяет имена учетных записей и пароли в базе данных с такой информацией. Контроллер домена позволит доступ к определенным ресурсам только в случае допустимой комбинации регистрационного имени и пароля. Изменять связанную с безопасностью информацию в базе данных контроллера домена может только сетевой администратор. Этот подход обеспечивает централизованную безопасность и позволяет управлять ресурсами с изменяющейся степенью контроля в зависимости от их важности и расположения.

В отличие от одноранговой модели, сеть с выделенным сервером обычно требует только один пароль для доступа к самой сети, что уменьшает количество паролей, которые пользователь должен помнить. Кроме того, сетевые ресурсы типа файлов и принтеров легче найти, потому что они расположены на определенном сервере, а не на чьей-то машине в сети. Концентрация сетевых ресурсов на небольшом количестве серверов также упрощает резервное копирование и поддержку данных.

Сети с выделенным сервером лучше масштабируются — в сравнении с одноранговыми сетями. С ростом размера одноранговые сети сильно замедляют свою работу и становятся неуправляемыми. Сети с выделенным сервером, наоборот, могут обслуживать от единичных пользователей до десятков тысяч пользователей и географически распределенных ресурсов. Другими словами, сеть с выделенным сервером может расти с ростом компании, не оказываясь сдерживающим фактором.

Подобно одноранговой модели, сеть с выделенным сервером также имеет недостатки. Первой в этом списке стоит необходимость дополнительных расходов на такие сети. Сеть с выделенным сервером требует наличия одного или более мощных — и, соответственно, более дорогих — компьютеров для запуска специального (и тоже дорогого) серверного программного обеспечения. Вдобавок серверное программное обеспечение требует квалифицированного персонала для его обслуживания. Подготовка персонала для овладения необходимыми для обслуживания сети с выделенным сервером навыками или наем на работу подготовленных сетевых администраторов также увеличивают стоимость такой сети.

Есть и другие негативные аспекты сетей с выделенным сервером. Централизация ресурсов и управления упрощает доступ, контроль и объединение ресурсов, но при этом приводит к появлению точки, которая может привести к неполадкам во всей сети. Если сервер вышел из строя, — не работает вся сеть. В сетях с



несколькими серверами потеря одного сервера означает потерю всех ресурсов, связанных с этим сервером. Также, если неисправный сервер является единственным источником информации о правах доступа определенной части пользователей, то эти пользователи не смогут получить доступ к сети.

Некоторые преимущества сетей с выделенным сервером:

- они обеспечивают централизованное управление учетными записями пользователей, безопасностью и доступом, что упрощает сетевое администрирование;

- более мощное оборудование означает и более эффективный доступ к сетевым ресурсам;

- пользователям для входа в сеть нужно помнить только один пароль, что позволяет им получить доступ ко всем ресурсам, к которым имеют права.

Недостатки сетей с выделенным сервером:

- неисправность сервера может сделать сеть неработоспособной; в лучшем случае это означает потерю сетевых ресурсов;

- такие сети требуют квалифицированного персонала для сопровождения сложного специализированного программного обеспечения, что увеличивает общую стоимость владения сетью;

- стоимость увеличивается из-за потребности в выделенном оборудовании и специализированном программном обеспечении.

**Выбор сети.** Выбор сети зависит от обстоятельств. Выбирать одноранговую сеть стоит только в том случае, если выполнены все следующие условия:

- имеется не более 10 сетевых пользователей (желательно не более пяти);

- все машины в сети находятся достаточно близко друг к другу, чтобы их можно было объединить в одну локальную сеть;

- финансовые причины требуют недорогого решения;

- не нужны специализированные серверы (например, факс-серверы, коммуникационные серверы или серверы приложений).

С другой стороны, сеть с выделенным сервером оправдана, если выполнено хотя бы одно из следующих условий:

- требуется одновременный доступ более 10 пользователей;

- требуется централизованное управление, безопасность, управление ресурсами или резервное копирование;

- требуется доступ к специализированным серверам или нагрузка на сетевые ресурсы слишком велика;

- используется объединенная сеть или требуется доступ к глобальной сети.

Когда в сети больше пяти, но меньше десяти пользователей, финансовые ограничения часто приводят организацию к одноранговому решению. Но если организация собирается расти или кажутся привлекательными специализированные сетевые серверы, лучше начать с сети с выделенным сервером.

**Типы кабелей.** Существует несколько различных типов кабелей, используемых в современных сетях. Линейные размеры, цена, скорость передачи данных, минимальная и максимальная длина, а также легкость установки у всех этих типов отличаются. Различные сетевые ситуации и требования могут потребовать различных типов кабелей.

**Кабель типа «витая пара».** Кабель типа «витая пара» (twisted pair) представляет собой сетевой носитель, используемый во многих сетевых топологиях, включая Ethernet, ARCNet и IBM Token Ring. Витая пара бывает двух типов: экранированная и неэкранированная. Будучи, вероятно, самым распространенным типом сетевого кабеля в США, витая пара первоначально была разработана для использования в телефонных линиях. Примером может послужить кабель CAT1, используемый для соединения телефона со стенной розеткой, — неэкранированная витая пара с двумя парами проводов, также известная в США как «silver satin».

**Неэкранированная витая пара.** Это один из типов кабеля «витая пара», обычно состоящий из двух пар изолированных медных проводов, завитых вокруг друг друга и покрытых пластиковой оболочкой. Закручивание проводов вокруг друг друга обеспечивает защиту от взаимных наводок и других типов внешнего воздействия.

Этот тип кабеля поддерживает скорость передачи данных от 10 до 100 Мбит/с и может подавать данные на расстояние до 100 м без повторителя. Во многих случаях также возможно проложить неэкранированную витую пару по уже существующим коробам для телефонных проводов, что еще более упрощает установку сети.

Несмотря на свою дешевизну и легкость в установке, кабели «неэкранированная витая пара» не лишены недостатков, а именно: чувствительности к помехам со стороны внешних электромагнитных источников и взаимное наложение сигналов между смежными проводами (crosstalk). В некоторой степени конструкция кабеля — закручивание проводов вокруг друг друга — в значительной степени устраняет естественное наложение сигнала и взаимовлияние, существующее между проводами. Хотя электромагнитная интерференция (electromagnetic interference) и взаимовлияние (crosstalk) существуют и в других типах носителя, неэкранированная витая пара особенно восприимчива к ним, поскольку в ней отсутствует экранирование, имеющееся в остальных типах кабелей. Это отсутствие экранирования также делает неэкранированную витую пару потенциально уязвимой для перехвата информации — момент, на который нужно обратить особое внимание, если безопасность данных имеет большое значение для будущей сети.

Неэкранированная витая пара также приводит к большему затуханию или ослаблению сигнала по пути, чем другие типы кабеля. Это означает, что кабельные сегменты, использующие неэкранированную витую пару, имеют серьезные ограничения по рас-

стоянию: длина сегмента не может превышать 100 м. Ограничение на длину кабеля существует потому, что сигнал ослабевает по пути и частично поглощается носителем, по которому перемещается. Это затухание делает сигнал нечитаемым после указанного расстояния, если не используется повторитель (устройство, которое очищает и заново передает сигнал).

**Экранированная витая пара.** Экранированная витая пара (Shielded Twisted Pair, STP) традиционно используется в нескольких типах сетей, включая AppleTalk и Token Ring. Экранированная витая пара, имеющая сходную внутреннюю конструкцию, подчиняется тому же 100-метровому ограничению, что и неэкранированная витая пара. Экранированная витая пара обычно содержит в середине четыре или более пары скрученных медных проводов. Экранированная витая пара отличается от неэкранированной витой пары тем, что она содержит защиту — электрически заземленную плетеную медную сетку или алюминиевую фольгу, которая окружает внутренние провода кабеля, отделяя их от внешней оболочки кабеля и обеспечивая противодействие внешней электромагнитной интерференции (Electromagnetic Interference, EMI). Некоторые типы экранированной витой пары также используют внутреннюю защиту вокруг каждой витой пары, чтобы держать каждую пару отдельно от других, что еще больше уменьшает взаимное влияние пар друг на друга. Кроме того, экранированную витую пару можно считать более защищенной, чем неэкранированную витую пару, поскольку ее защита делает такой кабель менее уязвимым к перехвату информации.

Скорость передачи данных и ограничения по расстоянию для экранированной витой пары идентичны тем же параметрам неэкранированной витой пары. Хотя экранированная витая пара обеспечивает лучшую защиту от электромагнитной интерференции, чем неэкранированная витая пара, она редко используется в новых сетевых проектах, поскольку сложнее в установке и поддержке. Одна из причин этого в том, что защита делает кабель менее гибким, другая — в том, что экранированная витая пара требует электрического заземления.

**Коаксиальный кабель.** Коаксиальный кабель был первым типом кабеля, использованным для соединения компьютеров в сеть, и он помог сформировать основу оригинального стандарта Ethernet. Этот тип кабеля состоит из центрального медного проводника, более толстого, чем провода в кабеле «витая пара». Центральный проводник покрыт слоем пеннистого пластикового изолирующего материала, который, в свою очередь, окружен вторым проводником, обычно плетеной медной сеткой или алюминиевой фольгой. Этот внешний проводник не используется для передачи данных и лишь обеспечивает электрическое заземление и защищает центральный проводник от внутренней и внешней интерференции.

Несмотря на то, что коаксиальный кабель не так широко используется во вновь устанавливаемых сетях, как неэкранированная витая пара, он по-прежнему остается основным в уже установленных сетях. Он также является тем типом кабеля, который используется для кабельного телевидения. Коаксиальный кабель может передавать данные со скоростью до 10 Мбит/с на максимальное расстояние от 185 до 500 м. Производители коаксиального кабеля создали спецификации, которые делят коаксиальный кабель на категории в зависимости от таких характеристик, как волновое сопротивление (impedance) и толщина кабеля. Схемы на основе коаксиального кабеля обычно требуют заглушек (terminator), сопротивление которых должно точно соответствовать типу кабеля.

**Опволоконный кабель.** Находясь на переднем крае сетевой кабельной технологии, опволоконные кабели (fiber optic cable) обеспечивают превосходную скорость передачи данных на большие расстояния. Опволоконный кабель состоит из центрального стеклянного или пластикового проводника, окруженного другим слоем стеклянного или пластикового покрытия, и внешней защитной оболочки. Данные передаются по кабелю с помощью лазерного (laser transmitter) или светодиодного (LED, light-emitting diode transmitter) передатчика, который посылает однонаправленные световые импульсы через центральное стеклянное волокно. Стеклянное покрытие помогает поддерживать фокусировку света во внутреннем проводнике. Сигнал принимается на другом конце фотодиодным приемником (photodiode receiver), преобразующим световые импульсы в электрический сигнал, который сможет использовать получающий его компьютер.

Скорость передачи данных для опволоконных сетей находится в диапазоне от 100 Мбит/с до 2 Гбит/с, а данные могут быть надежно переданы на расстояние до 2 км без повторителя. Опволоконный кабель может поддерживать передачу видео- и голосовой информации так же, как и передачу данных. Поскольку световые импульсы полностью закрыты в пределах внешней оболочки, опволоконный носитель фактически невосприимчив к внешней интерференции и подслушиванию. Эти качества делают опволоконный кабель привлекательным выбором для защищенных сетей или сетей, которые требуют очень быстрой передачи на большие расстояния.

Поскольку световые импульсы могут двигаться только в одном направлении, системы на базе опволоконных кабелей должны иметь входящий кабель и исходящий кабель для каждого сегмента, который будет посылать и получать данные. Волоконный кабель сложен в установке, это делает его самым дорогим типом сетевого носителя. Волоконный носитель требует специальных коннекторов и высококвалифицированной установки. Одним из



способов снижения расходов является ограничение использования волоконного кабеля сетевыми магистралями или теми областями, где имеют значение влияние электромагнитного наложения, возгораемость и пр.

**Беспроводные технологии.** В дополнение к традиционным физическим носителям, методы беспроводной передачи данных могут являться удобной, а иногда и неизбежной альтернативой кабельным соединениям. Беспроводные технологии различаются по типам сигнала, частоте (большая частота означает большую скорость передачи) и расстоянию передачи. Помехи и стоимость имеют большое значение. По причине роста количества глобальных сетей и потребности в мобильных компьютерных решениях беспроводные технологии включают в себя наиболее быстрорастущий сегмент сетевых технологий. Тремя главными типами беспроводной передачи данных являются радиосвязь, связь в микроволновом диапазоне и инфракрасная связь.

**Радиосвязь.** Технологии радиосвязи (Radio Waves) пересылают данные на радиочастотах и практически не имеют ограничений по дальности. Она используется для соединения локальных сетей на больших географических расстояниях. Радиопередача в целом имеет высокую стоимость, подлежит государственному регулированию и крайне чувствительна к электронному и атмосферному наложению. Она также подвержена перехвату, поэтому требует шифрования или другой модификации при передаче, чтобы обеспечить разумный уровень безопасности.

**Связь в микроволновом диапазоне.** Передача данных в микроволновом диапазоне (Microwaves) использует высокие частоты и применяется как на коротких расстояниях, так и в глобальных коммуникациях. Их главное ограничение заключается в том, что передатчик и приемник должны быть в зоне прямой видимости друг друга. Передача данных в микроволновом диапазоне обычно используется для соединения локальных сетей в отдельных зданиях, где использование физического носителя затруднено или непрактично. Хорошим примером могут послужить два расположенных рядом небоскреба, где использование кабеля было бы невозможно. Связь в микроволновом диапазоне также широко используется в глобальной передаче с помощью спутников и наземных спутниковых антенн, обеспечивающих выполнение требования прямой видимости. Передача данных в микроволновом диапазоне, особенно в случае использования спутников, может быть чрезвычайно дорогой.

**Инфракрасная связь.** Инфракрасные технологии (infrared transmissions), функционирующие на очень высоких частотах, приближающихся к частотам видимого света, могут быть использованы для установления двусторонней или широковещательной передачи на близких расстояниях. Они обычно используют светодиоды

(LED, light-emitting diode) для передачи инфракрасных волн приемнику. Поскольку они могут быть физически заблокированы и испытывать интерференцию с ярким светом, инфракрасная передача ограничена малыми расстояниями в зоне прямой видимости. Инфракрасная передача обычно используется в складских или офисных зданиях, иногда для связи двух зданий. Другим популярным использованием инфракрасной связи является беспроводная передача данных в портативных компьютерах. Инфракрасные технологии могут быть как недорогими, так и очень дорогими.

**Сетевые компоненты.** Существует множество сетевых устройств, которые можно использовать для создания, сегментирования и усовершенствования сети. В этом разделе мы рассмотрим несколько сетевых устройств, таких как сетевые адаптеры, повторители, усилители, мосты, маршрутизаторы и шлюзы.

**Сетевые карты.** Сетевой адаптер, или NIC (Network Interface Card), является тем устройством, которое физически соединяет компьютер с сетью. Прежде чем выполнить такое соединение, надо правильно установить и настроить сетевой адаптер. Простота или сложность этой установки и настройки зависит от типа сетевого адаптера, который собираются использовать. Для некоторых конфигураций достаточно просто вставить адаптер в подходящий слот. Автоматически конфигурирующиеся адаптеры, а также адаптеры, отвечающие стандарту Plug and Play (вставь и работай), автоматически производят свою настройку. Если сетевой адаптер не отвечает стандарту Plug and Play, нужно настроить IRQ (Interrupt Request Line — линия запроса на прерывание) и адрес ввода-вывода (Input/Output address — адрес ввода/вывода). IRQ представляет собой логическую коммуникационную линию, которую устройство использует для связи с процессором. Адрес ввода-вывода — это трехзначное шестнадцатеричное число, которое идентифицирует коммуникационный канал между аппаратными устройствами и центральным процессором. Чтобы сетевой адаптер функционировал правильно, должны быть правильно настроены как IRQ, так и адрес ввода-вывода.

**Повторители и усилители.** Как уже упоминалось, сигнал при перемещении по сети ослабевает. Чтобы противодействовать этому ослаблению, используют повторители и (или) усилители, которые усиливают сигналы, проходящие через них по сети.

Повторители (repeater) используются в сетях с цифровым сигналом для борьбы с ослаблением сигнала. Повторители обеспечивают надежную передачу данных на большие расстояния, нежели обычно позволяет тип носителя. Когда повторитель получает ослабленный входящий сигнал, он очищает его, увеличивает мощность и посылает этот сигнал следующему сегменту.

Усилители (amplifier), хоть и имеют сходное назначение, применяются для увеличения дальности передачи в сетях, использу-



ющих аналоговый сигнал. Аналоговые сигналы могут переносить как голос, так и данные одновременно — носитель делится на несколько каналов, так что разные частоты могут передаваться параллельно.

**Концентраторы.** Концентратор (hub) представляет собой сетевое устройство, служащее в качестве центральной точки соединения и связующей линии в сетевой конфигурации звезда (star), описанной в следующем подразделе. Существуют три основных типа концентраторов: пассивные (passive), активные (active) и интеллектуальные (intelligent). Пассивные концентраторы, не требующие электроэнергии, действуют просто как физическая точка соединения, ничего не добавляя к проходящему сигналу. Напротив, активные концентраторы требуют энергии, которую они используют для восстановления и усиления сигнала, проходящего через них. Интеллектуальные концентраторы могут предоставлять такие сервисы, как переключение пакетов (packet switching) и перенаправление трафика (traffic routing).

**Мосты.** Мост (bridge) представляет собой другое устройство, используемое для соединения сетевых сегментов. Мосты можно рассматривать как усовершенствование повторителей, так как они уменьшают загрузку сети: мосты считывают адрес сетевой карты компьютера-получателя из каждого входящего пакета данных и просматривают специальные таблицы, чтобы определить, что делать с пакетом.

Поскольку мост функционирует в первую очередь как повторитель, он может получать данные из любого сегмента, однако он более разборчив в передаче этих сигналов, чем повторитель. Если получатель пакета находится в том же физическом сегменте, что и мост, то мост знает, что этот пакет достиг цели и, таким образом, больше не нужен. Однако если получатель пакета находится в другом физическом сегменте, мост знает, что его надо переслать. Эта обработка помогает уменьшить загрузку сети. Например, сегмент не получает сообщений, не относящихся к нему.

Мосты могут соединять сегменты, которые используют разные типы носителей (кабелей). Они могут соединять сети с разными схемами доступа к носителю, например сеть Ethernet и сеть Token Ring. Примером таких устройств являются мосты-трансляторы (translating bridge), которые осуществляют преобразование между различными методами доступа к носителю, позволяя связывать сети разных типов. Другой специальный тип — моста прозрачный (transparent bridge), или интеллектуальный мост (learning bridge), периодически «изучает», куда направлять получаемые им пакеты. Он делает это посредством непрерывного построения специальных таблиц, добавляя в них по мере необходимости новые элементы.

Возможным недостатком мостов является то, что они передают данные дольше, чем повторители, так как проверяют адрес

сетевой карты получателя для каждого пакета. Они также сложнее в управлении и дороже, нежели повторители.

**Маршрутизаторы.** Маршрутизатор (router) представляет собой сетевое коммуникационное устройство, которое может связывать два и более сетевых сегментов (или подсетей). Маршрутизатор функционирует подобно мосту, но для фильтрации трафика он использует не адрес сетевой карты компьютера, а информацию о сетевом адресе, передаваемую в относящейся к сетевому уровню части пакета. После получения этой информации об адресе маршрутизатор использует таблицу маршрутизации (routing table), содержащую сетевые адреса, чтобы определить, куда направить пакет. Он делает это посредством сравнения сетевого адреса в пакете с элементами в таблице маршрутизации — если совпадение найдено, пакет направляется по указанному маршруту. Если же совпадение не найдено, обычно пакет отбрасывается.

Существуют два типа маршрутизирующих устройств: статические и динамические. Статические маршрутизаторы (static router) используют таблицы маршрутизации, которые должен создать и вручную обновлять сетевой администратор. С другой стороны, динамические маршрутизаторы (dynamic router) создают и обновляют свои собственные таблицы маршрутизации. Они используют информацию как найденную на своих собственных сегментах, так и полученную от других динамических маршрутизаторов. Динамические маршрутизаторы всегда содержат свежую информацию о возможных маршрутах по сети, а также информацию об узких местах и задержках в прохождении пакетов. Эта информация позволяет им определить наиболее эффективный путь, доступный в данный момент, для перенаправления пакетов данных к их получателям.

Поскольку маршрутизаторы могут осуществлять интеллектуальный выбор пути и отфильтровывать пакеты, которые им не нужно получать, они помогают уменьшить загрузку сети, сохранить ресурсы и увеличить пропускную способность. Кроме того, они повышают надежность доставки данных, поскольку маршрутизаторы могут выбрать для пакетов альтернативный путь, если маршрут по умолчанию недоступен.

Термин «маршрутизатор» (router) может обозначать элемент электронной аппаратуры, сконструированной специально для маршрутизации. Он также может означать компьютер (обеспеченный таблицей маршрутизации), подключенный к другим сегментам сети с помощью нескольких сетевых карт и, следовательно, способный выполнять функции маршрутизации между связанными сегментами.

Маршрутизаторы превосходят мосты своей способностью фильтровать и направлять пакеты данных по сети. И в отличие от мостов для них можно отключить пересылку ширококвещательных сообщений, что уменьшает сетевой ширококвещательный трафик.

Другое важное преимущество маршрутизатора как соединительного устройства заключается в том, что, поскольку он работает на сетевом уровне, он может соединять сети, использующие различную сетевую архитектуру, методы доступа к устройствам или протоколы. Например, маршрутизатор может соединять подсеть Ethernet и сегмент Token Ring. Он может связывать несколько небольших сетей, использующих различные протоколы, если используемые протоколы поддерживают маршрутизацию.

Маршрутизаторы по сравнению с повторителями дороже и сложнее в управлении. У них меньшая пропускная способность, чем у мостов, поскольку они должны производить дополнительную обработку пакетов данных. Кроме того, динамические маршрутизаторы могут добавлять излишний трафик в сети, поскольку для обновления таблиц маршрутизации постоянно обмениваются сообщениями.

**Мосты-маршрутизаторы.** Английский термин «brouter» (мост-маршрутизатор) представляет собой комбинацию слов «bridge» (мост) и «router» (маршрутизатор). Из этого можно сделать вывод, что мост-маршрутизатор сочетает функции моста и маршрутизатора. Когда мост-маршрутизатор получает пакет данных, он проверяет, послан пакет с использованием маршрутизируемого протокола или нет. Если это пакет маршрутизируемого протокола, мост-маршрутизатор выполняет функции маршрутизатора, посылая при необходимости пакет получателю вне локального сегмента.

Если же пакет содержит немаршрутизируемый протокол, мост-маршрутизатор выполняет функции моста, используя адрес сетевой карты для поиска получателя на локальном сегменте. Для выполнения этих двух функций мост-маршрутизатор может поддерживать как таблицы маршрутизации, так и таблицы мостов.

**Шлюзы.** Шлюз (gateway) представляет собой метод осуществления связи между двумя или несколькими сетевыми сегментами. В качестве шлюза обычно выступает выделенный компьютер, на котором запущено программное обеспечение шлюза и производятся преобразования, позволяющие взаимодействовать несходным системам в сети. Например, при использовании шлюза персональные компьютеры на базе Intel-совместимых процессоров на одном сегменте могут связываться и разделять ресурсы с компьютерами Macintosh.

Другой функцией шлюзов является преобразование протоколов. Шлюз может получить сообщение IPX/SPX, направленное клиенту, использующему другой протокол, например TCP/IP, на удаленном сетевом сегменте. После того как шлюз определяет, что получателем сообщения является станция TCP/IP, он действительно преобразует данные-сообщения в протокол TCP/IP. В этом отличие от моста, который просто пересылает сообщение, используя один протокол внутри формата данных другого протоко-

ла, — преобразование при необходимости происходит у получателя. Почтовые шлюзы производят сходные операции по преобразованию почтовых сообщений и других почтовых передач из родного формата вашего приложения электронной почты в более универсальный почтовый протокол, например SMTP, который может быть затем использован для направления сообщения в Интернет.

Хотя шлюзы имеют много преимуществ, нужно учитывать несколько факторов, когда принимается решение об их использовании в сети. Шлюзы сложны в установке и настройке. Они также дороже других коммуникационных устройств. Еще один момент: благодаря лишнему этапу обработки, связанному с процессом преобразования, шлюзы работают медленнее, чем маршрутизаторы и подобные устройства.

**Типы сетевой топологии.** Под топологией сети понимается описание ее физического расположения. То, как компьютеры соединены в сети друг с другом и с помощью каких устройств входят в физическую топологию. Существуют четыре основных топологии: Bus (шина), Ring (кольцо), Star (звезда) и Mesh (ячеистая топология). Другие топологии обычно являются комбинацией двух и более главных типов. Выбор типа физической топологии для сети является одним из первых шагов планирования сети. Выбор топологии основывается на множестве факторов, в число которых входят цена, расстояния, вопросы безопасности, предполагаемая сетевая операционная система, а также будет ли новая сеть использовать существующее оборудование, проводку и т. п.

**Шина.** Физическая топология шина (Bus), именуемая также линейной шиной (Linear Bus), состоит из единственного кабеля, к которому присоединены все компьютеры сегмента (рис. 5.66). Сообщения посылаются по линии всем подключенным станциям вне зависимости от того, кто является получателем. Каждый компьютер проверяет каждый пакет в проводе, чтобы определить получателя пакета. Если пакет предназначен для другой станции, компьютер отвергнет его. Соответственно, компьютер получит и обработает любой пакет на шине, адресованный ему.

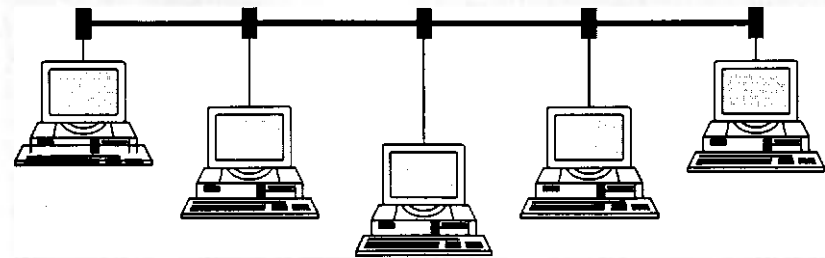


Рис. 5.66. Топология шина



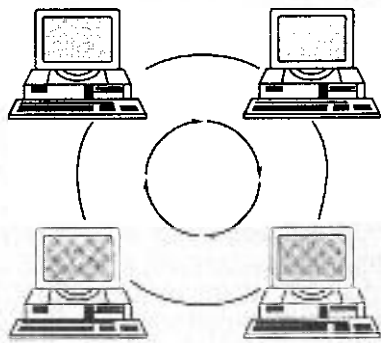


Рис. 5.67. Топология кольцо

Главный кабель шины, известный как магистраль (backbone), имеет на обоих концах заглушки (terminator) для предотвращения отражения сигнала. Без правильно установленных заглушек работа шины будет ненадежной или вообще невозможной.

Шинная топология представляет собой быстрейший и простейший способ установки сети. Она требует меньше оборудования и кабелей, чем другие топологии, и ее легче настраивать.

Это хороший способ быстрого построения временной сети, обычно лучший выбор для маленьких сетей (не более 10 компьютеров).

Имеется пара недостатков, о которых надо знать при решении вопроса об использовании шинной топологии для сети. Неполадки станции или другого компонента сети трудно изолировать. Кроме того, неполадки в магистральном кабеле могут привести к выходу из строя всей сети.

**Кольцо.** Топология кольцо (Ring) обычно используется в сетях Token Ring и FDDI (волоконно-оптических). В физической топологии Ring линия передачи данных фактически образует логическое кольцо, к которому подключены все компьютеры сети (рис. 5.67). В отличие от шинной топологии, которая использует конкурентную схему, чтобы позволить станциям получать доступ к сетевому носителю, доступ к носителю в кольце осуществляется посредством логических знаков — «маркеров» (token), которые пускаются по кругу от станции к станции, давая им возможность переслать пакет, если это нужно. Это дает каждому компьютеру в сети равную возможность получить доступ к носителю и, следовательно, переслать по нему данные. Компьютер может посылать данные только тогда, когда владеет маркером.

Так как каждый компьютер при этой топологии является частью кольца, он имеет возможность пересылать любые полученные им пакеты данных, адресованные другой станции. Получающаяся регенерация делает сигнал сильным и позволяет избежать необходимости в применении повторителей. Так как кольцо формирует бесконечный цикл, заглушки не требуются. Кольцевая топология относительно легка для установки и настройки, требуя минимального аппаратного обеспечения.

Топология физического кольца имеет несколько недостатков. Как и в случае линейной шины, неполадки на одной станции могут привести к отказу всей сети. Поддерживать логическое кольцо трудно, особенно в больших сетях. Кроме того, в случае необхо-

димости настройки и переконфигурации любой части сети придется временно отключить всю сеть.

Кольцевая топология даст всем компьютерам равные возможности доступа к сетевому носителю.

**Звезда.** В топологии звезда (Star) все компьютеры в сети соединены друг с другом с помощью центрального концентратора (рис. 5.68). Все данные, которые посылает станция, направляются прямо на концентратор, который затем пересылает пакет в направлении получателя. Как и при шинной топологии, компьютер в сети типа Star может попытаться послать данные в любой момент. Однако на деле только один компьютер может в конкретный момент времени производить посылку. Если две станции посылают сигналы на концентратор точно в одно время, обе посылки окажутся неудачными и каждому компьютеру придется подождать случайный период времени, прежде чем снова попытаться получить доступ к носителю. Сети с топологией Star обычно лучше масштабируются, чем другие типы.

Главное преимущество внедрения топологии звезда заключается в том, что в отличие от линейной шины неполадки на одной станции не выведут из строя всю сеть. В сетях с этой топологией проще находить обрывы кабеля и прочие неисправности. Это облегчает обнаружение обрыва кабеля и других неполадок. Кроме того, наличие центрального концентратора в топологии Star облегчает добавление нового компьютера и реконфигурацию сети. Топология Star проще для перенастройки, чем все остальные топологии.

Топологии Star присущи несколько недостатков. Во-первых, этот тип конфигурации требует больше кабеля, чем в некоторых других сетях, вследствие наличия отдельных линий, соединяющих каждый компьютер с концентратором. Кроме того, центральный

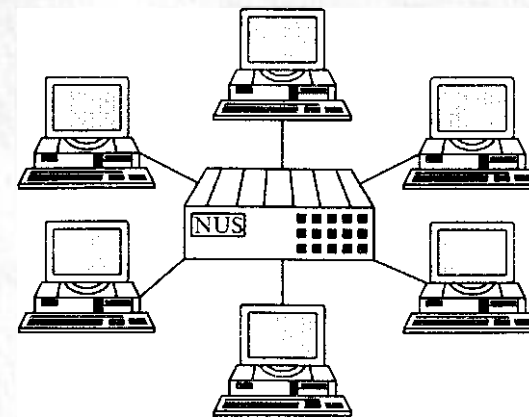


Рис. 5.68. Топология звезда



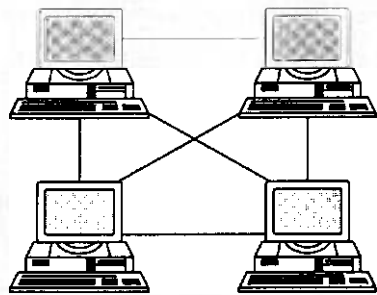


Рис. 5.69. Ячеистая топология

концентратор выполняет большинство функций сети, так что выход из строя одного этого устройства отключит всю сеть.

**Ячеистая топология.** Ячеистая топология (Mesh) соединяет все компьютеры попарно (рис. 5.69). Сети Mesh используют значительно большее количество кабеля, чем любая другая топология, что делает их дороже. Кроме того, такие сети значительно

сложнее устанавливать. Тогда почему некоторые используют топологию Mesh? Ответ заключается в устойчивости к сбоям (fault tolerance). Устойчивость к сбоям заключается в способности работать при наличии повреждений. В сети с поврежденным сегментом это означает обход сегмента.

**Смешанные сети.** Многие организации используют комбинации главных сетевых топологий. Обсудим три такие смешанные (hybrid) сети: Star Bus (звезда на шине), Star Ring (звезда на кольце), Hybrid Mesh (гибридная ячеистая структура).

**Звезда на шине.** Смешанная топология звезда на шине (Star Bus), показанная на рис. 5.70, объединяет топологии Bus и Star. Преимущество этой топологии заключается в том, что никакие неполадки на отдельном компьютере или в сегменте не могут вывести из строя всю сеть. Также в случае неисправности отдельного концентратора не смогут взаимодействовать по сети только

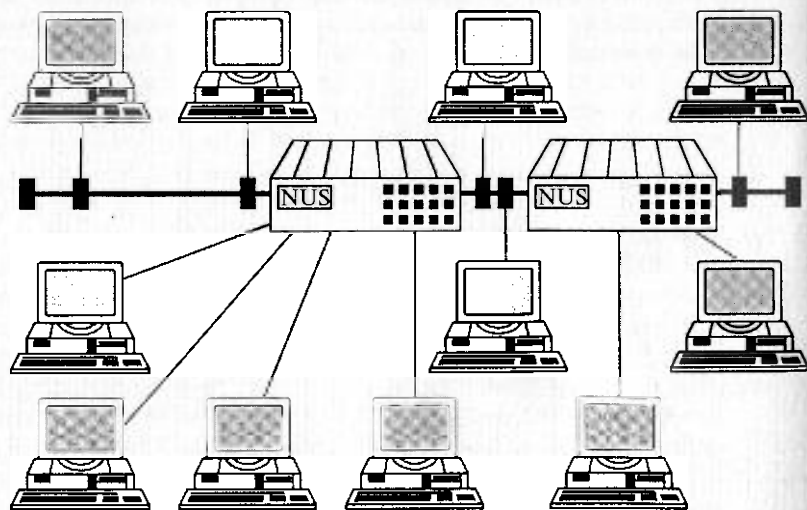


Рис. 5.70. Топология звезда на шине

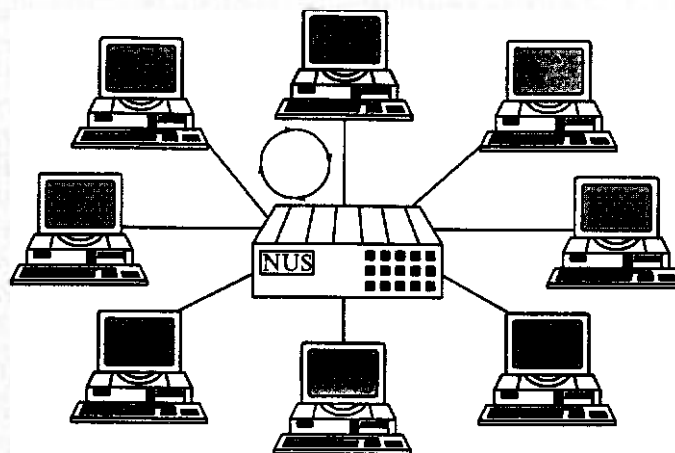


Рис. 5.71. Топология звезда на кольце

те компьютеры, которые присоединены к этому концентратору, а остальные компьютеры проблема не затронет.

**Звезда на кольце.** Топология звезда на кольце (Star Ring) известна также под названием Star-wired Ring. Как видно из рис. 5.71, сеть Star Ring внешне идентична топологии Star, но на самом деле концентратор соединен проводами как логическое кольцо.

Эта топология популярна для сетей Token Ring, поскольку легче в реализации, чем физическое кольцо. Как при топологии Ring, компьютеры имеют равный доступ к сетевому носителю за счет посылки «токенов». Повреждение отдельного компьютера не может привести к остановке всей сети, но, если выходит из строя концентратор, кольцо, которым управляет концентратор, тоже отключается.

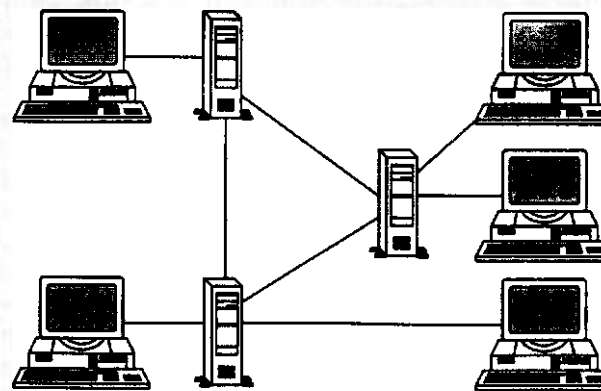


Рис. 5.72. Топология гибридная ячеистая

**Гибридная ячеистая.** Реализация настоящей гибридной ячеистой структуры (Hybrid Mesh) в крупных сетях может быть дорогой, требующей времени и непростой. Сеть топологии Hybrid Mesh может предоставить некоторые из существенных преимуществ настоящей сети Mesh без необходимости использовать столь много кабеля. В большинстве крупных организаций критически важные данные хранятся не на всех компьютерах сети. Вместо этого они хранятся на сетевых серверах. Компании, которые хотят обеспечить защиту от сбоев для своих сетей на уровне кабелей, могут ограничить их Mesh только компьютерами с критически важными данными. Это означает, что Mesh существует только на части сети (рис. 5.72).

Этот тип Mesh по-прежнему обеспечивает защиту от сбоев для серверов с важной информацией, но не добавляет защиты для отдельных клиентов сети. Hybrid Mesh должна стоить меньше, чем сеть, полностью построенная на топологии Mesh, но будет не столь защищенной от сбоев.

#### Контрольные вопросы

1. Перечислите основные внешние устройства ЭВС.
2. Для чего предназначены ПУ?
3. На какие три группы подразделяются ПУ по назначению?
4. На какие три группы подразделяются ПУ по быстродействию?
5. Охарактеризуйте принцип хранения информации на магнитных носителях.
6. Каково устройство НГМД?
7. Каково устройство НЖМД?
8. Охарактеризуйте принцип действия оптических ВЗУ.
9. Охарактеризуйте принцип действия магнитооптических ВЗУ.
10. Объясните принцип действия устройства ввода данных клавиатуры.
11. Объясните устройство и принцип действия дигитайзера.
12. Объясните устройство и принцип действия сканера.
13. Объясните принцип действия УВВ речевой информации.
14. Объясните устройство и принцип действия дисплеев на основе ЭЛТ.
15. Объясните устройство и принцип действия плоских дисплеев.
16. Перечислите и охарактеризуйте основные типы принтеров.
17. Объясните устройство и принцип действия графопостроителей.
18. Перечислите и охарактеризуйте основные топологии сетей ЭВМ.
19. Перечислите и охарактеризуйте основные интерфейсы, применяемые в ЭВМ.
20. Каково назначение системы прерываний программ и в каких ВС она применяется?
21. Охарактеризуйте режим прямого доступа к памяти ПДП.
22. Поясните понятия приоритета программы, глубины прерываний.
23. Поясните понятия «мультиплексный канал», подканалы.
24. В чем отличие селекторного канала от мультиплексного?

## Глава 6

### ТЕНДЕНЦИИ РАЗВИТИЯ АРХИТЕКТУРЫ И АППАРАТНОГО ОБЕСПЕЧЕНИЯ ЭВМ

#### 6.1. Требования различных задач к вычислительным ресурсам и ограничения фон неймановской архитектуры

В последнее время быстро развивающиеся технологические возможности привели к созданию мощных однокристалльных микропроцессоров, таких как Альфа, Power PC, Pentium, Intel 860 и др. Многие сложные вычислительные задачи удается решать на современных PC и рабочих станциях, объединенных в единую локальную сеть. Однако требования, предъявляемые к ЭВМ, при необходимости решения проблемных задач первостепенной важности со временем повышаются, а возможности ЭВМ традиционной фон неймановской архитектуры ограничены теоретическим и техническими пределами скорости выполнения операций. Принципиальные причины, не позволяющие построить ЭВМ требуемой производительности, заключаются в следующем. Ограничения накладываются возможностями отвода тепла от микросхем. Жидкостное охлаждение позволяет отвести 20 Вт с одного квадратного сантиметра. Следовательно, чем меньше выделяет энергии схема на одно логическое срабатывание  $A_3$ , тем меньше выделяется мощности при том же быстродействии

$$P_3 = A_3/t_3,$$

где  $t_3$  — время срабатывания элемента.

Следовательно, в одном и том же объеме можно сосредоточить большее количество логических элементов и элементов памяти. Если объем растет, растут потери во времени на связи между элементами  $t_{св}$ , и времена  $t_3$  становятся соизмеримы, а иногда много меньше, чем  $t_{св}$ . Поэтому, чем выше производительность элемента, тем меньший объем должна иметь вычислительная система.

Немалую роль при создании ЭВМ играет показатель надежности элемента  $\lambda$ . В зарубежных разработках этот параметр не ограничивает производительности, так как даже для схем большой интеграции величина  $\lambda$  не превышает  $10^{-8}$ . Отечественная элементная база, в особенности на первом этапе производства, имеет  $\lambda > 10^{-6}$ ,

это заставляет вводить резервирование и аппаратный контроль во все устройства машины, что увеличивает объем устройств и потребляемую мощность. Предельная производительность одного процессора, работающего по фон неймановскому принципу, вряд ли превысит  $10^8$  опер/с на скалярных операциях. В конечном счете требования к развитию вычислительных средств сводятся к увеличению производительности при уменьшении потребляемой мощности и повышению ожидаемой надежности функционирования.

Традиционная модель вычислителя сформулирована в работах Тьюринга, Клини, Колмогорова, Глушкова и наиболее полно представлена в работах фон Неймана в 1969 г. Фон неймановская архитектура имеет четыре характеристики:

- 1) наличие единого вычислительного устройства, включающего процессор, средства передачи информации и память;
- 2) линейная структура адресации памяти, состоящей из слов фиксированной длины;
- 3) низкий уровень машинного языка, команды которого осуществляют операции над элементарными операндами;
- 4) централизованное последовательное управление.

Один из способов повышения скорости передачи и обработки информации — увеличение числа битов, передаваемых по шине параллельно, поэтому разрядность МП увеличилась с 8 до 16, затем до 32 бит, а в больших ЭВМ — до 64 разрядов.

## 6.2. Распараллеливание процессов обработки информации

Первый путь повышения производительности вычислительных средств увеличением быстродействия элементной базы практически исчерпал все возможности, натолкнувшись на теоретический предел  $10^9 \dots 10^{16}$  опер/с. Дальнейшее продвижение в этом направлении требует открытия новых физических принципов обработки информации.

Второй путь, в основе которого лежит распараллеливание процессов обработки информации на всех уровнях решения задачи, широко использовался на всех этапах развития вычислительных средств (рис. 6.1):

- переход от последовательного счета к параллельному;
- параллельная работа основных устройств ЭВМ;



Рис. 6.1. Многопроцессорная архитектура с общей шиной

- введение пакетного режима работы и режима разделения времени;
  - мультиплексный режим обработки данных;
  - конвейерный режим;
  - многопроцессорные и многомашинные комплексы;
  - режим управления потоками данных и др.
- Второй путь не исчерпал себя и с успехом будет развиваться в перспективных ЭВМ, особенно в суперЭВМ.

## 6.3. Принцип совмещения операций. Конвейерная обработка информации

Элементной базой ЭВМ многопроцессорной конфигурации с успехом могут служить наборы многокристальных секционных микропроцессоров. Автономность работы и большое быстродействие БИС управляющего процессора позволяют выбирать команды из памяти с большей скоростью, чем скорость их исполнения БИС операционного процессора. При этом в управляющем процессоре образуется очередь еще не исполненных команд. Напомним, что очередь — последовательный список, в котором все дополнения вносятся в один конец, а доступ для выборки осуществляется к другому концу списка (FIFO). Одновременно либо заранее подготавливаются те данные, которые потребуются операционному процессору в следующих циклах работы. Это так называемая опережающая выборка команд, которая экономит время процессора на ожидание операндов, необходимых для выполнения команд программы. В этом случае проводится совмещение операций выборки команд и подготовки операндов.

В многокристальном секционном микропроцессоре выбираемые из памяти команды распознаются и выполняются каждой частью МП автономно, поэтому может быть обеспечен режим одновременной работы всех БИС МП, т. е. конвейерный поточный режим исполнения последовательности команд программы с небольшим временным сдвигом.

При конвейерной обработке последовательность выполнения команды разбивается на этапы, которые могут выполняться одновременно.

Для выполнения операций каждого типа служат специализированные блоки. Во время выполнения предыдущей команды производится опережающая выборка из памяти очередной команды.

### Пример 6.1

Разбить на этапы команду «Умножение двух десятичных чисел  $3,8 \cdot 10^2$  и  $9,6 \cdot 10^3$ ».

Процессор выполняет три операции:

- а) перемножает мантиссы ( $3,8 \cdot 9,6 = 36,48$ );



б) складывает порядки ( $2 + 3 = 5$ );  
 в) нормализует результат, т. е. размещает в необходимом месте запятую (точку) ( $3,648 \cdot 10^6$ ).

Эти три операции (этапа) реализуются тремя отдельными исполнительными блоками; во время выполнения операции в) блоки а) и б) не заняты никакими действиями. При применении конвейера эти исполнительные блоки могли бы обрабатывать очередную пару чисел одновременно с выполнением операции в) над данной парой.

Рассмотрим теперь функционирование конвейера МП Z8000 фирмы Zilog, 16-разрядного, степень интеграции — 20000 транзисторов на кристалле. Цикл выполнения команды состоит из 6-ти этапов:

- а) выборка команды;
- б) декодирование команды;
- в) вычисление адреса операнда;
- г) выборка операнда;
- д) собственно выполнение команды;
- е) запоминание результата.

Когда в конвейере заканчивается выполнение определенного этапа предыдущей команды, высвобождается соответствующий исполнительный блок, и может быть начато выполнение аналогичного этапа следующей команды. В идеале очередная команда поступает на конвейер в тот момент, когда предыдущая команда с него сходит, что определяет шестикратное повышение производительности.

В случае, если команды не выполняются одна за другой или не в каждой команде используются все исполнительные блоки конвейера, а также если этапы имеют различную длительность, оптимальное повышение производительности не достигается. На рис. 6.2 представлено во времени конвейерное выполнение последо-

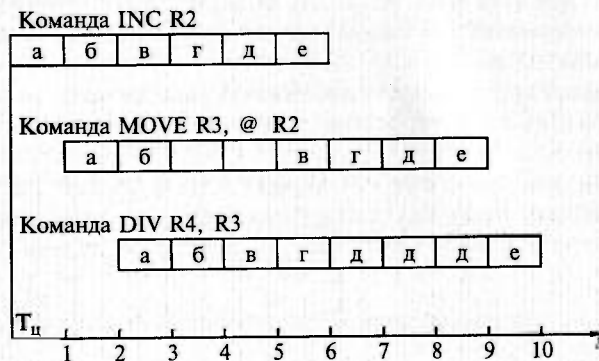


Рис. 6.2. Конвейерное выполнение последовательности команд

вательности трех команд. Видны потери производительности вследствие того, что отдельные команды используют не все блоки конвейера.

#### Пример 6.2

Выполняется следующая последовательность команд:

Мнемокод	Комментарий	Необходимые этапы и блоки
INC R2	Инкремент регистра 2	а), б), д)
MOVE R3, @ R2	Пересылка операнда из R3 по адресу, находящемуся в R2	а), б), в), г), д)
DIV R4, R3	Деление содержимого R4 на содержимое R3	а), б), д), е)

При выполнении команды пересылки MOVE используется содержимое регистра R2, которое формируется лишь по окончании его обработки командой инкремента, поэтому два цикла — четвертый и пятый затрачены на ожидание. Конвейер останавливается на время двух этапов арифметической обработки, происходящей на этапе команды деления.

Для грубой оценки повышения производительности при использовании конвейеризации принимается, что самым медленным участком конвейера является блок (д), в котором происходит выполнение операции, предписываемой командой. Увеличение производительности равно сумме средних значений времени работы всех блоков конвейера, деленной на среднюю величину времени работы самого медленного участка конвейера для заданной комбинации команд.

При выполнении команд переходов или обращений к подпрограммам нарушается последовательность событий.

Тогда необходимо конвейер частично или полностью опорожнить от команд, выполнение которых началось, и повторно его заполнить.

При выполнении «коротких» команд некоторые блоки конвейера не используются.

Кроме того, одни из блоков работают медленнее других, снижая производительность конвейера.

Доля команд, на которых нарушается естественная последовательность выполнения программ, обычно составляет 15... 30 % от общего количества.

Недостатком вычислительных средств с конвейерной обработкой является сложность управляющего устройства — управляющего процессора. Он должен вести учет операций, выполняемых одновременно.

Конвейерная обработка применяется в ВС, построенных на базе секционированных многокристалльных МП типа K1810, K1804, в суперЭВМ Cray 4-MP и др.

#### 6.4. Архитектура процессоров с сокращенным набором команд

Набор команд центрального процессорного устройства — операционного процессора микропроцессорных комплексов содержит от 90 до 303 команд (супермини-ЭВМ VAX 11/780). Компьютеры с сокращенным набором команд (КСНК), или иначе RISC-процессоры (от Reduced instruction set command) содержат 30...50 команд в наборе. Это достигается тем, что для выполнения наиболее часто встречающихся в алгоритмах микропрограммного управления переходов в устройство управления введены некоторые дополнительные средства. Наиболее часто повторяющиеся переходы приведены на рис. 6.3, где  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\phi$  — некоторые действия в программе.

На рис. 6.3, а представлен безусловный переход от адреса  $A_i$  по адресу  $A_j$ . Условный переход по адресу  $A_j$  при выполнении некоторого условия  $C$  и по адресу  $A_k$  при невыполнении этого условия (см. рис. 6.3, б). На рис. 6.3, в, г выполнение циклов по условию реализуется с использованием условного перехода по условию (в) или с помощью стека (г), т.е. безадресной памяти. Выполнение циклов по заданному числу их повторений реализуется со счетчиком  $R$  числа повторений  $N$ , входящим в состав устройства управления (см. рис. 6.3, д). Условный переход к микропрограмме  $A_j$  и возврат из нее по значениям условий  $C$  (см. рис. 6.3, е). На рис. 6.3, ж реализуется условный переход к микропрограмме  $A_j$  и возврат из нее по счетчику количества циклов. Безусловный переход к микропрограмме  $A_j$  и возврат после ее выполнения к основной программе (см. рис. 6.3, з).

Для выполнения этих переходов устройство управления должно содержать: средства мультиплексирования внутренних источников адреса (переходы типа (см. рис. 6.3, а...е), регистр управления микропрограммного счетчика с инкрементором и детектором нуля (см. рис. 6.3, д), сумматор (см. рис. 6.3, в...д), средства маскирования полей адреса (в зоне адреса в формате микрокоманды). Маскирование — логическая операция над байтом, словом или полем данных с целью модификации или выделения их части. Для этого битовая комбинация той же длины, что и выделяемая комбинация, генерируется и хранится в регистре, как маска. Маска используется для подавления некоторых разрядов данных, установки их в «0», для выявления признаков высокого приоритета в байтах состояния или в операциях обработки прерывания.

Если сократить набор команд, можно снизить еложность и стоимость ЭВМ. У КСНК получается регулярная структура простых команд, поэтому комбинация из нескольких команд выполняется быстрее, чем одна, эквивалентная этой комбинации сложная команда. Такие ЭВМ работают быстрее, команда выполняется за один машинный цикл, тогда как в других машинах — за несколько ма-

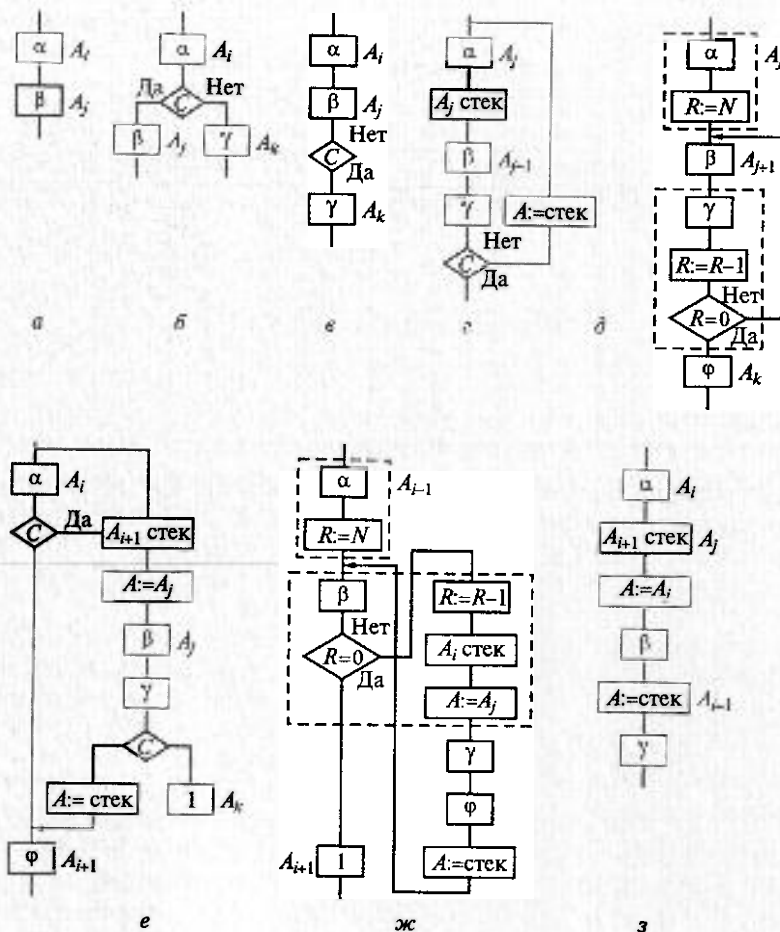


Рис. 6.3. Переходы в алгоритмах микропрограммного управления КСНК

шинных циклов. Большинство операций в КСНК — типа «регистр-регистр», а обращения к основной памяти происходят только для выполнения простых операций загрузки в регистр и занесения в память. Уменьшается число логических вентилях и объем микропрограммного ПЗУ, необходимых для декодирования и реализации сложных команд, поэтому размеры МП с RISC-архитектурой меньше и, соответственно, стоимость его тоже ниже. Так, RISC-процессору, состоящему из 50 000 транзисторов, соответствует 32-разрядный МП традиционной архитектуры на 150 000 транзисторов.

Цикл разработки RISC-процессора короче, чем у МП традиционной архитектуры. Так, МП ARM фирмы Acorn был разработан всего за 18 месяцев.

Частота выполнения команд в ЭВМ

Операции	Команды языков высокого уровня, %		Машинные команды, %		Обращение к памяти, %	
	Паскаль	Си	Паскаль	Си	Паскаль	Си
Вызов, возврат, запоминание и восстановление	15	12	31	33	44	45
Цикл	5	3	42	32	33	26
Присвоение	45	38	13	13	14	15
Условные	29	43	11	21	7	13
Другие	6	4	2	1	2	1

Программы, написанные на языках высокого уровня, с помощью программы компилятора транслируются в абсолютный машинный двоичный код или иногда в язык Ассемблер. В табл. 6.1 представлены полученные из наблюдений частоты выполнения команд в ЭВМ со сложным набором команд.

Затем разработчиками была предпринята попытка исключения тех команд, без которых можно обойтись, повысив скорость выполнения наиболее часто используемых команд путем реализации управляющего процессора частично на ПЗУ микрокоманд, а частично на блоках с «жесткой» логикой управления.

КСНК типа III-83 — K1883 был выпущен совместно СССР и ГДР на базе  $8 \cdot n$ -разрядных секций с микропрограммным управлением, выполненных по  $n$ -МДП-технологии. Фирмой TI&SOC в 1988 г. выпущен RISC-процессор с конвейерной обработкой, производительностью  $200 \cdot 10^6$  опер/с на базе арсенид-галлиевых интегральных микросхем.

### 6.5. Развитие новых архитектурных принципов

В процессоре традиционной фон неймановской архитектуры последовательно обрабатываются одиночные элементы данных — скалярные величины. Такие процессоры получили название скалярных процессоров. Архитектура скалярного процессора представлена на рис. 6.4, а. Поскольку на процессор поступают одиночный поток команд и одиночный поток данных, такая архитектура называется ОКОД (SISD — Single Instruction Single Data).

При реализации принципа параллельной обработки возможны три варианта архитектур: ОКМД — одиночный поток команд, множественный поток данных (SIMD — Single Instruction Multiple Data); МКМД (MIMD — Multiple Instruction Multiple Data) — мно-



Рис. 6.4. Типы архитектуры ЭВМ:

а — ОКОД; б — ОКМД; в — МКМД

жественный поток команд, множественный поток данных; и архитектура МКМД с коммутационной матрицей (рис. 6.4 и 6.5).

В ОКМД-машине одна и та же операция выполняется одновременно над разными данными.

В МКМД-машине содержится множество пар процессор — память и отсутствует центральное устройство управления.

МКМД-машины с коммутационной матрицей, иногда называемые *Connection machine*, служат для обработки больших информационных массивов данных, чаще всего встречающихся при манипулировании матрицами в научных исследованиях. Элементами матриц являются векторы в отличие от скалярных величин, обрабатываемых на ЭВМ традиционной ОКОД архитектуры. В случае единообразной обработки таких упорядоченных последовательностей данных существенный выигрыш дает использование конвейерной обработки. Векторная обработка увеличивает производительность процессорных элементов, но требует наличия полного параллелизма в ходе обработки задач. Векторная ЭВМ имеет отдельные векторные процессоры или конвейеры, одна команда выполняется в машине над несколькими элементами данных — векторами.

В суперЭВМ нового типа применяются одновременно и скалярные, и векторные процессоры (рис. 6.6).

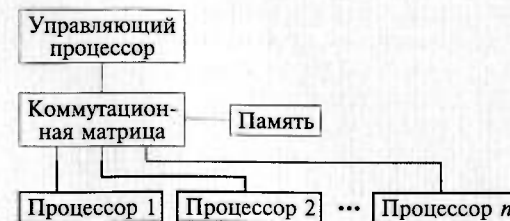


Рис. 6.5. Архитектура ЭВМ с параллельной обработкой типа МКМД с коммутационной матрицей



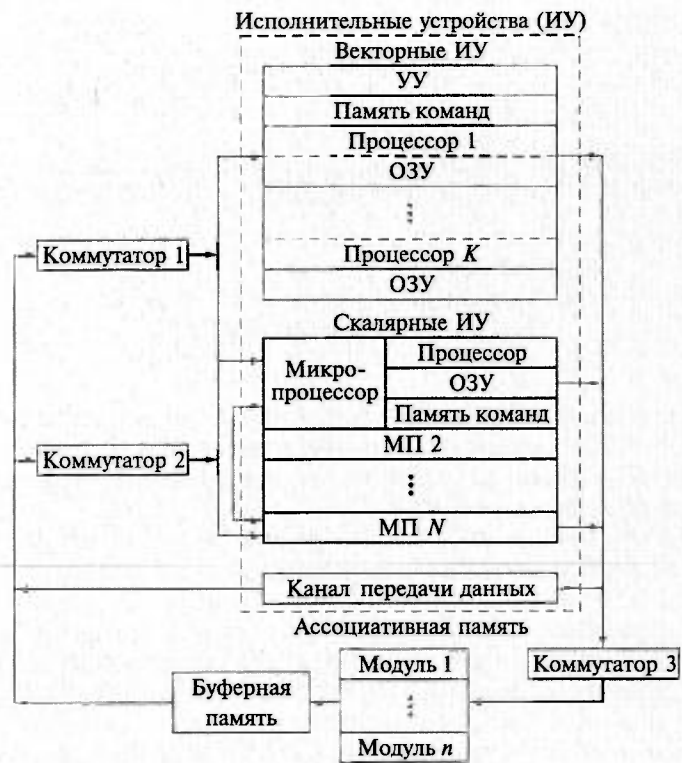


Рис. 6.6. Структурная схема суперкомпьютера нового типа

#### Контрольные вопросы

1. Какими причинами объясняется принципиальная невозможность создания ЭВМ, удовлетворяющей высоким требованиям различных задач по производительности и объему памяти?
2. Назовите четыре основные характеристики фон неймановской архитектуры ЭВМ.
3. Чему равна предельная производительность процессора фон неймановской архитектуры?
4. Перечислите этапы ввода в архитектуру ЭВМ в процессе их развития методов распараллеливания процессов обработки информации.
5. Что является элементной базой многопроцессорной ЭВМ?
6. В чем состоит опережающая выборка команд?
7. В чем состоит конвейерный поточный режим исполнения команд?
8. Во сколько раз в пределе может быть повышена производительность при конвейерной обработке?
9. За счет чего достигнуто сокращение числа команд в наборе команд КСНК (RISK-процессоров)? Перечислите достоинства КСНК.

## Глава 7 ПРОГРАММНОЕ (МАТЕМАТИЧЕСКОЕ) ОБЕСПЕЧЕНИЕ ЭВМ

### 7.1. Операционная система, ее назначение, состав и функции

Центральное место в структуре программного обеспечения (ПО) занимает операционная система.

*Операционная система (ОС)* — совокупность программных средств, обеспечивающая управление аппаратной частью компьютера и прикладными программами, а также их взаимодействие между собой и пользователем.

Операционная система образует автономную среду, не связанную ни с одним из языков программирования. Любая прикладная программа связана с операционной системой и может эксплуатироваться только на тех компьютерах, где имеется аналогичная системная среда. Прикладные программные средства, разработанные в среде одной операционной системы, не могут быть использованы для работы в среде другой операционной системы, если нет специального комплекса программ (конвертера), позволяющего это сделать. В таком случае говорят о программной несовместимости компьютеров.

Операционная система выполняет следующие функции:

- управление работой каждого блока персонального компьютера и их взаимодействием;
- управление выполнением программ;
- организацию хранения информации во внешней памяти;
- взаимодействие пользователя с компьютером, т. е. поддержку интерфейса пользователя.

Обычно операционная система хранится на жестком диске, а при его отсутствии выделяется специальный гибкий диск, который называется системным диском. При включении компьютера операционная система автоматически загружается с диска в оперативную память и занимает в ней определенное место. Операционная система создается не для отдельной модели компьютера, а для серии компьютеров, в структуре которых заложена и развивается во всех последующих моделях определенная концепция.

Для каждого типа ЭВМ возможно использование нескольких типов ОС. Все они имеют несколько версий. Для IBM PC распространение получили MS DOS фирмы Microsoft, OS/2Warp и DOS фирмы IBM, DR DOS фирмы Digital Research, DOS фирмы Novell. Отличия ОС определяются составом и детализацией системных функций. Более распространенной является MS DOS, она используется в большинстве компьютеров. Система DR DOS имеет очень развитые средства защиты информации и разграничение доступа, что предопределяет ее использование в системах с закрытием обрабатываемой информации. OS/2Warp позволяет более полно использовать возможности самых мощных микропроцессоров при организации вычислительного процесса. Novell DOS ориентирована на работу ЭВМ в сети.

Основу любой ОС составляет управляющая программа, основными функциями которой являются: управление заданиями, управление задачами — управление ходом выполнения отдельных программ, и управление данными.

Задание — это требование пользователя на выполнение некоторого объема вычислительных работ. Процедуры управления заданиями обеспечивают предварительное планирование работы ЭВМ и оперативную связь пользователя и оператора с машиной во время работы. Планирование работы включает: ввод пакетов или одиночных заданий, формирование очередей заданий в соответствии с их приоритетами, активизацию (запуск) и завершение заданий.

Каждое задание реализуется как определенная последовательность отдельных программ — задач. Задачи образуют отдельные программы вместе с обрабатываемыми ими данными. Например, типовое задание пользователя включает этапы трансляции, редактирования и собственно выполнения сформированной машинной программы. На каждом из этих этапов выполняется некоторая программа (задача), обрабатывающая определенные данные. Комплекс программ управления задачами обеспечивает автоматическое выполнение последовательности программ каждого задания пользователя.

Управление задачами требует распределения и назначения ресурсов (управления временем работы процессора, распределения оперативной памяти для программ пользователей и программ ПО, синхронизации выполнения задач и организации связей между ними, управления очередностью задач, внешними устройствами, защиты задач от взаимных помех). Ведущей программой управления задачами является управляющая программа-диспетчер: супервизор, базовый модуль дисковой операционной системы (ДОС) или др. Часто используемые модули образуют ядро ОС, которое постоянно находится в оперативной памяти и быстро реагирует на изменяющиеся условия функционирования. Примером такой программы может служить командный процессор `command.com`

для ПЭВМ типа IBM PC. Остальные программы ОС вызываются из ВЗУ в оперативную память ЭВМ по мере их надобности в вычислительном процессе.

Набор программ управления данными обеспечивает процессы организации, идентификации, размещения в ОП и на ВЗУ, хранения, построения библиотек и выборки всех данных, которые могут обрабатываться в ЭВМ.

В ПЭВМ программы управления заданиями представлены достаточно слабо, так как они изначально создавались как однопользовательские и однозадачные ЭВМ. С появлением ОС типа Windows, ориентированных на многозадачные и многопользовательские режимы, появились и эти процедуры. Программы управления задачами и данными представлены достаточно полно. Так, ядро MS DOS включает следующие системы: файловую, управления памятью, управления программами, связи с драйверами устройств для управления серийной аппаратурой, обработки ошибок, службы времени, общения с оператором.

Структурно ОС IBM PC состоит из следующих элементов.

Программа начальной загрузки (Boot Record) находится в первом секторе на нулевой дорожке системного диска. Она занимает объем 512 байт. После включения компьютера и его проверки постоянный модуль BIOS формирует вызов данной программы и ее запуск. Назначением программы начальной загрузки является вызов модуля расширения `IO.sys` и базового модуля ДОС `MS DOS.sys`.

Базовая система ввода-вывода (BIOS) является надстройкой аппаратурной части компьютера. Постоянный модуль BIOS отвечает за тестирование компьютера после его включения, вызов программы начальной загрузки. Модули BIOS обрабатывают прерывания вычислительного процесса нижнего уровня и обслуживают стандартную периферию: дисплей, клавиатуру, принтер, дисководы.

Модуль расширения BIOS обеспечивает подключение к компьютеру дополнительных периферийных устройств, изменение некоторых параметров ДОС, замещение некоторых стандартных функций, загрузку командного процессора и его запуск.

Базовый модуль ДОС (`MS DOS.sys` или `IBM DOS.com`) отвечает за работу файловой системы, обслуживает прерывания верхнего уровня (32...63), обеспечивает информационное взаимодействие с внешними устройствами.

Командный процессор (`COMMAND.com`) предназначен для выполнения команд, загружаемых в командную строку ДОС. Все команды ДОС делятся на внутренние и внешние. Внутренние команды содержатся внутри самого файла `command.com`. Внешние команды — это требования запуска каких-либо программ, находящихся на дисках. Кроме того, командный процессор

выполняет команды файла `autoexec.bat`, если он находится на системном диске.

Файл `autoexec.bat` содержит список команд, выполнение которых позволяет развернуть в оперативной памяти компьютера некоторый набор вспомогательных программ или пакетов для обеспечения последующей работы пользователя.

Файл `config.sys` отражает специфические особенности формирования конфигурации компьютера, т. е. состава его технических и программных средств.

## 7.2. Организация файловой структуры и методы доступа к файлам

В основе любой операционной системы лежит принцип организации работы внешнего устройства хранения информации. Несмотря на то, что внешняя память может быть технически реализована на разных материальных носителях (например, в виде гибкого магнитного диска или магнитной ленты), их объединяет принятый в операционной системе принцип организации хранения логически связанных наборов информации в виде так называемых файлов.

*Файл* — логически связанная совокупность данных или программ, для размещения которой во внешней памяти выделяется именованная область.

Файл служит учетной единицей информации в операционной системе. Любые действия с информацией в MS DOS осуществляются над файлами: запись на диск, вывод на экран, ввод с клавиатуры, печать, считывание информации CD-ROM и пр.

На диске файл не требует для своего размещения непрерывного пространства, обычно он занимает свободные кластеры в разных частях диска. Кластер является минимальной единицей пространства диска, которое может быть отведено файлу. Самый маленький файл занимает один кластер, большие файлы — несколько десятков кластеров. Сведения о номерах этих кластеров хранятся в специальной FAT-таблице.

На гибком диске обычно 80 дорожек. На каждой дорожке для 3,5" диска — 18 секторов. Кластер — группа смежных секторов. Кластер для гибкого диска составляет 1; 2 сектора (0,512; 1 Кбайт). Кластер для жесткого диска составляет 4; 8; 16 секторов (2...8 Кбайт).

В файлах могут храниться разнообразные виды и формы представления информации: тексты, рисунки, чертежи, числа, программы, таблицы и т. п. Особенности конкретных файлов определяются их форматом. Под форматом понимается элемент языка, в символическом виде описывающий представление информации в файле.

Текстовая информация хранится в файле в кодах ASCII, в так называемом текстовом формате. Содержимое текстовых файлов можно просмотреть на экране дисплея с помощью разных программных средств, в том числе и в MS DOS.

Любой другой файл с нетекстовой информацией просмотреть теми же средствами, что и текстовый файл, не удастся. При просмотре на экран будут выводиться абсолютно непонятные символы. Для характеристики файла используются следующие параметры:

- полное имя файла;
- объем файла в байтах;
- дата создания файла;
- время создания файла;
- специальные атрибуты файла — R (Read only) — только для чтения, H (Hidden) — скрытый файл, S (System) — системный файл, A (Archive) — архивный файл.

С понятием файла в MS DOS тесно связано понятие логического диска. Логический диск создается и управляется специальной программой (драйвером). Он имеет уникальное имя в виде одной латинской буквы, например, C, D, E, F и т. д. Логический диск может реализовываться на жестком диске, на гибком диске, на CD-ROM, в оперативной памяти (электронный диск) и т. п. На одном физическом диске может быть создано несколько логических дисков.

К файлу можно обращаться с помощью имени, полного имени, спецификации. Имя файла всегда уникально и служит для отличия одного файла от другого. В операционной системе MS DOS имя файла образуется не более чем из восьми символов, причем используются только буквы латинского алфавита.

В качестве имени файла можно использовать символьное имя устройства:

PRN или LPT1 (2, 3) — принтер или любое устройство, подключенное к параллельному порту;

CON — консоль (клавиатура при вводе и дисплей при выводе);

COM1 (2, 3, 4) — внешнее устройство, подключенное к последовательному порту;

NUL — фиктивное устройство, вывод в файл NUL никуда не направляется, а просто уничтожается.

Обычно к файлу обращаются с помощью полного имени. Полное имя файла образуется из имени файла и расширения (типа), разделенных точкой.

Тип файла служит для характеристики хранящейся в файле информации и образуется не более чем из трех символов букв латинского алфавита. Например, файл `TEXT1.txt` представляет собой текстовый файл.

Спецификация файла может иметь сокращенную и полную формы. Сокращенная форма спецификации образуется из имени



логического диска и полного имени файла, разделенных двоеточием (например, В:ТЕХТ1.txt).

**Организация доступа к файлу.** *Доступ* — процедура установления связи с памятью и размещенным в ней файлом для записи и чтения данных.

Имя логического диска, стоящее перед именем файла в спецификации, указывает логический диск, на котором следует искать файл. На этом же диске организован каталог, в котором хранятся полные имена файлов, а также их характеристики: дата и время создания; объем (в байтах); специальные атрибуты. По аналогии с библиотечной системой организации каталогов полное имя файла, зарегистрированное в каталоге, будет служить шифром, по которому операционная система находит месторасположение файла на диске.

*Каталог* — справочник файлов с указанием месторасположения на диске. Различают два состояния каталога — текущее (активное) и пассивное. MS DOS помнит текущий каталог на каждом логическом диске.

*Текущий (активный) каталог* — каталог, в котором работает пользователь производится в текущее машинное время.

*Пассивный каталог* — каталог, с которым в данный момент времени не имеется связи.

В операционной системе MS DOS принята иерархическая структура организации каталогов. На каждом диске всегда имеется единственный главный (корневой) каталог. Он находится на 0-м уровне иерархической структуры и обозначается символом «\». Корневой каталог создается при форматировании (инициализации, разметке) диска, имеет ограниченный размер и не может быть удален средствами DOS. В главный каталог могут входить другие каталоги и файлы, которые создаются командами операционной системы и могут быть удалены соответствующими командами.

*Родительский каталог* — каталог, имеющий подкаталоги. *Подкаталог* — каталог, который входит в другой каталог. Таким образом, любой каталог, содержащий каталоги нижнего уровня, может быть, с одной стороны, по отношению к ним родительским, а с другой стороны, подчиненным по отношению к каталогу верхнего уровня. Каталоги на дисках организованы как системные файлы. Единственное исключение — корневой каталог, для которого отведено фиксированное место на диске. Доступ к каталогам можно получить, как к обыкновенному файлу. В структуре каталогов могут находиться каталоги, не содержащие ни одного файла или подкаталога. Такие подкаталоги называются пустыми.

Правила наименования подкаталогов такие же, как и правила наименования файлов. Для формального отличия от файлов обычно подкаталогам присваивают только имена, хотя можно добавить и тип по тем же правилам, что и для файлов.

Доступ к содержимому файла организован из главного каталога, через цепочку соподчиненных каталогов (подкаталогов) *n*-го уровня. В каталоге любого уровня могут храниться записи как о файлах, так и о каталогах нижнего уровня.

С понятием файловой системы тесно связано понятие файловой структуры диска, под которой понимают, как размещаются на диске: главный каталог, подкаталоги, файлы, операционная система, а также какие для них выделены объемы секторов, кластеров, дорожек.

При формировании файловой структуры диска операционная система MS DOS соблюдает ряд правил:

- файл или каталог могут быть зарегистрированы с одним и тем же именем в разных каталогах, но в одном и том же каталоге только один раз;
- порядок следования имен файлов и подкаталогов в родительском каталоге произвольный;
- файл может быть разбит на несколько частей, для которых выделяются участки дискового пространства одинакового объема на разных дорожках и секторах.

Доступ к файлу осуществляется через каталог благодаря зарегистрированному в нем имени данного файла. Если каталог имеет иерархическую структуру, то доступ к файлу операционная система организует в зависимости от положения подкаталога, в котором зарегистрировано имя искомого файла. Доступ к файлу можно организовать следующим образом:

- если имя файла зарегистрировано в текущем каталоге, то достаточно для доступа к файлу указать только его имя;
- если имя файла зарегистрировано в пассивном каталоге, то, находясь в текущем каталоге, нужно указать путь, т.е. цепочку соподчиненных каталогов, через которые следует организовать доступ к файлу.

*Путь* — цепочка соподчиненных каталогов, которую необходимо пройти по иерархической структуре к каталогу, где зарегистрирован искомый файл. При задании пути имени каталогов записываются в порядке следования и отделяются друг от друга символом \. Взаимодействие пользователя с операционной системой осуществляется с помощью командной строки, индуцируемой на экране дисплея. В начале командной строки всегда имеется приглашение, которое заканчивается символом >. В приглашении может быть отображено: имя текущего диска, имя текущего каталога, текущие время и дата, путь, символы-разделители.

*Приглашение операционной системы* — индикация на экране дисплея информации, означающей готовность операционной системы к вводу команд пользователя.

Возможны три варианта организации пути доступа к файлу в зависимости от места его регистрации:

- файл находится в текущем каталоге (путь отсутствует). При организации доступа к файлу достаточно указать его полное имя;
- файл находится в пассивном каталоге одного из нижних уровней, подчиненного текущему каталогу. При организации доступа к файлу необходимо указать путь, в котором перечислены все имена каталогов нижнего уровня, лежащих на этом пути (включая каталог, в котором зарегистрирован данный файл);
- файл находится в пассивном каталоге на другой ветке по отношению к местонахождению текущего каталога иерархической структуры. При организации доступа к файлу необходимо указать путь, начиная с главного каталога, т. е. с символа \. Это объясняется тем, что в иерархической структуре движение возможно только по вертикали сверху-вниз. Горизонтальные переходы из каталога в каталог недопустимы.

### 7.3. Основные команды операционной системы

*Команда* — это способ общения пользователя с компьютером в операционной системе.

Команда технически реализована программой в машинных кодах и хранится либо в файле на диске, либо входит в состав командного процессора операционной системы MS DOS. Как и любая другая программа, команда имеет уникальное имя и всегда типа .com или .exe.

Ввод команды осуществляется в командной строке в соответствии с определенными правилами, заданными в виде формата.

Командная строка — строка экрана, начинающаяся с приглашения операционной системы. Формат команды — правило формирования команды пользователем с клавиатуры.

При формировании команды в соответствии с установленным форматом запомните правила:

- формат команды состоит из имени команды латинскими буквами (без типа) и отделенных от нее одним пробелом параметров, уточняющих ее действие;
- в большинстве случаев параметры между собой пробелом не разделяются, а в качестве разделителя часто используется символ /;
- параметрами могут быть: имя дискового, путь, имя файла, тип файла, латинские буквы, символы, цифры;
- присутствие параметров в формате часто бывает необязательным, что указывается с помощью квадратных скобок [ ].

Обобщенный формат команды можно представить в следующем виде:

Имя команды [Параметры].

Процедура ввода команды состоит в следующем.

1. В соответствии с форматом в командной строке набирается ее имя и необходимые параметры.

2. Нажимается клавиша ввода, что служит сигналом начала анализа ее структуры: при отсутствии ошибок в формате команды будет выполнено указанное действие; при наличии ошибок на экране выдается сообщение

Bad command or filename (Неверная команда или имя файла).

3. При невыполнении команды внимательно просматривается заданная конструкция команды и вновь вводится, но уже в откорректированном варианте.

Все команды операционной системы можно классифицировать по двум признакам — по способу реализации и по функциональному назначению.

По способу реализации команды разделяются на две группы:

- резидентные (внутренние, встроенные);
- транзитные (внешние, утилиты, обслуживающие).

Резидентные команды входят в состав командного процессора COMMAND.com и после окончания загрузки операционной системы MS DOS обычно располагаются в оперативной памяти. Резидентные команды доступны в любой момент времени. Пользователю не надо заботиться об их наличии или отсутствии в составе операционной системы. После ввода пользователем резидентной команды операционная система проводит анализ ее структуры и при отсутствии ошибок вызывает ее на выполнение из оперативной памяти. При этом не требуется обращения к внешней памяти, что существенно сокращает время ее выполнения. В операционной системе MS DOS невозможны замена или добавление резидентных команд, так как они являются частью командного процессора COMMAND.com.

Транзитные команды реализуются в виде файлов типа .com или .exe и постоянно находятся на диске в области пользователя. После ввода транзитной команды организация работы будет аналогична обычной процедуре обработки программного файла. Командный процессор COMMAND.com анализирует структуру введенной команды, передает управление другим модулям операционной системы, которые организуют обращение к диску и считывание программы из файла в оперативную память. Имя этого файла соответствует имени введенной команды. После выполнения команды управление возвращается командному процессору.

Можно неограниченно расширять возможности операционной системы за счет введения новых транзитных команд. По сравнению с резидентными командами транзитные команды характеризуются большим временем выполнения из-за необходимости обращения к диску. Транзитные команды так же, как и резидентные, задаются именем без указания типа.



По функциональному назначению классификация команд может быть сделана весьма условно вследствие разнообразных возможностей многих команд. Основным функциональным признаком может служить объект, с которым производятся различные действия, например каталог, файл, диск. В этом случае выделяются следующие группы команд: для работы с каталогами, с файлами, с дисками, управления памятью и устройствами, конфигурирования системы и др.

**Основные команды для работы с каталогами.** Команда DIR — просмотр каталога. К наиболее часто используемым командам относится команда просмотра содержимого каталога DIR. Работа на персональном компьютере, как правило, начинается с просмотра каталога, чтобы убедиться в том, что нужный файл или подкаталог существует. Только после этого можно переходить в ту среду, где будет протекать работа. В процессе работы часто появляется необходимость просмотра содержимого пассивного каталога. Работа с новым диском также предваряется просмотром его каталога. Для всех этих ситуаций и многих других пользуются командой DIR.

В зависимости от параметров, допустимых в структуре команды, можно просмотреть записи каталога в стандартной или усеченной форме с выводом только полных имен файлов, а также при большом каталоге выводить его постранично. Формат команды

DIR [Имя дисковода:][Путь\][Имя файла][Параметры].

Команда MD — создание каталога. Новый каталог можно создать командой MD в текущем каталоге или, если указан путь, в пассивном каталоге. Формат команды

MD [Имя дисковода:][Путь\]Имя подкаталога.

Команда RD — уничтожение каталога. Командой RD уничтожается только пустой каталог. Предварительно вы должны удалить из него командой DEL все файлы, а затем командой DIR, просмотрев каталог, убедиться в том, что он пустой. Только после этого используйте команду RD. Формат команды

RD [Имя дисковода:][Путь\]Имя подкаталога.

Команда CD — переход в другой каталог. Часто возникает ситуация, когда следует перейти в другой каталог и сделать его текущим. В этом случае следует воспользоваться командой CD. Формат команды

CD [Имя дисковода:][Путь\]Имя подкаталога.

Для перехода в родительский каталог достаточно вместо имени каталога задать .. (две точки). Для перехода в главный каталог задается символ \.

**Основные команды для работы с файлами.** Команда TYPE — просмотр текстового файла. Командой TYPE удобно пользоваться для просмотра содержимого текстового файла на экране дисплея или для вывода на принтер. После запуска команды текст (содержимое файла) выводится непрерывным потоком, причем ско-

рость смены кадров с текстом на экране настолько велика, что прочесть его весьма затруднительно. Для приостановки вывода текста нажмите одновременно две клавиши <CTRL><S>. Нажатие любой клавиши возобновит вывод текста. Формат команды для вывода на экран

TYPE [Имя дисковода:][Путь\] Полное имя файла.

Формат команды для печати

TYPE [Имя дисковода:][Путь\] Полное имя файла > PRN.

Команда DEL — удаление файлов. Команда DEL используется для удаления файлов. Можно удалять как один файл, так и группу файлов, указывая в имени файла \* или ?. Формат команды

DEL [Имя дисковода:][Путь\] Полное имя файла [/P].

Параметр /P служит для вывода на экран запроса на подтверждение удаления.

Команда COPY — копирование файлов. Этой командой можно пользоваться для создания копий существующих файлов. Помимо этой основной функции, команда COPY будет полезна и для выполнения двух вспомогательных функций:

- вывода содержимого файла на внешнее устройство;
- объединения содержимого нескольких файлов.

Команда COPY допускает выполнение функций над группой файлов, и тогда в именах файлов используются символы \* и ?. Формат команды для копирования файлов

COPY [Имя дисковода источника:][Путь\] Полное имя файла источника [Имя дисковода приемника:][Путь\] Полное имя файла приемника.

**Основные команды для работы с дисками.** Команда FORMAT — форматирование диска. Новый диск представляет собой покрытую магнитным слоем поверхность, на которой не создана необходимая для работы структура, т. е. не имеется ни дорожек, ни секторов, ни специальных зон, требуемых файловой системой. Перед началом записи на диск следует создать на диске такую структуру, которая принята в операционной системе персонального компьютера. В любой операционной системе имеется для этого соответствующая команда, называемая командой форматирования (инициализации) диска. С помощью этой команды на поверхности диска образуется структура, принятая в данной операционной системе. Процедура форматирования диска после ввода команды протекает обычно в диалоговом режиме и предоставляет шанс отказаться от форматирования, если вы недостаточно обдумали свой шаг.

В операционной системе MS DOS 6.22 для форматирования диска используется команда FORMAT. Формат команды

FORMAT [Имя дисковода:] [Параметры].

Параметры, используемые в этой конструкции, имеют следующее назначение:



/F: число — указание объема дисковой памяти: (160; 180; 320; 360; 720) Кбайт, (1,2; 1,44; 2,88) Мбайт;

/T: число — задание количества дорожек на одной стороне диска;

/M: число — задание количества секторов на дорожке;

/S — запись основных двух модулей операционной системы MS DOS;

/B — резервирование места для размещения основных системных модулей (IO.sys и MS DOS.sys);

/V — задание имени диска;

/I — форматирование одной стороны диска;

/Q — быстрое форматирование.

Команда DISKCOPY — копирование содержимого дисков. Часто возникает ситуация, когда надо сделать копию содержимого гибкого диска. Например, рекомендуется иметь два комплекта дисков:

- один комплект — рабочие диски, с которыми вы постоянно работаете;
- второй комплект — архив с копиями программ рабочих дисков.

Целесообразно на дисках архива закрыть прорезь «защита от записи», чтобы при очередном копировании с них не повредить оригинальные версии программ.

Копирование содержимого гибкого диска производится командой DISKCOPY по дорожкам. Эта команда является транзитной, поэтому прежде чем ее вводить, надо просмотреть каталог системного диска и убедиться в ее наличии. После ввода команды DISKCOPY содержимое диска-оригинала переписывается через оперативную память по дорожкам на диск-копию.

Команда DISKCOPY предусматривает форматирование диска-копии в соответствии с форматом диска-оригинала. Поэтому отпадает необходимость в предварительном форматировании диска-копии. Формат команды

DISKCOPY [Имя дисковод-оригинала:] [Имя дисковод-копии:].

Копирование дисков можно производить как на двух дисководах, так и на одном.

## 7.4. Графическая операционная система Windows и ее особенности

Семейство операционных систем Microsoft Windows включает целый комплекс систем, сложность и функциональные возможности которых возрастают с каждой версией. Начиная с версии Windows 95 — это высокопроизводительная, многозадачная и многопотоко-

вая 32-разрядная операционная система с графическим интерфейсом и расширенными сетевыми возможностями, работающая в защищенном режиме, поддерживающая 16-разрядные приложения без всякой их модификации.

Windows — интегрированная среда, обеспечивающая эффективный обмен текстовой, графической, звуковой и видеоинформацией между отдельными программами.

Windows 95 — первая полномасштабная операционная система семейства Windows, не требующая MS DOS.

Она полностью совместима с используемыми в настоящее время программными и аппаратными средствами и практически во всех отношениях значительно усовершенствована по сравнению с Windows 3.1.

Ключевыми новинками по сравнению с Windows 3.1 являются следующие:

- простота в работе, которая достигается использованием нового пользовательского интерфейса, технологии поддержки самонастраиваемой аппаратуры Plug and Play, допустимостью применения длинных имен файлов, встроенной сетевой поддержкой, усовершенствованной обучающей и справочной системой и др.;

- повышенная производительность, обусловленная такими свойствами Windows, как истинная вытесняющая многозадачность и масштабируемость рабочих характеристик для 32-битовых приложений, повышенная устойчивость к сбоям, ускоренная печать, наличие высокоэффективных мультимедийных компонентов, увеличенный объем памяти для программ MS DOS и др.;

- совместимость с существующими приложениями MS DOS и Windows, поддержка любого существующего оборудования и драйверов устройств, сетевая совместимость с существующими клиентами реального времени и процедурами регистрации.

Windows 95 — первый представитель нового поколения мощных 32-битовых многопоточковых операционных систем.

Усовершенствования, внесенные в архитектуру Windows 95, предоставляют пользователю по сравнению с Windows 3.1 ряд преимуществ. Основные преимущества связаны со следующими понятиями:

- интегрированная операционная система;
- вытесняющая многозадачность;
- многопоточность.

*Интегрированная операционная система* — операционная система, ядро которой, загружаемое в момент включения компьютера, активизирует графический интерфейс пользователя и обеспечивает полную совместимость с операционной системой MS DOS.

При использовании Windows 95 отпадает необходимость в отдельной копии MS DOS.

## 7.5. Многозадачный режим работы Windows

Вытесняющая многозадачность — свойство операционной системы самостоятельно в зависимости от внутренней ситуации передавать или забирать управление у того или иного приложения.

В Windows 3.1 приложения работают в режиме кооперативной многозадачности, т. е. последовательно. Каждое приложение периодически самостоятельно проверяет очередь сообщений, чтобы при необходимости передать управление другому приложению. Приложения, редко проверяющие очередь сообщений, забирают себе практически все процессорное время.

В Windows 95 для 32-битовых приложений используется механизм вытесняющей многозадачности, основанный на многопоточности.

*Многопоточность* — свойство операционной системы выполнять операции одновременно над потоками нескольких 32-битовых приложений, называемых процессами.

Процесс состоит из потоков. *Поток* — это некоторая часть процесса, которой может быть выделено процессорное время для одновременного выполнения наряду с другими потоками того или иного процесса, 32-битовые приложения Windows способны породить или инициировать несколько потоков внутри данного процесса. Каждый процесс состоит как минимум из одного потока. Многопоточное приложение значительно эффективнее в работе, быстрее реагирует на действия пользователя и выполняет многие операции в фоновом режиме.

Распределение времени между активными приложениями в Windows осуществляет ядро операционной системы, а поддержка вытесняющей многозадачности обеспечивает плавное переключение между одновременно выполняемыми процессами и не позволяет одному приложению занять все системные ресурсы.

Отметим, что Windows выполняет 32-битовые и MS DOS приложения в режиме вытесняющей многозадачности, а 16-битовые приложения — в режиме кооперативной многозадачности.

Везде, где не ущемляется совместимость с 16-битовыми приложениями и повышается производительность, Windows использует 32-битовый код. Такие компоненты, как планировщик, диспетчер памяти, подсистемы ввода-вывода и драйверы устройств, являются 32-битовыми. Многие графические функции, включая печать, растеризацию шрифтов TrueType, ключевые графические операции, переведены в 32-битовый код. Вместе с тем значительная часть кода User, осуществляющая управление окнами, оставлена для совмещения с существующими приложениями в 16-битовом формате.

**Технология Plug and Play: включи и работай.** В Windows, начиная с версии Windows 95, широко используется технология Plug

and Play, обеспечивающая новые возможности интеграции программных продуктов и аппаратных средств. Она ориентирована на поддержку любого типа устройств, включая мониторы, видеоплаты, принтеры, звуковые карты, модемы, приводы CD-ROM, различные контроллеры жестких дисков.

Технология Plug and Play упрощает работу с компьютером за счет следующих сервисных функций:

- помощь при распознавании устройств для их установки и настройки;
- уведомление программных продуктов и приложений и динамическое изменение состояния системы;
- тесная интеграция драйверов устройств, системных компонентов и пользовательского интерфейса, облегчающая настройку и управление системой. В Windows пользователю для подключения нового устройства, поддерживающего технологию Plug and Play, достаточно его просто вставить в систему. Перераспределение и настройка системных ресурсов далее происходят автоматически. Например, настольный компьютер легко превращается в мультимедийную систему простым подсоединением звуковой платы и привода CD-ROM и последующим запуском Windows.

Windows обеспечивает также динамическое изменение конфигурации системы, построенной на базе технологии Plug and Play. Например, если в процессе работы от станции отключается портативный компьютер, то Windows в этом компьютере автоматически изменяет конфигурацию, настраиваясь на монитор низкого разрешения, отсутствие сетевой карты и большого жесткого диска.

Технология Plug and Play, заложенная в Windows, позволяет также работать с устройствами, не подчиняющимися спецификации Plug and Play, упрощая их настройку и управление оборудованием.

Для корректного обращения с системными ресурсами компьютера Windows отслеживает все устройства и выделяемые им ресурсы. Диспетчер устройств Windows позволяет пользователю получать информацию о всех найденных системой Windows устройствах, изменять при необходимости их конфигурацию.

Кроме просмотра устройств, сконфигурированных в Windows, диспетчер устройств позволяет увидеть системные ресурсы, выделенные этим устройствам.

## 7.6. Основные объекты оконного интерфейса: окна, меню, ярлыки, папки

Задача пользовательского интерфейса — сделать компьютер доступнее для всех. Представление о пользовательском интерфейсе Windows дают рабочий стол и панель задач, появляющиеся

на экране монитора после включения компьютера (рис. 7.1). Рабочий стол занимает всю плоскость экрана и по желанию пользователя может быть одноцветным или содержать какое-нибудь изображение в виде фона. Панель задач располагается вдоль одной из границ экрана и содержит кнопку <Пуск> с логотипом Windows и кнопки с именами запущенных программ и открытых документов. Щелчок мышью на кнопку <Пуск> приводит к появлению главного системного меню, которое позволяет выполнить различные операции: запустить программы, открыть документы, вызвать панель управления для настройки компьютера, произвести поиск файла для папки, получить справку и т.д. При остановке указателя мыши на одном из пунктов меню автоматически открывается подменю. После выбора нужной строки подменю щелчком мыши запускается на выполнение выбранная программа. Все программы в Windows имеют подобный оконный интерфейс, включающий окно программы и систему меню.

На рабочем столе могут располагаться графические объекты, ярлыки, с названиями, соответствующие программам, приложениям, документам и позволяющие запускать эти приложения или открывать документы. На рабочем столе также могут располагаться сами приложения или отдельные файлы, с которыми ассоциируются миниатюрные графические изображения, иконки, и

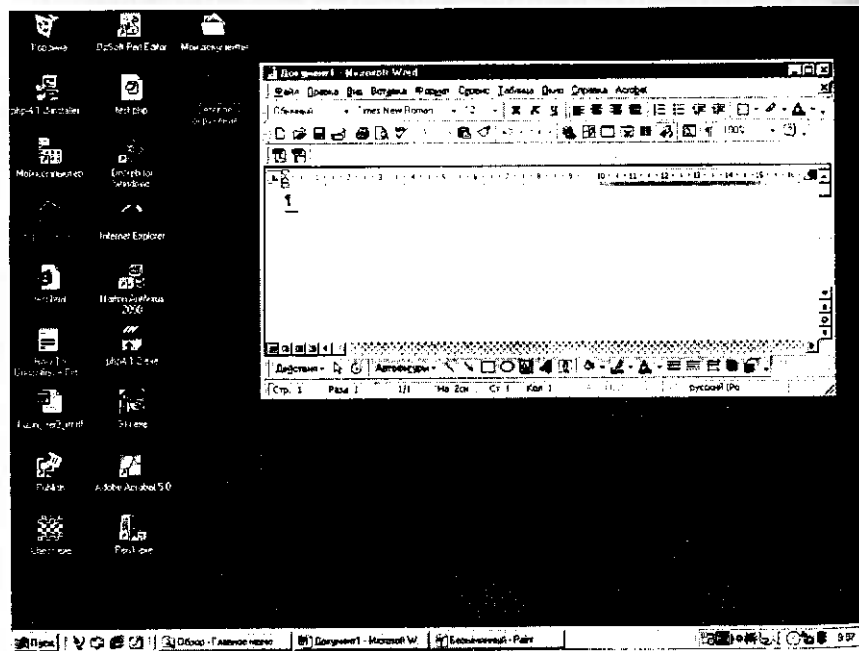


Рис. 7.1. Элементы интерфейса Windows

названия. Ярлыки отличаются от икон наличием стрелки в левом нижнем углу. В соответствии со своим вкусом пользователь может изменить состав графических объектов на рабочем столе, их внешний вид, расположение, форму, размеры и цвет.

В Windows большое значение имеет понятие папка. Папка — это, в частности, группа файлов, объединенных по какому-либо критерию, например «Мои документы». В Windows понятие папка сменило понятие каталог (директория), используемое в предыдущих версиях Windows и DOS и, как указано ранее, приобрело расширенное толкование.

#### Контрольные вопросы

1. Каково назначение операционной системы ЭВМ?
2. Каков состав и основные функции операционной системы?
3. Какова организация файловой системы ОС MS DOS?
4. Охарактеризуйте методы доступа к файлам ОС MS DOS.
5. Охарактеризуйте основные команды ОС MS DOS.
6. Сформулируйте и охарактеризуйте основные особенности ОС Windows.
7. Охарактеризуйте основные объекты оконного интерфейса Windows.
8. Расскажите о многозадачном режиме работы Windows.



## Глава 8

# АЛГОРИТМИЗАЦИЯ И ОСНОВЫ ПРОГРАММИРОВАНИЯ

### 8.1. Основные этапы решения задач на ЭВМ

Решение серьезной инженерной задачи с использованием ЭВМ — довольно длительный и сложный процесс. С определенной степенью условности его можно разбить на ряд последовательных этапов. Выделим следующие этапы:

- 1) постановка проблемы;
- 2) выбор или построение математической модели;
- 3) постановка вычислительной задачи;
- 4) предварительный (предмашинный) анализ свойств вычислительной задачи;
- 5) выбор или построение численного метода;
- 6) алгоритмизация и программирование;
- 7) отладка программы;
- 8) счет по программе;
- 9) обработка и интерпретация результатов;
- 10) использование результатов и коррекция математической модели.

**Постановка проблемы.** Первоначально прикладная задача бывает сформулирована в самом общем виде: исследовать некоторое явление, спроектировать устройство, обладающее заданными свойствами, дать прогноз поведения некоторого объекта в определенных условиях и т. д. На данной стадии происходит конкретизация постановки задачи, и первостепенное внимание при этом уделяется выяснению цели исследования. От исследователя требуется глубокое понимание существа задачи и умение сформулировать ее так, чтобы найденное решение было полезным и в то же время могло быть получено с помощью существующих методов и в реальные сроки. Неудачная постановка проблемы может привести к тому, что длительный и дорогостоящий процесс решения задачи завершится получением бесполезных или тривиальных результатов (в этом случае возможно и отсутствие каких-либо результатов).

Этот очень важный и ответственный этап завершается конкретной формулировкой проблемы на языке, принятом в данной предметной области. Знание возможностей, которые дает применение ЭВМ, может оказать существенное влияние на окончательную формулировку проблемы.

**Выбор или построение математической модели.** Для последующего анализа исследуемого явления или объекта необходимо дать его формализованное описание на языке математики, т. е. построить математическую модель. Часто имеется возможность выбора модели среди известных и принятых для описания соответствующих процессов, но нередко требуется и существенная модификация известной модели, а иногда возникает необходимость в построении принципиально новой модели.

Рассматриваемый этап — едва ли не самый важный и трудный. Часто удачный выбор математической модели является решающим шагом к достижению цели. Одна из существенных трудностей такого выбора состоит в объективном противоречии между желанием сделать описание явления как можно более полным (что приводит к усложнению модели) и необходимостью иметь достаточно простую модель (чтобы была возможность реализовать ее на ЭВМ). Важно, чтобы сложность математической модели соответствовала сложности поставленной проблемы. Если поставленных целей можно достичь, используя более простую математическую модель, то ей и следует отдать предпочтение. Как правило, полезно иметь несколько упрощенных вариантов принимаемой модели. Заметим, что грамотное упрощение модели — непростая задача, однако анализ упрощенных моделей весьма полезен в течение всего процесса решения задачи. Такие упрощенные модели часто позволяют ответить на многие принципиальные вопросы и понять основные закономерности поведения более сложной модели.

**Постановка вычислительной задачи.** На основе принятой математической модели формулируют вычислительную задачу (или ряд таких задач). Анализируя результаты ее решения, исследователь предполагает получить ответы на интересующие его вопросы.

**Предварительный анализ свойств вычислительной задачи.** На этом этапе проводят предварительное (предмашинное) исследование свойств вычислительной задачи. Большое внимание уделяют анализу корректности ее постановки, т. е. выяснению вопросов существования и единственности решения, а также исследованию устойчивости решения задачи к погрешностям входных данных. Такое исследование, как правило, относится к компетенции профессиональных математиков. Тем не менее инженеру полезно быть в курсе современного состояния названных проблем, уметь самостоятельно проводить простейшие исследования.

К сожалению, для многих имеющих практическую ценность задач их строгое исследование в полной постановке провести не

удается, и к решению приступают без детального анализа математических свойств этих задач. Это нежелательная, но вынужденная мера, так как в прикладных исследованиях существенное значение имеют конкретные (часто — весьма сжатые) сроки получения результата. На этом этапе полезным оказывается изучение упрощенных постановок задачи. Иногда для них удается провести исследование, позволяющее понять основные особенности исходной вычислительной задачи. Особую ценность имеют различные аналитические решения; они оказываются полезными не только для анализа явления, но и как основа для тестовых испытаний на этапе отладки программы.

**Выбор или построение численного метода.** Для решения вычислительной задачи на ЭВМ требуется использование численных методов.

Часто решение инженерной задачи сводится к последовательному решению стандартных вычислительных задач, для которых разработаны эффективные численные методы. В этой ситуации происходит либо выбор среди известных методов, либо их адаптация к особенностям решаемой задачи. Однако если возникающая вычислительная задача является новой, то не исключено, что для ее решения не существует готовых методов. Построение численного метода для такой задачи может оказаться очень трудной проблемой и потребовать привлечения специалиста по вычислительной математике. Умение различать отмеченные две ситуации необходимо, и наличие его уже говорит об определенной квалификации в области вычислительных методов.

Для решения одной и той же вычислительной задачи обычно может быть использовано несколько методов. Необходимо знать особенности этих методов, критерии, по которым оценивается их качество, чтобы выбрать метод, позволяющий решить проблему наиболее эффективным образом. Здесь выбор далеко не однозначен. Он существенно зависит от требований, предъявляемых к решению, имеющихся в наличии ресурсов, доступной для использования вычислительной техники и т. д.

**Алгоритмизация и программирование.** Как правило, выбранный на предыдущем этапе численный метод содержит только принципиальную схему решения задачи, не включающую многие детали, без которых невозможна реализация метода на ЭВМ. Необходима подробная детализация всех этапов вычислений, для того чтобы получить реализуемый на ЭВМ алгоритм. Составление программы сводится к переводу этого алгоритма на выбранный язык программирования.

**Отладка программы.** На этом этапе с помощью ЭВМ выявляют и исправляют ошибки в программе.

Как правило, начинающий пользователь ЭВМ убежден, что ошибок в составленной им программе нет или же они могут быть

легко обнаружены и исправлены. Однако совершенно неожиданно для него отладка программы и доведение ее до рабочего состояния нередко оказывается длительным и весьма трудоемким процессом. Приобретая определенный опыт в составлении и отладке сравнительно сложных программ, пользователь убеждается в справедливости популярного афоризма: «В любой программе есть по крайней мере одна ошибка».

Таким образом, наличие в программах ошибок — вполне нормальное и закономерное явление. Поэтому подготовку к отладке следует начинать уже на этапе алгоритмизации и программирования. Заметим, что эффективность отладки самым существенным образом зависит от общей методики разработки программ. После устранения ошибок программирования необходимо провести тщательное тестирование программы — проверку правильности ее работы на специально отобранных тестовых задачах, имеющих решения.

**Счет по программе.** На этом этапе происходит решение задачи на ЭВМ по составленной программе в автоматическом режиме. Этот процесс, в ходе которого входные данные с помощью ЭВМ преобразуются в результат, называют *вычислительным процессом*. Как правило, счет повторяется многократно с различными входными данными для получения достаточно полной картины зависимости от них решения задачи.

Первые полученные результаты тщательно анализируются, для того чтобы убедиться в правильности работы программы и пригодности выбранного метода решения. Счет по программе продолжается несколько секунд, минут или часов. Именно быстротечность этого этапа порождает распространенную иллюзию о возможности решать важные прикладные задачи на ЭВМ в очень короткое время. В действительности же, конечно, необходимо принимать во внимание весь цикл от постановки проблемы до использования результатов. Для серьезных задач часто полезные результаты получаются только в результате многолетней работы.

**Обработка и интерпретация результатов.** Для правильной интерпретации результатов расчетов и оценки их достоверности от исследователя требуется глубокое знание существа решаемой инженерной задачи, ясное представление об используемой математической модели и понимание (хотя бы в общих чертах) особенностей применяемого вычислительного метода.

**Использование результатов и коррекция математической модели.** Завершающий этап состоит в использовании результатов расчетов в практической деятельности, иначе говоря, во внедрении результатов. Не стоит огорчаться, если большинство полученных сначала результатов окажется бесполезным. Действительно полезные для практики результаты являются плодом серьезной целенаправленной работы, в процессе которой цикл решения задачи повторяется неоднократно.



Очень часто анализ результатов, проведенный на этапе их обработки и интерпретации, указывает на несовершенство используемой математической модели и необходимость ее коррекции. В таком случае математическую модель модифицируют (при этом она, как правило, усложняется) и начинают новый цикл решения задачи.

## 8.2. Правила составления программ







Правила, которые могут понадобиться программисту при составлении любой программы, можно сформулировать следующим образом. Прежде всего программисту необходимо четко представить алгоритм решения поставленной задачи и попытаться разделить его по возможности на отдельные законченные части. Такое разбиение достигается составлением *схемы программы* — графического представления программных блоков (модулей) в виде геометрических фигур, помеченных своими порядковыми номерами и соединенных между собой связями, отражающими последовательность использования их машиной. Вычерчивание блоков производится в соответствии с ЕСКД.

В табл. 8.1 даны обозначения некоторых блоков в соответствии с ГОСТ 19002—80, 19003—80.

Блоки соединяются линиями потока информации. Внутри блоков записываются выполняемые действия. Линии определяют направление вычислений, причем сверху вниз и слева направо. Если необходимо отразить другое направление (снизу вверх или справа

Таблица 8.1

Обозначения блоков алгоритма программы

Наименование символа	Обозначение символа	Примечание
Процесс		Арифметический блок, определяющий действия, которые необходимо выполнить
Предопределенный процесс		Обращение к подпрограмме
Принятие решения		Логический блок, проверяющий истинность или ложность некоторого условия
Передача данных		Ввод или вывод информации
Прерывание		Начало, конец, пуск, останов, вход в подпрограмму
Модификация		Организация циклического процесса

налево), то на линиях ставятся стрелки. Блоки на схеме нумеруются цифрами, которые ставятся в разрыве верхней линии слева.

Схема алгоритма программы вначале может быть укрупненной, т.е. структурной, и отражать технологию вычислительного процесса. Если отдельные блоки будут разрабатываться различными программистами (принцип модульного программирования), то программисты сразу же должны установить связи между блоками. После этого содержимое крупных блоков детализируется до уровня, определяемого языком программирования, который предполагается использовать для составления программы. Уровень языка программирования определяется классом задачи (программы). В том случае, когда отдельная программа входит в комплекс программ, реализующий какой-то проект, следует ориентироваться на язык, который был выбран общим для всей разработки. Если задача носит вычислительный характер, программист должен обратить внимание на выбор численного метода ее решения и при необходимости принять меры к тому, чтобы получить максимальную точность вычислений. При описании данных необходимо быть внимательным к формату данных и их масштабированию.

## 8.3. Структура программы. Отладка программы

По своей структуре программы могут быть линейными, циклическими и ветвящимися.

*Линейная программа* — программа, команды которой исполняются машиной в порядке последовательности их записи (рис. 8.1, а). При большом числе операций такая программа окажется очень громоздкой. Во избежание этого повторяющиеся блоки можно объединить в цикл, который будет написан один раз, а машина будет повторять выполнение операций в цикле столько раз, сколько укажет программист. Такие программы называются *циклическими программами* (рис. 8.1, б).

Любой циклический алгоритм содержит несколько типовых блоков. Основной блок, называемый *телом цикла*, производит требуемые вычисления. Остальные блоки имеют вспомогательное значение, они организуют циклический процесс: устанавливают начальные и новые значения данных, проверяют условия окончания или продолжения циклического процесса. Различают три типа структур цикла: цикл с предусловием, цикл с постусловием, цикл с параметром или с повторением. Циклический алгоритм позволяет компактно описать большое число одинаковых вычислений над разными данными для получения необходимого результата.

*Разветвляющийся алгоритм* содержит блок (или блоки) проверки некоторого условия, и в зависимости от результата провер-



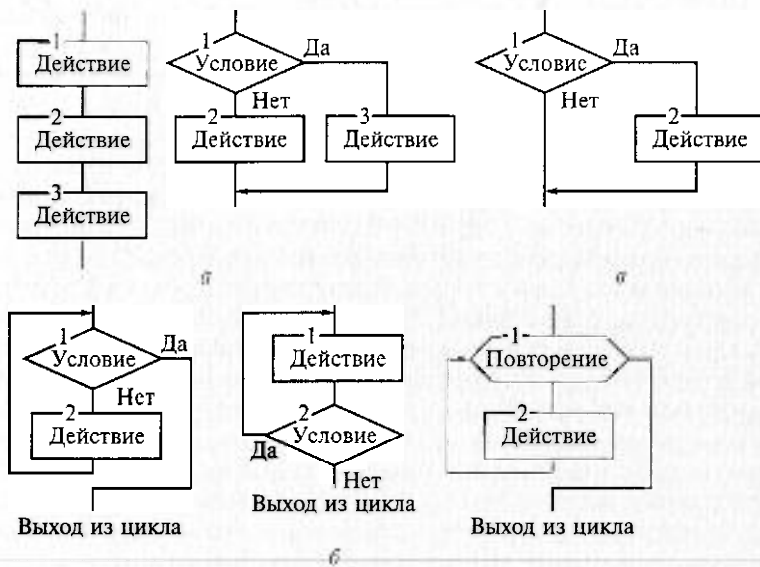


Рис. 8.1. Структура программ:

а — линейная; б — циклическая; в — разветвляющаяся

ки выполняется та или иная последовательность операций, называемая *ветвью*. При этом форма разветвления может быть как полная, так и сокращенная (рис. 8.1, в).

#### 8.4. Подпрограммы и встроенные функции

С ростом числа разрабатываемых программ было замечено, что во многих программах можно выделить участки, повторяющиеся в одной программе или являющиеся общими для различных программ. Эти участки могут быть записаны в программе отдельным блоком, а в ходе выполнения основной программы их можно вызывать по мере необходимости и исполнять. Такие программные блоки называются *подпрограммами* основной программы. Если подпрограммы должны использоваться в различных программах, то они могут храниться отдельно от программ. В этом случае их называют *стандартными программами*. Примерами стандартных программ могут быть программы ввода-вывода данных, программы вычисления тригонометрических функций, логарифма и др. Большинство из этих программ в настоящее время считается элементами алгоритмических языков высокого уровня и используется в качестве *встроенных функций* языка.

Стандартные программы объединяются в библиотеку программ и составляют неотъемлемую часть машины. Возможность исполь-

зования заранее подготовленных программ значительно ускорила процесс разработки пользовательских программ. Но при этом возникла необходимость в наличии общей программы, которая бы обслуживала библиотеку стандартных программ и осуществляла объединение пользовательской программы со стандартной в один загрузочный модуль.

#### 8.5. Понятие о структурном программировании

Структурное программирование основано на модульной структуре программного продукта и типовых управляющих структурах алгоритмов обработки данных различных программных модулей

*Модуль* — логически взаимосвязанная совокупность функциональных элементов, оформленных в виде отдельных программных модулей. Модуль характеризуют:

- один вход и один выход — на входе программный модуль получает определенный набор исходных данных, выполняет содержательную обработку и возвращает один набор результатных данных, т. е. реализуется стандартный принцип IPO (Input — Process — Output) — вход — процесс — выход;

- функциональная завершенность — модуль выполняет перечень регламентированных операций для реализации каждой отдельной функции в полном составе, достаточных для завершения начатой обработки;

- логическая независимость — результат работы программного модуля зависит только от исходных данных, но не зависит от работы других модулей;

- слабые информационные связи с другими программными модулями — обмен информацией между модулями должен быть по возможности минимизирован;

- обозримый по размеру и сложности программный элемент.

Таким образом, модули содержат определение доступных для обработки данных, операции обработки данных, схемы взаимосвязи с другими модулями.

Состав и вид программных модулей, их назначение и характер использования в программе в значительной степени определяются инструментальными средствами. Например, применительно к средствам систем управления базами данных (СУБД) отдельными модулями могут быть: экранные формы ввода и (или) редактирования информации базы данных; отчеты генератора отчетов; макросы; стандартные процедуры обработки информации; меню, обеспечивающее выбор функции обработки и др.

При структурном программировании любой блок, кроме условного, имеет только один вход и выход, безусловный переход на блок с нарушением иерархии запрещен (оператор типа GoTo в

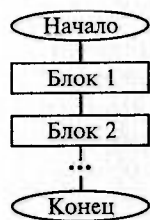


Рис. 8.2. Управляющая структура Последовательность

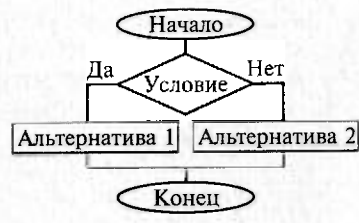


Рис. 8.3. Управляющая структура Альтернатива



Рис. 8.4. Управляющая структура Цикл

структурном программировании не используется). Виды основных управляющих структур алгоритма делятся на последовательность, условие выбора, цикл.

**Последовательность** (рис. 8.2). Последовательность включает в себя фиксированный перечень блоков (операторов). Каждый очередной блок обрабатывается после завершения предыдущего без дополнительных условий. Для изменения порядка обработки блоков редактируется последовательность выполняемых блоков (операторов).

**Условие выбора, или альтернатива** (рис. 8.3). В блоке **Условие** содержится условие выбора альтернативы обработки. Каждая альтернатива выполняется 1 раз; выполнение одной из двух альтернатив — обязательно. Развитием данного типа структуры является множественная альтернатива, когда последовательно проверяются условия выполнения определенных альтернатив. Если очередное условие истинно, обрабатывается соответствующая ему альтернатива, после чего происходит выход. В противном случае — переход к проверке условия следующей альтернативы. Если ни одно из условий не выполнилось, происходит выход.

**Цикл** (рис. 8.4). В блоке **Условие** задается условие **Тело цикла** — определенной обработки. Если условие не выполняется, цикл прерывается и осуществляется выход. Условие может содержать счетчик повторений тела цикла либо логическое условие. **Тело цикла** — произвольная последовательность блоков (операторов) обработки.

## 8.6. Понятие об объектно-ориентированном программировании

Объектно-ориентированный подход в программировании предлагает все, что входит в состав приложения, считать объектами, которые взаимодействуют друг с другом и с пользователем в со-

ответствии с заданными в программе свойствами и поведением, выполняя необходимые функции приложения. Таким образом, любое приложение при данном подходе представляет собой набор взаимосвязанных объектов, реализующих необходимые функциональные требования, предъявленные к приложению.

**Объект** — это некая отдельная сущность, выделяющаяся среди других сущностей особыми свойствами, поведением, взаимодействием с себе подобными объектами приложения. Объект всегда конкретен и реально существует в форме или в приложении, обладает при этом только ему присущими свойствами и поведением. Признаками объектов, отличающих их друг от друга, являются их свойства и поведение. Объект характеризуется следующими основными понятиями.

**Метод** — это функция или процедура, которая реализует возможные с объектом действия.

**Событие** — это средство взаимодействия объектов друг с другом. Объекты генерируют заданные события и выполняют действия в ответ на заданные события. События — это аналог сообщений, которые получают и отправляют объекты.

**Состояние** — каждый объект всегда находится в определенном состоянии, которое характеризуется набором свойств объекта. Под воздействием событий объект переходит в другие состояния. При этом объект может сам генерировать события о переходе в другое состояние.

**Свойство** — признак, некоторое отдельное качество (параметр) объекта. Например, свойствами могут быть размеры объекта, заголовков, его наименование. Совокупность свойств объекта определяет его состояние. Как правило, свойства — это набор переменных и констант, в которых хранятся значения, определяющие параметры объекта.

Объекты, имеющие общие свойства и поведение, объединяются в классы. Таким образом, **класс** можно определить как некую общность конкретных объектов, как описание объекта — каким он должен быть и что должен делать. Если объекты существуют в приложениях, то класс — это абстракция, объединяющая объекты в одну группу согласно их свойствам и поведению в среде окружения, в которой они существуют и взаимодействуют. Класс, таким образом, является некоей абстракцией, так как он не существует в работающем приложении. Например, кнопка в форме со всеми своими конкретными свойствами и действием является объектом некоторого класса.

Класс является носителем общих для объектов этого класса признаков. Класс характеризуется следующими основополагающими понятиями (ООП).

**Наследование.** Объект класса обладает всеми свойствами, методами и событиями класса.

**Иерархия.** Все классы образуют иерархическую систему и могут быть дочерними по отношению к вышестоящим классам.

**Инкапсуляция.** Скрытие сложного механизма действия объекта.

**Полиморфизм.** Объекты разных классов могут использовать одноименные методы, работающие по-разному для разных объектов.

**Методы класса** — это выполняемые классом функциональные «обязанности», задаваемые при его проектировании. Практически они представляют собой процедуры и функции, которые реализуют все функциональные требования к объектам класса и, соответственно, к классу как описанию объектов.

**События класса** — это сообщения, при помощи которых класс управляется и реагирует на воздействия внешней среды. Все события определяются в классе при его проектировании. В дальнейшем при использовании объектов этого класса они будут воспринимать и генерировать соответствующие классу события. Классам присущи разнообразные события и они, конечно, различны для каждого класса, однако в языках высокого уровня имеются два специальных события, которые присущи всем классам. Это события создания и уничтожения (прекращения работы) объектов класса.

**Свойства объекта.** Объект обладает всеми свойствами класса. Однако это не означает, что объекту нельзя добавить какое-либо дополнительное свойство, событие или метод. Если необходимо использовать в объекте дополнительные методы или другие члены, их можно добавить в объект или создать дочерний класс с требуемыми членами на базе класса этого объекта.

**Понятие коллекции объектов.** Коллекция — это набор объектов, объединенных общим именем, причем необязательно это объекты одного класса. Для типовых пользовательских объектов характерны (и обычно присутствуют при создании коллекции без специального их объявления) следующие свойства и методы:

- метод Add — добавляет объект в коллекцию;
- метод Remove — исключает объект из коллекции;
- свойство Count — возвращает общее число объектов в коллекции;
- метод Item — возвращает объект из коллекции.

Программный продукт, созданный с помощью инструментальных средств объектно-ориентированного программирования, содержит объекты с их характерными свойствами, для которых разработан графический интерфейс пользователя. Как правило, работа с программным продуктом осуществляется с помощью экранной формы, с объектами управления, которые содержат методы обработки, вызываемые при наступлении определенных событий. Экранные формы также используются для выполнения за-

даний и перехода от одного компонента программного продукта к другому. Каждый объект управления обладает определенными свойствами, значения которых могут изменяться. Для объектов управления уточняется перечень событий и создаются пользовательские методы обработки — программный код на языке программирования в виде *событийных процедур*.

Существуют различные объектно-ориентированные технологии и методики проектирования программных продуктов, для которых характерны следующие черты:

- объект описывается как модель некоторой сущности реального мира;
- объекты, для которых определены места хранения, рассматриваются во взаимосвязи, и применительно к ним создаются программные модули системы.

В процессе объектно-ориентированного анализа:

- осуществляется идентификация объектов и их свойств;
- устанавливается перечень операций (методов обработки), выполняемых над каждым объектом, в зависимости от его состояния (событий);
- определяются связи между объектами для образования классов;
- устанавливаются требования к интерфейсу с объектами.

Выделено четыре этапа объектно-ориентированного проектирования:

- 1) разработка диаграммы аппаратных средств системы обработки данных, показывающей процессоры, внешние устройства, вычислительные сети и их соединения;
- 2) разработка структуры классов, описывающей связь между классами и объектами;
- 3) разработка диаграмм объектов, показывающих взаимосвязи с другими объектами;
- 4) разработка внутренней структуры программного продукта.

## 8.7. Понятие о базах данных, базах знаний, экспертных системах, интегрированных системах

Программные продукты можно разделить на три класса:

- 1) системное программное обеспечение;
- 2) инструментарий технологии программирования;
- 3) пакеты прикладных программ.

Системное программное обеспечение было кратко рассмотрено в предыдущей главе.

**Инструментарий технологии программирования** — это совокупность программных средств, обеспечивающих технологию разработки, отладки и внедрения создаваемых программных продуктов.



Сюда относятся всевозможные алгоритмические языки низкого и высокого уровней. Данные программные средства требуют самостоятельного рассмотрения и выходят за рамки настоящего учебника.

**Пакет прикладных программ** — комплекс взаимосвязанных программ для решения задач определенного класса конкретной предметной области. Среди пакетов прикладных программ можно выделить программы автоматизированного проектирования, системы управления базами данных, интеллектуальные, экспертные и интегрированные системы.

Программы автоматизированного проектирования предназначены для поддержания работы конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм, графическим моделированием и конструированием, созданием библиотеки стандартных элементов чертежей и их многократным использованием, созданием демонстрационных иллюстраций и мультфильмов.

Отличительной особенностью этого класса программных продуктов являются высокие требования к технической части системы обработки данных, наличие библиотек встроенных функций, объектов, интерфейсов с графическими системами и базами данных.

Еще одной группой являются СУБД. Они появились, когда ЭВМ стали использоваться в контуре управления технологическими процессами и людскими коллективами. Разработка различных автоматизированных систем управления предполагает создание в памяти ЭВМ информационных моделей объектов управления — больших информационных массивов, получивших название «база данных».

**База данных (БД)** — это совокупность взаимосвязанных данных, хранящихся совместно в памяти ЭВМ. Каждая БД состоит из *записей*. Запись образует подмножество данных, служащих для описания единичного объекта. Например, фамилия, имя, отчество, год рождения, адрес, место работы, номер телефона могут составлять одну запись и характеризовать одного человека. Информационный массив может содержать записи по отдельным цехам, службам, отделам всего предприятия. Назначением БД является удовлетворение информационных потребностей пользователей. СУБД автоматизирует работу пользователей с хранящимися данными. Количество информационных массивов в БД и их объем зависят от сложности создаваемой системы. Ядро БД составляет информация, наиболее часто используемая в процессах управления. Согласно принципу В. Парето (итальянский экономист XIX в.) 20 % всей информации обеспечивают более 80 % всех задач управления. Эта часть в первую очередь и подлежит автоматизации.

Достаточно мощные СУБД позволяют значительно автоматизировать процессы управления и удовлетворять до 90...95 % потребностей управленческого аппарата. Одним из основных назна-

чений СУБД является *автоматизация документооборота*. На основе хранящейся информации можно автоматически формировать любые стандартные документы. Дополнительно к этому СУБД позволяет обращаться к данным и с нестандартными запросами для получения каких-либо справок, обобщений. СУБД поддерживает диалоговый режим работы пользователей, в котором запросы данных и реакция системы побуждают к формированию более точных запросов и исследованию данных.

СУБД обеспечивает ввод, поиск, сортировку данных, составление отчетов. Они имеют возможность сопряжения с табличными процессорами для специфической обработки и графического представления данных. В настоящее время широко используются СУБД: Dbase IV (V), Fox Pro, Paradox, Clipper, Access и др. Все они состоят из языковых и программных средств. Различие между ними состоит в предлагаемом сервисе и удобствах работы.

**Системы искусственного интеллекта.** *Искусственный интеллект* — это одно из направлений информатики, цель которого разработка аппаратно-программных средств, позволяющих пользователю-непрограммисту ставить и решать свои задачи, традиционно считающиеся интеллектуальными, общаясь с ЭВМ на ограниченном подмножестве естественного языка.

Основное направление искусственного интеллекта — это представление знаний и разработка систем, основанных на знаниях. Оно связано с разработкой моделей представления знаний, созданием баз знаний, образующих ядро экспертных систем.

*Экспертные системы* — это сложные программные комплексы, аккумулирующие знания специалистов в конкретных предметных областях и тиражирующие этот опыт для нужд менее квалифицированных пользователей.

Для хранения данных используются *базы данных* (для них характерны большой объем и относительно небольшая удельная стоимость информации), для хранения знаний — *базы знаний* (небольшого объема, но исключительно дорогие информационные массивы). База знаний — основа любой интеллектуальной системы.

Данный класс программных продуктов реализует отдельные функции интеллекта человека. Основными компонентами систем искусственного интеллекта являются база знаний, интеллектуальный интерфейс с пользователем и программа формирования логических выводов. Их разработка идет по следующим направлениям:

- программы-оболочки для создания экспертных систем путем наполнения баз знаний и правил логического вывода;
- готовые экспертные системы для принятия решений в рамках определенных предметных областей;
- системы управления базами знаний для поддержания семантических моделей (процедурной, семантической сети, фрейм-вой, продукционной и др.);

- системы анализа и распознавания речи и др.
- Как правило, интеллектуальный интерфейс включает в себя:
- диалоговый процессор на естественном языке;
  - планировщик, преобразующий описание задачи в программу решения на основе информации базы знаний;
  - монитор, осуществляющий управление компонентами интерфейса.

*Интегрированные пакеты* — набор нескольких программных продуктов, функционально дополняющих друг друга, поддерживающих единые информационные технологии, реализованные на общей вычислительной и операционной платформе.

Наиболее распространены интегрированные пакеты, компонентами которых являются: СУБД; текстовый редактор; табличный процессор; органайзер; средства поддержки электронной почты; программы создания презентаций; графический редактор. Примером может являться пакет MS Office. Компоненты интегрированных пакетов могут работать изолированно друг от друга, но основные достоинства интегрированных пакетов проявляются при их разумном сочетании друг с другом. Пользователи интегрированных пакетов имеют унифицированный для различных компонентов интерфейс, тем самым обеспечивается относительная легкость процесса их освоения.

Отличительными особенностями данного класса программных средств являются:

- полнота информационных технологий для конечных пользователей;
- однотипный интерфейс конечного пользователя для всех программ, входящих в состав интегрированного пакета — общие команды в меню, стандартные пиктограммы одних и тех же функций (сохранение на диске, печать, проверка орфографии, шрифтовые оформления и т. п.), стандартное построение и работа с диалоговыми окнами и др.;
- общий сервис для программ интегрированного пакета (например, словарь и средства орфографии для проверки правописания, построитель диаграмм, конвертер данных и др.);
- легкость обмена и ссылок на объекты, созданные программами интегрированного пакета, единообразный перенос объектов (метод drag-and-drop);
- наличие единой языковой платформы для разработки макрокоманд, пользовательских программ;
- возможность создания документов, интегрирующих в себе возможности различных программ, входящих в состав интегрированного пакета.

Интегрированные пакеты эффективны и при групповой работе в сети многих пользователей. Так, из прикладной программы, в которой находится пользователь, можно отправить документы и

файлы данных другому пользователю, при этом поддерживаются стандарты передачи данных в виде объектов по сети или через электронную почту.

## 8.8. Перспективы использования ЭВМ в различных отраслях промышленности

Рынок современных компьютеров отличается разнообразием и динамизмом, каких еще не знала ни одна область человеческой деятельности. Каждый год стоимость вычислений сокращается примерно на 25...30%, стоимость хранения единицы информации — до 40%. Практически каждое десятилетие меняется поколение машин, каждые два года — основные типы микропроцессоров — СБИС, определяющих характеристики новых ЭВМ. Такие темпы сохраняются уже многие годы.

Вместе с тем существует целый ряд закономерностей развития вычислительной техники, которые позволяют предвидеть и предсказывать основные результаты этого поступательного движения.

Существуют три глобальные сферы деятельности человека, которые требуют использования качественно различных типов ЭВМ.

Первое направление является традиционным — применение ЭВМ для автоматизации вычислений. Научно-техническая революция во всех областях науки и техники постоянно выдвигает новые научные, инженерные, экономические задачи, которые требуют проведения крупномасштабных вычислений (задачи проектирования новых образцов техники, моделирования сложных процессов, атомная и космическая техника и др.). Отличительной особенностью этого направления является наличие хорошей математической основы, заложенной развитием математических наук и их приложений. Первые, а затем и последующие вычислительные машины классической структуры в первую очередь и создавались для автоматизации вычислений.

Вторая сфера применения ЭВМ связана с использованием их в системах управления. Она родилась примерно в 1960-е гг., когда ЭВМ стали интенсивно внедряться в контуры управления автоматических и автоматизированных систем. Математическая база этой новой сферы практически отсутствовала, в течение последующих 15—20 лет она была создана. Новое применение вычислительных машин потребовало видоизменения их структуры. ЭВМ, используемые в управлении, должны были не только обеспечивать вычисления, но и автоматизировать сбор данных и распределение результатов обработки. Сопряжение с каналами связи потребовало усложнения режимов работы ЭВМ, сделало их многопрограммными и многопользовательскими. Для исключения взаимных по-



мех между программами пользователей в структуру машин были введены средства разграничения: блоки прерываний и приоритетов, блоки защиты и т. п. Для управления разнообразной периферией стали использоваться специальные процессоры ввода-вывода данных или каналы, появился дисплей как средство оперативного человеко-машинного взаимодействия пользователя с ЭВМ. Этой сфере работ в наибольшей степени отвечали мини-ЭВМ. Они стали использоваться для управления отраслями, предприятиями, корпорациями. В настоящее время использование мини-ЭВМ сокращается. Исчезает и термин мини-ЭВМ. На смену им приходят ЭВМ других типов; серверы, обеспечивающие диспетчерские функции в сетях ЭВМ, средние ЭВМ или старшие модели персональных ЭВМ. Одновременно со структурными изменениями ЭВМ происходило и качественное изменение характера вычислений. Доля чисто математических расчетов постоянно сокращалась, и в настоящее время она составляет около 10 % от всех вычислительных работ. Машины все больше стали использоваться для новых видов обработки: текстов, графики, звука и др.

Третье направление связано с применением ЭВМ для решения задач искусственного интеллекта. Задачи искусственного интеллекта предполагают получение не точного результата, а усредненного в статистическом, вероятностном смысле. Примеров подобных задач много: задачи робототехники, доказательства теорем, машинного перевода текстов с одного языка на другой, планирования с учетом неполной информации, составления прогнозов, моделирования сложных процессов и явлений и т. д. Это направление все больше набирает силу. Во многих областях науки и техники создаются и совершенствуются базы данных и базы знаний, экспертные системы. Для технического обеспечения этого направления нужны качественно новые структуры ЭВМ с большим количеством вычислителей (ЭВМ или процессорных элементов), обеспечивающих параллелизм в вычислениях. По существу, ЭВМ уступают место сложнейшим вычислительным системам.

Анализ областей применения ЭВМ показывает, что для решения различных задач нужна и различная вычислительная техника. Поэтому рынок компьютеров постоянно имеет широкую градацию классов и моделей ЭВМ. Можно предложить следующую классификацию средств вычислительной техники, используемых в различных сферах человеческой жизнедеятельности.

- СуперЭВМ для решения крупномасштабных вычислительных задач, обслуживания крупнейших информационных банков данных. С развитием науки и техники постоянно выдвигаются новые крупномасштабные задачи, требующие выполнения больших объемов вычислений. Особенно эффективно применение суперЭВМ при решении задач проектирования, в которых натурные эксперименты оказываются дорогостоящими, недоступными или прак-

тически неосуществимыми. В этом случае ЭВМ позволяет методами численного моделирования получить результаты вычислительных экспериментов, обеспечивая приемлемое время и точность решения, т. е. решающим условием необходимости разработки и применения подобных ЭВМ является экономический показатель «производительность/стоимость». Примером крупномасштабных задач следует считать задачу разработки новых схем СБИС для следующих поколений ЭВМ. СуперЭВМ позволяют точнее, быстрее и качественнее решать подобные задачи, обеспечивая необходимый приоритет в разработках перспективной вычислительной техники. Дальнейшее развитие суперЭВМ связывается с использованием направления массового параллелизма, при котором одновременно могут работать сотни и даже тысячи процессоров.

- Большие ЭВМ для комплектования ведомственных, территориальных и региональных вычислительных центров. Развитие ЭВМ данного класса имеет большое значение для России. В 1970—1990 гг. основные усилия нашей страны в области вычислительной техники были сосредоточены на программе ЕС ЭВМ (Единой системы ЭВМ), заимствовавшей архитектуру IBM 360/370. Было выпущено несколько десятков тысяч ЭВМ этой системы. Более пяти тысяч ЭВМ серии ЕС еще продолжают работать в различных учреждениях. Большинство АСУ верхнего уровня государственного управления в РФ (в силовых структурах, банках, на транспорте, связи и т. д.) оснащено этими машинами. Накоплен громадный программно-информационный задел, который следует рассматривать как элемент национального достояния (по стоимости) и элемент национальной безопасности (по стратегической значимости). Принято решение о дальнейшем развитии этого направления.

- Средние ЭВМ широкого назначения для управления сложными технологическими производственными процессами. ЭВМ этого типа могут использоваться и для управления распределенной обработкой информации в качестве сетевых серверов.

- Персональные и профессиональные ЭВМ, позволяющие удовлетворять индивидуальные потребности пользователей. На базе этого класса ЭВМ строятся автоматизированные рабочие места для специалистов различного уровня. IBM-совместимые компьютеры на платформе микросхем фирмы Intel составляют примерно 50 % рынка всей компьютерной техники.

- Встраиваемые микропроцессоры, осуществляющие автоматизацию управления отдельными устройствами и механизмами. Успехи микроэлектроники позволяют создавать миниатюрные вычислительные устройства, вплоть до однокристалльных ЭВМ.

Эти устройства, универсальные по характеру применения, могут встраиваться в отдельные машины, объекты, системы. Они находят все большее применение в бытовой технике (телефонах, телевизорах, электронных часах, микроволновых печах и т. д.), в го-



родском хозяйстве (энерго-, тепло-, водоснабжении, регулировке движения транспорта и т. д.), на производстве (робототехнике, управлении технологическими процессами). Постепенно они входят в нашу жизнь, все больше изменяя среду обитания человека.

Высокие скорости вычислений, обеспечиваемые ЭВМ различных классов, позволяют перерабатывать и выдавать все большее количество информации, что, в свою очередь, порождает потребности в создании связей между отдельно используемыми ЭВМ.

Поэтому все современные ЭВМ в настоящее время имеют средства подключения к сетям связи и комплексирования в системы.

Перечисленные типы ЭВМ, которые должны использоваться в индустриально развитых странах, образуют некое подобие пирамиды с определенным соотношением численности ЭВМ каждого слоя и набором их технических характеристик.

Требуемое число суперЭВМ для такой страны, как Россия, должно составлять 100... 200 шт., больших ЭВМ — тысячи, средних — десятки и сотни тысяч, ПЭВМ — миллионы, встраиваемых микроЭВМ — миллиарды. Все используемые ЭВМ различных классов образуют машинный парк страны, жизнедеятельность которого и его информационное насыщение определяют успехи информатизации общества и научно-технического прогресса страны. Формирование сбалансированного машинного парка является сложной политической, экономической и социальной проблемой, решение которой требует многомиллиардных инвестиций.

Для этого должна быть разработана соответствующая структура: создание специальных производств (элементной базы ЭВМ, программного обеспечения и технических связей), смена поколений машин и технологий, изменение форм экономического и административного управления, создание новых рабочих мест и т. д.

#### Контрольные вопросы

1. Охарактеризуйте основные этапы решения задач на ЭВМ.
2. Охарактеризуйте основные правила составления программ.
3. Каковы основы структурного проектирования алгоритмов и программ?
4. Что такое функциональная структура алгоритма?
5. Что такое модульное программирование?
6. Каковы свойства программного модуля?
7. Назовите управляющие структуры, применяемые в структурном программировании.
8. Что такое СУБД?
9. Охарактеризуйте интеллектуальные системы.
10. Что такое базы знаний?
11. Охарактеризуйте экспертные системы.
12. Что такое интегрированные системы?

## ЗАКЛЮЧЕНИЕ

Основным содержанием современного этапа развития науки и техники считается определяющая роль информации, которая становится стратегическим ресурсом. Инфраструктуру общества формируют способы и средства сбора, обработки, хранения и распределения информации, а затраты в сфере информатизации превысили затраты в других сферах деятельности. Информационные услуги стали товаром, приносящим значительные прибыли. Происходит быстрый рост доли трудовых ресурсов, занятых в информационной сфере. Это привело к феноменальному по масштабам и скорости прогрессу в области электронных средств компьютерной техники и средств передачи данных, массовому применению новых информационных технологий.

Огромные средства вкладываются развитыми странами мира в проведение исследований в области информатики и вычислительной техники, включая такие, как микро- и нанoeлектроника, оптоэлектроника, молекулярная электроника, проектирование интегральных схем, логика и логическое программирование, параллельные системы и системы реального времени, базы данных и информационные системы, методы и средства обработки информации, представление знаний и инженерия знаний, человеко-машинное взаимодействие, нейронные сети.

По уровню информатизации наша страна существенно отстает от ведущих и даже некоторых развивающихся стран мира. Объем производства средств вычислительной техники составляет ничтожную долю от уровня США и продолжает сокращаться, надежность отечественных электронных средств в сотни раз уступает зарубежным образцам и отстает от потребностей общества в развитии сетей связи и передачи данных, служащих базой информационной инфраструктуры регионов страны.

В то же время наши информационно-вычислительные системы обеспечивают вывод и слежение за орбитами спутников и много-разовых космических кораблей, осуществляют контроль космического, земного и морского пространства.

Это говорит о том, что во многих направлениях архитектурных и схемотехнических решений, системного программирования, математических методов решения прикладных задач наша наука и техника находятся на современном мировом уровне.

## ПРИЛОЖЕНИЯ

Приложение 1

## Характеристики зарубежных суперЭВМ

Производитель, тип	Год поставки	Производительность, млн опер/с <sup>1)</sup>	Максимальное число ЦП	$T_{цикла}$ , нс	Емкость ОЗУ <sup>2)</sup> , Мбайт	Тактовая скорость, млн цикл/с	Производительность/тактовая скорость	Архитектура
Cray-1	1976	160	1	12,5	8	80	2	Однопроцессорная векторно-конвейерная типа «регистр — регистр»
CDC Cyber 205	1981	400	1	20	64	50	8	Однопроцессорная векторно-конвейерная типа «память — память»
Denelkor НЕР	1981	160	16	100 <sup>3)</sup>	1024	10	16	Многопроцессорная скалярная с конвейеризацией
Cray X-MP/24	1983	470	2	8,5	32 <sup>4)</sup>	117,6	4	Многопроцессорная векторно-конвейерная типа «регистр — регистр»
Fujitsu VP-200	1983	570	1	7	256	142,9	4	Однопроцессорная векторно-конвейерная типа «регистр — регистр»
Hitachi S-810/20	1984	630	1	14	256	71,4	9	То же

Cray X-MP/416	1985	940	4	8,5	128 <sup>4)</sup>	117,6	8	Многопроцессорная векторно-конвейерная типа «регистр — регистр»
Fujitsu VP-400	1985	1140	1	7	256	142,9	8	Однопроцессорная векторно-конвейерная типа «регистр — регистр»
Cray 2/4-256	1985	1950	4	4,1	2048	243,9	8	Многопроцессорная векторно-конвейерная типа «регистр — регистр»
NEC SX-2	1985	1300	1	6	256	166,7	8	Однопроцессорная векторно-конвейерная типа «регистр — регистр»
IBM 3090-600E/VF	1987	696	6	17,2	256	58,1	12	Многопроцессорная скалярная с встроенными векторно-конвейерными устройствами VF
ETA-10G <sup>5)</sup>	1987	10285	8	7	2048	142,9	72	Многопроцессорная векторно-конвейерная типа «память — память»
Cray 2/4-512	1988	1950	4	4,1	4096	243,9	8	Многопроцессорная векторно-конвейерная типа «регистр — регистр»
Gray Y-MP/832	1988	2667	8	6	256	166,7	16	То же

Производитель, тип	Год поставки	Производительность, млн опер/с <sup>1)</sup>	Максимальное число ЦП	$T_{цикла}$ , нс	Емкость ОЗУ <sup>2)</sup> , Мбайт	Тактовая скорость, млн цикл/с	Производительность/тактовая скорость	Архитектура
Cray-3 <sup>6)</sup>	1990	16000	16	2	16384	500	32	Многопроцессорная векторно-конвейерная типа «регистр—регистр»
NEC SX-3	1990	22000	4	2,9	—	345	64	То же
Cray-4	1993	128000	64	1	—	1000	128	»
SS-1	1993	128000	64	—	—	—	—	Многопроцессорная векторно-конвейерная

<sup>1)</sup> Для ЭВМ IBM 3090-600 E/VE операции 32-разрядные, для всех других ЭВМ операции 64-разрядные. Для многопроцессорных ЭВМ дана суммарная пиковая производительность их центральных процессоров.

<sup>2)</sup> Ряд ЭВМ имеют также вторичную (расширенную) память (ЭВМ Cray X-MP, Hitachi S-810/20, NEC SX-2, IBM 3090-600E, Cray Y-MP оборудованы такой памятью емкостью 1...4 Гбайт).

<sup>3)</sup> Длительность цикла дана для одиночной команды.

<sup>4)</sup> Имеются также общие кластеры регистров для межпроцессорной связи.

<sup>5)</sup> Имеется общая память для многопроцессорной связи емкостью 8 Мбайт. Величина производительности — ориентировочная.

<sup>6)</sup> Применены арсенид-галлиевые схемы.

### Основные технические характеристики статических микросхем памяти

Тип	Информационная емкость, слов×разряд	Время выборки, не более, нс	Напряжение питания, (номинальное), В	Потребляемая мощность, не более, мВт	Технология	Синхронизация	Достоинства	Недостатки
M185PY7 K589PY01 KP185PY9	256×4 16×4 64×9	45 35 45	5 5 5	710 56 1050	ТТЛ ТТЛШ ТТЛ	— — —	Низкая стоимость изготовления, высокая помехоустойчивость и нагрузочная способность	Ограниченность быстродействия, создают помехи
K1500PY474 K1500PY480A	1024×4 16384×1	15 20	-4,5 -4,5	950 1100	ЭСЛ ЭСЛ	— —	Невысокая стоимость изготовления, хорошая помехоустойчивость, высокая нагрузочная способность	
KP53PY10  KP537PY14A, Б	2К×8  К×4	220  220	5  5	Хранение: 5,25; обращение: 370 Хранение: 0,135; обращение: 330	КМОП  КМОП	Асинхронная  Асинхронная	Низкая $P_{потр}$ в режиме хранения, малая задержка переключения, высокие помехоустойчивость и коэффициент разветвления	Зависимость от температуры, необходимость защиты от статического электричества



Тип	Информационная емкость, слов × ряд	Время выборки, не более, нс	Напряжение питания, (номинальное), В	Потребляемая мощность, не более, мВт	Технология	Синхронизация	Достоинства	Недостатки
KP537PY14A, Б	4K × 1	220	5	Хранение: 0,135; обращение: 230	КМОП	Асинхронная		
KP537PY17A, Б	8K × 8	220	5	Хранение: 22; обращение: 470	КМОП	Асинхронная		
KP537PY; А, Б	4K × 1	150	5	Хранение: 0,55; обращение: 30	КМДП	Синхронная		
KP537PY11A, Б	256 × 16	120	5		КМДП	Синхронная		
K6500PY1	1K × 1	4	4; -2,5	1,6	на GaAs	—	Высокое быстродействие, малая работа переключения, большой диапазон входных сигналов	Сложная технология производства, высокая стоимость, низкая теплопроводность

Основные технические характеристики динамических микросхем памяти

Тип	Информационная емкость, слов × ряд	Время выборки, ки, нс	Мощность потребления, мВт		Тип корпуса	Страна-изготовитель, фирма
			хранения	обращения		
K565PY1A	4K × 1	200	33	720	DIP	Россия
KP565PY6B	16K × 1	250	40	460	»	»
K565PY5B	64K × 1	120	22	250	»	»
MB81C4256A	256K × 4	80	75	600	DIP, SOJ	США
MB81C1000/1/2	1M × 1	80	75	600	»	»
NM571000JP	1M × 1	35/45	—	—	SOJ	Япония
2116	16K	150	—	720	—	Intel
TMS4070	16K	150	—	550	—	TI
MK4116	16K	120	—	600	—	Mostek
MCM6616	16K	250	—	500	—	Motrola
41256	256K × 4	60	—	—	—	GoldStar
41258	256K × 1	60	—	—	—	»
411000	1M × 1	60	—	—	—	NEC

Приложение 3

Основные технические характеристики ВИС РЛЗУ

Приложение 4

Тип	Информационная емкость, слов × ряд	Время считывания, мкс	Мощность потребления, мВт	Напряжение питания, (номинальное), В	Технология изготовления
558RP1	256 × 8	5	307	5; -12	МНОП
558RP23A	2K × 4	0,35	480	5	»
1601RP11	512 × 4	1,8	625	5; -12	»
1601RP31	1K × 8	1,6	850	5; -12	»
1609RP1	2K × 8	0,35	525	5	n-МОП
573RP2	2K × 8	0,35	620	5	»
573RP1	1K × 8	0,45	1100	5; -5; -12	ЛИЗМОП
573RФ3	4K × 16	0,45	450	5	n-МОП
573RФ5	2K × 8	0,45	580	5	»
573RФ6	8K × 8	0,3	870	5	»

## Общая характеристика представителей семейств универсальных микропроцессоров

Производитель, тип	Тактовая частота, МГц	Производительность на тестах SPEC-int92; -fr92 UNIX	Технология изготовления	Достоинства	Недостатки	Назначение
Digital Equipment, Alpha21164A	300	500; 700	КМОП 0,35 мкм	Для повышения тактовой частоты архитектура с 4-мя потоками и простыми маршрутами данных, RISC	Высокая стоимость, зависимость от качества компилятора	Рабочие станции, серверы NT и UNIX, параллельные суперЭВМ
SGI, Mips R 10000	200	300; 600	КМОП 0,35 мкм	64-разрядный МП с 5-ю функциональными конвейерами, на каждом такте может выполнять до 4-х команд	Вспомогательная КЭШ на дорогостоящей статической МС ОЗУ (SRAM)	Графические рабочие станции, SMP-серверы и суперЭВМ
Motorola, Power PC 620	133	225; 300	КМОП 0,5 мкм	Программное переключение между 64-разрядными и 32-разрядными режимами	Производительность ниже, чем у других RISC-процессоров	Графические рабочие станции, SMP-серверы, параллельные суперЭВМ и ПК Macintosh

Hewlett-Packard, PA-8000	200	360; 550	КМОП 0,5 мкм	RISC-процессор с расположенными вне его кэшами команд и данных до нескольких Мбайт	RISC-ядро дает неудовлетворительные показатели, если наборы данных больше размера кэша; кэши — на дорогостоящих сверхбыстрых SRAM	Коммерческая обработка данных, рабочие станции для инженерных и научных расчетов
Sun Microsystems, UltraSPARC II	250...300	350...420; 550...660	КМОП 0,3 мкм	64-разрядный суперскалярный МП для программ мультимедиа и сетевых приложений	Требует перекомпиляции старых программ	Рабочие станции мультимедиа
Intel, Pentium PRO	150; 200; 233; 333; 433	220; 215	БиКМОП 0,6 мкм, 0,35 мкм	Многокристалльный корпус с кэшем 256 (512) Кбайт для повышения скорости обмена. Архитектура с изменением порядка выполнения команд, конвейерная обработкой команд на высокой тактовой частоте	Сложность изготовления корпуса, потребляемая мощность и рассеивание тепла велики и не подходят для портативных ПК	Серверы и высокопроизводительные ПК, суперЭВМ

Производитель, тип	Тактовая частота, МГц	Производительность на тестах SPEC-int92; -fr92 UNIX	Технология изготовления	Достоинства	Недостатки	Назначение
ADM, K5	75 ... 166	109 ... 115; 109 ... 115	КМОП 0,35 мкм	Суперскалярная архитектура с ядром типа RISC, изменение порядка выполнения команд	Низкая тактовая частота	Массовые настольные ПК
Cyrix, 6 × 86	100; 120; 133; 150	176 ... 203; 176 ... 203	КМОП 0,6 мкм	Производительность выше, чем у Pentium с той же частотой при меньшей стоимости МП	Большой размер по сравнению с вариантами Pentium	Массовые ПК

### Характеристики медийных микропроцессоров

Тип	Особенности
Trimedia	Блоки ввода-вывода аудио/видео (DMA); интерфейсы с шиной PCI, цифровыми камерами, стерео-, аудиосистемами
Mediaprocessor	Оптимизирован для мультимедиа и широкополосных коммуникаций; тактовая частота 300 МГц... 1 ГГц; набор 32-разрядных команд с обработкой сигналов и расширенными математическими операциями; дополнительная кэш MediaBridge, микросхема ввода-вывода MediaCodec и интерфейсы ввода-вывода с пропускной способностью до 1 Гбит/с
Mpract Media	Кодирование/декодирование видео/аудио MPEG-1 в реальном времени; декодирование MPEG-2, синтез звука; поддержка видеоконференций (рекомендации H.320 (ISDN) и H.324)
NVI	MIDI, табличный синтез звука; ускорители трехмерной графики; до 32 16-разрядных аудиоканалов
MediaGX	Интегрирует функции видеоконтроллера и универсального процессора с системой команд × 86. Используется с сателлитной микросхемой Sx 5510, реализующей функции контроллера шин и UVB



## Характеристики серийных быстродействующих процессоров

Производитель, тип	Тактовая частота, МГц	Быстродействие, млн опер/с		Внутренняя память, Кбайт	Подключенная память, Гбайт	Интерфейс				Количество транзисторов, млн шт.	Потребляемая мощность, Вт
		с фиксированной запятой	с плавающей запятой			с памятью		с ВУ			
						ство	разрядность	количество	пропускная способность, Мбайт/с		
Inmos-Thomson, Транспьютеры:											
T 805	30	15	2	4	4	1	32	—	4	4	0,3
T 9000	50	200	25	16	4	1	64	—	4	25	3,2
Intel:											
860 XR	40	80	80	12	4	1	64	—	—	1,2	6
860 XP	50	15	100	32	4	1	64	—	—	2,6	8
Pentium	90	180	90	16	4	1	64	—	—	3,1	10
SUN-Texas Instruments, super spark 60	—	180	—	36	4	1	64	—	—	3,1	—
TMS 320-C40	40	270	50	8,5	64	2	32	6	20	1,1	—
TMS 320-C80	—	2000	5	64			64	—	—	4	—
Motorola-Apple: DSP-96002	40	200	60	8	4	2	32	—	—	0,7	—

88110	50	100	60	16	4	1	64	—	—	1,3	—
Power PC 601	80	160	80	32	—	1	64	—	—	—	12
DEC, Alpha 200	400	100	16	4	1	128	64	—	2	20	—
Analog Devices, ADSP 21060 (Sharc) 40	40	120	512	4	1	48	6	—	40	—	—

## Характеристики НЖМД типа винчестер

Производитель, тип	Емкость	Тактовая частота, МГц	Примечание
Seagate:			
ST1917 N/W/FC	9,1 Гбайт	—	Магнитно-резистивные головки
ST 42100N	1830 Мбайт	—	
ST 238R	30 Мбайт	—	Оксидное покрытие
ST34340A (Medalist 4340)	4,3 Гбайт	—	Время наработки на отказ 300 тыс.ч
ST 2777R-1	65 Мбайт	8...12	RLL-кодирование; $t_{\text{дост}} = 28$ мс
Core International:			
HC315	315 Мбайт	20	Интерфейс SCSI; $t_{\text{дост}} = 16$ мс
HC655	655 Мбайт	20	То же
HC1000	1 Гбайт	20	»
CDCC: Wren III	90 Мбайт	—	Для портативных ПК

Производитель, тип	Емкость	Тактовая частота, МГц	Примечание
Fujitsu: M2949 S/Q/R/E	9,1 Гбайт	—	10 дисков; 10 ошибок на 10 <sup>15</sup> считанных бит
IBM: Deskstar 16GP:			
DTTA 351010	10,1 Гбайт	—	1 ошибка на 10 <sup>15</sup> считанных бит
DTTA 351680	16,8 Гбайт	—	То же
Ultrastar 18ZX DRVS	18,2 Гбайт	—	Скорость вращения 10 тыс. об/мин

Приложение 9

## Основные технические характеристики магнитных лент

Производитель, тип	BASF			Seagate Tape Operations	
	Computer Tape	Tape Cartridge	Compusette	Travan DDS-3	TR-4
Емкость носителя	146 Мбайт	210 Мбайт	700 Кбайт	8 Гбайт	24 Гбайт
Продольная плотность записи, бит/дюйм	800...6250	38 000	800	—	—
Ширина носителя, мм	12,7	—	3,81	—	—
Длина носителя, м	730	—	500	—	—
Вид носителя	Бобина	Картридж	Миникассета	Мини-картридж	Картридж
Стандарты	MIL, DI IV, ISO, GSA	ECMA, ISO, ANSI		—	—
Примечание		Для НМЛ IBM 3480; до 50000 протяжек		Совместимость с Novell, Windows NT, Unix, Mac	

Приложение 10

## Основные характеристики записываемых оптических дисков

Производитель	Модель	Емкость, Мбайт	Время, мин	Размер, дюйм	Количество поверхностей	Технология	Примечание
BASF	CD-R 580MB	540... 640	63	5,25	1	Органический краситель с золотым рефлектором	
	CD-R 650MB	650... 800	74	5,25	1		
Fujitsu	128MB	128	—	3,5	—	MSR (Magnetically-induced Super Resolution)	
	230MB	230	74	3,5	—		
	540MB	540	74	3,5	—		
	640MB	640	74	3,5	—		
MEMOREX	1300MB	1300	—	3,5	—	Основание — поликарбонат, толщина 1,2 мм	
	DVD-R	3950	—	5	—		
	CD-R	700	80	5	—	—	Совместим с приводами CD-R, CD-RW, DWD
	CD-R	650	74	5	—	Основание — поликарбонат, толщина 1,2 мм	Стандарт Оранжевой книги II
	CD-RW	650	74	5	—	То же	Стандарт Оранжевой книги III Более 1000 циклов перезаписи

## Основные технические характеристики сканеров

Изготовитель, тип	Скорость сканирования	Разрешение, dpi	Разрядность цвета, бит	Размер документа, мм	Примечание
1. Планшетные					
HP ScanJet 3200C	2,5 мин	600	—	216×297	Pentium 586, ОЗУ от 16 Мбайт, интерфейс: параллельный порт для Windows 95 и 98
HP ScanJet 4100C	3...6 мс/строка	600	36	216×297	Pentium 586, ОЗУ от 16 Мбайт, интерфейс: порт USB
HP ScanJet 4200C	2 мин	600	36	216×297	То же
HP ScanJet 5200C	90 с	600	36	216×356	»
Mustek Paragon 1200 A3 Pro	—	600×1200	—	304×431	Windows xx, Mac OS, интерфейс: SCSI-II
Mustek Power Pro	—	1200×4800	—	211×297	То же
Mustek Scan Express 1200 CP	—	600×600	—	211×297	Windows xx, интерфейс: LTP(ECP/EPP)
Mustek A3 P	—	300×600	—	297×432	То же

Microtec ScanMaker E3	—	600×300	24	—	Windows xx, интерфейс: SCSI Адаптер, модуль для сканирования слайдов
Microtec ScanMaker E6	—	1200×600	30	—	То же
Olivetti-logabax PS-102	10 с	300	64 оттенка	—	
Olivetti-logabax PS-103	11 с	300	256 оттенков	—	

## 2. Листовые

Microtec PageWiz	38 с	300	—	A4	Windows 3.x/9x, интерфейс LTP
Microtec Color	38 с	300×600	—	A4	Windows 3.x/9x, интерфейс: SCSI/EPP
UMAX Astra 1220P	18 мс/линия	600×1200	36	A4	Параллельный порт SPP/EPP/ECP
UMAX Astra 2400S	4,2 с	600×2400	36	216×356	
UMAX PowerLooc III	—	1200×2400	42	216×297	
UMAX Mirage II и Iise	—	700×1400	36	290×432 1400×2800	



Изготовитель, тип	Скорость сканирования	Разрешение, dpi	Разрядность цвета, бит	Размер документа, мм	Примечание
<b>3. Ручной</b>					
HP CapShare 910	6 с	—	—	До А4	Имеет 4 Мбайт памяти на 50 страниц текста
<b>4. Барабанный</b>					
Mustek Paragon Page Express	—	300 × 300	—	216 × 1200	Windows xx, интерфейс: EPP
<b>5. Комбинированный — сочетание барабанного и планшетного</b>					
UMAX PowerLooc 3000	—	1220 × 3048 (объектив 1) 3048 × 3048 (объектив 2)	42	216 × 297 (объектив 1) 86 × 297 (объектив 2)	Для плакатов, рекламы

## Основные технические характеристики дигитайзеров

Производитель, тип	Число точек по координате		Разрешение, мм	Оцифровываемая площадь, мм <sup>2</sup>	Точность, мм	Скорость передачи	Интерфейс
	x	y					
Graphitec Corporation KD3200	2970	2100	0,1	297 × 210	0,025	150 позиций/с	RS-232C
KD3300	3050	3050	0,1	305 × 305	0,025	»	»
KD4300	3880	2600	0,1	380 × 260	0,025	»	»
KD4600	4600	3100	0,1	460 × 310	0,025	»	»
KD3800	3810	3810	0,1	381 × 381	0,025	»	»
Olivetti-logabax 9012	—	—	—	305 × 305	0,254	200 коорд/с	»
Olivetti-logabax 9236	—	—	0,1	610 × 914	0,254	»	»
Olivetti-logabax 9248	—	—	0,1	914 × 1219	0,254	»	»
Olivetti-logabax 9260	—	—	0,1	1118 × 1524	0,254	»	»
BENSON 6301	—	—	0,02	1300 × 870	0,005	100 точек/с	»
BENSON 6440	—	—	0,02	305 × 305	—	»	»
OCE Graphics G 6421	—	—	0,05	304,8 × 304,8	0,5	—	»
OCE Graphics G 6452	—	—	0,025	279,4 × 279,4	0,25	—	»
OCE Graphics G 6301	—	—	0,02	1300 × 870	0,1	—	»
АРИСТОГРИД CD 200	—	—	—	650 × 900 1000 × 1500	—	—	—
GGT AccuMarc 300	—	—	—	1115 × 1520	—	—	—

## Основные технические характеристики мониторов на базе ЭЛТ

Производитель, тип	Диагональ, дюйм	Тип ЭЛТ	Горизонтальный шаг теневой маски, мм	Частота развертки		Разрешение и частота сканирования кадров, рекомендуемая, точек на дюйм/Гц	Тип фокусирующей системы	Угол отклонения луча, °	Полоса пропускания видеосигнала, МГц	Примечание
				горизонтальной, кГц	вертикальной, Гц					
<b>I. HITACHI</b>										
CM641ET	17	Высококонтрастный	0,22	31 ... 95	50 ... 130	1280 × 1024/85	Эллиптическая и динамическая (EADF)	90	150	
CM650ET	17	Супервысококонтрастный укороченный	0,21	31 ... 69	50 ... 130	1024 × 768/85	Многошаговая динамическая (MDF)	100	110	
CM811ET	21	С усовершенствованным точечным шагом (EDP)	0,21	31 ... 96	50 ... 160	1280 × 1024/85	То же	100	200	Для графических и дизайнерских работ, в том числе web-дизайна
CM813ET	21	То же	0,21	31 ... 115	50 ... 160	1600 × 1280/85	»	100	250	Для обработки изображений



## II. MITSUBISHI ELECTRIC Diamond

Scan 70	17	Антибликовое, антистатическое покрытие	0,28	30 ... 70	50 ... 100	1280 × 1024/66	—	—	100	Экранное меню
Pro 900u NF	22	То же	0,25	30 ... 121	50 ... 160	1800 × 1440/80	—	—	240	*
Plus 100e	21	*	0,28	30 ... 108	50 ... 152	1600 × 1200/85	—	—	180	Фосфор с большим послесвечением

## III. SONY Multiscan

100EST	15	Trinitron укороченный	0,25	30 ... 70	50 ... 120	1280 × 1024/65	Корректирует четкость в углах	—	—	Интерфейс: DDC 1/2 В
400PST	19	То же	0,25	30 ... 94	48 ... 160	1600 × 1200/75	Линзовые системы	—	—	То же
F500	21	Плоский тринитрон	0,22	30 ... 121	48 ... 160	1800 × 1440/81	То же	—	—	Интерфейс: DDC 1/2B и 2AB, USB
W900	24	Повышенная контрастность и яркость	0,25	30 ... 96	50 ... 160	1920 × 200/6	—	—	—	Для САПР, видеомонтажа, анимации, многооконных приложений

Производитель, тип	Диагональ, дюйм	Тип ЭЛТ	Горизонтальный шаг теневой маски, мм	Частота развертки		Разрешение и частота сканирования кадров, рекомендуемая, точек на дюйм/Гц	Тип фокусирующей системы	Угол отклонения луча, °	Полоса пропускания видеосигнала, МГц	Примечание
				горизонтальной, кГц	вертикальной, Гц					
<b>IV. ViewSonic</b>										
E655	15	Плоский экран	0,28	30 ... 70	50 ... 100	1024 × 768/87	—	90	110	Экономическая серия
G790	19	Повышенная контрастность и четкость	0,26	30...95	50...180	1280 × 1024/88	—	90	200	Для графических и бизнес-приложений
PS790	19	Повышенная точность и яркость цветопередачи, укороченный	0,25	30...95	50 ... 180	То же	—	100	202,5	Профессиональная серия
<b>V. LG Electronics Studioworks</b>										
57M	15	—	0,28	30...70	50 ... 120	1024 × 768/85	Одинарная	—	110	Мультимедийный

78FT	17	Flatron	0,24	30...85	50...160	1280 × 1024/75	Двойная динамическая	—	135	То же
795FT plus	17	•	0,24	30...96	50...160	1600 × 1200/75	То же	—	135	Выбор восьми языков на экране

## Приложение 14

## Основные технические характеристики мониторов на базе плоских панелей (LCD)

Изготовитель, тип	Диагональ, дюйм	Технология	Частота развертки		Разрешение и частота обновления кадров	Угол обзора, °	Видеосигнал, мГц	Примечание
			горизонтальной, кГц	вертикальной, Гц				
<b>Настольные</b>								
<b>I. HITACHI</b>								
DT3140E	14,1	Жидкие кристаллы (LCD), активная матрица SuperTFT	24 ... 60	56 ... 75	1024 × 768/75 Гц	160	—	—
DT3180E	18	То же	24...80	60 ... 75	1280 × 1024/75 Гц	160	—	Низкие потребление энергии и тепловыделение



Изготовитель, тип	Диагональ, дюйм	Технология	Частота развертки		Разрешение и частота обновления кадров	Угол обзора, °	Видеосигнал МГц	Примечание
			горизонтальной, кГц	вертикальной, Гц				
<b>II. MITSUBISHI ELECTRIC</b>								
LCD 51	15	LCD	30...61	30...85	1024×768/75 Гц	—	80	Контрастность 400:1
LCD 80	18	»	30...61	30...85	1280×1024/75 Гц	—	80	Контрастность 230:1
<b>III. SONY</b>								
L150	15	TFT LCD	30...70	50...85	0,297 мм 1024×768/85 Гц	140	—	Контрастность 300:1; может быть прикреплен к стене
<b>IV. ViewSonic</b>								
VPA145	14,5	LCD	30...62	50...75	1024×768	100	82	Мультимедийный. Толщина 6,7", масса 5,5 кг
<b>V. LG Electronics Studioworks</b>								
500LC	15,1	TFT LCD	31...61	56...75		120	—	Пиксель 0,3×0,3 мм
<b>VI. Silicon Graphics</b>								
1600SW	17,3	LCD	—	—	1600×1024	100	—	Контрастность 350:1. Для профессиональной работы с графикой

## Портативные

## VII. Белый ветер ДВМ

	6,1	»			640×480		—	В ноутбуке: Partner AT
	13,3	»			1024×768		—	Voyager MT5
	14,1	»			»		—	Explorer FT6
	15,1	»			»			
<b>VIII. IBM</b>								
	11,3	»			800×600		—	IBM Think Pad 365XD
	12,1	»			1024×768		—	IBM Think Pad 760 ED
	11,3	LCD			800×600		—	IBM Think Pad 2640—100

Приложение 15

## Основные технические характеристики графопостроителей

Производитель, тип	Скорость вычерчивания, мм/с	Размеры рабочего поля, мм	Шаг пишущего узла, мм	Количество головок записи	Тип рабочего инструмента	Примечание
Рулонные						
<b>I. HOUSTON INSTRUMENT</b>						
EDMP-GONP	800	A4, A3	0,0127	8	—	

Производитель, тип	Скорость вычерчивания, мм/с	Размеры рабочего поля, мм	Шаг пишущего узла, мм	Количество головок записи	Тип рабочего инструмента	Примечание
EDMP-62	600	A4...A0	0,127	1...6	—	
<b>II. BENSON</b>						
1645-R	600	930	0,0125	8	Шариковый фломастер, трубчатые перья	Встроенный интеллект: генерация дуг, пунктиров, наклон и утолщение знаков, штриховка, закрапка
1642-SB	500	625	0,025	8	То же	То же
1122	150	340	0,0125	3	Перья	Двигатели постоянного тока с автоматической регулировкой оптически кодирующими устройствами
1332	250	940	0,0125	3	»	То же
<b>III. GERBER-marker</b>						
AccuPlot 700	2300	202	—	1	»	Пневмоперьевая технология

<b>IV. Oce Graphics</b>						
G1820 Range	760	A4...A0	0,0125	8	Перья	Автоматический контроль за перьевой головкой
Планшетные						
<b>V. BENSON</b>						
222	75	1200×840	0,05	4	»	Носители: бумага, калька, полиэфирный лист Цикл подъема/опускания пера 30 мс. Встроенный интеллект, интерактивные функции
254	250	1600×1200	0,025	4	»	
1455	500	1400×2000	0,0125	—	Шариковый фломастер, трубчатые перья	
<b>VI. ARISTO</b>						
АРИСТОМАТ						
205 S	—	1200×1500	—	4	—	Устройство Зензотроник инк-крайбер — регулятор потока туши
<b>VII. SEIKO INSTRUMENTS &amp; ELECTRONICS LTD</b>						
XP-100	1000	630×1000	0,010	4	—	Головка передвигается на воздушной подушке
XP-130	1000	1580×2980	0,010	4	—	То же

Производитель, тип	Скорость вычерчивания, мм/с	Размеры рабочего поля, мм	Шаг пишущего узла, мм	Количество головок записи	Тип рабочего инструмента	Примечание
Барабанные						
VIII. BENSON						
15.1565	800	1189 × 841	0,0125	4	Шариковый фломастер, трубчатые перья, трубчатые под давлением	Занимает малую площадь

## СПИСОК ЛИТЕРАТУРЫ

1. Аванесян Г. Р., Левшин В. П. Интегральные микросхемы ТТЛ, ТТДШ: Справочник. — М.: Машиностроение, 1993.
2. Брамм П., Брамм Д. Микропроцессор 80386 и его программирование: Пер. с англ. — М.: Мир, 1990.
3. Бурцев В. С. О необходимости создания суперЭВМ в России // Информационные технологии и вычислительные системы. — 1995. — № 1.
4. Головкин Б. А. Вычислительные системы с большим числом процессоров. — М.: Радио и связь, 1995.
5. Гореликов С. Х. IBM PC. Дисковая система: контроллеры, накопители и их обслуживание. — М.: Изд-во АО «Звезды и С», 1993.
6. Гук М. Аппаратные средства IBM PC. — СПб.: Питер, 1997.
7. Жаров А. «Железо» IBM. — М.: МИКРОАРТ, 1996.
8. Затуливетер Ю. А. Компьютерные архитектуры: Неожиданные повороты // HARD'n SOFT. — М., 1996. — № 2.
9. Иванников А. Д. Моделирование микропроцессорных систем. — М.: Энергоатомиздат, 1990.
10. Каган Б. М. Электронные вычислительные машины и системы. — М.: Энергия, 1987.
11. Интерфейсы систем обработки данных: Справочник / Под ред. А. А. Мячева. — М.: Радио и связь, 1989.
12. Информатика / Под ред. Н. В. Макаровой. — М.: Финансы и статистика, 1998.
13. Корриган Дж. Компьютерная графика: секреты и решения: Пер. с англ. — М.: ЭНТРОП, 1995.
14. Коутс Р., Влейминк И. Интерфейс «человек-компьютер». — М.: Мир, 1990.
15. Ларионов А. М., Горнец Н. Н. Периферийные устройства в вычислительных системах. — М.: Высш. шк., 1991.
16. Манпиль Л. И. Устройства регистрации графической информации ЭВМ и систем: Учеб. пособие для вузов. — М.: Энергоатомиздат, 1995.
17. Морисита И. Аппаратные средства микроЭВМ: Пер. с япон. — М.: Мир, 1988.
18. Мячев А. А. Мини- и микроЭВМ систем обработки информации: Справочник. — М.: Энергоатомиздат, 1991.
19. Мячев А. А. Персональные ЭВМ: Краткий энциклопедический справочник. — М.: Финансы и статистика, 1992.
20. Мячев А. А., Степанов В. Н. Персональные ЭВМ и микроЭВМ. Основы организации / Под ред. А. А. Мячева. — М.: Радио и связь, 1991.
21. Нешумова К. А. Электронные вычислительные машины и системы. — М.: Высш. шк., 1989.



## ОГЛАВЛЕНИЕ

22. *Огнев И. В., Борисов В. В.* Интеллектуальные системы ассоциативной памяти. — М.: Радио и связь, 1996.
23. *Пескова С. А., Гуров А. И., Кузин А. В.* Центральные и периферийные устройства электронных вычислительных средств. — М.: Радио и связь, 2000.
24. *Пятибратов А. П., Гудыно Л. П., Кириченко А. А.* Вычислительные системы, сети и телекоммуникации. — М.: Финансы и статистика, 1998.
25. *Савета Н. Н.* Периферийные устройства ЭВМ. — М.: Высш. шк., 1987.
26. СуперЭВМ. Аппаратная и программная организация / Под ред. С. Фернбаха: Пер. с англ. — М.: Радио и связь, 1991.
27. *Титтел Э., Хадсон К., Стюарт Дж. М.* Networking Essentials. Сертификационный экзамен — экстерном (экзамен 70-058). — СПб.: Питер Ком, 1999.
28. Толковый словарь по вычислительным системам / Под ред. В. Иллингворта и др.: Пер. с англ. — М.: Машиностроение, 1989.
29. *Фергусон Дж., Макари Л., Уильямз П.* Обслуживание микропроцессорных систем: Пер. с англ. — М.: Мир, 1989.
30. *Фрир Дж.* Построение вычислительных систем на базе перспективных микропроцессоров: Пер. с англ. — М.: Мир, 1990.
31. *Чернега В. С., Василенко В. А., Бондарев В. Н.* Расчет и проектирование технических средств обмена и передачи информации: Учеб. пособие для вузов. — М.: Высш. шк., 1990.
32. *Шевкопляс Б. В.* Микропроцессорные структуры. Инженерные решения. Дополнение первое: Справочник. — М.: Радио и связь, 1993.
33. *Шевкопляс Б. В.* Микропроцессорные структуры. Инженерные решения. — М.: Радио и связь, 1990.

Предисловие .....	3
<b>Глава 1. Основные сведения о микропроцессорах и микроЭВМ .....</b>	<b>4</b>
1.1. Классификация ЭВМ .....	4
1.2. Базовые параметры и технические характеристики ЭВМ .....	6
1.3. Архитектура вычислительной системы. Структура аппаратной части и назначение основных функциональных узлов ЭВМ .....	9
<b>Глава 2. Информационно-логические основы ЭВМ .....</b>	<b>14</b>
2.1. Системы счисления .....	14
2.1.1. Перевод целых чисел из десятичной системы счисления в двоичную .....	17
2.1.2. Перевод дробных чисел .....	17
2.1.3. Представление информации в ЭВМ .....	18
2.2. Арифметические основы ЭВМ .....	21
2.2.1. Машинные коды .....	21
2.2.2. Арифметические операции над числами с фиксированной точкой .....	22
2.2.3. Арифметические операции над двоичными числами с плавающей точкой .....	25
2.3. Логические основы ЭВМ .....	27
2.3.1. Основные сведения из алгебры логики .....	27
2.3.2. Законы алгебры логики .....	33
2.3.3. Понятие минимизации логических функций .....	34
2.3.4. Техническая реализация логических функций .....	42
<b>Глава 3. Типовые логические элементы и устройства ЭВМ .....</b>	<b>45</b>
3.1. Классификация элементов и устройств ЭВМ .....	45
3.2. Типовые функциональные узлы комбинационных логических устройств .....	45
3.3. Цифровые автоматы (триггеры, регистры, счетчики) .....	59
3.4. Запоминающие устройства ЭВМ .....	66
3.5. Организация безадресной и виртуальной памяти .....	77
<b>Глава 4. Структура и функционирование процессора .....</b>	<b>96</b>
4.1. Классификация и типовая структура микропроцессоров .....	96
4.2. Микропроцессоры с «жестким» и программируемым принципами управления .....	99
4.3. Рабочий цикл процессора. Микропрограммная интерпретация команд центрального процессора .....	102
4.4. Микроконтроллеры. Особенности организации однокристалльных и секционных микропроцессоров .....	106
<b>Глава 5. Периферийные устройства ЭВМ, методы и средства сопряжения, сети ЭВМ .....</b>	<b>111</b>
5.1. Классификация периферийных устройств .....	111

5.2. Внешние запоминающие устройства на магнитных, оптических, магнитооптических носителях .....	116
5.3. Устройство ввода данных — клавиатура .....	128
5.4. Устройства ввода изображений — дигитайзеры, сканеры .....	130
5.5. Устройства ввода-вывода речевой информации .....	138
5.6. Устройства отображения информации — дисплеи .....	146
5.7. Устройства вывода информации — принтеры и графопостроители .....	155
5.8. Функции и типы интерфейсов. Способы организации связи между МП и устройством ввода-вывода .....	161
5.9. Автоматические устройства ввода-вывода аналоговой информации в ЭВМ .....	176
5.10. Каналы ввода-вывода и аппаратура сопряжения .....	191
5.11. Сети ЭВМ, назначение, характеристики .....	207
<b>Глава 6. Тенденции развития архитектуры и аппаратного обеспечения ЭВМ .....</b>	<b>227</b>
6.1. Требования различных задач к вычислительным ресурсам и ограничения фон неймановской архитектуры .....	227
6.2. Распараллеливание процессов обработки информации .....	228
6.3. Принцип совмещения операций. Конвейерная обработка информации .....	229
6.4. Архитектура процессоров с сокращенным набором команд .....	232
6.5. Развитие новых архитектурных принципов .....	234
<b>Глава 7. Программное (математическое) обеспечение ЭВМ .....</b>	<b>237</b>
7.1. Операционная система, ее назначение, состав и функции .....	237
7.2. Организация файловой структуры и методы доступа к файлам .....	240
7.3. Основные команды операционной системы .....	244
7.4. Графическая операционная система Windows и ее особенности .....	248
7.5. Многозадачный режим работы Windows .....	250
7.6. Основные объекты оконного интерфейса: окна, меню, ярлыки, папки .....	251
<b>Глава 8. Алгоритмизация и основы программирования .....</b>	<b>254</b>
8.1. Основные этапы решения задач на ЭВМ .....	254
8.2. Правила составления программ .....	258
8.3. Структура программы. Отладка программы .....	259
8.4. Подпрограммы и встроенные функции .....	260
8.5. Понятие о структурном программировании .....	261
8.6. Понятие об объектно-ориентированном программировании .....	262
8.7. Понятие о базах данных, базах знаний, экспертных системах, интегрированных системах .....	265
8.8. Перспективы использования ЭВМ в различных отраслях промышленности .....	269
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>273</b>
<b>Приложения .....</b>	<b>274</b>
<b>Список литературы .....</b>	<b>301</b>