

ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ

Ю. М. Келим

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

9-е издание

ЭЛЕКТРОНИКА
И МИКРОЭЛЕКТРОНИКА

ACADEM'A

УЧЕБНИК



Ю. М. КЕЛИМ

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Учебник

Рекомендовано

*Федеральным государственным автономным учреждением
«Федеральный институт развития образования» (ФГАУ «ФИРО»)
в качестве учебника для использования в учебном процессе
образовательных учреждений, реализующих программы
среднего профессионального образования*

*Регистрационный номер рецензии 771
от 26 декабря 2012 г. ФГАУ «ФИРО»*

9-е издание, стереотипное



Москва
Издательский центр «Академия»
2014

УДК 681.3(075.32)
ББК 32.97я723
К341

Рецензент —

доцент кафедры «Автоматизация и информационные технологии» МТУСИ,
канд. техн. наук *А. А. Андрюков*

Келим Ю. М.

К341 Вычислительная техника : учебник для для студ. учреждений сред. проф. образования / Ю. М. Келим. — 9-е изд., стер. — М. : Издательский центр «Академия», 2014. — 368 с.
ISBN 978-5-4468-1399-5

Изложены сведения об электронной вычислительной технике: классификация, характеристики, принцип действия цифровых вычислительных машин (компьютеров); виды информации и способы представления ее в ЭВМ; системы счисления, логические основы ЭВМ; основы микропроцессорных систем (архитектура микропроцессора и ее элементы, система команд микропроцессора, процедура выполнения команд, рабочий цикл микропроцессора); типовые узлы и устройства вычислительной техники (регистры, дешифраторы, счетчики, сумматоры); принципы построения и классификация устройств памяти; организация интерфейсов, периферийные устройства вычислительной техники; взаимодействие аппаратного и программного обеспечения в работе ЭВМ; основы программирования. Рассмотрены компьютерные сети, коммуникационные возможности компьютеров и основы работы на персональном компьютере.

Учебник может быть использован при изучении общепрофессиональной дисциплины «Вычислительная техника» в соответствии с ФГОС СПО для групп специальностей по направлениям «Электронная техника, радиотехника и связь», «Автоматизированные технологии и производства».

Для студентов образовательных учреждений среднего профессионального образования.

УДК 681.3(075.32)
ББК 32.97я723

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

© Келим Ю. М., 2012
© Образовательно-издательский центр «Академия», 2012
ISBN 978-5-4468-1399-5 © Оформление. Издательский центр «Академия», 2012

ПРЕДИСЛОВИЕ

Вычислительная техника в настоящее время представляет собой наиболее динамичную и быстро развивающуюся область техники, которая затронула практически все виды человеческой деятельности.

На работе, в учебных заведениях, на отдыхе, в быту — везде в настоящее время используются компьютеры и другие средства вычислительной техники.

На использовании компьютеров основаны информационные и коммуникационные технологии, без которых уже немыслима жизнь современного человека. Особенно важно иметь глубокие знания по вычислительной технике специалистам в области автоматизации.

В процессе развития человеческого общества было создано великое множество разнообразных устройств для удовлетворения самых разных потребностей человека, для обеспечения его жизни. Сюда следует отнести различные постройки жилого и производственного назначения; машины для обработки материалов (рабочие машины); машины для получения энергии (энергетические машины); машины для обработки и передачи информации (информационные машины); транспортные средства для перемещения по суше, воде, воздуху и в космосе. Перечислить все созданные человечеством устройства не представляется возможным. Обобщенно все они могут быть названы техническими устройствами, или кратко обозначены понятием «техника». *Техника* — это совокупность средств человеческой деятельности, создаваемых для осуществления процессов производства и обслуживания производственных потребностей общества.

Вычислительная техника в настоящее время используется не только для выполнения вычислений с огромной скоростью, она в еще большей степени нужна для обработки и поиска информации, для передачи информации на огромные расстояния. Поэтому вычислительная техника относится к информационным машинам.

Со времени первого издания этой книги прошло более трех лет. За это время произошли большие изменения в вычислительной технике. Прежде всего существенно улучшились технические характеристики компьютеров: во много раз увеличилась их производительность, стал больше объем памяти, появилось новое про-

граммное обеспечение. Надо также отметить, что подобная техника все шире применяется для связи, т.е. для коммуникации. Поэтому все чаще те устройства, которые носили общее название вычислительная техника, теперь называются информационно-коммуникационная техника.

Эти изменения нашли свое отражение в 4-м издании книги. Поскольку основные принципы работы вычислительной (информационно-коммуникационной) техники остались прежними, то построение книги изменилось мало.

Учебник состоит из четырех разделов.

В разделе I рассмотрены назначение вычислительной техники, ее классификация и характеристики, принцип действия ЭВМ и способы представления информации в них.

В разделе II изложены математические и логические основы работы ЭВМ.

Раздел III посвящен элементам и устройствам, из которых состоит ЭВМ, их взаимодействию в процессе работы. В нем рассмотрены типовые элементы вычислительной техники (логические элементы, триггеры, регистры, счетчики, сумматоры, различные преобразователи, устройства сравнения и коммутирующие элементы, а также элементы аналоговых вычислительных машин). Большое внимание уделено основам микропроцессорных систем, арифметико-логическим устройствам процессора, устройствам управления процессора, работе микропроцессора, запоминающим устройствам, организации интерфейсов в вычислительной технике, периферийным устройствам.

В разделе IV рассмотрены операционные системы, программное обеспечение и основы программирования. Отдельные главы посвящены принципам построения компьютерных сетей и основам работы на персональном компьютере. Добавлен подраздел о коммуникационных возможностях компьютерной техники.

Автор выражает благодарность своим многочисленным коллегам, аспирантам и студентам Московского технического университета связи и информатики, оказавшим ему помощь в работе над книгой своими советами и участием в обсуждении. После выхода в свет трех изданий этой книги многие читатели прислали свои замечания и предложения. Всем им автор выражает свою благодарность, а их замечания были учтены при работе над 4-м изданием.

Автор будет благодарен всем читателям, которые пришлют свои замечания и предложения на его электронный адрес yuri.kelim@gmail.com

ВВЕДЕНИЕ

Понятие «вычислительная техника» (ВТ) имеет два значения. Во-первых, это область техники, объединяющая средства автоматизации математических вычислений и обработки информации в различных сферах человеческой деятельности. Во-вторых, это наука о принципах построения, действия и проектирования этих средств.

Знание технических и научных основ вычислительной техники жизненно необходимо специалисту в XXI в.

Вычислительная техника — наиболее бурно развивающаяся область техники. От единичных экземпляров в середине XX в. до более чем миллиарда различных устройств и средств ВТ в наши дни — таковы темпы и масштабы ее развития.

Электронные вычислительные машины (ЭВМ) появились чуть более полувека тому назад. За это время их размеры уменьшились в тысячи раз, а производительность увеличилась в миллионы раз. Если в первые 10 лет своего развития ЭВМ создавались из отдельных (дискретных) элементов, то затем научные и технологические достижения микроэлектроники позволили в одном элементе (т. е. в одном корпусе) размещать сразу несколько полупроводниковых компонентов.

От одного транзистора в корпусе до десятков миллионов — в современном микропроцессоре. Никакая другая область техники, кроме космической, не может похвастаться такими бурными, поистине революционными изменениями за последние полвека.

Впрочем своими успехами космическая техника в немалой степени обязана вычислительной технике. Ведь без бортовых ракетных и спутниковых компьютеров и мощнейших ЭВМ на станциях наземного слежения и управления космическими объектами развитие современной космической техники было бы невозможно.

Будет справедливым отметить, что космонавтика не осталась в долгу. Именно в условиях невесомости на искусственных спутниках Земли были получены сверхчистые материалы, необходимые для создания полупроводниковых элементов микроэлектроники.

Помимо быстрого развития средств вычислительной техники надо отметить и существенные изменения в сфере ее приложе-

ния. Создававшиеся именно как средства для облегчения вычислений ЭВМ уже давно используются в большей степени как средства обработки информации в самом широком смысле этого слова.

Все, что может узнать человек благодаря своим пяти органам чувств, да то, что он может узнать благодаря самым разнообразным измерительным методам и приборам (а они чувствительны и к тем явлениям, которые человек не воспринимает), ЭВМ может обработать (т.е. преобразовать) и запомнить. Поэтому в настоящее время ЭВМ — это прежде всего средство обработки информации.

В качестве такового ЭВМ является основой современных информационных технологий, с помощью которых обеспечивается ускоренное развитие самых разных областей человеческой деятельности.

Широкое применение ЭВМ (прежде всего персональных компьютеров) — это то новое, что меняет образ жизни всего человечества.

В обиходе начинают использоваться новые слова, а некоторые прежние меняют свое содержание. Поэтому необходимо уточнить некоторые термины и понятия, используемые в вычислительной технике.

В нашей стране действует специальный стандарт на термины и определения в области систем обработки информации (ГОСТ 15971 — 90). Согласно этому стандарту *вычислительная машина* (ВМ) — это совокупность технических средств, создающая возможность проведения обработки информации и получение результата в необходимой форме.

Электронная ВМ — это вычислительная машина, основные функциональные устройства которой выполнены на электронных компонентах.

Совокупность технических средств и программного обеспечения, а также методов обработки информации и действий персонала, обеспечивающая выполнение автоматизированной обработки информации, составляет *систему обработки информации*. *Обработка информации* — это систематическое выполнение операций над данными, представляющими предназначенную для обработки информацию.

Понятие «данные» является очень важным в вычислительной технике. *Данные* — это информация, представленная в виде, пригодном для обработки автоматическими средствами при возможном участии человека.

Следует также различать понятия «автоматический» и «автоматизированный». Первое понятие означает, что действие происходит без участия человека. Второе понятие используется для обозначения совместных действий автоматических устройств и человека.

Таким образом, в систему обработки информации включаются автоматические устройства (ЭВМ), а также человек — пользователь (или оператор) ЭВМ и человек — разработчик программ, которые управляют работой ЭВМ.

Существуют различные ЭВМ, которые принято подразделять на классы. С точки зрения производительности существующие ЭВМ можно подразделить на следующие четыре класса:

1) суперЭВМ — ЭВМ, относящаяся к классу вычислительных машин, имеющих самую высокую производительность, которая может быть достигнута на данном этапе развития технологии, и в основном предназначенная для решения сложных научно-технических задач;

2) ЭВМ общего назначения — ЭВМ, относящаяся к классу вычислительных машин, занимающих на шкале производительности широкий диапазон и предназначенных для решения большого круга задач. В литературе встречается и иное, устаревшее наименование такой машины: универсальная ЭВМ. В английском языке такую машину называют *mainframe computer*, или коротко *mainframe*. Поэтому и в нашей научно-технической и популярной литературе все чаще применяется слово «мэйнфрейм»;

3) мини-ЭВМ — это ЭВМ, относящаяся к классу вычислительных машин, разработанных исходя из требования минимизации стоимости и предназначенных для решения достаточно простых задач. Как правило, мини-ЭВМ устанавливаются в обычных рабочих помещениях;

4) микроЭВМ — ЭВМ, относящаяся к классу вычислительных машин, центральная часть которых построена на одном или нескольких микропроцессорах, разработанных исходя из требования минимизации физического объема. К микроЭВМ относится персональная ЭВМ, или персональный компьютер (ПК). Стандарт определяет ПК как настольную ЭВМ, имеющую эксплуатационные характеристики бытового прибора и универсальные функциональные возможности.

По принципу физической формы представления обрабатываемой информации различают аналоговые, цифровые и аналого-цифровые (гибридные) средства ВТ. В аналоговой ВТ обработке подвергаются физические величины (токи, напряжения и др.), которые в определенном непрерывном диапазоне моделируют математические величины. В цифровых средствах ВТ обработке подвергаются цифровые (дискретные) коды математических величин.

По степени универсальности обработки информации средства ВТ подразделяются на машины общего назначения (универсальные) и специализированные. Первые служат для решения широкого класса задач, вторые — для решения узкого класса или даже единственной задачи.

По степени автоматизации обработки информации различают вычислительные инструменты (линейки, счеты и т.п.), приборы (планиметры, арифмометры и т.п.) и машины.

На современном этапе развития ВТ широко используются вычислительные машины и их комплексы. Кроме настольного ПК получил большое распространение ноутбук — компьютер в виде небольшой папки или портфеля. Питается он от аккумулятора, поэтому работать на нем можно где угодно. Появились карманные ПК. Некоторыми способностями компьютеров теперь обладают и сотовые телефоны (их обычно называют смартфонами, т.е. разумные телефоны).

За последние 15—20 лет сотни миллионов компьютеров объединились в сети, что еще более расширило область применения ВТ. Возросла роль ЭВМ как средства связи.

С помощью компьютеров стало возможным пересылать информацию на тысячи километров за доли секунды, с помощью электронной почты — тексты и картинки, с помощью интернет-телефонии — голосовые сообщения. Теперь можно с помощью компьютера не только разговаривать с далеким собеседником, но и видеть его лицо.

Таким образом, стали использоваться и коммуникационные возможности компьютеров, возникли новые информационно-коммуникационные технологии (ИКТ).

РАЗДЕЛ I

ОСНОВНЫЕ СВЕДЕНИЯ ОБ ЭЛЕКТРОННОЙ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ

Глава 1

НАЗНАЧЕНИЕ И РАЗВИТИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

1.1. Краткий исторический очерк развития вычислительной техники

В истории вычислительной техники можно выделить три не равные по длительности этапа. Первый этап — от глубокой древности до появления первых электронных вычислительных машин. В это время создавались различные устройства и методы, облегчавшие счет, вычисления. Началом второго этапа явилось создание в середине XX в. первой цифровой вычислительной машины, работавшей на основе тех же принципов, что и современные компьютеры. Вычислительная техника на этом этапе еще не стала массовой. Третий этап, начавшийся в 1980-х годах с создания персонального компьютера, продолжается в настоящее время. Вычислительная техника получила массовое распространение. Так же, как в прошлом цивилизованный человек был обязан уметь читать и писать, современный человек обязан владеть навыками использования вычислительной техники. Рассмотрим подробнее эти три этапа.

Первый этап. История развития счета уходит в глубь тысячелетий. Люди в древности (так же, как, порой, и мы) при подсчете загибали пальцы на руках. Отсюда, видимо, и пошла десятичная система счисления, т.е. подсчет десятками. Для больших чисел пальцев не хватало, и люди стали придумывать различные вычислительные приборы. В Древнем Египте на песке проводили полосы, а на них раскладывали камешки. Подобный принцип записи чисел был использован в самом первом из известных вычислительных приборов, который носит название *абак*.

Абак использовался очень широко в древней Греции и Римской империи. Он представлял собой доску, разделенную на ряд вертикальных полос. В каждой полосе раскладывалось столько камешков, сколько ими хотели обозначить единиц в соответствующем разряде. Позднее камешки были заменены жетонами, на которых стали писать цифры. Разновидность абака — проволочки с бусинками на них, которые при подсчете передвигались. От по-

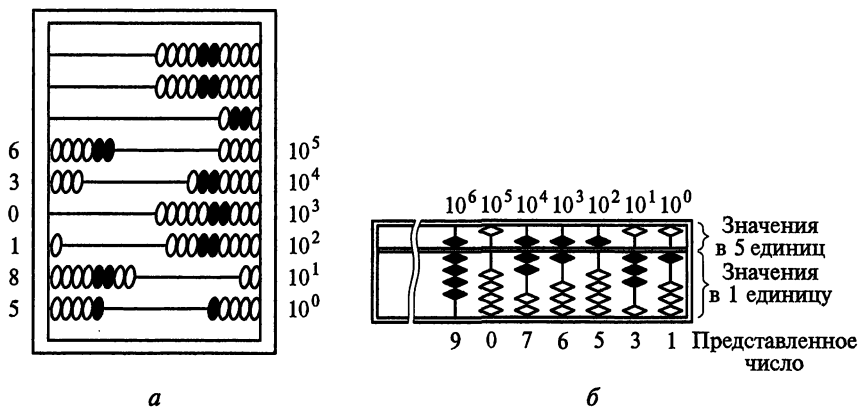


Рис. 1.1. Ручные вычислительные устройства:

a — русские счеты; *б* — японские счеты

явления песчаного абака до создания абака с бусинками прошло более трех тысячелетий. А от абака с бусинками недалеко и до сохранившегося до наших дней (хотя уже практически вышедшего из применения) устройства, которое называется *счетами*. И абак, и счеты — это вычислительные приборы для выполнения сложения и вычитания (с помощью некоторых приемов на них можно выполнять и другие арифметические действия). В России счеты (рис. 1.1, *a*) были известны с XIV в. В Китае и Японии тоже пользуются счетами, но они отличаются от наших (рис. 1.1, *б*).

Примерно с XVI в. ускорилось развитие науки и техники, что в значительной степени было связано с изобретением книгопечатания и, следовательно, с многократным увеличением информации и скорости ее распространения. В то же время были разработаны методы и устройства для облегчения вычислений. Например, изобретенные Джоном Непером палочки (рис. 1.2) служили для облегчения умножения и деления. Они представляли собой пластинки прямоугольной формы из кости или дерева. На них записывались произведения каждого числа от 1 до 9 на 1, 2, ..., 8 и 9, причем единицы стояли под десятками и были сдвинуты вправо.

С помощью таких пластинок легко выполнить умножение любых чисел. Например, положим рядом пластинки чисел 4, 7, 3, 8, 9, 7 таким образом, чтобы результаты умножения каждого из этих чисел на 6 находились в одной строке (на одной горизонтали). Тогда произведение числа 473 897 на 6 получится в таком виде:

$$\begin{array}{cccccc}
 2 & 4 & 1 & 4 & 5 & 4 \\
 4 & 2 & 8 & 8 & 4 & 2
 \end{array}$$

0	1	0	2	0	3	0	4	0	5	0	6	0	7	0	8	0	9
0	2	0	4	0	6	0	8	1	0	1	2	1	4	1	6	1	8
0	3	0	6	0	9	1	2	1	5	1	8	2	1	2	4	2	7
0	4	0	8	1	2	1	6	2	0	2	4	2	8	3	2	3	6
0	5	1	0	1	5	2	0	2	5	3	0	3	5	4	0	4	5
0	6	1	2	1	8	2	4	3	0	3	6	4	2	4	8	5	4
0	7	1	4	2	1	2	8	3	5	4	2	4	9	5	6	6	3
0	8	1	6	2	4	3	2	4	0	4	8	5	6	6	4	7	2
0	9	1	8	2	7	3	6	4	5	5	4	6	3	7	2	8	1

Рис. 1.2. Счетные палочки Непера

Сложив каждое число нижней строки с числом верхней строки, находящимся вправо наискосок от первого, получим искомое произведение 2 843 382. При этом если в результате сложения получается число 10 или больше, то единица добавляется к результату следующего сложения. Сложение начинают с младшего разряда. Именно Джон Непер (1550—1617), великий шотландский математик, изобрел логарифмы. Благодаря логарифмированию можно сложные вычисления сводить к выполнению простых арифметических операций — сложению и вычитанию. Как видно, неперовы палочки позволяли при умножении использовать простое сложение. Этот принцип положен в основу счетных (логарифмических) линеек, которые применяли для ручных расчетов инженеры и техники большую часть XX в. до появления электронных калькуляторов. Кстати, в компьютерах все вычисления обычно сводятся именно к сложению.

В XVII в. французский ученый, математик и физик Блез Паскаль (1623—1662) плодотворно работал над изобретением вычислительной машины. Он сделал около полусотни различных моделей и в 1642 г. создал машину, позволявшую складывать и вычитать многозначные числа. Дальнейшее развитие вычислительных машин связано с именем немецкого ученого Готфрида Лейбница (1646—1716). В конце XVII в. он изобрел ступенчатый валик, благодаря которому построил машину, выполнявшую четыре арифметических действия.

Более двух веков механические вычислительные машины использовали принцип передачи угла поворота с помощью редукторов с передаточным числом 10. В каждом разряде десять цифр от 0 до 9 изображались на ободе колеса. Когда колесо низшего разряда

поворачивалось на один оборот (т. е. цифры последовательно возрастали от 0 до 9), колесо следующего разряда поворачивалось на одну десятую оборота (т. е. показания в старшем разряде возрастали на одну единицу).

Подлинным переворотом в развитии вычислительной техники явилось создание арифмометра. В 1874 г. русский инженер В. Т. Однер сконструировал первый арифмометр (рис. 1.3), который сразу же завоевал всеобщее признание. На нем можно было производить сложение, вычитание, умножение, деление, возведение в степень и извлечение корня. Более половины вычислительных машин, которые применялись во всем мире до 1960-х годов, были основаны на принципе действия колес Однера. Особенность передачи вращения с помощью колеса Однера заключается в том, что при завершении одного оборота в низшем разряде колесо старшего разряда поворачивается на одну десятую оборота скачком, а не плавно (как в обычном зубчатом редукторе). Следовательно, в арифмометре существуют только фиксированные положения, что упрощает считывание информации.

Умножение в арифмометрах осуществляется примерно тем же методом, что и при умножении на бумаге. При умножении на цифру Ц младшего разряда надо повернуть валик с колесами Ц раз. Затем, прежде чем умножить на цифру следующего разряда, надо сдвинуть результат на один разряд влево. Сдвиг результата влево означает увеличение в 10 раз, а сдвиг вправо — уменьшение в 10 раз. Надо заметить, что сдвиг используется и в современных компьютерах, только в арифмометрах он выполнялся механически, а в компьютерах для этого используются электрические схемы, называемые регистрами сдвига.

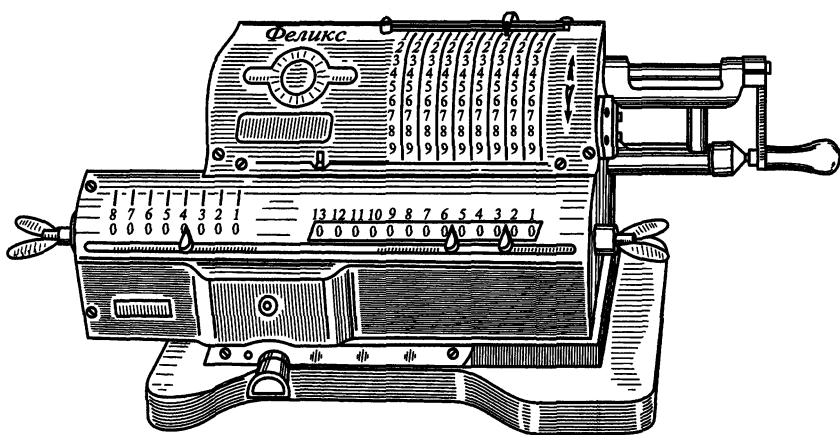


Рис. 1.3. Механический арифмометр

Прообразом современных малых вычислительных машин явились машины академика П.Л. Чебышева (1821 — 1894). Одна из них, построенная в 1878 г., замечательна тем, что в ней впервые была осуществлена постепенная передача десятков из низших разрядов в высшие.

Наряду с развитием и усовершенствованием механических вычислительных машин в XIX в. были созданы вычислительные машины нового типа, получившие название *табуляторов*. Это были уже электромеханические машины. Первый табулятор был построен Г. Холлеритом для ускорения обработки переписи населения США. В нем перфокарта прощупывалась тонкими щеточками из упругих проволочек. Если щеточки попадали в отверстия перфокарты, вставленной в табулятор, то замыкались электрические цепи и возникал электрический ток, который использовался для ввода чисел в счетчики и управления работой машины. Таким образом, стала возможной запись программы работы вычислительной машины, т. е. последовательности действий. Первые перфокарты для записи последовательности команд появились в ткацких машинах (так называемых машинах Жаккарда) в начале XIX в., а первым применил перфокарты для управления процессом вычислений английский математик Чарльз Бэббидж в середине XIX в. Интересно отметить, что изобретатель табулятора Г. Холлерит основал фирму для их производства, которая позднее превратилась в корпорацию IBM (International Business Machines Corporation) — один из гигантов компьютерного производства.

Применение электричества позволило улучшить характеристики вычислительных машин, основанных на механическом перемещении, что видно на примере перехода от арифмометра к табулятору. Кроме того, появилась возможность с помощью электрических схем создавать вычислительные устройства, выполняющие такие сложные математические операции, как интегрирование и дифференцирование. Ведь процессы в электрических цепях, содержащих катушки индуктивности и конденсаторы, описываются дифференциальными уравнениями, аналогичными тем, которые служат для описания самых разных физических процессов. Можно сказать, что существует определенная аналогия между, например, передачей электрической энергии и теплоты, между механическим движением и движением электрических зарядов. Поэтому электрические схемы моделируют самые разные физические устройства. В таких схемах исходные данные для расчета и сами результаты расчета представляются в виде электрических сигналов (обычно в виде напряжения). Поскольку эти сигналы могут принимать любые значения, т. е. изменяются не дискретно (скачком), а непрерывно, их называют аналоговыми, а вычислительные машины с такими сигналами — *аналоговыми вычислительными машинами* (АВМ). Если же сигнал может принимать только фиксированные значения, т. е.

от одного до другого соседнего значения изменение происходит скачком (как, например, между цифрой 4 и цифрой 5), то вычислительные машины с такими сигналами называются *цифровыми вычислительными машинами* (ЦВМ).

Простейшим примером аналогового вычислительного инструмента является логарифмическая линейка, изобретенная еще в средние века. В дальнейшем многие ученые (физики и математики) внесли свой вклад в развитие аналоговых вычислительных инструментов. К таким инструментам относятся планиметр (прибор для подсчета площадей плоских фигур) Дж. Германа, фрикционный интегратор Джозефа Джона Томсона, аналоговый интегратор (интеграф) польского математика Б. Абданк-Абакановича. Идеи последнего были положены в основу первой АВМ, созданной в 1904 г. А. Н. Крыловым для решения дифференциальных уравнений при строительстве кораблей.

Во втором десятилетии XX в. был разработан метод моделирования, на основе которого получили развитие вычислительные устройства, использующие электропроводящую бумагу. Начало работ по АВМ в СССР относится к третьему десятилетию XX в., когда С. А. Гершгорин заложил основы построения сеточных электроинтеграторов для решения уравнений в частных производных. В 1930-х годах С. А. Лебедев разработал методику моделирования электросетей переменного тока и построил электрическую модель для их расчета.

Потребность в промышленном выпуске аналоговых вычислительных устройств возникла прежде всего при решении штурманских задач для судовождения и расчете параметров прицеливания для артиллерийских стрельб (особенно в случае подвижной цели). В начале XX в. стали появляться (а затем и серийно выпускаться) счетно-решающие устройства специализированного назначения: автосчислители и путепрокладчики для судовождения, приборы для управления артиллерийским огнем. В них для преобразования механического угла поворота в электрическое напряжение и для выполнения арифметических и тригонометрических преобразований применялись микрорелемашины — сельсины и поворотные трансформаторы. В нашей стране такими вычислительными устройствами были, например, автопрокладчик «Путь», автоматический счислитель координат АС, прибор для управления зенитным артиллерийским огнем ПУАЗО. Все они являлись по существу АВМ специализированного назначения.

При массовом использовании каких-либо технических устройств особую важность приобретает проблема надежности. Поэтому широкому применению таких счетно-решающих устройств во многом способствовало изобретение в 1938 г. отечественными учеными А. Г. Иосифьяном и Д. В. Свечарником высоконадежных бесконтактных сельсинов.

Создание первого аналогового компьютера относят к 1927 г. (США, Массачусетский технологический институт).

Первые модели ЦВМ также разрабатывались на электромеханических элементах, в качестве которых использовались реле, т. е. устройства, которые могут находиться только в двух устойчивых состояниях: «включено» или «выключено». Надежность таких элементов крайне низка, а для ЦВМ их требуется сотни и тысячи. Первую ЦВМ построил в 1938 г. немецкий инженер К. Цузе.

Особенно бурное развитие получила вычислительная техника (в том числе АВМ и особенно ЦВМ), когда на помощь ей пришла электроника и начали создаваться электронные вычислительные машины. В 1947 г. советскими учеными был создан электроинтегратор, на котором можно было быстро решать сложные дифференциальные уравнения. Если раньше для решения таких задач требовались месяцы, то на электроинтеграторе они решались за несколько часов.

Вместе с развитием электроинтеграторов появилась целая группа быстродействующих цифровых электронных машин. Цифровые вычислительные средства развивались параллельно с аналоговыми. Наиболее близким прообразом современных ЦВМ следует считать «аналитическую машину» Ч. Бэббиджа. В 1937—1944 гг. под руководством американского ученого Г. Эйкена была создана электромеханическая ЦВМ «Mark-1».

Второй этап. Революционным поворотом в развитии цифровой вычислительной техники явилось создание электронных ЦВМ с программным управлением. Только с появлением подобных машин оказалось возможным решать такие сложные математические задачи, как, например, управление движением спутников и ракет. За несколько десятков минут указанные машины делают вычисления, которые с помощью арифмометра пришлось бы выполнять в течение 15—20 лет. Все созданные быстродействующие электронные ЦВМ имеют устройство управления, арифметическое и запоминающее устройства. Для того чтобы такие машины решали определенный тип задач и выполняли определенный вид работы, предварительно составляется программа работы машины, которая при помощи, например, перфокарт вводится в устройство управления. Большой вклад в создание первых цифровых машин внес американский математик Джон (Янош) фон Нейман (1903—1957), сформулировавший основные принципы, на которых основывалась работа практически всех программируемых ЦВМ XX в.

Первая быстродействующая цифровая электронная машина ЭНИАК (ENIAC — Electronic Numeral Integrator and Computer) была построена в США в 1946 г. Эта ЦВМ размерами 30×2,5 м весила 30 т. За 1 с она делала 5 тыс. операций сложения. Машина ЭНИАК содержала около 18 тыс. электронных ламп и потребляла

более 100 кВт. Она работала в десятичной системе счисления. Сложение и вычитание производилось за 200 мкс, умножение — за 2 800 мкс.

В дальнейшем в нашей стране и за рубежом было создано большое число разнообразных конструкций таких машин. В 1947 г. под руководством С.А.Лебедева начались работы по созданию универсальной ЦВМ, работающей по хранимой программе. В 1948 г. при Академии наук СССР был создан специальный НИИ точной механики и вычислительной техники. В конце того же года вышло постановление Совета Министров СССР о создании специального конструкторского бюро для разработки и обеспечения изготовления средств вычислительной техники для систем управления оборонными объектами. В обстановке холодной войны началась гонка в развитии тех направлений науки и техники, которые могли обеспечить превосходство в войне или сдерживание противника, замышляющего развязать войну. К таким направлениям относятся, в первую очередь, создание ядерного оружия, средств его доставки и быстродействующих систем управления на основе вычислительной техники.

По всем этим трем направлениям между основными соперниками (США и СССР) существовало примерное равенство, хотя в целом промышленно-технический уровень США был выше, в области оборонной техники разница в уровнях развития между США и СССР была не столь заметна, поскольку оборонным отраслям промышленности в СССР уделялось особое внимание. В области вычислительной техники научные разработки советских ученых были близки к мировому уровню, а вот элементная база отставала. И все же средства советской вычислительной техники для управления оборонными системами примерно до середины 1960-х годов не уступали американским, а кое в чем и превосходили их.

В 1950 г. в СССР под руководством С.А.Лебедева была создана первая в континентальной Европе (не считая Великобритании) малая электронная счетная машина МЭСМ. Она содержала около 2 тыс. электронных ламп, работала по параллельно-последовательному принципу выполнения операций, имела быстродействующую память на ламповых регистрах и внешнюю память на магнитном барабане. Структура и основные схемы этой машины стали классическими и были положены в основу быстродействующих электронных счетных машин семейства БЭСМ, созданных также под руководством С.А.Лебедева. Первая модель БЭСМ, которая была построена и начала работать в конце 1952 г., за 1 с выполняла в среднем 7—8 тыс. арифметических действий. Для сравнения напомним, что с помощью арифмометра можно выполнить за 8 ч лишь около 2 тыс. действий. В последующие годы машина была усовершенствована: вместо прежнего запоминающего устройства

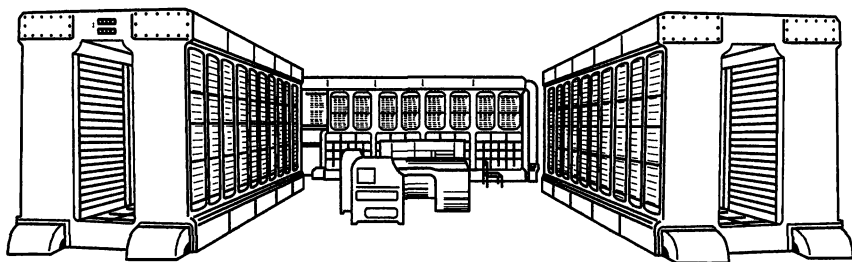


Рис. 1.4. Цифровая ЭВМ «Стрела» (1953 г.)

(ЗУ) на электронно-лучевых трубках было установлено построенное на ферритовых сердечниках оперативное ЗУ емкостью 2 048 чисел; скорость работы была повышена до 10 тыс. операций в секунду. Машина имела и внешнее ЗУ на магнитной ленте (четыре магнитофона общей емкостью 120 тыс. чисел и магнитный барабан емкостью до 5 тыс. чисел).

Большое распространение получили машины типа «Урал», модификация которой «Урал-4» выполняла 5 тыс. операций в секунду. Были созданы разнообразные по конструкции и назначению ЭЦВМ: «Стрела» (рис. 1.4), УМШН (универсальная машина широкого назначения), «Киев», «Сетунь», «Минск» и ряд других, причем «Сетунь» была единственной в мире машиной, в которой использовалась троичная система счисления — более экономная с точки зрения количества элементов для представления больших чисел, чем двоичная система. Характеристики машин семейства БЭСМ последовательно улучшались. Самая мощная из них БЭСМ-6 (рис. 1.5) имела быстродействие 1 млн операций в секунду. Все перечисленные машины имели большие размеры и были сложны в эксплуатации. Ими оснащались вычислительные центры. В рассматриваемый период стали появляться и ЭВМ для использования непосредственно на рабочем месте научного сотрудника, например ЦВМ «Проминь» (рис. 1.6).

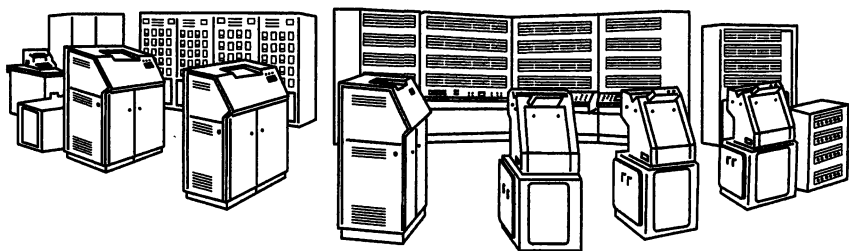


Рис. 1.5. Цифровая ЭВМ БЭСМ-6 (1967 г.)

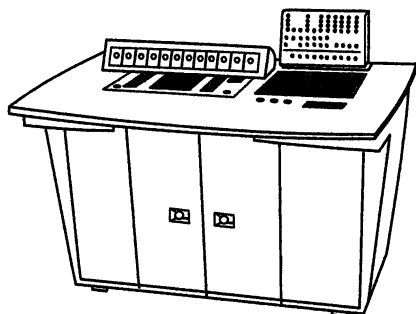


Рис. 1.6. Цифровая ЭВМ для инженера (1962 г.)

«Мы были на передней линии развития ЭВМ», — таково мнение академика Н.Н.Моисеева, под руководством которого средствами вычислительной техники было проведено моделирование последствий ядерной войны для планеты Земля. Уместно напомнить, что специалист по истории развития компьютеров, куратор Музея вычислительной техники в Великобритании Д.Свейд в 1996 г. опубликовал сенсационную статью,

в которой называл технологическое превосходство США в области компьютерной техники мифом и с удивлением констатировал, что БЭСМ-6, созданная в СССР в середине 1960-х годов, не уступала западным аналогам.

Во второй половине 1960-х годов фирма ИВМ создала семейство ЦВМ универсального назначения; их можно было с одинаковой эффективностью использовать в различных областях экономики, науки и техники. Различные модели семейства ИВМ-360 представляли собой ряд ЦВМ с последовательно возрастающей производительностью. Соответственно росту производительности возрастала и цена, но это не была прямо пропорциональная зависимость. Например, для моделей, отличавшихся по производительности в 100 раз, цена отличалась только в 50 раз. Таким образом, чем мощнее была машина, тем меньше была цена отдельной выполненной на ней операции. Для всех моделей использовалась универсальная операционная система, обладающая широкими возможностями и предоставляющая большие удобства для пользователя.

Для некоторых моделей, например, эта операционная система содержала около 2 млн команд. Имелись в системе и трансляторы для наиболее распространенных языков программирования. Главное, что определило популярность семейства ИВМ-360, заключалось в программной совместимости всех моделей. Архитектура ИВМ-360 оказала сильное влияние на разработки многих зарубежных фирм, которые начали производить вычислительные машины и системы, полностью или в значительной степени совместимые с ней. Логическая структура ИВМ-360 стала самой распространенной в мире.

Принципы программной совместимости были положены и в основу созданной в СССР при участии стран — участниц Совета Экономической Взаимопомощи (ГДР, Польша, Венгрия и др.) ЦВМ единой системы, получившей название ЕС ЭВМ (одна из



Рис. 1.7. Цифровая ЭВМ единой серии ЕС-1020

моделей этой системы показана на рис. 1.7). Ряд моделей был построен по принципу возрастания производительности от нескольких тысяч до миллиона операций в секунду.

Последовательно увеличивалась и емкость оперативной памяти — от 64 Кбайт до 1 Мбайт. Это были машины третьего поколения (на гибридных интегральных схемах), поэтому они потребляли значительную мощность (от 4 до 60 кВт·А) и занимали большую площадь (от 12 до 230 м²). Большим достоинством ЕС ЭВМ была возможность использования программного обеспечения, созданного разными странами. Надо заметить, что разработка программного обеспечения стоит порой не меньше, чем создание самой вычислительной машины.

В развитии электронных ЦВМ принято выделять четыре периода, соответствующие различным электронным приборам, на основе которых строилась вычислительная техника. В соответствии с этими четырьмя периодами различают ЭЦВМ четырех поколений.

В первый период (с 1945 до конца 1950-х годов) ЦВМ (первого поколения) создавались на электронных лампах. Эти машины потребляли много электроэнергии, выделяли много теплоты, требовали специальных мер для охлаждения, занимали большую площадь.

Во второй период (конец 1950-х — середина 1960-х годов) на смену электронным лампам пришли полупроводниковые элементы (транзисторы и диоды), в десятки раз меньшие по размеру, не требующие подогрева и обладающие высокой надежностью. Это позволило существенно увеличить их число в одной машине и соответственно увеличить ее производительность. В силу этого в десятки раз уменьшились размеры самих ЦВМ и потребляемая ими мощность. Примерами машин второго поколения являются IBM-1620 и IBM-1790.

Третий период (середина 1960-х — начало 1970-х годов) ознаменован появлением интегральных схем — микроминиатюрных электронных устройств с высокой плотностью упаковки связанных между собой элементов (диодов, транзисторов, резисто-

ров и др.). Использование интегральных схем с несколькими десятками транзисторов на кристалле позволило еще больше уменьшить размеры ЦВМ с одновременным повышением производительности и надежности. Из ЦВМ третьего поколения особенно сильное влияние на развитие вычислительной техники оказали машины семейства IBM-360.

Четвертый период начался с появлением больших интегральных схем, когда на одном кристалле размещаются тысячи полупроводниковых переходов. В 1969 г. в одном полупроводниковом устройстве удалось объединить все элементы вычислительной машины, осуществляющие арифметические и логические преобразования, временное хранение и вызов данных и команд, управление устройствами ввода и вывода информации. Такое полупроводниковое интегральное устройство получило название *микропроцессор*. В 1970 г. была создана ЦВМ IBM-370, которую принято считать первой машиной четвертого поколения. С этого времени развитие вычислительной техники в наибольшей степени стало определяться достижениями технологии производства больших интегральных схем.

Третий этап. Настоящая информационная революция была связана с появлением первого персонального компьютера, выпущенного в 1981 г. фирмой IBM. (Вообще-то, первый ПК придумали и создали в 1976 г. два молодых энтузиаста Стив Джобс и Стефан Возняк (см. подразд. 2.3), но это была, скорее, игрушка, чем серьезная машина. Правда, спустя некоторое время такая игрушка трансформировалась в знаменитый Apple Macintosh.) В сущности, его разработчики не придумали ничего нового в области компонентов ПК.

Революционной была принципиально новая архитектурная система благодаря введению в конструкцию системной платы ПК разъемов расширения для подключения дополнительных устройств непосредственно к шине. В результате этого простого решения и было достигнуто главное преимущество ПК — открытая архитектура.

Дело в том, что до этого все модели микрокомпьютеров имели закрытую архитектуру, т.е. неизменяемую конструкцию, и были обречены на неизбежное моральное старение. Открытая же архитектура не только допускает, но и стимулирует замену дополнительных устройств при старении прежних. Причем все новые устройства и программы должны быть совместимыми по принципу «сверху вниз», т.е. последующие версии должны обслуживать все ранее существовавшие.

Например, пользователь приобрел ПК с цветным монитором и таким же видеоконтроллером (электронной платой для управления монитором). Даже неспециалист в состоянии извлечь из системы устаревшее устройство и заменить его новым. Через два

года с появлением улучшенного цветного графического контроллера пользователь заменяет лишь один из компонентов ПК. Преимущества подобного подхода очевидны.

Во-первых, нет необходимости в замене системы в целом, если возможно ее обновление по частям (особенно если учесть, что разные блоки ПК устаревают с разной скоростью).

Во-вторых, совершенствование ПК становится делом самого пользователя, который вследствие своей близости к конкретному применению ПК лучше представляет себе, что требуется от системы.

В-третьих, процесс ремонта сводится к замене не устройства в целом, а отдельного элемента, что можно сделать намного быстрее.

Именно благодаря удачному конструктивному решению началось лавинообразное нарастание производства ПК и их отдельных узлов, совместимых с оригинальной конструкцией. За многолетнюю историю развития микропроцессорной техники ведущие позиции в этой области занимает американская фирма Intel (INTEgreted Electronics).

В 1971 г. она разработала и выпустила первый в мире 4-битный микропроцессор (МП) 4004. Подлинный успех ей принес 8-битный МП 8080, который был создан в 1973 г. Этот процессор получил очень широкое распространение во всем мире. В 1980-е годы выпускавшийся в нашей стране его аналог микропроцессор КР580ИК80 применялся во многих бытовых персональных компьютерах и разнообразных контроллерах.

С середины 1970-х годов владельцами и пользователями компьютеров становились уже не только организации, но и отдельные граждане. Этому способствовала не только доступная цена и несложное техническое обслуживание, но и возможность для пользователя самостоятельно программировать работу компьютера. Для этого был создан интерпретатор языка Бейсик (Basic), который позволил пользователям достаточно просто общаться с компьютером и решать на нем разнообразные задачи. Возможности ПК еще более расширились, популярность возросла, число пользователей увеличилось до сотен тысяч.

В 1979 г. фирма Intel первой выпустила 16-битный МП типа 8086. Он стал прародителем целого семейства, которое обычно называют семейством 80×86 . (Знак \times в середине данного обозначения по мере развития микропроцессоров заменяется на очередную цифру: 80186, 80286, 80386, 80486, ...). Аналог этого микропроцессора КР1810ВМ86 применяется в ПК, выпускаемых в нашей стране.

Вслед за МП 8086 появился МП 8088, архитектурно повторяющий МП 8086 и имеющий 16-битные внутренние регистры, но его внешняя шина данных составляет 8 бит. Широкой популярно-

сти МП 8088 способствовало его применение фирмой IBM в персональных компьютерах PC и PC-XT. Очень быстро для этих компьютеров был накоплен такой огромный объем программного обеспечения, что в последующих, более совершенных процессорах фирмы Intel пришлось предусматривать специальный режим эмуляции (см. подразд. 3.3) этого МП. Обычно такой режим называют режимом реального адреса (Real Address Mode), или R-режимом.

В 1981 г. появился МП 80186, который сохранил базовую архитектуру МП 8086, но содержал на кристалле контроллер прямого доступа к памяти, счетчик-таймер и контроллер прерываний. Кроме того, была несколько расширена система команд. Однако большого распространения этот МП (как и ПК на его основе) не получил.

В 1982 г. был создан МП 80286 (далее этот и последующие МП, принимая во внимание их широкие возможности, будем называть процессорами). При его разработке были учтены достижения в архитектуре мини-компьютеров и больших компьютеров. Процессор 80286 может работать в двух режимах. В режиме реального адреса он эмулирует МП 8086, а в защищенном режиме виртуального адреса (Protected Virtual Address Mode), или P-режиме, предоставляет программисту много новых возможностей и средств. Среди них следует отметить расширенное адресное пространство памяти емкостью 16 Мбайт, появление дескрипторов сегментов и дескрипторных таблиц, наличие защиты по четырем уровням привилегий, поддержку организации виртуальной памяти и мультизадачности. Процессор 80286 применялся в ПК PC/AT и младших моделях PS/2.

При разработке 32-битного процессора 80386 потребовалось решить две основные задачи: совместимости и производительности. Первая из них была решена путем внедрения трех режимов работы.

В R-режиме, который действует после включения питания или системного сброса, процессор копирует работу МП 8086 и использует 16-битные регистры; адресное пространство составляет 1 Мбайт.

В P-режиме процессор может выполнять 16-битные программы процессора 80286. Вместе с тем, в этом же режиме он может выполнять свои 32-битные программы, что обеспечивает повышение производительности системы.

Именно в этом режиме реализуются все новые возможности и средства процессора 80386, среди которых можно выделить более удобную индексную адресацию памяти, более быстрый доступ к регистрам общего назначения, новые команды, средства отладки и т. д.

Следующей разработкой фирмы Intel стал процессор i486, содержащий более 1 млн транзисторов. Два главных его отличия от предыдущих процессоров: математический сопроцессор реализо-

ван на одном кристалле вместе с центральным процессором в виде устройства с плавающей точкой FPU (Floating Point Unit); имеется внутренняя совмещенная кэш-память команд и данных емкостью 8 Кбайт. В этом процессоре реализовано много аппаратных и программных новинок.

Последнее десятилетие XX в. было отмечено появлением микропроцессоров Пентиум. Сначала предполагалось, что после 486-го микропроцессора появится 586-й. Он появился в 1993 г., но под другим названием (Пентиум), поскольку произошло резкое, кардинальное улучшение характеристик. В первом процессоре Пентиум по сравнению с МП 486 частота возросла втрое, разрядность шины данных — вдвое, емкость адресуемой памяти — в 16 раз. Затем были созданы Пентиум II (1997 г.), Пентиум III (1999 г.) и Пентиум 4 (2000 г.), который до сих пор является одним из наиболее распространенных среди ПК. В одном таком процессоре размещаются более 40 млн транзисторов.

В настоящее время появились микропроцессоры, превосходящие Пентиум по своим характеристикам. Значительно увеличивается и количество транзисторов в одном микропроцессоре. А на 2008 г. запланирован выпуск высокопроизводительного микропроцессора (кодовое имя Alpha EV10), в котором будет 1,5 млрд транзисторов. Уже сейчас появились многоядерные микропроцессоры, выполняющие операции параллельно, т.е. одновременно.

Разместить в одной микросхеме огромное количество транзисторов удастся за счет очень малого расстояния между ними. Если в первых микропроцессорах это расстояние измерялось микронами (микрометрами, т.е. миллионными долями метра), то теперь оно составляет десятки нанометров (миллиардных долей метра). Стала более совершенной технология изготовления микросхем. Прежде ее называли микронной технологией, теперь используется нанотехнология.

Развитие микропроцессоров напоминает спортивные соревнования, поскольку происходит в непрерывном соперничестве фирм-производителей. Кроме процессоров Пентиум 4 фирма Intel выпускает МП Celeron и Xeon, а фирма AMD — МП Athlon и Duron.

Возможности персональных компьютеров превзошли возможности больших и средних ЭВМ, которые использовались в организациях и вычислительных центрах. Теперь персональные компьютеры используются и в организациях для обработки самой разной информации. ЭВМ не только производят вычисления, но и обрабатывают информацию: с их помощью пишут и редактируют тексты, переводят с иностранных языков, расшифровывают секретные сообщения, записывают и обрабатывают рисунки, записывают и обрабатывают речь и музыку.

Другими словами, ЭВМ теперь — это не просто вычислительная, а скорее информационная машина, а после объединения ПК

во всемирную сеть — еще и телекоммуникационная. Поэтому частью истории развития вычислительной техники следует считать и основные моменты из истории электросвязи, а именно: создание телефона (передача информации в виде электрических сигналов по проводам), телеграфа (тоже передача информации по проводам, но сигнал дискретный и двоичный: точка и тире в азбуке Морзе), радио (беспроводная передача информации), телевидения (передача изображения). Использование всех этих способов передачи информации совместно с вычислительной техникой для обработки информации привело к созданию цифровых сетей интегрального обслуживания (ЦСИО, а на английском ISDN — Integrated Service Digital Net).

Продолжается развитие и сверхмощных ЭВМ, так называемых суперЭВМ. Мощность (быстродействие) ЭВМ за последние 40 лет увеличилась почти в 1 млрд раз. Ни в одной из других областей техники человечество не достигло такого значительного прогресса за столь малый период времени. Для сравнения укажем, что благодаря развитию ракетно-космической отрасли скорость транспортных средств за полвека возросла с 500 до примерно 40 000 км/ч, т. е. менее чем в 100 раз.

Наряду с разделением истории развития ВТ на три этапа можно выделить принципиально важный поворотный момент в развитии этой техники — появление СБИС (сверхбольших интегральных схем), а точнее — появление микропроцессора.

1.2. Области применения вычислительной техники

Вычислительная техника создавалась для облегчения и ускорения сложных математических расчетов, т. е. именно для вычислений, что и нашло свое отражение в самом ее названии. Первые крупные ЭВМ (1940-х годов и последующих десятилетий XX в.) использовались в области атомной физики, аэродинамики, баллистики и ряде других областей, в которых расчеты протекающих процессов очень сложны и трудоемки. Уже в этот период успехи вычислительной техники были весьма впечатляющими. Появились сенсационные сообщения о том, что ЭВМ за несколько часов выполнила работу, которую несколько десятков инженеров и техников выполняли с помощью логарифмических линеек и арифмометров несколько месяцев.

С легкой руки журналистов вычислительные машины стали называть не иначе, как «думающие машины» или «электронный мозг», что, строго говоря, не соответствует истине, поскольку ЭВМ не «думает», а последовательно выполняет команды программы, составленной человеком.

Само же составление программы хотя и требовало больших затрат времени, но обходилось во много раз дешевле, поскольку сто-

имость первых ЭВМ была очень высока. Такие дорогостоящие высокопроизводительные ЭВМ могли приобрести только крупные государственные вычислительные центры и очень богатые фирмы. Поэтому те предприятия и организации, которым необходимо было производить расчеты на ЭВМ, готовили программы для своих расчетов, а затем сдавали их для выполнения в вычислительные центры, оплачивая каждый час работы ЭВМ. На языке того периода — «покупали машинное время». Таким образом, в ту пору пользователь и владелец ЭВМ были разными субъектами.

По мере развития вычислительной техники стоимость ее существенно снижалась. С 1980-х годов персональный компьютер стал доступен не только любому специалисту в какой угодно области человеческой деятельности, но и любому частному лицу. В те же годы область применения вычислительной техники значительно расширилась, поскольку оказалось возможным представлять в цифровом виде самую разнообразную информацию.

Как писал Билл Гейтс, впервые стало возможным представление любого вида информации: чисел, текстов, звука, изображения — в цифровом формате, пригодном для хранения и обработки в любом компьютере, а также для передачи с компьютера на компьютер.

С этого времени вычислительная техника стала больше применяться не столько для математических вычислений, сколько для обработки информации, все больше превращаясь из вычислительной в информационную технику. Поэтому когда говорят о широком применении информационных технологий, имеют в виду именно широкое применение компьютеров.

Человек получает информацию с помощью своих пяти органов чувств: зрения, слуха, обоняния, осязания, вкуса. В цифровом формате сейчас можно представить в основном только ту информацию, которую получаем с помощью зрения и слуха. А вот возможности представления в цифровой форме (формате) запахов и вкусовых ощущений пока еще весьма ограничены. Хотя уже и сейчас имеются различные газоанализаторы, в которых датчики (сенсоры) определяют наличие и процентное содержание в воздухе различных газов и паров (в том числе и не только пахнущих). Точно так же есть и датчики солености, сахаристости. А в роботах и манипуляторах используются датчики, дающие информацию о противодействующих усилиях. Разработаны устройства, определяющие твердость, шероховатость, гладкость и иные характеристики, которые человек определяет с помощью органов осязания, т.е. ощупыванием.

Следовательно, ЭВМ может обрабатывать и ту информацию, которую мы получаем с помощью органов обоняния, вкуса и осязания. Поскольку эта информация преобразуется в электрический сигнал, то можно «оцифровать» этот сигнал и он будет воспринят

ЭВМ. Но в целом пока надо отметить, что с представлением информации, которую человек получает «на вкус, на запах, на ощупь», в понятном для ЭВМ виде дело обстоит не столь хорошо, как с «оцифровкой» изображения и звука. А ведь существует еще множество внешних воздействий и природных явлений, которые человек может воспринимать только с помощью приборов. Его пять органов чувств здесь бессильны.

Для примера здесь достаточно упомянуть о радиоактивном излучении — его не видно, не слышно, оно не пахнет и вкуса не имеет.

«Научить» ЭВМ воспринимать информацию в самом широком смысле этого слова — задача, которую человек стремится решить.

До конца XIX в. свыше 90 % трудоспособного населения США было занято физическим трудом и только менее 10 % — работой с информацией. К началу XXI в. соотношение кардинально изменилось: в информационной сфере занято около 80 % трудоспособного населения (рис. 1.8). С появлением первых ЭВМ в середине XX в. началась компьютерная эпоха. К этому времени 30 % трудоспособного населения было занято в информационной сфере. Производительность труда этой категории людей была крайне невелика, поскольку в их распоряжении находились только простейшие вычислительные устройства, а для сохранения информации и необходимой при этом сортировки использовались в основном бумажные носители (таблицы, картотеки и т. п.), в лучшем случае — перфокарты. С появлением ЭВМ производительность труда в информационной сфере многократно возросла. Несмотря на это число людей, занятых в информационной сфере, к настоящему времени почти утроилось. Это означает, что потребность в работе

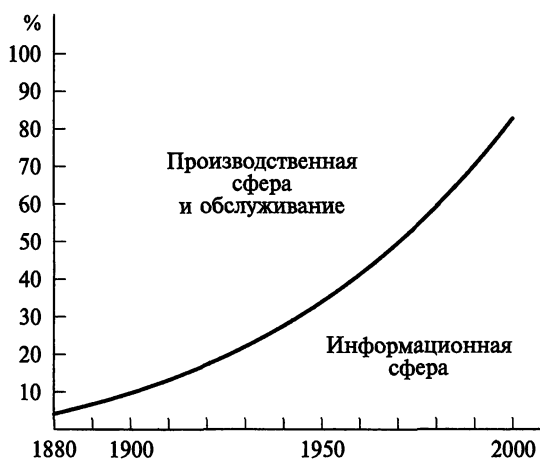


Рис. 1.8. Перераспределение трудовых ресурсов в XX в.

с информацией для современного общества становится весьма высокой.

В течение всей предшествующей XX в. истории развития цивилизации основным предметом труда оставались материальные объекты. Экономическая мощь государства оценивалась материальными ресурсами. В конце XX в. впервые в истории развития цивилизации именно информация становится основным предметом труда в общественном производстве промышленно развитых стран.

Постоянная тенденция перекачивания трудовых ресурсов из сферы материального производства в информационную сферу является сейчас наиболее заметным, хотя далеко не единственным симптомом приближающихся «гигантских потрясений», которые получили пока общее и несколько туманное название «информационный кризис».

Техника непрерывно совершенствуется, создаются все новые и новые технические устройства. Появление новых технических устройств вызвано и развитием науки (знания человечества расширяются и на их основе можно создать новое), и расширением потребностей человека (при этом не только в материальной, но и духовной сфере).

Новые технические устройства выполняют все более сложные функции, соответственно они требуют более сложного управления. Для того, чтобы освободить человека от многих задач по управлению огромным количеством технических устройств, и служит автоматическое управление.

В системах автоматического управления производственными и технологическими процессами еще до появления цифровой вычислительной техники использовались законы управления, в основе которых лежало формирование управляющих воздействий, находящихся в определенном соотношении с сигналом ошибки (разности между желаемым и истинным значениями регулируемой величины). Это соотношение в общем случае включало в себя три составляющих: пропорциональную, интегральную и дифференциальную. Иными словами, надо было рассчитать произведение сигнала ошибки на некоторый коэффициент, интеграл сигнала ошибки по времени, скорость изменения сигнала ошибки.

Все эти операции выполнялись на аналоговых элементах, т.е. фактически средствами аналоговой вычислительной техники. С появлением цифровых вычислительных машин эти операции стали поручать им. Тем более, что были разработаны специальные так называемые численные методы интегрирования и решения дифференциальных уравнений. Поэтому перед ЭВМ открылась широкая область использования в теплоэнергетике, металлообработке, химическом производстве, переработке минерального и органи-

ческого сырья, словом, во всех отраслях производственной деятельности человека.

После появления микропроцессоров и персональных компьютеров, когда вычислительная техника стала доступна практически каждому человеку, сформировалась такая широчайшая область использования вычислительной техники, как игры и развлечения. Надо сказать, что создатели первого персонального компьютера (молодые, еще не достигшие 30-летнего возраста С. Джобс и С. Возняк) и замыслили его именно как универсальную (с возможностью перепрограммирования) игровую машину.

Популярность компьютерных игр оказалась столь ошеломляющей, что теперь в стандартном наборе программ, которые закладываются в любой ПК еще до его продажи, обязательно имеется несколько игр.

Практически любой пользователь ПК часть времени проводит, играя, а получившие распространение во всем мире компьютерные клубы и кафе заполнены молодежью, играющей в компьютерные игры.

Эти игры непрерывно совершенствуются, добавляются звуковые и видеоэффекты, изображение становится трехмерным, похожим на реальное. Возникает ощущение виртуальной реальности. Цифровая запись и обработка звука и изображения произвели подлинный переворот в производстве кинофильмов и телефильмов, видеофильмов на компакт-дисках и магнитной ленте, да и во всей индустрии развлечений.

Элементы цифровой вычислительной техники очень широко используются в разнообразной бытовой технике: сотни миллионов микропроцессоров встроены в карманные калькуляторы, часы, телевизоры, стиральные машины и кухонные комбайны. Объединение вычислительной техники и техники связи, их взаимное влияние друг на друга привели к тому, что микропроцессоры и другие элементы вычислительной техники стали неотъемлемой частью телекоммуникационных приборов. Например, мобильных телефонов сейчас в мире около миллиарда, а в каждом находится микропроцессор.

Объединение компьютеров в мировую сеть расширило сферу применения вычислительной техники. Наиболее популярными и общественно значимыми можно назвать следующие области: информационно-поисковые системы, электронная почта, дистанционное обучение, электронная коммерция, телемедицина.

Дистанционное обучение, использующее средства вычислительной техники, основанное на применении информационно-коммуникационных технологий, позволяет получать образование (и расширять его) без отказа от производительного труда.

Широкое распространение получила электронная коммерция. С помощью подключенного к сети Интернет персонального ком-

пьютера покупатель «посещает» электронный магазин, знакомится с предлагаемыми товарами, их характеристиками и ценами. Выбрав нужный ему товар, он сообщает об этом, указывая адрес, куда товар должен быть доставлен, и номер своего банковского счета, с которого будет произведена оплата покупки.

Аналогично могут быть произведены выбор и покупка лекарств, а также консультации по их применению в электронной аптеке. Но этим возможности телемедицины не исчерпываются. Через компьютер можно не только посетить врача, рассказать ему о своих недугах, ответить на его вопросы и получить советы по лечению, но и «осмотреть» пациента.

С помощью компьютера можно на расстоянии выполнить и некоторые анализы, необходимые для диагностики заболевания (например, измерить пульс, прослушать шумы в легких, даже снять электрокардиограмму). Уже известны случаи, когда при выполнении сложных и уникальных операций для консультации привлекались крупнейшие специалисты, находившиеся за многие тысячи километров от операционной и наблюдавшие за операцией через Интернет.

Надо признать, что нет такой области человеческой деятельности, в которой не нашлось бы применение вычислительной технике (если не сейчас, то в близком будущем).

1.3. Перспективы информатизации общества

Развитие и широкое применение информационно-коммуникационных технологий (ИКТ) является всеобщей тенденцией мирового развития в последние десятилетия. Их применение имеет решающее значение для повышения конкурентоспособности экономики страны, расширения возможностей ее интеграции в мировую систему хозяйства, повышения эффективности государственного управления и местного самоуправления.

Несмотря на высокие темпы развития информационных и коммуникационных технологий в последнее десятилетие Россия не смогла сократить в этой области отставание от промышленно развитых стран. Отчасти это вызвано длительным экономическим кризисом.

В то же время недостаточное развитие ИКТ в России усугубляется рядом других факторов. Это, к примеру, несовершенная нормативная правовая база, отсутствие целостной информационной инфраструктуры и эффективной информационной поддержки рынков товаров и услуг, недостаточный уровень подготовки кадров в этой области, недостатки в регулировании экономической деятельности при выходе предприятий и организаций сферы ИКТ на российский и мировой рынки, высокий уровень монополизации сетей связи.

Особенности географического положения России, размещения производства и расселения населения, отставание развития транспортной инфраструктуры определяют особую важность ИКТ и систем удаленного доступа к информации для обеспечения населения и бизнеса необходимыми консультационными услугами. Развитие ИКТ позволит всем гражданам получать дистанционно медицинские, юридические и иные виды консультационных услуг, существенно повысит возможности дистанционного обучения и повышения квалификации.

А это важный шаг на пути развития демократии и реального обеспечения равных прав всех граждан в области информации. Индикатором выступает число пользователей сетью Интернет и объемы услуг, предоставляемых с помощью систем удаленного доступа.

На базе развития современных методов информационного обмена между государством и обществом будут созданы принципиально новые возможности для обеспечения информационной открытости и гласности принятия решений, для повышения уровня доверия и взаимодействия между обществом и государством. У граждан России появятся условия для сокращения затрат времени на реализацию своих конституционных прав и обязанностей.

Решение этих задач предусмотрено Федеральной целевой программой «Электронная Россия 2002—2010 гг.», утвержденной в 2001 г. постановлением Правительства РФ. В ней четко сформулированы задачи в сфере ИКТ.

Это, в первую очередь, совершенствование законодательства и системы государственного регулирования деятельности органов государственной власти и местного самоуправления, а также взаимодействия органов государственной власти и местного самоуправления с хозяйствующими субъектами и внедрение ИКТ в реальный сектор экономики.

В связи с этим не менее важным является развитие системы подготовки в этой области специалистов и квалифицированных пользователей, развитие независимых средств массовой информации, телекоммуникационной инфраструктуры и создание пунктов подключения к открытым информационным системам, разработка и создание системы электронной торговли.

Программа «Электронная Россия...» рассчитана на девять лет, в течение которых предстоит предпринять меры, направленные на информатизацию России. На ее выполнение выделяется около 2,4 млрд дол.

К 2007 г. доля продукции индустрии информационных технологий в российском валовом внутреннем продукте должна возрасти в 4 раза (до 2%), а объем экспорта высоких технологий увеличится в 15—20 раз (до 2,5 млрд дол).

Создание единого информационного пространства на всей территории страны создаст предпосылки для преодоления экономического отставания отдельных территорий, обеспечит гражданам и организациям, независимо от места проживания и расположения равную общедоступность всех отечественных и мировых открытых информационных ресурсов.

Повышение спроса на продукцию ИКТ обеспечит прирост отечественного производства товаров и услуг в этой сфере и рост экспорта в размере 15... 20 % ежегодно. Предполагается, что объем рынка информационных услуг и программного обеспечения возрастет к 2010 г. в 5—6 раз. Доля ИКТ в экономике увеличится в несколько раз.

Парк персональных компьютеров, используемых в сфере бизнеса, увеличится в 6 раз (используемых населением — в 4 раза) и существенно обновится.

Прогнозируется, что примерно каждый второй компьютер будет иметь выход в сеть Интернет.

Предполагаемые программой темпы распространения ИКТ в экономике и обществе резко увеличат потребность в специалистах этой области. К 2010 г. потребность в специалистах с высшим и средним специальным образованием составит более 130 тыс. в год.

Программа «Электронная Россия...» коснется всех граждан России. В рамках программы для них предполагается создать инфраструктуру новых технологий и образовательные программы, призванные научить этими технологиями пользоваться. По всей стране создается широкая сеть пунктов доступа в Интернет. В частности, так называемые коллективные центры доступа в Сеть появятся в каждом населенном пункте, где население превышает 10 тыс. человек.

Располагаются такие центры в почтовых отделениях, библиотеках, школах, вузах и других государственных и муниципальных учреждениях. На образование и переподготовку кадров «Электронная Россия...» потратит 26 млрд рублей, т. е. треть всего своего бюджета.

К концу 2010 г. все школы страны будут оснащены компьютерами: в среднем по одному компьютеру на четырех учащихся. Уже сейчас около 30 тыс. образовательных учреждений подключено к сети Интернет.

Во многих почтамтах и почтовых отделениях установлены персональные компьютеры, подключенные к Интернету. За небольшую плату каждый человек теперь может воспользоваться услугами электронной почты, получить интересующие его сведения через поисковую систему и богатейшие информационные ресурсы Интернета. На близкое будущее поставлена задача, чтобы в каждой семье был компьютер с выходом в Интернет.

Внедрение и массовое распространение ИКТ позволят создать условия для развития демократии, повышения эффективности функционирования экономики, государственного управления и местного самоуправления, а также обеспечат права граждан на свободный поиск, получение, передачу, производство и распространение информации, создадут условия для расширения подготовки не только специалистов по ИКТ, но и квалифицированных пользователей.

Значение данной программы трудно переоценить. Она окажет воздействие на все сферы общественной жизни, прежде всего позволит эффективно использовать интеллектуальный и кадровый потенциал России, обеспечить гармоничное вхождение нашей страны в мировую экономику на основе кооперации и информационной открытости, преодолеть отставание от развитых стран в уровне использования и развития ИКТ, обеспечить равноправное присутствие России в глобальном информационном сообществе.

Внедрение информационных технологий в организацию работ министерств и ведомств, а также региональных органов управления позволит сократить издержки на управление, в том числе и за счет высвобождения части технического персонала этих служб. Целевым индикатором информатизации работы органов управления может выступать доля безбумажного документооборота внутри ведомств и между ними.

Предполагается довести долю безбумажного документооборота в среднем до 65 % внутри ведомств и до 40 % в межведомственном обороте.

Эта интеграция государственных информационных ресурсов в единую систему позволит также радикально сократить возможности для нарушений законов, уклонения от уплаты налогов и т. п.

Реализация программы существенно повысит конкурентоспособность экономики страны за счет снижения издержек и улучшения качества продукции и услуг, эффективности государственного управления и местного самоуправления, обеспечит рост качества жизни граждан, сократит отставание от развитых стран мира, позволит избежать информационной и экономической изоляции от мировой экономики и мирового сообщества, обеспечит развитие процессов международной интеграции.

Контрольные вопросы

1. Какое вычислительное устройство было первым в истории человечества?
2. В какой последовательности выполняются действия на русских счетах при сложении чисел 6 237 и 5 964?

3. Выполните умножение числа 5 307 на 8 с помощью палочек Непера.
4. На каких элементах были построены первые электронные вычислительные машины?
5. В какой последовательности развивалась элементная база ЭВМ?
6. Приведите пример использования вычислительной техники в той области, в которой намерены трудиться.
7. Какие задачи будут решены в ходе выполнения программы «Электронная Россия 2002 — 2010 гг.»?

ХАРАКТЕРИСТИКИ И КЛАССИФИКАЦИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

2.1. Основные характеристики ЭВМ

В процессе развития цивилизации создавались самые разнообразные технические средства, помогавшие человеку в обеспечении его жизненных потребностей, облегчавшие его труд. Многие из таких устройств называют общим словом «машина». В процессе развития машинной техники можно выделить три этапа.

На первом этапе создавались машины, помогавшие человеку получать тот или иной продукт из исходного материала. К машине отошла прежде всего функция воздействия на объект труда. Мельничные жернова перемалывали зерно на муку, плугом пахали землю, на ткацком станке делали ткань, на гончарном круге — посуду. Этот класс машин можно считать рабочими машинами. Основным показателем совершенства рабочих машин является производительность, определяемая отношением количества изготовленного продукта к затраченному времени.

На втором этапе потребовались машины, создающие не сами изделия, а механическую энергию, посредством которой приводятся в действие рабочие машины. Этот класс машин можно считать энергетическими машинами. К ним относятся паровые и электрические двигатели, вырабатывающие механическую энергию за счет сжигания топлива. Основным показателем совершенства энергетических машин также является производительность, определяемая количеством выработанной энергии к затраченному времени, т. е. мощностью.

Объединение энергетических машин с рабочими машинами привело к появлению сложных устройств — систем, состоящих из трех частей: машина-двигатель (энергетическая), передаточный механизм, машина-орудие (рабочая). Дальнейшее их развитие вызвало необходимость создания машин для управления такими устройствами, т. е. потребовались машины для управления машинами.

С этого начался третий этап — бурное развитие средств облегчения умственного труда, который заключается в обработке информации. Выдающимся достижением на этом этапе явилось создание совершенно нового типа машин — электронных вычислительных машин. Основным показателем совершенства вычислительных машин также является производительность, которая определяется отношением количества выполненных операций к

затраченному времени. Таким образом, для всех машин самого разного назначения наиболее важной характеристикой является производительность. А вот единица измерения для этой характеристики зависит от назначения машины. При этом по аналогии с энергетическими машинами для вычислительных машин иногда вместо термина «производительность» используют термин «вычислительная мощность». Появление ЭВМ заставило пересмотреть и само определение машины. Она теперь может рассматриваться как техническое устройство, осуществляющее преобразование вещества (материала), энергии или информации.

Как уже говорилось, основной характеристикой ЭВМ является ее производительность, измеряемая количеством выполняемых операций за единицу времени. Однако операции различаются по сложности; естественно, что простые выполняются быстрее, а сложные — медленнее. Чтобы можно было сравнивать различные модели ЭВМ по производительности, в настоящее время принято производительность оценивать количеством операций, производимых над числами с плавающей запятой за одну секунду. Такую единицу производительности ЭВМ называют *флопс* (от *англ.* floating point operations per second — операции с плавающей точкой за секунду (во многих странах целую и дробную части числа разделяют не запятой, а точкой)). Современные ЭВМ выполняют миллиарды операций в секунду, поэтому приходится пользоваться такими единицами производительности, как *гигафлопс* — миллиард флопс и *терафлопс* — тысяча миллиардов флопс (приставкам «гига» и «тера», сокращенно обозначаемым Г и Т, соответствуют множители 10^9 и 10^{12}). При сравнении ЭВМ, разных по производительности, но принадлежащих примерно к одному классу, им предлагают специально разработанные одинаковые тестовые задания, т.е. программы вычислений.

Другими характеристиками ЭВМ являются тактовая частота, объем памяти, разрядность, надежность, вес и габаритные размеры, потребляемая мощность, цена.

Тактовая частота в значительной степени определяет производительность ЭВМ, поскольку фактически показывает число микроопераций, выполняемых за секунду. Но отдельные операции состоят из разного количества микроопераций, поэтому имеют значение и эффективность составленной программы выполнения той или иной операции, и время обращения к памяти, и разрядность ЭВМ. Например, ЭВМ, оперирующая числами, имеющими 64 разряда, будет вдвое производительнее 32-разрядной ЭВМ (при одинаковой тактовой частоте). В настоящее время тактовая частота достигает нескольких гигагерц (ГГц), т.е. за секунду выполняется несколько миллиардов микроопераций. Увеличение производительности до нескольких тысяч миллиардов операций достигается за счет параллельного их выполнения несколькими

процессорами. Например, во входящей в первую десятку по производительности суперЭВМ фирмы Hewlett Packard использованы 4 096 процессоров с тактовой частотой 1,25 ГГц.

Рассмотрим основные показатели надежности, по которым оцениваются элементы автоматики (к ним относятся и элементы, из которых состоит ЭВМ). При оценке надежности используется термин «отказ». Отказами в работе элемента называют как выход его из строя, так и изменение параметров, приводящее к неудовлетворительному выполнению элементом его функций. Отказы, как правило, появляются внезапно, т.е. подчиняются законам, свойственным случайным величинам. Их изучают с помощью математической статистики. Для количественной оценки надежности элементов автоматики обычно используют следующие показатели: $p(t)$ — вероятность безотказной работы в течение заданного отрезка времени; $\lambda(t)$ — интенсивность отказов; T_{cp} — среднее время безотказной работы.

Основной количественной характеристикой надежности является вероятность безотказной работы $p(t)$, т.е. вероятность того, что за время t не произойдет отказа в работе. Ее значение может находиться в пределах от 0 до 1 (рис. 2.1): $p(0) = 1$; $p(\infty) = 0$; $0 \leq p(t) \leq 1$.

Вероятность безотказной работы элемента автоматики можно определить по результатам испытаний большого числа одинаковых элементов в течение заданного промежутка времени t :

$$p(t) = (N - n) / N,$$

где N — общее число испытанных элементов; n — число элементов, вышедших из строя за время испытаний.

Интенсивность отказов $\lambda(t)$, или λ -характеристика, очень часто используется для количественной оценки надежности элементов и при расчете надежности системы автоматики, состоящей из нескольких элементов. Величину λ можно оценить как отношение числа отказавших элементов к числу элементов, оставшихся к данному моменту времени работоспособными, полученное за единицу времени. Обычно единицей измерения интенсивности отказов является число отказов в час.

На рис. 2.2 приведена типичная кривая интенсивности отказов в зависимости от времени эксплуатации для большого числа однотипных элементов, изготовленных на одном и том же заводе по одинаковой технологии. На этой кривой можно выделить три

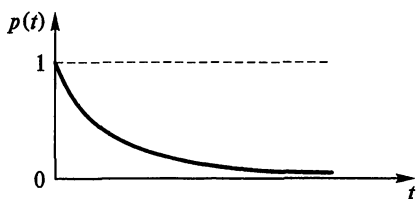


Рис. 2.1. Зависимость вероятности безотказной работы от времени



Рис. 2.2. Типичная зависимость интенсивности отказов изделия от времени эксплуатации

характерных участка. Первый участок от (0 до t_1) называется периодом приработки и тренировки. В этот период выходят из строя некачественно изготовленные элементы. Обычно этот период проходит на заводе-изготовителе, дорожащем своей репутацией. Дефектные элементы заранее, как говорится, «выжигают», не пуская их в продажу. Второй участок (от t_1 до t_2) — это период нормальной эксплуатации элемента, в течение которого интенсивность отказов низкая и примерно постоянная. На этом участке вероятность безотказной работы определяется по формуле $p(t) = \exp(-\lambda t)$.

Третий участок начинается с момента t_2 и характеризуется нарастанием интенсивности отказов, что объясняется старением и износом элементов. Обычно рекомендуется производить замену элементов до наступления момента времени t_2 .

Среднее время безотказной работы при постоянной интенсивности отказов

$$T_{\text{ср}} = 1/\lambda.$$

Например, если известно, что какой-либо элемент имеет интенсивность отказов $\lambda = 2 \cdot 10^{-6}$ отказов в час, то из миллиона таких элементов каждый час будут в среднем выходить из строя два элемента. Среднее время безотказной работы такого элемента составит 500 тыс. часов.

Следует отметить, что на величину интенсивности отказов и, следовательно, на среднее время безотказной работы очень сильно влияют условия эксплуатации. В вычислительных центрах создаются наилучшие условия для эксплуатации ЭВМ: поддерживается комфортная температура; помещение вентилируется, а воздух очищается; обслуживающий персонал одет в чистую одежду. Компьютеры, установленные в офисах и квартирах, в большей степени подвержены таким отрицательным воздействиям, как колебания температуры и особенно запыленность. Еще более сильным является воздействие окружающей среды на вычислительную технику, используемую для автоматизации производства. В этом случае к перечисленным ранее отрицательным факторам добав-

ляются вибрации и влияние электромагнитных полей. В наиболее сложных условиях работает так называемая бортовая вычислительная техника, устанавливаемая на движущихся объектах: автомобилях, кораблях, летательных аппаратах.

Важным показателем надежности работы ЭВМ является также среднее время восстановления работоспособности после отказа. Сокращению этого времени во многом способствуют развитые системы самодиагностики ЭВМ, позволяющие определить элемент, подлежащий замене. Процентное соотношение времени безотказной работы к сумме этого времени со временем восстановления работоспособности называют коэффициентом готовности (или доступности). Для высоконадежных ЭВМ этот коэффициент достигает 99,999 %, т.е. на простой уходит не более тысячной доли процента. Иными словами, за год работы только 5 мин уходит на простой.

2.2. Классификация ЭВМ

Классифицировать ЭВМ можно по различным признакам.

Прежде всего все вычислительные машины подразделяются на *аналоговые* (АВМ) и *цифровые* (ЦВМ). С 50-х годов XX в., когда пошел процесс бурного развития цифровых вычислительных машин на базе электронных схем, стали использовать термин «электронные цифровые вычислительные машины» (ЭЦВМ). Затем, когда благодаря успехам микроэлектроники эти машины получили преимущественное (по сравнению с аналоговыми) распространение, прилагательное «цифровые» отбросили, как само собой разумеющееся. Теперь, когда говорят ЭВМ, то имеют в виду именно цифровые машины, хотя и АВМ тоже чаще всего являются электронными. В этой книге термин ЭВМ используется применительно к цифровым вычислительным машинам, если нет специальной оговорки.

В наиболее общей классификации ЭВМ подразделяют по мощности, т.е. по производительности.

Самые мощные ЭВМ получили название *суперЭВМ*. Для оценки их производительности используется единица измерения терафлопс (см. подразд. 2.1). Такие машины представляют собой по существу вычислительные комплексы, состоящие из нескольких шкафов или стоек, в которых размещены сотни или тысячи процессоров. Потребляемая мощность таких комплексов достигает сотен киловатт. Они способны решать самые сложные вычислительные и моделирующие задачи в реальном масштабе времени. Между фирмами, разрабатывающими и производящими такие суперЭВМ, идет своеобразное соревнование за лидерство. В сети Интернет на специальном сайте можно найти информацию о 500 самых мощных в мире на данное время суперЭВМ.

На ступеньку ниже по сравнению с суперЭВМ можно поместить *мэйнфреймы* (см. Введение) и *серверы*. Они содержат несколько микропроцессоров и способны взаимодействовать одновременно с несколькими ЭВМ. Вообще-то понятие «сервер» было введено для того, чтобы характеризовать функции наиболее мощной ЭВМ в сети, объединяющей несколько ЭВМ обычной производительности. Сервер призван обслуживать эти ЭВМ, предоставлять им свои ресурсы (например, снабжая их сведениями из хранящихся на сервере баз данных, решая для них сложные задачи), регулировать взаимодействие этих ЭВМ в сети и т. п. Именно для выполнения функций сервера начали разрабатывать ЭВМ с большим объемом памяти и высоким быстродействием. Такие ЭВМ стали называть серверами. Но в небольших сетях роль сервера иногда можно поручить и обычной ЭВМ, не отличающейся от других ЭВМ этой же сети ни по производительности, ни по объему памяти. В этом случае термин «сервер» относится именно к функциональному назначению ЭВМ. Часто при этом к слову «сервер» добавляются еще слова, характеризующие назначение данного сервера: сервер файлов, сервер печати, сервер баз данных, сервер удаленного доступа и др.

Наиболее распространенный вид ЭВМ — *персональные компьютеры*. Современный ПК имеет столь высокую производительность, что может использоваться в качестве сервера при объединении в сеть нескольких ПК. Благодаря весьма развитому программному обеспечению на ПК можно выполнять и создавать самые разнообразные программы.

При объединении нескольких ПК в сеть каждый из них может использовать память и программное обеспечение, которые имеются в мэйнфрейме или сервере данной сети. В этом случае собственная память каждого ПК в сети может быть очень небольшой по объему. Главное, что требуется от этого ПК, это возможность ввода информации для управления процессом вычислений и вывод информации, т. е. отображение результатов процесса вычислений. Проще говоря, достаточно иметь клавиатуру и экран. Такое устройство называется терминальным, или коротко терминалом. Довольно часто терминалом (или рабочей станцией) называют подключенный к сети ПК.

Современные ПК имеют различные конструктивные исполнения. Наиболее распространенной и привычной является настольная конструкция ПК. В ее состав входит системный блок (собственно говоря, именно он и является компьютером), устройство вывода информации — монитор (его экран называют также дисплеем) и устройство ввода, т. е. клавиатура и мышь. Все эти составные части могут быть объединены в одном небольшом наколенном ПК, называемом *лэптоп*. Он представляет собой переносной компьютер.

Еще легче и меньше по размерам *ноутбук*. Он имеет аккумулятор, способный обеспечивать ПК электроэнергией в течение полтора часов и более. Следовательно, пользоваться таким ПК можно где угодно.

Следующий вид — *встраиваемые ЭВМ*, которые предназначены для управления технологическими и производственными процессами. Они получают информацию от датчиков, контролирующих ход того или иного процесса, и выдают управляющие сигналы на исполнительные устройства, изменяющие ход этого процесса в нужном направлении. Эти ЭВМ «встраиваются» в систему автоматического управления (САУ), являются ее частью и взаимодействуют с другими элементами САУ. Обычно алгоритм выработки управляющих сигналов уже заложен в ЭВМ (аппаратным или программным путем), а человек-оператор может изменять настройки (т. е. отдельные количественные показатели) этого алгоритма. Встраиваемые ЭВМ называют также управляющими.

В настоящее время получили очень широкое распространение различные бытовые приборы с процессорным управлением (стиральные машины, микроволновые печи, кухонные автоматы и т. п.). В них заложен небольшой набор фиксированных программ, выбор между которыми осуществляет пользователь.

Процессор (важнейший элемент ЭВМ) используется и в многочисленных игровых приставках, в которых заложены несколько неизменяемых программ. В процессе игры входные сигналы вводятся игроком-пользователем с помощью небольшой клавиатуры, а на небольшом экране-дисплее отображается игровая ситуация. Мобильные телефоны также включают в себя элементы вычислительной техники, обеспечивающие соединение с абонентом, хранение в памяти информации и т. п. Современные мобильные телефоны по существу уже являются многофункциональными устройствами и содержат большое количество элементов вычислительной техники.

Многочисленные калькуляторы также являются простыми ЭВМ со сравнительно низким быстродействием и небольшим набором выполняемых математических операций.

На нижнюю ступень можно поставить простые игры для маленьких детей всего с единственной программой и миниатюрным дисплеем.

Четкую грань между всеми рассмотренными видами ЭВМ провести трудно. Нарращивать информационную и вычислительную мощности можно путем объединения нескольких ЭВМ. Например, из нескольких мэйнфреймов можно получить суперЭВМ, добавив несколько стоек или шкафов, в которых смонтированы элементы вычислительной техники. Порой так делают лишь на период решения какой-либо сложной задачи, а после ее выполнения составные части суперЭВМ снова эксплуатируют порознь.

Такую «сборную» ЭВМ иногда называют *кластер*. Из нескольких серверов (даже из двух) можно образовать мэйнфрейм, если с помощью специальных программ обеспечивается одновременное выполнение нескольких задач с нескольких терминалов (т.е. обычных ПК). В свою очередь, сервером можно считать и высокопроизводительный ПК, в который можно быстро вставлять несколько дополнительных жестких дисков. Еще раз напомним, что функции сервера может в локальной сети выполнять и обычный персональный компьютер.

Примерная классификация вычислительных машин показана на рис. 2.3.

Первыми вычислительными машинами были аналоговые вычислительные устройства. Их возможности не столь универсальны, как у цифровых машин, но они обладают определенными достоинствами, благодаря которым достаточно широко применяются в производственной и научной сферах деятельности человека. В быту человек с ними встречается не часто, поэтому они менее известны, чем цифровые.

Механические аналоговые вычислительные устройства благодаря своей простоте (а следовательно, низкой стоимости и малым расходам на эксплуатацию) достаточно широко используются при подсчете изделий на конвейере, определении площадей на картах и графиках, типовых расчетах по заранее известным простым формулам и т.п.

Электромеханические АВМ до сих пор широко применяются в судовождении для навигации (прокладка пути) и обсервации (определение места), при управлении технологическими процессами, требующими определенных соотношений между небольшим числом регулируемых параметров, и в иных подобных слу-

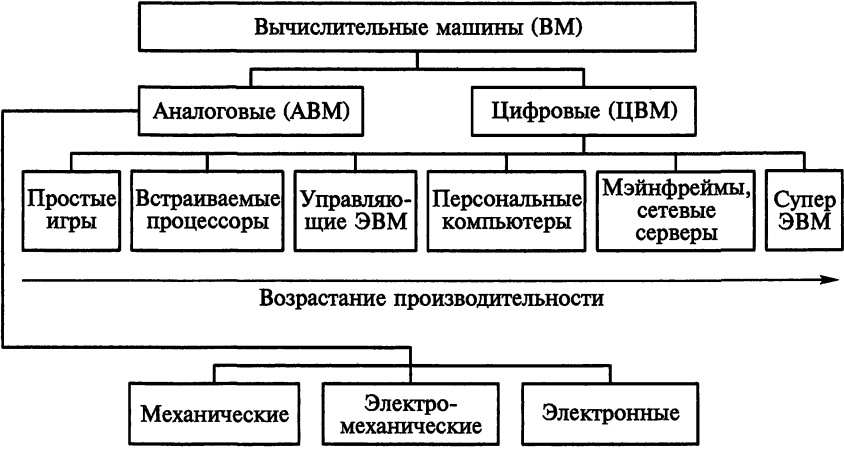


Рис. 2.3. Классификация вычислительных машин

чаях, когда необходим расчет по заранее известным простым формулам.

Электронные ЭВМ весьма широко используются для моделирования процессов, описываемых дифференциальными уравнениями. Это обширнейшая область применения. К тому же разработаны способы моделирования и для нелинейных процессов.

На рис. 2.3 цифровые ЭВМ расположены слева направо в порядке возрастания производительности. Но и здесь возможны исключения. Например, управляющие ЭВМ для пилотирования сверхзвуковых аппаратов или для слежения за подобными объектами должны работать в реальном масштабе времени и иметь очень высокую производительность, хотя большой набор программ для них и не требуется.

Классифицируют ЭВМ и по таким признакам, как режимы работы и обслуживания, структурный состав, способы функционирования и представления информации, размещение, назначение и др.

По режиму работы ЭВМ подразделяются на однопрограммные и мультипрограммные.

Однопрограммные ЭВМ могут работать только в режиме, при котором выполняется не более одной программы пользователя. Они переходят к выполнению следующей программы лишь по окончании предыдущей. Ресурсы ЭВМ используются только действующей программой. При работе ЭВМ возможны простои на период ввода или вывода информации.

Мультипрограммные ЭВМ могут работать в режиме, при котором одновременно выполняется несколько программ пользователя. Применение этого режима позволяет более эффективно использовать оборудование машины путем совмещения операций (например, ввода, вывода, счета), уменьшить время, потребное для решения комплекса задач, с некоторым увеличением времени для решения каждой отдельной задачи. Последнее достигается за счет уменьшения времени ожидания очереди на решение.

По режимам обслуживания ЭВМ подразделяют на ЭВМ индивидуального пользования, пакетной обработки и коллективного пользования.

ЭВМ *индивидуального пользования* наиболее удобны в эксплуатации. Все ресурсы ЭВМ предоставляются одному пользователю, но при этом эффективность применения аппаратных средств и программного обеспечения снижается.

ЭВМ с режимом *пакетной обработки* выполняют программы, записанные заранее во внешней памяти. Пользователи при этом не имеют непосредственного доступа к ЭВМ, т. е. программы не могут выполняться в режиме диалога. Для обработки программ их приходится несколько раз (после исправлений) последовательно вводить в пакеты программ. Эффективность работы при этом снижается.

ЭВМ с режимом *коллективного пользования* предоставляют аппаратные и программные ресурсы одновременно нескольким пользователям. При этом каждому пользователю выделяется терминал, с помощью которого он связывается с ЭВМ, следит за выполнением своих программ, осуществляет запросы к ЭВМ. Режим коллективного пользования может быть реализован только высокопроизводительными вычислительными системами.

По структурному составу ЭВМ подразделяют на *однопроцессорные* (одномашинные), *многомашинные* и *многопроцессорные*. Многопроцессорные ЭВМ в отличие от многомашинных могут работать с общим полем оперативной памяти.

Многопроцессорные и многомашинные ЭВМ создаются как для увеличения производительности, так и для повышения надежности. В ЭВМ повышенной надежности выход из строя одной машины не приводит к полному отказу всей вычислительной системы, так как функции отказавшего устройства могут взять на себя другие. Многопроцессорные и многомашинные ЭВМ повышенной надежности обычно имеют избыточные резервные устройства и специальные средства управления ресурсами для отключения неисправных устройств и подключения резервных.

По способу размещения ЭВМ и вычислительные системы подразделяют на *сосредоточенные*, с телеобработкой и вычислительные сети.

Оборудование *сосредоточенных* ЭВМ расположено практически в одном помещении, а устройства связываются между собой по стандартным внутренним интерфейсным линиям связи.

У вычислительных систем с *телеобработкой* некоторые источники и потребители информации, в том числе и терминалы, могут располагаться на удалении от вычислительного ядра. Для связи с удаленными терминалами и другими периферийными устройствами используются стандартные и нестандартные линии связи.

Вычислительные сети представляют собой территориально разнесенные многомашинные системы, объединенные каналами связи.

По способу функционирования ЭВМ подразделяются на ЭВМ, работающие в *реальном* масштабе времени, и ЭВМ, работающие *независимо от реального времени*.

ЭВМ реального времени способны к быстрой реакции на запросы, обладают высоким быстродействием и имеют в своем составе таймерное (временное) устройство. Таймерное устройство (таймер) необходимо для отслеживания времени и установки задаваемых временных интервалов. ЭВМ реального времени используются обычно в системах управления объектами и технологическими процессами. Результаты обработки параметров состояний управляемых объектов и процессов в таких системах непосредственно используются для воздействия на эти объекты или процессы. Чем меньше время реакции на таймерный или внешний

запрос, чем быстрее ЭВМ определяет воздействие на объекты управления после запросов, тем лучше будет функционировать система управления.

По назначению ЭВМ подразделяются на управляющие, информационно-справочные и общего назначения.

Управляющие (а некоторые из них ранее были упомянуты как встраиваемые) ЭВМ работают обычно в реальном масштабе времени. Часто они являются структурными составляющими специализированных систем управления, например корабельных, самолетных, ракетно-космических и т. п. В этих случаях специализированные управляющие ЭВМ отличаются от обычных повышенными требованиями к массогабаритным характеристикам и условиями эксплуатации, но, в первую очередь, — надежностью. В связи со спецификой использования и стабильностью программного обеспечения в специализированных ЭВМ сокращены или полностью отсутствуют некоторые программные средства управления вычислительным процессом, связанные с загрузкой в память программ, динамическим распределением памяти и т. п. В них шире используется постоянная память для хранения программ и констант, уменьшен формат операндов (обрабатываемых данных), сокращен список команд. Управляющие ЭВМ предназначены для автоматизации различных процессов, позволяют изменять (перенастраивать) программы управления, но набор программ у них существенно меньше, чем у ПК. Поскольку благодаря массовому выпуску ПК их стоимость непрерывно снижается, порой целесообразно не создавать специализированную ЭВМ, а использовать в качестве таковой обычный ПК.

Специализация ЭВМ в смысле их проблемной ориентации в некоторых случаях ограничивается выбором конфигурации и состава аппаратных средств, а также программ операционных систем и пакетов прикладных программ. Подобным способом может быть достигнута, например, ориентация (в смысле — предназначение) ЭВМ для информационно-справочных систем, т. е. *информационно-справочных* ЭВМ. Они в основном предназначаются для большого числа неквалифицированных пользователей, таких как покупатели супермаркетов, читатели библиотек, пассажиры на вокзалах и т. п.

ЭВМ *общего назначения* состоят из максимального количества предусмотренных для данного типа ЭВМ устройств, а их математическое обеспечение должно быть приспособлено для решения задач любого типа.

По способам передачи и обработки информации различают:

- *параллельные* ЭВМ, ведущие передачу и обработку параллельно по всем разрядам;
- *последовательные* ЭВМ, в которых передача и обработка чисел идет последовательно разряд за разрядом;

- *смешанные ЭВМ*, осуществляющие передачу и обработку параллельно по группе разрядов и последовательно по группам разрядов, как, например, в байте параллельно обрабатываются все 8 двоичных разрядов, а в слове, состоящем из нескольких байтов, обрабатывается последовательно байт за байтом.

По числу адресов, содержащихся в составе команды, различают *трех-, двух-, одно- и нульадресные ЭВМ*.

По системам счисления различают *двоичные ЭВМ*, в которых все вычисления ведутся в двоичной системе счисления, имеющей только две цифры — 0 и 1, и *десятичные ЭВМ*.

По форме представления чисел различают:

- *ЭВМ с фиксированной запятой*, позволяющие вести счет при условии, что запятая для всех вводимых в данной задаче чисел будет фиксирована на строго определенном месте, которое может быть выбрано перед началом решения задачи;

- *ЭВМ с плавающей запятой*, не требующие фиксации запятой в определенном месте. Для фиксации порядка числа выделяются специальные разряды, косвенно определяющие место расположения запятой для каждого числа в отдельности. Цифровая ЭВМ с плавающей запятой может работать в режиме с фиксированной запятой. При этом разряды, отводимые для представления порядка числа, используются для расширения диапазона представления чисел с фиксированной запятой.

По структуре различают:

- *ЭВМ с жесткой структурой*, не позволяющие изменять структуру, вводить дополнительные устройства или заменять одни устройства другими;

- *ЭВМ с модульной структурой*, строящиеся на основе легко заменяемых модулей. В качестве модуля может рассматриваться арифметико-логическое устройство, часть оперативной памяти или даже весь процессор. Такая структура позволяет варьировать состав модулей и тем самым изменять основные параметры ЭВМ в довольно широких пределах.

По характеру управления различают:

- *синхронные ЭВМ*, работающие в жестко заданных временных интервалах, которые вырабатываются в виде тактовых импульсов специальным генератором;

- *асинхронные ЭВМ*, работающие без жестко заданных интервалов. Время начала следующей операции определяется окончанием предыдущей.

2.3. Развитие производства ЭВМ

Первые модели ЭВМ, появившиеся в середине XX в., зачастую создавались в единственном экземпляре, в процессе испытаний которого выявлялись недостатки и определялись пути их

устранения. Таким образом, следующая модель уже существенно отличалась от предыдущей. Наиболее удачные модели выпускались в нескольких экземплярах, но размеры серии были невелики — в пределах сотни.

В нашей стране в 1960-е годы серийно выпускались машины семейства «Днепр», «Проминь» и др. Последовательное улучшение характеристик БЭСМ привело к созданию самой производительной модели БЭСМ-6 (см. подразд. 1.1), которая также выпускалась серийно.

Подлинный рывок в производстве ЭВМ произошел после появления персональных компьютеров. В 1976 г. двое молодых людей Стив Джобс и Стефан Возняк увлеклись идеей создать доступную по цене ЭВМ, в которой можно было бы устанавливать программы для любых игр. Оба они посещали клуб любителей компьютерных игр (в то время очень простых). Для каждой игры существовало отдельное устройство, перепрограммировать которое на другую игру было невозможно. Играть в несколько игр можно было только на довольно большой стационарной ЭВМ, установленной в клубе. С. Джобс и С. Возняк видели огромную увлеченность молодежи компьютерными играми и поэтому считали, что спрос на небольшую ЭВМ с возможностью перепрограммирования на различные игры будет обеспечен.

Технической основой для реализации проекта должна была стать созданная к тому времени сверхбольшая интегральная схема, объединяющая в себе столько полупроводниковых элементов, что они могли бы выполнять функции всего процессорного устройства ЭВМ. Это был микропроцессор фирмы Intel. Для реализации проекта нужны были деньги — начальный капитал. Друзья продали все, что у них было более-менее ценного (подержанный автомобиль и калькулятор), заняли на месяц еще 15 тыс. дол. и купили на 20 тыс. дол. радиодетали для сборки полусотни компьютеров. За месяц упорной работы они собрали 50 компьютеров, быстро их продали, рассчитались по долгам и остались с прибылью. Для организационного подкрепления всей этой затеи 1 апреля 1976 г. была создана фирма Apple Computer. В январе 1977 г. фирма взяла кредит уже на 250 тыс. дол. и выпустила первый в истории персональный компьютер в пластиковом корпусе и с цветной графикой. Стоил он 1 300 дол. Через шесть лет объем производства ПК фирмы Apple Computer достиг миллиарда долларов. Это — рекорд в мире бизнеса.

Популярность яблочного компьютера (*англ.* apple — яблоко) привлекла к идее выпуска ПК крупные фирмы, способные вкладывать в это дело большие деньги. Лидером здесь была (и остается) фирма IBM. Свой персональный компьютер эта фирма выпустила в 1981 г. на базе микропроцессора Intel 8088 с тактовой частотой 4,77 МГц и памятью 16 Кбайт. За месяц было продано

более 200 тыс. ПК. К концу 1980-х годов ежегодный выпуск ПК фирмой IBM превысил миллионный рубеж, а число компьютеров в США составило более 50 млн.

Во всем мире к началу 1990-х годов пользователей ПК насчитывалось уже более 100 млн. В наши дни общее число компьютеров во всем мире превысило миллиард, при этом половина из них объединены в мировую сеть, т. е. имеют выход в сеть Интернет. Число владельцев мобильных телефонов также близко к миллиарду.

Следует отметить, что по отдельным странам компьютеры распределены весьма неравномерно. В развитых странах на каждую сотню человек приходится около 50 компьютеров (или даже больше).

В среднем по всем странам мира только каждый десятый человек имеет компьютер. Наша страна по числу компьютеров на душу населения пока занимает место в третьем десятке среди более чем 200 стран мира. Но и у нас распределение компьютеров по отдельным регионам весьма неравномерно. В Москве, например, средняя обеспеченность компьютерами такая же, как в среднем во всей Европе.

Средства вычислительной техники очень широко применяются теперь в устройствах связи, т. е. для телекоммуникации. Фактически произошло слияние (объединение, интеграция, конвергенция) вычислительной техники и телекоммуникационной техники. Например, современные мобильные телефоны и телефонные станции, обеспечивающие их взаимное соединение (коммутацию), включают в себя такие элементы вычислительной техники, как микропроцессоры и запоминающие устройства. Поэтому и телекоммуникационное оборудование с такими элементами может быть отнесено к средствам вычислительной техники. А это означает, что общее количество таких средств в мире надо удвоить, поскольку количество мобильных телефонов уже достигло миллиарда.

Таким образом, можно считать, что во всем мире сейчас существует несколько миллиардов различных средств вычислительной техники — от самых простых (детские игрушки и дешевые калькуляторы) до суперкомпьютеров.

Современные ПК по своим вычислительным и информационным возможностям уже давно превосходили вычислительные машины, созданные в середине прошлого века и составлявшие гордость вычислительных центров, которые обслуживали сотни и тысячи пользователей. Причем, как правило, пользователями являлись не отдельные ученые и инженеры, а целые коллективы — институты и лаборатории. С каждым годом быстродействие и объем памяти ПК возрастают. Соответственно прежние модели ПК устаревают, хотя остаются еще вполне работоспособными. Можно сказать, что для ПК моральный износ наступает раньше физического.

Интересно рассмотреть самые мощные из существующих ЭВМ (т.е. суперЭВМ), поскольку их характеристики являются тем ориентиром, к которому стремятся при создании новых ПК.

В 2002 г. самая мощная ЭВМ была изготовлена в Японии фирмой NEC, ее быстродействие составляет 35,86 терафлопс. Создав такую ЭВМ, японская фирма NEC совершила гигантский скачок, поскольку ее машина впервые возглавила список самых производительных суперЭВМ за все годы. До 2002 г. первенство в этом списке принадлежало США (при этом особенно большое внимание суперЭВМ уделяла фирма Стру). Да и в десятке по состоянию на лето 2004 г. семь суперЭВМ — это американские машины. По одной суперЭВМ из этого списка принадлежали Японии (1-е место), Франции (7-е место) и Великобритании (9-е место). Надо отметить, что Япония лидировала с большим отрывом. Второе и третье места занимали американские машины фирмы Hewlett Packard, имеющие производительность 7,727 терафлопс, т.е. примерно в 4,5 раза менее быстродействующие. На четвертом месте стояла также американская машина, но уже фирмы IBM, которая является мировым лидером в производстве ПК. Производительность указанной машины 7,226 терафлопс, она возглавляла аналогичный список в 2000 г. «Чемпион мира» 1999 г. занимал летом 2004 г. 15-е место (2,379 терафлопс), чемпион 1998 г. — 28-е место (1,608 терафлопс), 1997 г. — 65-е место (0,815 терафлопс), а чемпион 1996 г. был на 167-м месте (0,3682 терафлопс). Интересно отметить, что лидер списка 1995 г. не попал в список лучших уже в 2002 г. Вот как быстро развивается производство ЭВМ и растет их мощность! Замыкала список 500 самых лучших суперЭВМ машина фирмы Hewlett Packard. Машин такого типа изготовлено уже более двух десятков, т.е. можно сказать, что суперЭВМ теперь выпускаются серийно. Нашей стране в списке 500 лучших летом 2004 г. принадлежало скромное, но все же почетное 74-е место. Его занимала машина MVS1000M производительностью 0,7346 терафлопс.

Список лучших суперЭВМ непрерывно меняется, поскольку каждый год появляются все более совершенные машины. Японская суперЭВМ держалась на первом месте более двух лет. В конце 2004 г. список возглавила американская машина производительностью 70,72 терафлопс. В десятке лучших суперЭВМ уже все из них имели производительность более 10 терафлопс. Суммарная же производительность всех суперЭВМ, входивших в список 500 лучших, превысила отметку 1 000 терафлопс и составила на начало 2005 г. 1,127 петафлопс (пета — 10^{15}). К середине 2007 г. такую суммарную производительность имеют уже первые 15 суперЭВМ из списка лучших 500 суперЭВМ.

К 2007 г. на первое место вышла американская ЭВМ, изготовленная фирмой IBM, с производительностью 280 терафлопс, т.е.

за 2 года производительность увеличилась в 4 раза. В этой машине более 130 тыс. процессоров. В десятке лучших все суперЭВМ имеют производительность более 40 терафлопс, т. е. и по этому показателю за 2 года произошло увеличение в 4 раза.

Японский суперкомпьютер, изготовленный фирмой NEC, находится на 9-м месте с производительностью более 47 тыс. терафлопс. Он содержит более 11 тыс. процессоров. Вообще характерной чертой всех суперкомпьютеров является наличие большого количества процессоров.

Самый мощный российский суперкомпьютер МВС-15000ВМ содержит 1 148 процессоров и имеет производительность более 6 терафлопс (т. е. по сравнению с 2004 г. этот показатель вырос почти в 10 раз). Он занимает 99-е место в списке суперкомпьютеров мира. В этом списке появился и еще один работающий в России суперкомпьютер (407-е место), но он изготовлен фирмой Hewlett-Packard.

По количеству ПК наша страна пока существенно отстает от США и ряда других развитых стран. Однако по темпам роста числа ПК в стране Россия выглядит совсем неплохо: каждые пять лет число ПК увеличивается втрое. Но догонять придется еще долго. Ведь по количеству ПК на душу населения мы отстаем от США более чем в 10 раз, а от Японии — в 5 раз. По этому показателю мы находимся на уровне таких стран, как Аргентина, Малайзия, Польша, Чили, и опережаем Индию, Индонезию, Перу, Румынию, Турцию, Украину и ряд других.

Чем же объяснить такое состояние производства вычислительной техники в России? Ведь на заре развития ЭВМ наши машины были по крайней мере лучше европейских и японских. Причин много, некоторые из них связаны с вынужденной изоляцией СССР от мирового рынка в период холодной войны, экономическими трудностями, возникшими в затянувшийся период перехода от плановой экономики к рыночной. Но конкретная техническая причина заключается в слабости нашей микроэлектронной промышленности. Ведь все успехи мирового производства ПК определяются высокими технологиями (хай-тек) в производстве сверхбольших интегральных схем микропроцессоров, являющихся сердцем, а точнее — мозгом ЭВМ. Примерно за 30 лет производства микропроцессоров число транзисторов в одном кристалле возросло с 2 тыс. до 40 млн (в 20 тыс. раз!), тактовая частота выросла в 15 тыс. раз. И все это в значительной степени благодаря высоким технологиям. В первых микропроцессорах расстояние между отдельными транзисторами составляло 3 мкм (три микрометра, или микрона; отсюда появилось выражение «трехмикронная технология»). Первые микропроцессоры Пентиум уже использовали технологию 0,5 и даже 0,35 микрона, а в Пентиум 4 использована технология 0,18 микрона. Уже есть сообщения об использовании

технологии 0,08 и даже 0,065 микрона. В одном микрометре (микроне) — 1 000 нанометров ($1 \text{ мкм} = 10^3 \text{ нм}$), поэтому технологию 0,065 микрона можно называть технологией 65 нанометров. Другими словами, речь идет уже о так называемой нанотехнологии — технологии ближайшего будущего, которая позволит еще более повысить возможности ПК и суперЭВМ. Но завод, выпускающий микроэлектронные изделия по такой технологии, обойдется в несколько миллиардов долларов. Одного желания построить такой завод мало, надо, чтобы страна была богатой.

Почему же уменьшение расстояния между транзисторами в микросхеме позволяет так существенно повысить вычислительную мощность, а точнее — быстродействие? Напомним, что в ЭВМ сигналы распространяются по электрическим проводникам со скоростью света, которая составляет примерно 300 000 км/с. Чем короче проводник, тем быстрее идет электрический сигнал от одной части схемы к другой. Но бесконечно уменьшать расстояние между транзисторами невозможно. Меньше размеров атома оно быть не может. Поэтому ученые работают над новыми принципами построения микропроцессоров.

Базовый диэлектрик (gate dielectric) в современных компьютерных чипах (микросхемах) представляет собой слой оксида толщиной около десятка атомов, и дальнейшее существенное уменьшение размеров, а значит, и повышение быстродействия чипов на основе кремниевых технологий не представляются возможными.

Одна из стратегий для создания следующего поколения на порядок более производительных вычислительных устройств, так называемых квантовых компьютеров, основана на использовании квантовой механики для решения сверхсложных вычислительных задач. Наш соотечественник Кирилл Болотин, работающий в области физики наносистем в Колумбийском университете в США, объясняет принцип действия квантового компьютера: «В классическом компьютере (таком, как, например, ваш Пентиум) информация хранится в битах, где каждый бит представляет собой единицу или ноль. В квантовом компьютере единица хранения информации, квантовый бит, наряду с единицей и нулем может находиться в состоянии квантовой суперпозиции между этими двумя значениями, например квантовый бит может быть на 25 % единицей и на 75 % нулем. Суперпозиция между различными «квантовыми битами» обуславливает возможность принципиально нового класса вычислений при помощи квантового компьютера: Например, объем информации, который можно закодировать всего лишь в 300 квантовых битах эквивалентен примерно 10^{90} классическим битам, больше, чем суммарный объем всей информации на Земле. Несмотря на столь очевидную привлекательность квантовые компьютеры пока еще не вышли из лабораторий и на пути к их практической реализации все еще большое число нерешенных проблем».

шенных технических проблем. Первые шаги, однако, уже сделаны: канадская компания D-Wave Systems анонсировала первый коммерческий прототип квантового компьютера Oqion, который можно будет арендовать для проведения ресурсоемких вычислений. Заявленный размер «памяти» этого компьютера — 16 квантовых бит».

Контрольные вопросы

1. Какой смысл вкладывается в понятие «машина»?
2. Какие бывают машины?
3. Что означает понятие «производительность» для любой машины?
4. Какие показатели определяют надежность?
5. Какой вид ЭВМ наиболее распространен в настоящее время?
6. Что такое суперЭВМ?
7. Какое электронное вычислительное устройство самое дешевое?
8. В каких бытовых приборах имеются вычислительные устройства?
9. Примерно сколько всего в мире персональных компьютеров?
10. Примерно сколько ПК приходится на одного жителя России?

ПРИНЦИП ДЕЙСТВИЯ ЭВМ

3.1. Функциональная схема ЭВМ

Немногим более полувека прошло после появления первой цифровой ЭВМ. За это время во всем мире было создано не менее миллиарда различных ЭВМ, которые отличаются друг от друга по конструкции, конфигурации (т.е. набору устройств), техническим характеристикам. Однако функциональная схема каждой ЭВМ (рис. 3.1) содержит три основных устройства: процессорное (ПУ), запоминающее (ЗУ), ввода — вывода (УВВ), взаимодействие которых обеспечивает выполнение основной задачи — обработку информации. Для снабжения всех устройств ЭВМ напряжениями нескольких стандартных уровней служит блок питания, который на рис. 3.1 не показан, поскольку он выполняет вспомогательную функцию, а не основную. Взаимосвязь трех основных функциональных устройств ЭВМ осуществляется по общей магистральной шине ОМШ, представляющей собой большое число проводов, по которым передаются информационные и управляющие сигналы. К ОМШ относятся также электросхемы, необходимые для коммутации (переключения) этих проводов. Каждое из трех основных функциональных устройств может и принимать, и передавать сигналы по ОМШ.

Процессорное устройство называется *центральным процессором*, или просто *процессором*. Это главное устройство ЭВМ, именно в нем происходит необходимая обработка информации, которая заключается в выполнении арифметических и логических операций над данными, представляющими собой кодовые комбинации. Последовательность этих операций задается командами. Команды также представляют собой кодовые комбинации. Набор команд для выполнения определенной задачи называется *программой*. Процес-

сор не только осуществляет обработку информации, но и управляет этим процессом. Управляющая часть процессора соединяет необходимые для решения данной задачи элементы во всех устройствах вычислительной машины. Эта часть контролирует номер выполняемого шага программы, вызывает из запо-

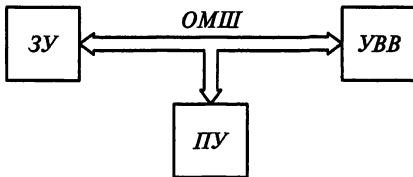


Рис. 3.1. Функциональная схема цифровой ЭВМ

минающего устройства очередную команду, настраивает элементы процессора на выполнение требуемой операции, вызывает из запоминающего устройства участвующие в данной операции данные, а по окончании операции направляет результат в запоминающее устройство.

Запоминающее устройство хранит всю информацию, необходимую для дальнейшего использования в процессоре. Поэтому данное устройство для краткости называют *памятью*, или *внешней памятью*, так как оно находится вне процессора. Некоторую часть информации необходимо сохранять и в самом процессоре, например очередную команду, промежуточные результаты операций. Поэтому процессор также имеет запоминающее устройство, но его называют *оперативной памятью*. Во внешней памяти хранится во много раз больше информации, чем в оперативной памяти, но для передачи информации из внешней памяти в процессор требуется больше времени, чем при использовании оперативной памяти. Поэтому при выполнении той или иной задачи процессор предварительно обращается к внешней памяти и переносит из нее необходимую информацию в оперативную память.

Устройство ввода — вывода обеспечивает ввод всей необходимой для выполнения задачи информации (прежде всего исходных данных и программ) и вывод результатов в удобной для пользователя форме. Ввод информации может осуществляться с помощью клавиатуры, читающего устройства (преобразователя графической информации в электрический сигнал), микрофона (преобразователя звуковой информации в электрический сигнал) и иных устройств, которые с каждым годом становятся все разнообразнее.

Все три основных функциональных устройства ЭВМ соединены между собой проводами. Чтобы увеличить скорость обмена информацией между устройствами, используются несколько параллельных проводов. Такое объединение проводов (или печатных проводников на электронной плате) называется *шиной*. В ЭВМ имеются несколько шин различного назначения: шины управления, шины данных, шины адресов и др.

Рассмотренная структура ЭВМ является универсальной, т. е. общей для ЭВМ разных конструкций и назначений. Общим является то, что в основу работы всех ЭВМ положен принцип программного управления, основанный на операционно-адресной организации процесса обработки информации. Поэтому ЭВМ относятся к классу универсальных программных автоматов, для которых разработана соответствующая теория. Информация, перерабатываемая автоматом, состоит из данных и программы. Информация, воспринимаемая автоматом, делится на данные (например, числа) и команды. И данные, и команды представлены в виде набора определенных знаков, в качестве которых чаще всего

используются два: ноль (0) и единица (1). В таком условном (кодированном) виде информация является кодовой комбинацией и носит название *слово*. Каждое слово хранится в запоминающем устройстве автомата в ячейках памяти, любая из которых имеет соответствующий адрес (номер). Каждая команда выполняет определенное преобразование информации. Сам акт преобразования называется *операцией*. Операции выполняются над словами, заданными адресами ячеек, в которые они помещены. Команда содержит кодовое обозначение выполняемой операции и адреса тех слов, которые участвуют в данной операции.

В 1945 г. Джоном фон Нейманом были разработаны пять основных принципов организации работы ЭВМ как универсального программного автомата:

- 1) информация представляется в двоичном коде, отдельные ее элементы называются словами;
- 2) при обращении к словам разного назначения их различают не по способу кодирования, а по необходимости использования;
- 3) слова размещаются в памяти и определяются адресами соответствующих ячеек;
- 4) алгоритм представляется в виде последовательности команд, определяющих наименование операции и адрес слова;
- 5) команды выполняются в той последовательности, в какой они размещены в памяти.

Эти принципы были положены в основу разработки самых первых ЭВМ и сохраняют свою актуальность до сих пор.

Долгое время все ЭВМ строго соответствовали всем пяти неймановским принципам, команды выполнялись ими последовательно, одна за другой. Про такие машины говорят, что в них реализуется *управление потоком команд*. В последние два-три десятилетия в связи с развитием микроэлектронных технологий произошло усложнение процессорных устройств и соответствующее расширение их возможностей. Стало доступным параллельное выполнение нескольких команд, т. е. пятый принцип перестал быть обязательным. При решении некоторых задач команды выполняются по мере готовности необходимых для их выполнения исходных данных. Такая последовательность выполнения получила название *управление потоком данных*.

3.2. Основные узлы ЭВМ

Основными узлами ЭВМ являются: центральный процессор; внешние запоминающие устройства; устройство управления; периферийные устройства, к которым относятся устройства ввода—вывода. Необходимыми вспомогательными узлами являются блок питания и соединительные провода (кабели) между отдельными основными узлами.

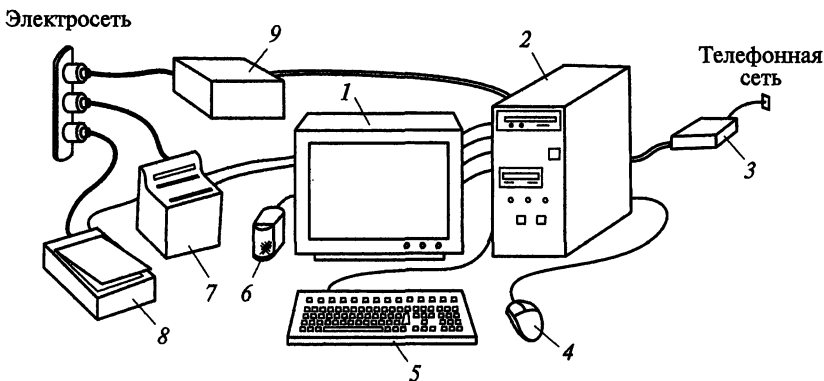


Рис. 3.2. Персональный компьютер с периферийными устройствами:
 1 — монитор; 2 — системный блок; 3 — модем; 4 — мышь; 5 — клавиатура;
 6 — звуковая колонка; 7 — принтер; 8 — сканер; 9 — источник бесперебойного питания

Рассмотрим эти основные узлы на примере наиболее широко распространенной ЭВМ — персонального компьютера. В минимальный состав ПК (рис. 3.2) входят системный блок 2, монитор 1 и клавиатура 5. В большинстве компьютеров теперь используется удобное устройство ввода — манипулятор типа мышь 4. Для ввода информации непосредственно с бумажных носителей применяется сканер 8. Устройства вывода информации являются принтер 7 и звуковые колонки 6. Для работы компьютера в сети служит модем 3. Для повышения надежности работы компьютера используют источник бесперебойного питания 9. В случае внезапного отключения напряжения в электросети это устройство короткое время (не менее 5 мин) обеспечивает питание ПК электроэнергией.

Системный блок со снятой крышкой показан на рис. 3.3, который состоит из двух видов (*a* и *б*), позволяющих лучше представить компоновку элементов в блоке. Части блока, изображенные на рис. 3.3, *a*, не показаны на рис. 3.3, *б*, чтобы они не загромождали другие элементы.

С каждым годом появляются все новые устройства, расширяющие возможности персонального компьютера. Для приема и просмотра на мониторе телевизионных программ к ПК подсоединяется ТВ-тюнер с телеантенной. Для обмена голосовыми сообщениями (телефонная связь) подсоединяется микрофон, а для участия в телеконференциях — видеочкама, с помощью которой собеседники видят друг друга. Большинство этих периферийных устройств подключается к системному блоку проводами через разъемы (гнезда) со стороны задней стенки блока. Там же находится и гнездо так называемой шины USB. Теперь к этой шине подключается очень много различных устройств, поэтому в современных

ПК такое гнездо (и чаще всего — не одно) вынесено на переднюю стенку системного блока. Самое широкое распространение получили устройства для хранения информации (накопители) типа флэш-память. Они как раз очень часто подключаются к ПК и именно через разъем USB. Широкое распространение флэш-памяти скорее всего приведет к тому, что от гибких дисков откажутся. Уже сейчас во многих новых ПК нет дисководов гибких дисков. А на его месте установлен небольшой блок, через который подключаются и диск типа флэш-память, и самые разные карты памяти, хранящие видеoinформацию, сделанную с помощью цифровых камер и мобильных телефонов, т.е. на месте устройства под номером 7 (или наряду с ним) может находиться устройство для подключения карт памяти и USB-устройств. В частности очень перспективно использование подключения инфракрасного порта (ИК-порт) через разъем USB. ИК-порт позволяет осуществить беспроводную связь ПК с самыми разными устройствами. Уже в настоящее время возможно использование клавиатуры и мыши без проводов.

Довольно часто системный блок называют компьютером, поскольку именно в нем сосредоточены те узлы, в которых осуществляется обработка информации. Центральный процессор в со-

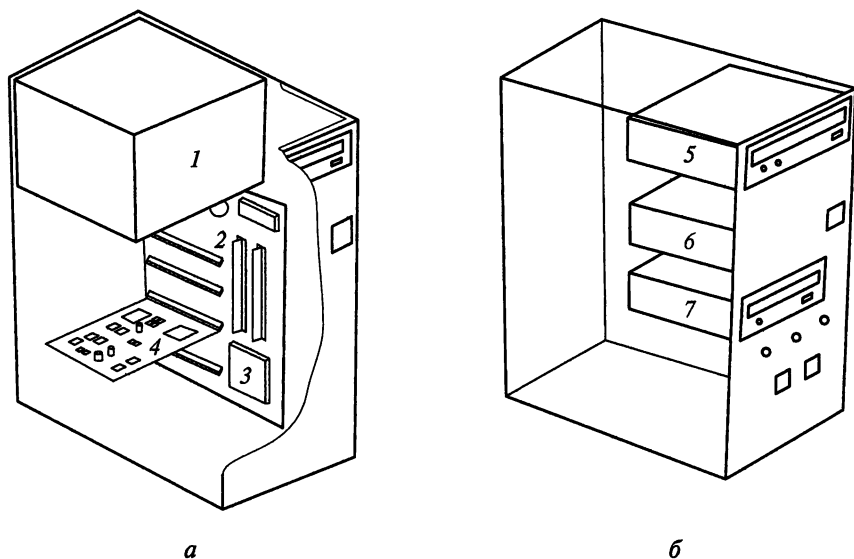


Рис. 3.3. Системный блок персонального компьютера:

а — вид с системной платой; *б* — вид с запоминающими устройствами; 1 — блок питания; 2 — системная плата; 3 — микропроцессор; 4 — электронная плата; 5 — дисковод компакт-дисков (CD-ROM); 6 — дисковод жестких дисков; 7 — дисковод гибких дисков

временном компьютере представляет собой единый электронный элемент, в котором с помощью современной технологии созданы миллионы полупроводниковых переходов и соединений между ними. Другими словами, это сверхбольшая интегральная микросхема, называемая микропроцессором. В микропроцессоре выполняются арифметические и логические операции по обработке информации, он содержит запоминающие устройства, в которых на время выполнения указанных операций хранится необходимая для этого информации. Кроме того, в микропроцессоре есть схемы управления, обеспечивающие взаимодействие его с другими устройствами компьютера. Микропроцессор 3 имеет несколько сотен выводов-ножек, с помощью которых он устанавливается в главную электронную плату 2 системного блока. По-английски эта плата называется *mainboard*, по-русски — *системной*, или *материнской, платой*. На системной плате размещены также специальные схемы для взаимодействия микропроцессора с другими узлами компьютера: монитором, клавиатурой, иными устройствами ввода и вывода информации. Эти схемы тоже являются интегральными, т. е. в состав каждой из них может входить много полупроводниковых переходов (но не так много, как в микропроцессоре). Раньше компьютер имел более 100 различных микросхем, обеспечивающих взаимодействие всех компонентов системного блока. Такой набор называется *чипсет* (*chip set*). В современных компьютерах благодаря высокой степени интеграции полупроводниковых элементов удается сосредоточить все необходимые функции взаимодействия микропроцессора с остальными устройствами всего лишь в двух сверхбольших интегральных микросхемах, которые получили названия *северный мост* и *южный мост*. Слова «северный» и «южный» обусловлены расположением этих микросхем на системной плате. Ближе к микропроцессору (как бы сверху) находится северный мост, южный мост расположен ниже — в районе разъемов для подключения внешних запоминающих устройств и устройств ввода и вывода. Слово «мост» означает, что данные микросхемы представляют собой систему соединений, т. е. как бы мостов, связывающих микропроцессор с другими устройствами.

На системной плате имеются также разъемы, называемые *слотами* (от *англ. slot* — щель), для подключения микросхем полупроводниковой оперативной памяти. У этой памяти быстродействие выше, чем у внешней, но ниже, чем у памяти, размещенной в самом микропроцессоре. Оперативная память может быть использована микропроцессором в процессе обработки информации. Обмен с ней микропроцессор осуществляет через северный мост. Кроме того, обычно через северный мост результаты обработки информации передаются на экран монитора, который подсоединен через электронную схему, называемую *видеокартой*.

На рис. 3.3, *а* показана вставленная в слот системной платы электронная схема (плата) 4 для дополнительных функций. Таких плат может быть несколько.

Южный мост содержит схемы управления (контроллеры) для подсоединения (объединения в компьютерную сеть) дисковых запоминающих устройств, клавиатуры, мыши и звуковых колонок. Через южный мост с помощью быстродействующего USB-порта могут быть подключены цифровые фото- и видекамеры, внешнее полупроводниковое запоминающее устройство типа флэш-памяти.

Микропроцессор обменивается информацией с северным мостом напрямую, а с южным — через северный. Для ускорения процесса обработки информации некоторые функции южного моста могут передаваться северному. Такая возможность появляется по мере совершенствования технологии производства наборов микросхем (чипсетов).

На системной плате размещены разъемы для подключения различных устройств: клавиатуры, мыши, принтера, сканера, игровой приставки, модема и т. п. Эти разъемы позволяют выполнить подключение нужного устройства без разборки системного блока, не снимая его корпуса. На системной плате обычно имеются переключатели (*джамперы*), с помощью которых выполняется настройка компьютера на работу при той или иной конфигурации устройств. Однако для большинства пользователей предпочтительной является работа на компьютере без какой-либо предварительной настройки с помощью переключателей. Такой режим называют «включил и играй» (*plug and play*).

В системном блоке предусмотрена возможность установки нескольких внешних запоминающих устройств, которые обычно называют *дисководами*. К ним относятся: дисковод жестких дисков *б* (см. рис. 3.3, *б*), на котором хранится вся информация, используемая при повседневной работе на компьютере; дисковод *7* гибких дисков, с помощью которого могут выполняться чтение (т. е. ввод в ЭВМ) и запись информации сравнительно небольшого объема; дисковод *5* компакт-дисков для большого объема информации (английское сокращение CD-ROM от compact disk read only memory — компакт-диск только для чтения из памяти). В настоящее время получают распространение дисководы компакт-дисков с возможностью не только чтения, но и записи информации.

В системном блоке располагается также блок питания *1* (см. рис. 3.3, *а*), обеспечивающий подачу напряжения питания нужного уровня для всех потребителей электроэнергии внутри системного блока и для некоторых потребителей, подключаемых к этому блоку через разъемы.

Даже кратковременный перерыв в подаче питания приводит к потере введенной перед этим информации. Чтобы избежать этого

применяют так называемые источники бесперебойного питания 9 (см. рис. 3.2), в состав которых входит аккумулятор, обеспечивающий подачу напряжения сразу после того, как произошло отключение питающей электросети. Даже если этот источник маломощный и способен обеспечивать питание компьютера всего лишь в течение нескольких минут, пользователь успеет дать команду на сохранение информации.

Пользователь компьютера управляет его работой и вводит в него информацию с помощью клавиатуры 5 и ручного манипулятора 4, который широко известен под названием «мышь».

Основное устройство для вывода информации из компьютера — монитор 1 (его называют также дисплей, хотя точнее дисплеем является сам экран монитора). На мониторе отображается видеоинформация, окончательные (а если требуется, то и промежуточные) результаты обработки информации. Очень распространен и удобен режим работы, при котором на экране монитора появляются вопросы, обращенные к пользователю и требующие его ответов. Таким образом осуществляется общение человека с компьютером, что помогает ускорить обработку информации, исключить некоторые лишние операции, предотвратить возможные ошибки. Указанный способ общения пользователя с компьютером носит название *интерактивный режим*. Обычно пользователь реагирует на вопросы компьютера нажатием на ту или иную клавишу клавиатуры или с помощью мыши. Однако имеются конструкции мониторов, позволяющие реагировать на такие вопросы непосредственно с помощью самого монитора, прикасаясь к его экрану пальцем (это очень удобно для справочно-информационных систем с возможностью доступа широкого круга пользователей). Для переносных компьютеров, используемых вне рабочего помещения (в цеху, на складе, в библиотеке, в автомобиле или самолете) предпочтительно применение электронного пера, которым информация вводится непосредственно на дисплее.

Для вывода звуковой информации служат звуковые динамики (колонки) 6, которые могут помещаться и в системный блок, и в корпус монитора. Для подачи простых сигналов обычно используется небольшой динамик в самом системном блоке.

Для печати документов, подготовленных на компьютере, и распечатки результатов выполненной на нем обработки информации применяются печатающие устройства — принтеры 7.

Для ввода графической (текст и рисунки) информации в компьютер служат сканеры 8. Они преобразуют эту информацию в понятные компьютеру кодовые комбинации.

Обмен информацией с другими компьютерами через телефонную сеть осуществляется с помощью специального устройства 3, называемого *модемом*, которое выполняет необходимое согласо-

вание электрических сигналов, направляемых в сеть и получаемых из нее.

Системный блок более крупных и производительных компьютеров, таких как серверы, имеет большие (по сравнению с персональным компьютером) размеры. Это позволяет вставлять несколько жестких дисков и производить их замену без разборки системного блока (не снимая его крышку).

Суперкомпьютеры обычно имеют стоечную конструкцию, т. е. собираются в специальных стандартных стойках. Эти стойки позволяют дополнять ЭВМ: несколькими электронными платами с микропроцессорами, составляющими вычислительный узел; коммутаторами, обеспечивающими взаимодействие нескольких узлов; электронными платами для обмена информацией с иными устройствами по определенным правилам (эти платы в комплексе с правилами носят общее название *интерфейс*). Например, в состав самого мощного отечественного суперкомпьютера МВС-1000/М входят шесть стоек, каждая из которых содержит несколько вычислительных узлов, имеющих по два процессора. Всего в суперкомпьютере 768 процессоров. Для обеспечения их параллельной работы служат специальный управляющий узел и соответствующее программное обеспечение.

В процессе отладки и проведения испытаний суперкомпьютеров можно наращивать число вычислительных узлов, вставляя их в стойки. Такие ЭВМ, предусматривающие добавление однотипных вычислительных элементов, называют *кластерами*.

Блок питания суперкомпьютера должен иметь очень большую мощность — несколько сотен киловольт-ампер. Для надежной работы суперкомпьютера обычно используют дополнительный аварийный блок питания (им может быть небольшая автономная дизель-электростанция, которая автоматически берет на себя питание суперкомпьютера сразу после исчезновения напряжения в электросети).

3.3. Принцип открытой архитектуры

В литературе по ЭВМ часто используется понятие «архитектура». Под этим термином подразумевается и структура (т. е. устройства, входящие в состав ЭВМ), и организация работы ЭВМ. Сюда же следует отнести и вопросы взаимодействия ЭВМ с операционными системами и прикладными программами. Принцип *открытой архитектуры* заключается в том, что технические характеристики, информационные и вычислительные возможности ЭВМ можно улучшать, дополняя существующую конструкцию новыми узлами, заменяя прежние узлы более совершенными, вводя в машину новые программы. Впервые этот принцип был предложен и реализован фирмой IBM в персональных компьютерах, которые

получили широчайшую известность под общим названием IBM PC (ай-би-эм пи-си).

Если бы IBM PC был сделан по тому же принципу, что и другие существовавшие во время его появления компьютеры, он бы быстро устарел, и ему на смену пришли бы другие ЭВМ. Но уже при создании IBM PC была заложена возможность усовершенствования его отдельных частей и использования новых устройств. Фирма IBM не сделала компьютер единым неразъемным устройством (как первый «яблочный» компьютер), а предусмотрела возможность его сборки из независимо изготовленных частей аналогично детскому конструктору. При этом методы совместного использования других устройств с компьютером IBM PC не только не держались в секрете, но и были доступны всем желающим. Принцип открытой архитектуры наряду с другими достоинствами обеспечил выдающийся успех компьютеру IBM PC. При этом фирма IBM лишилась возможности единолично пользоваться плодами этого успеха, подключив к процессу создания персональных компьютеров множество других фирм.

Фирма IBM стремилась к популярности своего компьютера и рассчитывала, что открытость архитектуры IBM PC позволит независимым производителям разрабатывать различные дополнительные устройства, что увеличит эту популярность. Расчет оказался верным, и уже через пару лет на рынке предлагались сотни разных устройств и комплектующих для IBM PC.

Наибольшую выгоду от открытости архитектуры IBM PC получили миллионы пользователей. Они смогли самостоятельно расширять возможности своих компьютеров, покупая соответствующие устройства и подсоединяя их через свободные разъемы на системной плате. При этом пользователи не были связаны ассортиментом моделей, предлагаемых фирмой IBM, так как могли покупать дополнительные устройства, производимые другими фирмами. Они могли экономить деньги, ориентируясь при покупке компьютеров на свои сегодняшние, а не на будущие потребности, поскольку при необходимости компьютер можно модернизировать.

Многие пользователи последовательно улучшали характеристики своих компьютеров, заменяя системную плату с микропроцессором по мере появления на рынке новой модели. Таким образом, 286-й компьютер становился 386-м, а затем и 486-м. Сам корпус системного блока (см. рис. 3.3) оставался прежним, но в него можно было добавлять новые дисководы, а на системной плате можно было увеличивать объем оперативной памяти, вставляя в разъемы дополнительные микросхемы.

Дополняя компьютер новыми устройствами, порой надо вставлять в разъемы на системной плате соответствующие этим устройствам электронные платы и записывать на жесткий диск необходимые управляющие программы — *драйверы*.

Принцип открытой архитектуры может быть реализован лишь при условии строгого выполнения целого ряда правил, которые получили общее название *протоколы*.

Не менее важным, чем совместимость различных блоков и устройств (в том числе изготовленных разными производителями и даже в разных странах), является возможность использования разных программ в персональных компьютерах. При этом должен соблюдаться следующий принцип. Программы, предназначенные для менее производительных компьютеров, должны успешно работать и в более производительных, выпущенных позднее. С улучшением технических характеристик новых компьютеров расширяются их возможности, появляется способность выполнять обработку информации по новым, более совершенным программам. В компьютерах более раннего выпуска использование этих новых программ оказывается невозможным: не хватает объема памяти или недостаточно быстродействие. Но программы, работающие в старых компьютерах, должны работать и в новых. Иногда для этого в новых компьютерах предусматривается режим эмуляции. Это означает, что новый компьютер в каком-то смысле «притворяется» старым, делается на него похожим для того, чтобы успешно заработала хорошо зарекомендовавшая себя старая программа.

Таким образом, принцип открытой архитектуры заключается как в аппаратной, так и в программной совместимости. Принципы открытой архитектуры и блочно-модульного построения являются весьма перспективными и целесообразными при создании различных технических устройств, предназначенных для совместного использования, т. е. для создания *системы*. Это понятие является очень важным в современной технике. Системой называют объединение функционально связанных элементов, приобретающее новое качество, т. е. такое, какого не было у отдельных элементов.

3.4. Понятие о программном обеспечении

В процессе работы ЭВМ выполняют различные задачи, которые в общем виде можно назвать *обработкой информации*. К этим задачам относятся: бухгалтерские расчеты; вычисления по математическим формулам (включая сложные операции из высшей математики — интегрирование и дифференцирование); решение систем уравнений; проектирование самых разнообразных устройств; моделирование физических и химических процессов; управление производством и т. д. Очень распространенной задачей является, например, составление и редактирование текстовых материалов, т. е. различных документов. Все эти задачи имеют конкретное приложение, а программы, т. е. последовательности команд (инструкций) для их выполнения называются *прикладными*.

При выполнении указанных задач все узлы ЭВМ взаимодействуют, т. е. функционируют совместно. Для обеспечения их совместной работы, а также для того, чтобы пользователь ЭВМ получил доступ ко всем прикладным программам, мог вводить новые, использовать все возможности ЭВМ в полной мере и, вообще, чувствовать себя удобно, комфортно при работе на ЭВМ, необходимы еще и специальные программы, которые в совокупности называются *операционными системами*.

Все прикладные программы и операционные системы для ЭВМ носят название *программное обеспечение*. Таким образом, для полноценного использования ЭВМ нужны как сами устройства, так и программы, руководящие работой этих устройств. Иными словами, нужно аппаратное и программное обеспечение. В английском языке для обозначения аппаратного обеспечения используют слово *hardware*, а программного — *software*. Соответственно на профессиональном жаргоне компьютерщики иногда пользуются терминами «железо» и «софт».

В подразд. 3.2 была рассмотрена аппаратная реализация компьютера. Рассмотрим теперь, каким образом компьютер обрабатывает информацию.

В 1950—1960-е годы, когда компьютер назывался электронно-вычислительной машиной, он мог только вычислять. Обработка информации заключалась в операциях над числовыми данными.

В 1970-е годы на компьютерах стали работать и с текстом. Пользователь получил возможность создавать, редактировать и форматировать текстовые документы. В настоящее время большая часть компьютеров и большая часть времени пользователей используется для работы именно с текстовыми данными.

В 1980-е годы появились первые компьютеры, способные работать с графической информацией. Сейчас компьютерная графика широко используется в деловой графике (построение диаграмм, графиков и т. д.), компьютерном моделировании, при подготовке презентаций, создании веб-сайтов (информации, размещаемой в сети), в рекламе на телевидении, анимационном кино и т. д. Постоянно расширяется применение компьютеров для обработки графических данных.

В 1990-е годы компьютер научился обрабатывать звуковую информацию. Любой пользователь современного персонального компьютера может воспользоваться стандартными приложениями Windows для прослушивания, записи и редактирования звуковых файлов. В так называемых мультимедийных технологиях используются звуковые данные.

Для того чтобы числовая, текстовая, графическая и звуковая информация могла обрабатываться на компьютере, она должна быть представлена в форме данных. Данные хранятся и обрабаты-

ваются в компьютере на машинном языке, т. е. в виде последовательностей нулей и единиц.

Для того чтобы процессор компьютера «знал», что ему делать с данными, как их обрабатывать, он должен получить определенную команду (инструкцию). Такой командой может быть, например, «сложить два числа» или «заменить один символ на другой».

Обычно для решения какой-либо задачи процессору требуется не единичная команда, а их последовательность — программа.

На первом этапе развития ЭВМ, в 1940—1950-е годы, программы писались непосредственно на машинном языке, т. е. на том языке, который «понимает» процессор. Они представляли собой очень длинные последовательности нулей и единиц, в которых человеку разобраться было очень трудно.

В 1960-е годы началась разработка языков программирования высокого уровня (Алгол, Фортран, Бейсик, Паскаль и др.), которые позволили существенно облегчить работу программистов. Писать программы стали те же люди, которые и ставили задачу. С появлением систем визуального программирования создание программ стало доступно даже для начинающих пользователей компьютера.

Программная обработка данных на компьютере реализуется следующим образом. После запуска на выполнение программы, хранящейся во внешней долговременной памяти, она загружается в оперативную память.

Процессор последовательно считывает команды программы и выполняет их. Необходимые для выполнения команды данные загружаются из внешней памяти в оперативную и над ними производятся необходимые операции. Данные, полученные в процессе выполнения команды, записываются процессором обратно в оперативную или внешнюю память.

В процессе выполнения программы процессор может запрашивать данные с устройств ввода информации и пересылать данные на устройства вывода информации.

Операционные системы подробно рассмотрены в гл. 15. Ознакомимся предварительно с назначением и составом этих систем. Наибольшее распространение получили операционные системы Windows, разработанные фирмой Microsoft. Они используются в персональных компьютерах IBM и совместимых с ними. Эти операционные системы непрерывно совершенствуются, и в каждой версии указывается год ее создания.

Для компьютеров, объединенных для совместного использования в сети, применяется также свободно распространяемая система Linux.

В персональных компьютерах фирмы Apple используются различные версии операционной системы Mac OS. В рабочих станциях и серверах наибольшее распространение получили операцион-

ные системы Windows NT/2000/XP (а с 2007 г. новая система Виста) и UNIX. Для суперЭВМ создаются специальные операционные системы.

Хотя операционные системы разные, но основное их функциональное назначение одно и то же. Операционная система является необходимой составляющей программного обеспечения компьютера.

Именно операционная система обеспечивает взаимодействие всех узлов компьютера и работающих с ним совместно периферийных устройств. Она же помогает пользователю в его общении с компьютером.

Современные операционные системы имеют сложную структуру, каждый элемент которой выполняет определенные функции по управлению компьютером. Прежде всего операционная система организует распределение и хранение информации в компьютере.

Количество информации, содержащейся в ЭВМ, очень велико, оно соизмеримо с количеством информации, содержащейся во всех книгах какой-нибудь библиотеки. Для того чтобы ориентироваться во всей этой информации, ее необходимо определенным образом организовать, распределить. Единицей организации информации в ЭВМ является *файл*, т.е. набор данных в ЭВМ, посвященный определенной теме и хранящийся отдельно от других. Файлы, близкие по теме, хранятся в общей для них папке (ее также называют каталогом, или директорией). В свою очередь, группа папок по общей теме хранится в особой папке и т.д.

Подобный принцип организации информации в ЭВМ называется *файловой системой*. Она имеет древовидную структуру: от общего ствола отходят толстые ветки, которые, в свою очередь, делятся на более мелкие, ну а на самых тонких ветках растут листья — это и есть файлы (подробнее см. подразд. 15.1).

Управление файловой системой составляет одну из самых главных задач операционной системы. Процесс работы компьютера в определенном смысле сводится именно к обмену файлами между различными устройствами. В операционной системе имеются соответствующие программные модули, управляющие таким обменом файлами.

Пользователь может дать команду на вызов и запуск программы для выполнения какой-либо операции над файлами (копирование, удаление, переименование, сохранение измененного или вновь созданного), вывода документа на печать и т.д. Операционная система обеспечивает выполнение этой команды.

К системному блоку и системной плате компьютера подключаются различные устройства (дисководы, монитор, клавиатура, мышь, принтер, сканер, модем и др.). В состав операционной системы входят драйверы этих устройств, т.е. специальные програм-

мы, которые обеспечивают управление работой устройств и согласование информационного обмена с другими устройствами, а также позволяют производить настройку некоторых параметров устройств.

Для удобства работы пользователя в состав современных операционных систем, в частности в состав Windows, входят программные модули, создающие *графический пользовательский интерфейс*.

Слово «интерфейс» часто используется в литературе и в разговорах про компьютеры. Обычно при этом речь идет об условиях и правилах общения какого-либо устройства с другими. Пользовательский интерфейс предполагает общение пользователя с компьютером. В операционных системах с графическим интерфейсом пользователь может вводить команды с помощью мыши, тогда как в режиме командной строки необходимо вводить команды с помощью клавиатуры.

В состав операционной системы входят также *сервисные программы*, или *утилиты*.

Такие программы позволяют обслуживать диски (проверять, сжимать, разделять информацию на каталоги, файлы и т.д.), выполнять операции с файлами (архивировать и т.д.), работать в компьютерных сетях и т.д.

Для удобства пользователя в состав операционной системы обычно входит справочная система, которая позволяет оперативно информировать о работе всей операционной системы и ее отдельных модулей.

Операционная система позволяет также найти нужный файл по его имени или по его содержанию.

Контрольные вопросы

1. Какие устройства входят в состав ЭВМ?
2. Перечислите основные составные части персонального компьютера.
3. В чем состоит основное достоинство принципа открытой архитектуры?
4. Какое программное обеспечение требуется для персонального компьютера?
5. В чем разница между системными и прикладными программами?
6. Что такое интерфейс?
7. Для чего нужны драйверы подключаемых к персональному компьютеру устройств?
8. Что такое система?

СПОСОБЫ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ В ЭВМ**4.1. Виды информации**

По мере развития вычислительной техники все более расширялись границы применения ЭВМ. Если первоначально они использовались для математических вычислений, то в настоящее время применяются преимущественно для обработки информации. *Информация* — это одно из наиболее общих понятий, обозначающее некоторые сведения, совокупность каких-либо данных, знаний и т. п.

Все мы существуем в мире информации. Человек непосредственно воспринимает окружающий мир, т. е. *получает информацию* с помощью органов чувств. Наибольшее количество информации (примерно 80—90 %) человек получает с помощью зрения, примерно 10—15 % — с помощью слуха и только 1—2 % — с помощью других органов чувств (обоняния, осязания и вкуса). Полученная в форме зрительных, слуховых и других образов информация сохраняется в памяти человека и используется в повседневной жизни. При этом в мозгу человека происходит *обработка информации*. На основе информации, полученной с помощью органов чувств, и теоретических знаний, приобретенных в процессе обучения и накопления жизненного опыта, человек создает информационные модели окружающего мира. Такие модели позволяют ему ориентироваться в окружающем мире и принимать разумные решения для достижения поставленных целей.

В процессе общения с другими людьми человек передает и получает информацию. Обмен информацией между людьми может осуществляться в различных формах (письменной, устной или с помощью жестов). Для обмена информацией всегда используется определенный язык (русский, азбука Морзе и т. д.). Для того чтобы информация была понятна, язык должен быть известен всем людям, участвующим в общении. Чем большее количество языков вы знаете, тем шире может быть круг вашего общения.

История человеческого общества — это, в определенном смысле, история накопления и преобразования информации. Весь процесс познания является процессом получения, преобразования и накопления информации (знаний). Но информация может быть получена не только с помощью органов чувств и через общение.

Сам человек может быть творцом информации. Осмысливая (обдумывая) сведения, полученные с помощью собственных ор-

ганов чувств, различных измерительных приборов, из письменных источников, общения с другими людьми (как непосредственного, так и с помощью радио, телевидения и других средств телекоммуникаций), человек может получать какие-то новые сведения, т. е. переработка информации приводит к появлению новой информации. Таким образом, существует *входная* и *выходная* информация. Понятие информации является одним из основных в кибернетике — науке об управлении в самых разных системах — технических, биологических, социальных, административных. Важность этого понятия настолько возросла, что появились (выделились из кибернетики) самостоятельные разделы науки, занятые исследованиями самой информации (информатика, информатиология) и ее машинной обработки (компьютерная наука — computer science).

Понятие информации обычно предполагает наличие двух объектов: источника информации и ее потребителя (адресата). Для того чтобы информация могла быть передана от источника к адресату, состояние источника должно быть каким-то образом отражено во внешней среде, воздействующей на приемные органы адресата. Материальные объекты, посредством которых происходит перенос информации от источника к адресату, называются *носителями информации*. Отображение множества состояний источника в множество состояний носителя называется *способом кодирования*. В общем случае код представляет собой набор (ансамбль) некоторого конечного числа символов (условных знаков) и систему правил, по которым формируется последовательность этих символов — так называемая *кодовая комбинация*. С этой точки зрения алфавит, т. е. набор букв, представляет собой код, с помощью которого можно преобразовать устную речь в рукописный или печатный текст. К буквам надо добавить также знаки препинания и пробел, а системой правил можно считать правила грамматики. Для записи мелодии используется набор символов в виде нотных знаков. Информацию о ходе технологического процесса получают с помощью измерительных приборов, показания которых можно записать в виде чисел, а для передачи и хранения этой информации удобно представить ее в виде электрических сигналов. При этом каждому значению напряжения или тока этих сигналов соответствует определенное значение температуры, давления, скорости или иного параметра, характеризующего технологический процесс.

В вычислительной технике, использующей цифровые вычислительные машины, применяются двоичные (бинарные) коды, для которых число символов равно двум (0 и 1). Из этих символов составляется кодовая комбинация — последовательность нулей и единиц. Используя двоичную систему счисления, числа записывают в виде последовательности этих символов. Таким образом, в

вычислительной технике с помощью всего двух символов можно и производить вычисления, и обрабатывать любую другую информацию.

В частном случае термин «кодирование» можно определить как операцию установления однозначного соответствия между символами или группами символов некоторого кода с символами или группами символов другого кода. В этом смысле перевод чисел из одной системы счисления в другую (например, десятичных чисел в двоичные) — это кодирование.

4.2. Количественные характеристики информации

Использование двоичного кода для записи информации позволяет дать количественную оценку информации, т. е. измерить ее. В результате можно определить, много ли мы узнали, получив ту или иную информацию, или сколько информации содержит то или иное сообщение.

Всякая информация извлекается в результате опыта (действия). Таким опытом может быть чтение книги, измерение прибором, визуальное наблюдение событий, консультация со специалистом и т. д. Пусть опыт, который производится в целях получения информации, обусловлен вопросом: «Какой сегодня день недели?». Предположим, что мы не знаем этого и до получения ответа на свой вопрос не можем сами ответить на него однозначно. Таким образом, до опыта (по латыни *a priori*) имеет место бóльшая или меньшая неопределенность в интересующей нас ситуации. После опыта (*a posteriori*), т. е. когда мы уже получили ответ, ситуация становится более определенной. На наш вопрос можно ответить либо однозначно (например, «сегодня среда»), либо приблизительно (например, «то ли среда, то ли вторник»). При однозначном ответе неопределенность полностью устраняется (было семь возможных ответов, остался единственный). В случае приблизительного ответа нам число наших возможных ответов уменьшается с семи до двух, т. е. в результате опыта существовавшая ранее неопределенность снижается.

Для количественной оценки информации обычно анализируется именно переход от неопределенности к определенности. Уменьшение неопределенности в результате опыта может быть принято за наиболее общую меру количества информации. Количество получаемой в результате опыта информации тем больше, чем больше число N равновозможных ответов до опыта и меньше число M равновозможных ответов после опыта или получения сообщения, т. е. оно зависит от отношения N/M . Например, если в одном случае $N = 2$, а в другом $N = 20$, то очевидно, что вторая ситуация является более неопределенной. Если в результате опыта эта неопределенность в обоих ситуациях полностью снимается

($M = 1$), то следует считать, что количество полученной информации в случае $N = 20$ больше.

Пользуясь терминами теории вероятностей, вместо понятия «равновероятные ответы» или «события», будем пользоваться понятием вероятности этих событий.

При равновероятных ответах априорная (доопытная) вероятность события x_i есть $p(x_i) = 1/N$, а апостериорная (после опыта или получения сообщения) вероятность $p_m(x_i) = 1/M$. В таком случае количество информации, которую мы получаем о событии x_i в результате опыта, есть функция отношения $N/M = p_m(x_i)/p(x_i)$.

Наиболее удобной для подсчета количества информации является логарифмическая функция, поэтому указанное выше количество информации можно принять равным $I(x_i) = k \log_a [p_m(x_i)/p(x_i)]$.

Благодаря применению логарифмов, количества информации, содержащиеся в независимых сообщениях, обладают свойством аддитивности, т.е. общее количество информации равно сумме количеств информации в сообщениях, так как вероятность $p(x_i, z_i)$ совместного события в этом случае равна $p(x_i)p(z_i)$. Выбор коэффициента k и основания a логарифмов не имеет принципиального значения, поскольку они определяют лишь масштаб или единицу количества информации.

Обычно выбирают $a = 2$ и $k = 1$. При таком выборе единица количества информации называется двоичной, т.е. количество информации составляет $I(x_i) = \log_2 [p_m(x_i)/p(x_i)]$ двоичных единиц.

Пусть апостериорная вероятность $p_m(x_i) = 1$, тогда $I(x_i) = \log_2 [1/p(x_i)]$. В этом случае при двух равновероятных событиях, например, падении монеты вверх «орлом» (событие A) или «решкой» (событие B), $p(A) = p(B) = 0,5$ и $p_m(A) = p_m(B) = 1$. В результате опыта подбрасывания монеты извлекается одна двоичная единица информации: $I(A) = I(B) = \log_2(1/0,5) = \log_2 2 = 1$ двоичная единица.

Двоичную единицу информации часто называют битом. Получение одной двоичной единицы количества информации соответствует случаю, когда мы узнаём, какое из двух равновероятных событий имело место. Если ответом на заданный вопрос может быть с одинаковой априорной вероятностью «да» или «нет», то, получив его, мы тем самым получим одну двоичную единицу информации, или один бит информации.

Удобство применения двоичной единицы информации, в частности, связано с тем, что устройство с двумя устойчивыми состояниями может хранить одну двоичную единицу, а n таких устройств могут хранить n бит, так как полное число возможных состояний при этом равно 2^n , а $\log_2 2^n = n$.

Для ввода текстов в вычислительную машину используются разные правила кодирования. Очень широко применяется кодирование всех знаков (символов), которые могут быть введены с

помощью клавиатуры персонального компьютера, восьмиразрядными двоичными комбинациями нулей и единиц. Всего таких комбинаций может быть $2^8 = 256$. Разные кодовые комбинации соответствуют строчным и прописным буквам латинского и русского алфавитов, знакам препинания и пробелу, знакам различных математических операций и некоторым специальным знакам.

В первых трех изданиях этой книги была приведена таблица кодовых комбинаций символов, вводимых с помощью клавиатуры персонального компьютера. Это был один из наиболее распространенных кодов, называемый код обмена информацией КОИ-8. Цифра 8 означает, что код содержит 8 двоичных разрядов.

Таким образом, при вводе текста с клавиатуры для текстовых символов (буквы русского и латинского алфавитов, цифры, знаки препинания) можно использовать восьмиразрядные кодовые комбинации, т. е. $n = 8$, $\log_2 2^8 = 8$. Следовательно, количество информации в каждом знаке печатного текста можно оценить в 8 бит. Для удобства подсчета количества информации ввели более крупную единицу информации — *байт*. В одном байте — 8 бит.

Подсчитаем для примера количество информации на одной странице текста, содержащей 30 строк, в каждой из которых 60 знаков. Поскольку в каждом знаке 1 байт, а всего знаков $60 \cdot 30 = 1\,800$, то на одной странице помещается 1 800 байт.

Часто при измерении длины, массы, напряжения, сопротивления и многих других физических величин пользуются добавляемыми к наименованию единицы измерения приставками «кило» (к) и «мега» (М), указывающими на увеличение соответственно в тысячу (10^3) и миллион (10^6) раз. Аналогичными приставками пользуются и при подсчете большого количества информации, но при этом, поскольку за основу взята не десятичная, а двоичная система счисления, приставка «кило» (в данном случае сокращенное обозначение К) указывает на увеличение в $2^{10} = 1\,024$, а мега — $2^{20} = 1\,048\,576$ раз. Для приблизительной оценки количества информации вполне можно принимать, что один килобайт (1 Кбайт) — это примерно тысяча байт, а один мегабайт (1 Мбайт) — примерно миллион байт. На одной странице текста количество информации составляет примерно 2 Кбайт, в книге из 250 страниц — примерно 500 Кбайт, или 0,5 Мбайт.

В связи с увеличением числа разнообразных символов появляется необходимость применять более длинные двоичные кодовые комбинации. Например, кодировка Юникод (Unicode) использует 16 разрядов, что соответствует $2^{16} = 65\,536$ различным комбинациям. В этой кодировке возможен ввод букв алфавитов разных языков, математических, декоративных и иных символов.

В виде двоичных кодовых комбинаций можно представить и рисунки. Черно-белый рисунок на экране компьютера — это совокупность черных и белых точек. Пусть на экране 200 строк, а в

каждой строке 300 точек, т.е. общее число точек $300 \cdot 200 = 60\,000$. Для обозначения каждой точки можно использовать двоичные цифры (например, 1 — белая, 0 — черная). Весь рисунок займет 60 000 бит, т.е. примерно 7,5 Кбайт.

Всю информацию, которую мы получаем благодаря зрению, тоже можно закодировать двоичными комбинациями и количественно оценить. При этом надо учитывать, что мы видим движущееся изображение. По аналогии с телевизионным изображением можно считать, что за 1 с перед нашими глазами сменяют друг друга 50 неподвижных кадров. Если воспользоваться результатом предыдущего примера, то можно предположить, что благодаря зрению мы воспринимаем информацию в объеме, как минимум,

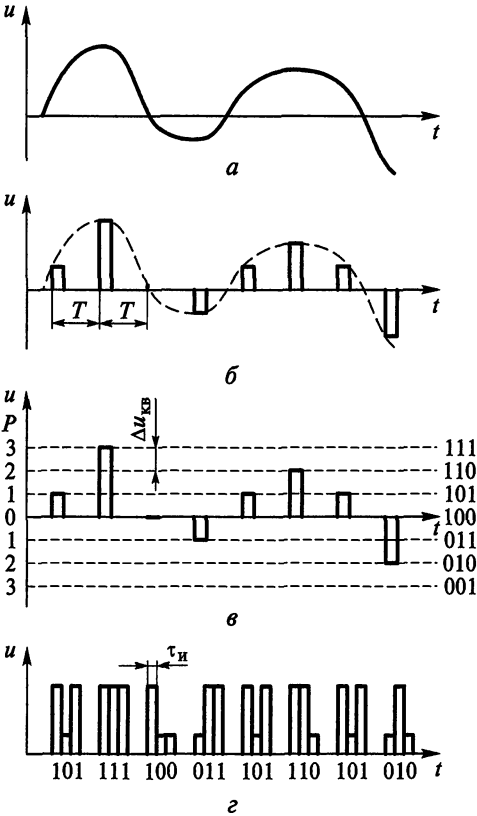


Рис. 4.1. Преобразование непрерывного (аналогового) сигнала в цифровой код:

a — непрерывный (аналоговый) сигнал; *б* — дискретные отсчеты (квантование по времени); *в* — квантование по уровню (семь уровней); *г* — цифровые (кодированные трехразрядные) сигналы

$7,5 \cdot 50 = 375$ Кбайт за 1 с. Но мы ведь видим не черно-белое, а цветное изображение. Пусть каждая точка будет окрашена в один из 16 цветовых оттенков (цветные рисунки на компьютере могут иметь 16, или 256, или и даже большее число градаций, т.е. оттенков). В этом случае можно считать, что каждую секунду через глаза в наш мозг поступает поток информации в количестве $375 \cdot 16 = 6\,000$ Кбайт ≈ 6 Мбайт.

Звуковую информацию тоже можно кодировать. Например, для записи двоичным кодом человеческой речи используется такая последовательность преобразований. Сначала с помощью микрофона звуковые колебания преобразуются в изменение электрического сигнала. Затем непрерывно изменяющийся во времени сигнал делают прерывистым (дискретным). Этот процесс называется *дискретизацией*, или *квантованием*. Выполняются два процесса квантования: по уровню и по времени. Исходный непрерывный электрический сигнал (рис. 4.1, *а*) имеет бесконечное множество значений и определен в каждый момент времени. Квантование по времени заключается в том, что берутся отсчеты с интервалом времени T (рис. 4.1, *б*). Все возможные непрерывные значения электрического сигнала (а их, как уже говорилось, бесконечное множество) разбиваются на 256 уровней. Следовательно, в каждый момент времени звуковому сигналу может быть присвоено значение максимального достигнутого в данный момент уровня (на рис. 4.1, *в* для простоты показано всего семь уровней). В этом и заключается квантование по уровню. Уровни P обозначены на рис. 4.1, *в* цифрами. Соответствующие этим уровням кодовые комбинации показаны на этом же рисунке справа. Более подробно рис. 4.1 рассмотрен в подразд. 7.7.

Дискретные отсчеты берутся с интервалом в одну восьмьюсекундную секунды ($1/8\,000$ с). Поясним, почему выбран именно такой интервал. Звуки человеческой речи (звуковые волны), передающиеся через воздушную среду, представляет собой сгущения и разрежения воздуха. Другими словами, это распространяющиеся в воздушной среде механические колебания, частота которых находится в диапазоне от нескольких десятков герц (очень низкий голос — бас) до 3,4 кГц. Указанные колебания в микрофоне преобразуются в электрические колебания такой же частоты. Академик В. А. Котельников доказал, что для передачи непрерывно изменяющихся по периодическому закону сигналов достаточно передавать их мгновенные значения с интервалом, который вдвое меньше, чем период колебаний при максимальной частоте изменения этих сигналов. Для звуковых колебаний приняли (с некоторым запасом) максимальную частоту 4 кГц. Следовательно, интервал между дискретными колебаниями на основании теоремы Котельникова должен составлять $1/8\,000$ с = 0,125 мкс. После квантования по уровню и по времени получают отсчеты в диапазоне от

нулевого до 255-го уровня, т.е. всего 256 различных вариантов. Для их представления в двоичной системе счисления достаточно иметь 8 разрядов, поскольку $2^8 = 256$.

Таким образом, для передачи текста, графической информации и звука можно применять двоичные кодовые комбинации. Такое представление информации и используется в вычислительной технике.

4.3. Достоинства дискретного сигнала

Первым, главным, достоинством дискретного двоичного сигнала с точки зрения его применения в вычислительной технике является то, что технически очень просто представить два используемых при этом символа (0 и 1). Для этого необходимо электрическое устройство, способное находиться только в двух устойчивых состояниях, которые легко различимы. Такое устройство называется *триггером*. Триггер содержит два транзистора. В одном состоянии первый транзистор открыт, а второй — заперт, в другом состоянии — наоборот. Иные состояния невозможны. Благодаря успехам микроэлектроники в настоящее время возможно в одной микросхеме разместить миллионы транзисторов. Для записи и хранения большого количества информации используют магнитные и оптические диски. В первых каждый бит информации представлен в виде намагниченной или размагниченной точки, во вторых — в виде наличия или отсутствия крошечной ямки. Современные запоминающие устройства позволяют записать миллиарды бит информации, поэтому, когда речь идет о ее количестве, все чаще используются приставки «гига» (Г), указывающая на увеличение в миллиард (10^9) раз (для двоичной системы счисления — $2^{30} = 1\,073\,741\,824$ раз), и «тера» (Т), указывающая на увеличение в тысячу миллиардов (10^{12}) раз (для двоичной системы — 2^{40} раз).

Вторым важным достоинством дискретного сигнала является то, что в промежутках между соседними отсчетами одной величины можно передавать отсчеты других величин. Следовательно, по одному и тому же каналу связи можно передавать множество самых разных сообщений. Нередко говорят, что они передаются одновременно, хотя на самом деле один замер от другого отделен некоторым очень малым промежутком времени. При указании величин этих крошечных промежутков времени требуются иные приставки, чем те, что используются при количественной оценке информации: «милли» (тысячная доля, или 10^{-3}), «микро» (миллионная доля, или 10^{-6}), «нано» (миллиардная доля, или 10^{-9}), «пико» (10^{-12}).

Благодаря передаче по одному каналу информации от большого числа источников оказывается возможным объединение нескольких вычислительных машин в сеть, создание глобальных се-

тей. В этих сетях промежутки между отдельными сигналами, соответствующими 0 и 1, составляют наносекунды.

Третье важное достоинство дискретного сигнала обусловлено тем, что он короткий и отделен от соседнего промежутком (перерывом). Поэтому нагрев электрических схем значительно меньше, чем при передаче непрерывных сигналов, и не требуется большая теплоотводящая поверхность. Значит, размеры элементов могут быть резко уменьшены. Отчасти благодаря этому удастся в больших интегральных микросхемах разместить миллионы элементов, хотя главная причина столь большого уровня интеграции — передовые технологии.

Контрольные вопросы

1. Каким образом человек получает информацию?
2. Что такое носители информации?
3. Какие существуют единицы для оценки количества информации?
4. Сколько бит в одном килобите?
5. Сколько бит в одном килобайте?
6. Почему передача дискретного цифрового сигнала обеспечивает повышение точности по сравнению с передачей аналогового сигнала?

РАЗДЕЛ II

ОСНОВЫ РАБОТЫ ЭВМ

Глава 5

МАТЕМАТИЧЕСКИЕ ОСНОВЫ РАБОТЫ ЭВМ

5.1. Системы счисления

Человек (по классификации живых существ Гомо Сапиенс — Человек Разумный) мыслит и обменивается мыслями с себе подобными. Обмен этот осуществляется в основном с помощью звуков — речи, состоящей из отдельных слов. Для сохранения мыслей, выраженных словами, служит письмо, т.е. запись речи определенными значкам. Наибольшее распространение получила буквенная запись, когда для обозначения каждого из звуков, составляющих слово, служит определенный знак. Для обозначения количественных соотношений между объектами, ситуациями, процессами используются числа, которые также могут быть выражены словами. Но для записи чисел используется более экономный способ, чем буквенная запись слов. Числа могут быть записаны с помощью специальных знаков — цифр.

Современное написание цифр десятичной системы счисления пришло к нам из Индии через арабов. Само слово «цифра» происходит от арабского слова «сифр». В XIII в. этими цифрами стали пользоваться в Италии благодаря флорентийским купцам, торговавшим с арабскими. Использование так называемых арабских цифр стало повсеместным в XV в.

При создании ЭВМ надежнее пользоваться схемами, которые могут быть только в двух устойчивых положениях. Например, электромагнитное реле может иметь замкнутый или разомкнутый контакт, определенный участок магнитной ленты может быть намагничен или размагничен и т.д. Поэтому, как уже отмечалось в гл. 4, большинство ЭВМ оперирует числами, записанными при помощи только двух цифр: 0 и 1. Как же представить любое число при помощи только двух цифр? Для этого необходимо использовать двоичную систему счисления.

Счислением называется совокупность приемов наименования и обозначения чисел. Существуют различные системы счисления. Прежде всего они различаются по количеству используемых зна-

ков, т. е. цифр. При записи чисел в десятичной системе счисления используются десятью цифрами: 0, 1, 2, 3, 4, 5, 6, 7, 8 и 9. Десятичная система является *позиционной*, поскольку значение каждой цифры в числе зависит от ее положения (позиции) среди других цифр этого числа. Например, в числе 2724,25 имеются три цифры 2. Все они имеют разные значения. Левая цифра указывает значение тысяч (2 тысячи), средняя — значение десятков (2 десятка), а правая (справа от запятой) — значение десятых долей (2 десятых доли). Указанное число является сокращенной записью следующей суммы:

$$2724,25 = 2 \cdot 10^3 + 7 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0 + 2 \cdot 10^{-1} + 5 \cdot 10^{-2}.$$

В числе имеется шесть позиций, т. е. шесть разрядов. Единица каждого разряда равна десяти единицам предыдущего разряда.

В двоичной системе счисления единица каждого разряда равна двум единицам предыдущего разряда. Слева от разряда единиц расположены разряды двоек, четверок, восьмерок и т. д. Чтобы записать число в двоичной системе, нужно представить его в виде суммы последовательных степеней числа 2, умноженных на 0 или 1:

$$2724,25 = 1 \cdot 2^{11} + 0 \cdot 2^{10} + 1 \cdot 2^9 + 0 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}.$$

Вместо того чтобы записывать десятичное число как сумму последовательных степеней числа 10, умноженных на одну из цифр от 0 до 9, мы пользуемся сокращенной записью (2724,25), т. е. выписываем подряд те цифры, на которые умножаются различные степени числа 10. Аналогично при записи чисел в двоичной системе счисления, вместо того чтобы записывать сумму последовательных степеней числа 2, умноженных на 0 или 1, выписывают только те цифры, на которые эти степени умножаются. Таким образом, $(2724,25)_{10} = (101010100100,01)_2$.

При двоичной системе счисления приходится записывать большое количество нулей и единиц. Это очень неудобно, особенно при ручном счете или вводе в ЭВМ. Поэтому в вычислительной технике довольно широко применяются и иные системы счисления, прежде всего восьмеричная и шестнадцатеричная.

При записи чисел в восьмеричной системе единица в каждом разряде равна восьми единицам предыдущего разряда. Поэтому в этой системе счисления используются только восемь цифр: 0, 1, 2, 3, 4, 5, 6, 7. Запишем число $(9775)_{10}$ в восьмеричной системе счисления:

$$(9775)_{10} = 2 \cdot 8^4 + 3 \cdot 8^3 + 0 \cdot 8^2 + 5 \cdot 8^1 + 7 \cdot 8^0 = (23057)_8.$$

Шестнадцатеричная система нередко оказывается еще более удобной при составлении программ. Поскольку арабских цифр у нас только десять (от 0 до 9), то вместо недостающих шести

цифр применяют прописные буквы латинского алфавита. Весь набор цифровых символов для шестнадцатеричной системы счисления выглядит так: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

При записи чисел в шестнадцатеричной системе единица в каждом разряде равна шестнадцати единицам предыдущего разряда. Запишем число $(9775)_{10}$ из предыдущего примера в шестнадцатеричной системе счисления:

$$(9775)_{10} = 2 \cdot 16^3 + 6 \cdot 16^2 + 2 \cdot 16^1 + F \cdot 16^0 = (262F)_{16}.$$

Представление информации в шестнадцатеричной системе гораздо компактнее, поскольку каждая группа из четырех двоичных цифр заменяется одним символом. Запомнить информацию в шестнадцатеричной системе (например, из трех символов) человеку легче, чем в двоичной (из 12 символов).

5.2. Перевод чисел из одной системы счисления в другую

Перевод целых чисел из десятичной системы в двоичную можно осуществить при помощи многократного деления на 2.

Для записи, например, числа $(173)_{10}$ в двоичной системе нужно найти такие цифры $A_0, A_1, A_2, \dots, A_n$, равные 0 или 1, чтобы

$$A_0 2^n + A_1 2^{n-1} + \dots + A_{n-1} 2 + A_n = 173. \quad (5.1)$$

Разделим правую и левую части равенства (5.1) на 2. Так как A_i равно 0 или 1, то в частном от деления левой части на 2 получим $A_0 2^{n-1} + A_1 2^{n-2} + \dots + A_{n-2} 2 + A_{n-1}$, а в остатке число A_n .

Получившиеся частное и остаток должны соответственно равняться частному и остатку от деления правой части равенства (5.1) на 2, поэтому

$$\begin{aligned} A_n &= 1; \\ A_0 2^{n-1} + A_1 2^{n-2} + \dots + A_{n-2} 2 + A_{n-1} &= 86. \end{aligned} \quad (5.2)$$

Разделим теперь на 2 обе части равенства (5.2) и приравняем получившиеся частные и остатки. В результате будем иметь

$$\begin{aligned} A_{n-1} &= 0; \\ A_0 2^{n-2} + A_1 2^{n-3} + \dots + A_{n-3} 2 + A_{n-2} &= 43. \end{aligned} \quad (5.3)$$

Разделим еще раз на 2 обе части равенства (5.3) и, сравнив частные и остатки, получим

$$\begin{aligned} A_{n-2} &= 1; \\ A_0 2^{n-3} + A_1 2^{n-4} + \dots + A_{n-4} 2 + A_{n-3} &= 21. \end{aligned}$$

Аналогичным образом найдем значения остальных цифр A_i . В результате получим: $A_0 = 1$; $A_1 = 0$; $A_2 = 1$; $A_3 = 0$; $A_4 = 1$; $A_5 = 1$; $A_6 = 0$ и $A_7 = 1$.

В левой части равенства (5.5) цифра A_1 есть целая часть числа в правой части (т.е. 1), а подчеркнутая сумма составляет дробную часть, поэтому $A_1 = 1$,

$$A_2 2^{-1} + A_3 2^{-2} + \dots + A_{n-2} 2^{-(n-3)} + A_{n-1} 2^{-(n-2)} + A_n 2^{-(n-1)} = 0,375. \quad (5.6)$$

Теперь умножим на 2 обе части равенства (5.6):

$$A_2 + \underbrace{A_3 2^{-1} + \dots + A_{n-2} 2^{-(n-4)} + A_{n-1} 2^{-(n-3)}}_{+ A_n 2^{-(n-2)}} = 0,75. \quad (5.7)$$

Отсюда $A_2 = 0$, а подчеркнутая часть равна 0,75.

Умножим на 2 обе части равенства (5.7):

$$A_3 + \underbrace{A_4 2^{-1} + \dots + A_n 2^{-(n-3)}} = 1,5. \quad (5.8)$$

Следовательно, $A_3 = 1$, а подчеркнутая часть равна 0,5, т.е.

$$A_4 2^{-1} + \dots + A_n 2^{-(n-3)} = 0,5. \quad (5.9)$$

После умножения на 2 равенства (5.9) получим $A_4 = 1$. Поскольку дробная часть равна 0 (результат умножения 0,5 на 2 можно записать как 1,000...00), то и все остальные цифры A_i равны 0, т.е. $A_5 = A_6 = \dots = A_n = 0$.

Таким образом, $(0,6875)_{10} = (0,1011)_2$.

Приведенные вычисления удобнее записать в следующем виде:

$$\begin{array}{r} 0,6875 \\ \times \quad 2 \\ \hline 1,3750 \\ 0,3750 \\ \times \quad 2 \\ \hline 0,750 \\ \times \quad 2 \\ \hline 1,50 \\ 0,50 \\ \times \quad 2 \\ \hline 1,0 \end{array}$$

Цифры, выделенные полужирным шрифтом, и являются искомыми двоичными цифрами, причем первой получается цифра, стоящая сразу после запятой.

Наше дробное число в двоичной системе можно представить в виде суммы: $(0,1011)_2 = (0,1000)_2 + (0,0010)_2 + (0,0001)_2$.

Каждому слагаемому соответствует своя десятичная дробь: $(0,1000)_2 = (0,5000)_{10}$; $(0,0010)_2 = (0,1250)_{10}$; $(0,0001)_2 = (0,0625)_{10}$. Сложив эти дроби, получим исходное число в десятичной системе: $0,5 + 0,125 + 0,0625 = 0,6875$. Это свидетельствует о том, что перевод из десятичной системы в двоичную выполнен верно.

Правило перевода чисел из десятичной системы счисления в восьмеричную состоит в делении переводимого числа и получающихся частных на 8. Остатки от деления и последнее частное, которые при этом получаются, и являются искомыми восьмеричными цифрами. Иными словами, алгоритм (правило) перевода аналогичен используемому для перевода десятичного числа в двоичное, только вместо деления на 2 выполняется деление на 8. Точно так же при переводе дробной части десятичного числа надо выполнять последовательное умножение на 8. Получаемые после каждой операции умножения цифры слева от запятой (т.е. целые части произведения) и есть искомые цифры для записи дробной части в восьмеричной системе.

Перевод числа из восьмеричной системы в двоичную и обратно очень прост. Чтобы число, записанное в восьмеричной системе счисления, записать в двоичной системе, нужно каждую восьмеричную цифру заменить тройкой двоичных цифр: $(0)_8 = (000)_2$; $(1)_8 = (001)_2$; $(2)_8 = (010)_2$; $(3)_8 = (011)_2$; $(4)_8 = (100)_2$; $(5)_8 = (101)_2$; $(6)_8 = (110)_2$; $(7)_8 = (111)_2$.

При переводе из двоичной системы в восьмеричную разбивают двоичное число справа налево на группы из трех двоичных цифр каждая. Сначала выделяют крайнюю правую группу (последние три цифры двоичной записи), затем следующую группу (три цифры слева от крайней группы) и т.д. Если в последней группе остается менее трех цифр, то вместо недостающих цифр ставят нули. Заменяв каждую группу соответствующей восьмеричной цифрой, получают число, записанное в восьмеричной системе счисления.

Например, двоичное число 11001101 разбивается на следующие группы: 011; 001; 101. Поскольку $(011)_2 = (3)_8$, $(001)_2 = (1)_8$, $(101)_2 = (5)_8$, то в восьмеричной системе это будет число 315, т.е. $(11001101)_2 = (315)_8$.

Аналогичные правила действуют и при использовании шестнадцатеричной системы счисления.

При переводе из двоичной системы в шестнадцатеричную двоичное число разбивают на группы из четырех цифр каждая. Такие группы называются *тетрадами*. Тетрады для шестнадцатеричных цифр от 0 до 7 подобны тем группам, что приведены выше для этих же восьмеричных цифр (только добавляется 0 слева). Остальным шестнадцатеричным цифрам соответствуют следующие тетрады: $(8)_{16} = (1000)_2$; $(9)_{16} = (1001)_2$; $(A)_{16} = (1010)_2$; $(B)_{16} = (1011)_2$; $(C)_{16} = (1100)_2$; $(D)_{16} = (1101)_2$; $(E)_{16} = (1110)_2$; $(F)_{16} = (1111)_2$.

Пример сложения двух восьмеричных чисел:

11 — единицы переноса
(3447)₈ — первое слагаемое
(7045)₈ — второе слагаемое

(12514)₈

Сложение начинают с разряда единиц: $(7)_8 + (5)_8 = (14)_8$. Записывают цифру 4 под чертой, а единицу переносят в следующий разряд — разряд восьмерок. Переходят к разряду восьмерок:

$$(1)_8 + (4)_8 + (4)_8 = (5)_8 + (4)_8 = (11)_8.$$

Одну единицу записывают, а другую переносят в следующий разряд. Переходя последовательно от разряда к разряду, определяют сумму $(12514)_8$.

Умножение двоичных и восьмеричных чисел производится аналогично умножению десятичных чисел. При этом пользуются соответствующими таблицами умножения чисел в двоичной (табл. 5.2) и восьмеричной системах счисления.

Вычитание двоичных чисел производится так же, как и десятичных, т. е. последовательно по разрядам от младшего к старшему. Если из меньшей цифры в данном разряде вычитается большая, то производится заем единицы из следующего старшего разряда, т. е. цифра этого старшего разряда становится на единицу меньше.

В вычислительной технике операции вычитания обычно заменяются операциями сложения. Рассмотрим пример такой замены. Вместо того чтобы из числа 85 вычитать число 37, к числу 85 прибавляется число $63 = 100 - 37$ (дополнительное к 37) и от результата 148 отнимается единица в старшем разряде. Получается число 48, которое является искомой разностью.

Аналогичным образом можно и в двоичной системе заменить вычитание сложением с использованием *дополнительного кода*. Саму операцию вычитания можно представить как сложение с отрицательным числом.

В вычислительной технике при использовании двоичной системы счисления крайний левый разряд служит для записи знака числа. Для положительного числа в этот разряд записывается 0, а для отрицательного — 1. Записанные таким образом двоичные числа будем называть записанными в *прямом коде*.

Рассмотрим составление дополнительного кода к прямому коду отрицательного числа.

Дополнительный код отрицательных двоичных чисел формируется по следующему правилу. Сначала цифры всех разрядов кроме знакового инвертируют (вместо 0 записывают 1, а вместо 1 — 0) и в младший разряд добавляют еди-

Таблица 5.2

Таблица умножения
двоичных чисел

Смножители	0	1
0	0	0
1	0	1

ницу. Если в младшем разряде уже стоит единица, то при этом приходится изменять цифру в следующем, а, возможно, и в более старших разрядах.

Например, при вычитании из числа 10110 числа 01101 уменьшаемое представляют как положительное число в прямом коде 010110, а вычитаемое — как отрицательное число, прямой код которого 101101 (полужирным шрифтом выделены цифры знакового разряда). Определяют дополнительный код вычитаемого. Сначала инвертируют цифры всех разрядов, кроме знакового (результат 110010), затем прибавляют единицу в младший разряд (110011). Выполняют операцию сложения уменьшаемого (в прямом коде) с вычитаемым (в дополнительном коде):

$$\begin{array}{r} 010110 \\ + \quad 110011 \\ \hline 001001. \end{array}$$

Число 01001 и есть результат вычитания, полученный в прямом коде. При сложении цифры знаковых разрядов складывают с отбрасыванием возникающего из этого разряда переноса. В данном примере в результате вычитания получилось положительное число, поскольку в знаковом разряде стоит 0. Это естественно, так как уменьшаемое больше вычитаемого. Если же из меньшего числа вычитать большее, то получается отрицательное число.

Убедимся в этом на примере, из числа 01101 (в прямом коде 001101) вычтем 10110. Для этого определим дополнительный код отрицательного числа 110110: сначала инвертируем цифры всех разрядов, кроме знакового (101001), потом добавим единицу в младший разряд (101010). Выполним сложение уменьшаемого в прямом коде и вычитаемого в дополнительном коде:

$$\begin{array}{r} 001101 \\ + \quad 101010 \\ \hline 110111. \end{array}$$

Результат есть отрицательное число (1 в знаковом разряде) и выражен он в дополнительном коде. Для получения его прямого кода убавим единицу в младшем разряде (110110), после чего инвертируем цифры всех разрядов, кроме знакового (101001).

Правильность вычислений проверим на десятичных числах: $(10110)_2 = (22)_{10}$; $(01101)_2 = (13)_{10}$; $(01001)_2 = (9)_{10}$; $22 - 13 = 9$; $13 - 22 = -9$.

При умножении двоичных многоразрядных чисел с учетом их знаков необходимо выполнить две операции: определить знак произведения и найти его абсолютную величину. Знаковый разряд может быть получен суммированием цифр знаковых разрядов сомножителей без формирования разряда переноса. При несовпаде-

нии складываемых цифр получается 1, что соответствует знаку произведения двух сомножителей с разными знаками. Абсолютная величина произведения определяется перемножением чисел без учета их знаков. Перемножение многоразрядных двоичных чисел производится с помощью табл. 5.2.

При умножении двух двоичных чисел множимое (первый сомножитель) последовательно умножают на каждую цифру множителя (второго сомножителя), начиная либо с младшего, либо со старшего разряда, и для учета веса соответствующей цифры множителя сдвигают либо влево (при начале умножения с младшего разряда множителя), либо вправо (при начале со старшего разряда) на такое число разрядов, на какое соответствующий разряд множителя сдвинут относительно младшего или старшего разряда.

При умножении вручную на бумаге мы привыкли начинать с младшей цифры второго сомножителя. При этом результат умножения на цифру следующего разряда записываем левее предыдущего результата на один разряд, т. е. тем самым производим сдвиг влево. Результаты умножения первого сомножителя на каждую цифру второго сомножителя называют частичными произведениями или промежуточными суммами. Получающиеся в результате умножения и сдвига частичные произведения после суммирования дают полное произведение. Особенность умножения двоичных чисел состоит в том, что частичное произведение может быть либо сдвинутым на соответствующее число разрядов множимым, если соответствующая цифра множителя равна 1, либо нулем, если соответствующая цифра множителя равна 0.

Рассмотрим пример:

10111	—	множимое
×		
<u>1101</u>	—	множитель
10111	—	первое частичное произведение
00000	—	второе частичное произведение
10111	—	третье частичное произведение
<u>10111</u>	—	четвертое частичное произведение
100101011	—	произведение

Тот же результат можно получить при умножении, начиная со старших разрядов множителя:

10111
×
1101
10111
10111
00000
<u>10111</u>
100101011.

В цифровых устройствах процессу суммирования частичных произведений придается последовательный характер: формируется одно из частичных произведений, к нему с соответствующим сдвигом прибавляется следующее частичное произведение, к полученной сумме с соответствующим сдвигом прибавляется очередное частичное произведение и так далее, пока не окажутся просуммированными все частичные произведения и не будет получено полное произведение.

Можно привести следующее обоснование тому, что умножение сводится к сдвигу и сложению. Пусть надо перемножить 101101 и 101101.

Запишем это в такой форме: $101101 \cdot 101101 = 101101(100000 + 1000 + 100 + 1) = 10110100000 + 101101000 + 10110100 + 101101$.

Таким образом, умножение на 100000 свелось к приписыванию пяти нулей (т. е. сдвигу на пять разрядов влево), на 1000 — трех (сдвиг на три разряда), на 100 — двух (сдвиг на два разряда). Иными словами, из первого сомножителя формируется столько частичных слагаемых, сколько единиц имеется во втором сомножителе. Сдвиг производится на столько разрядов влево, на каком месте (в каком разряде) находится соответствующая единица, минус один. Например, если единица есть в шестом разряде, сдвиг производится на пять разрядов, а если в четвертом, то на три. Если единица в первом разряде, то никакого сдвига делать не надо, в качестве одного из слагаемых берется сам первый сомножитель. Затем все полученные частные слагаемые складываются.

Операция деления в ЭВМ может быть сведена к нескольким операциям вычитаний и сдвигов. Результат деления (частное) определяется как число вычитаний с учетом сдвигов. Например, деление $132 : 11 = 12$ можно осуществить в виде такой последовательности вычитаний и сдвигов:

$$\begin{array}{r}
 \\
 \underline{} \\
 \\
 \underline{110} \text{ — первое вычитание} \\
 022 \\
 \underline{220} \text{ — сдвиг} \\
 110 \text{ — первое вычитание} \\
 110 \\
 \underline{ } \\
 110 \text{ — второе вычитание} \\
 000
 \end{array}$$

Ответ: 12 (одно вычитание до сдвига и два после).

Замена вычитания сложением остатка с дополнительным кодом вычитаемого сводит операцию деления к последовательности трех простейших операций.

Деление является весьма трудоемкой операцией. В ряде случаев в цифровых устройствах оно заменяется нахождением обратной величины делителя по специальной подпрограмме (на основе какой-либо быстро сходящейся итерационной формулы) и последующим умножением делимого на найденную обратную величину.

Иными словами, во многих машинах операция деления заменяется умножением, так как $a/b = a(1/b)$. По числу b машина автоматически вычисляет число $1/b$, которое затем умножается на a .

Довольно часто результат деления вычисляется не вполне точно, т.е. с некоторым приближением. Ведь деление без остатка не всегда возможно. В привычной нам десятичной системе это тоже часто бывает. Например, если разделить 2 на 3, то в ответе получится 0,666..., т.е. 6 в периоде. На практике принимают результат с округлением: 0,67, или 0,667, или 0,6667. Чем больше знаков после запятой, тем меньше ошибка вычисления.

5.4. Способы представления чисел в разрядной сетке ЭВМ

Вычислительные машины оперируют с числами, количество разрядов которых ограничено, поскольку электрическое устройство, предназначенное для хранения двоичного числа, имеет определенное число разрядов.

При выполнении операции сложения двоичных чисел возможно переполнение разрядной сетки, которое проявляется в том, что результат операции (получившаяся сумма) требует большего количества разрядов, чем имеется в устройстве для его хранения. Для выявления переполнения разрядной сетки используется *модифицированный код*. К двоичному числу добавляются два знаковых разряда, причем в обоих разрядах положительные числа содержат нули, а отрицательные числа — единицы. Выполнение операций суммирования с использованием модифицированного дополнительного кода производится по обычным приведенным выше правилам. Если результат суммирования содержит в знаковых разрядах комбинации **01** или **10** (т.е. разные цифры), то это служит признаком переполнения разрядной сетки.

Насмотрим пример сложения чисел:

	00	11011	— первое слагаемое в прямом модифицированном коде
+		11 01011	— второе слагаемое в дополнительном модифицированном коде
	00	00110	— сумма в прямом модифицированном коде

Переполнение разрядной сетки в данном случае не возникает (в знаковых разрядах комбинация **00**). Перенос из старшего знакового разряда при сложении отбрасывается.

Еще один пример:

$$\begin{array}{r} 00\ 10110 \\ + \\ 00\ 11011 \\ \hline 01\ 10001. \end{array}$$

В знаковых разрядах результата суммирования возникает комбинация **01**, что свидетельствует о переполнении разрядной сетки и ошибочности зафиксированного результата. Появление ошибки связано с тем, что при суммировании положительных чисел перенос из старшего разряда оказался зафиксированным во втором из знаковых разрядов. Для записи результата суммирования в данном примере потребуется уже шесть (а не пять, как в слагаемых) разрядов (кроме знаковых).

При суммировании отрицательных чисел также возможно переполнение разрядной сетки:

$$\begin{array}{r} 11\ 010011 \\ + \\ 11\ 100011 \\ \hline 10\ 110110. \end{array}$$

На переполнение указывает комбинация **10** в знаковых разрядах.

В общем случае каждое число состоит из целой и дробной частей. При записи чисел на бумаге эти части отделяют друг от друга запятой. Обычно в вычислительную машину вводят только значащие цифры исходных чисел без запятой. Как же указать машине, сколько целых разрядов содержат вводимые числа? Для различных типов машин это делается по-разному.

Некоторые машины устроены таким образом, что у всех вводимых в них чисел запятая стоит после заранее определенного фиксированного разряда. Они называются *машинами с фиксированной запятой*. К ним, например, относились первые модели ЭВМ «Минск-1» и многие другие. Такие машины оперируют с числами, которые по абсолютной величине меньше единицы, поэтому числа задаются своими дробными частями. Ноль, соответствующий разряду целых единиц, и запятая в машину не вводятся. Например, число 0,125 задается цифрами 125, а число 0,0625 — цифрами 0625. Дробные части десятичных чисел после ввода в электронную машину автоматически переводятся в двоичную систему счисления.

Для хранения одного двоичного числа в различных машинах используется разное количество разрядов, например, в «Минске-1» 31 разряд. Для хранения дробной части числа, переведенного в двоичную систему, используются 30 разрядов. Первый разряд является знаковым. Если число положительное, то машина задает в знаковый разряд 0, если отрицательное, то — 1. В машину «Минск-1»

вводятся десятичные числа, содержащие в дробной части семь цифр. Например, число $-0,0234375$ задается в машине «Минск-1» как -0234375 . После перевода в двоичную систему оно хранится в машине в следующем виде: $100000110000000000000000000000$.

Таким образом, машины с фиксированной запятой воспринимают только числа, у которых запятая стоит в строго фиксированном месте. Если при работе этих машин в промежуточных результатах получится число больше единицы, например при сложении, то его старшие (слева от запятой) разряды потеряются и будет неверный итоговый результат.

Во избежание этого предусмотрена автоматическая остановка машины, называемая остановкой по переполнению разрядной сетки машины. Это значит, что при подготовке задачи для решения на машине с фиксированной запятой нужно выбирать такие масштабы (единицы измерения) для исходных данных, чтобы все числа и все получающиеся промежуточные результаты были меньше единицы.

Другие машины, например БЭСМ, оперируют с числами, представленными в нормализованной форме.

Любое число можно представить в виде произведения правильной дроби и целой степени основания системы счисления, в которой это число записано.

Например, число $(N)_{10}$ можно представить в виде $N = B \cdot 10^p$, где $B < 1$. Число B называется *мантиссой* числа N , а число p — его *порядком*. Если $B > 0,1$, то говорят, что N записано в *нормализованной форме*; если $B < 0,1$, то — в *ненормализованной форме*.

Рассмотрим четыре примера:

$$36587,6 = 10^5 \cdot 0,365876;$$

$$36587,6 = 10^7 \cdot 0,00365876;$$

$$36587,6 = 10^3 \cdot 36,5876;$$

$$0,00365876 = 10^{-2} \cdot 0,365876.$$

Они показывают, что десятичное число может быть различными способами представлено в виде произведения степени 10 и десятичной дроби. В первом примере произведение является нормализованной формой числа $36587,6$, так как его мантисса $0,365876 > 0,1$. Порядок этого числа при такой мантиссе равен пяти. Во втором примере исходное число записано в ненормализованной форме, так как $0,00365876 < 0,1$. Порядок исходного числа стал теперь равен семи. В третьем примере число $36,5876$ не является мантиссой числа 36587 , так как $36,5876 > 1$. Последний пример показывает, что порядок числа может быть отрицательным.

Таким образом, заданное число может быть различными способами записано в ненормализованной форме и поэтому иметь различные порядки и мантиссы. Но записать его в нормализованной форме можно только одним способом. Порядок и мантисса

числа при записи его в нормализованной форме точно определяются по самому числу. Поэтому вместо числа можно вводить в электронную вычислительную машину его порядок и мантиссу, получающиеся при такой записи. В машине эти два числа переводятся в двоичную мантиссу и двоичный порядок.

Возьмем для примера число $(4,375)_{10} = (100,011)_2$. В десятичной записи мантисса $B = 0,4375$, а порядок $p = 1$. Эти два числа в машине переводятся в двоичную мантиссу $(0,100011)_2$ и двоичный порядок $(11)_2$, так как $(100,011)_2 = (0,100011)_2 \cdot (10)_2^{(11)_2}$.

Машины, в которых числа задаются мантиссами и порядками, называются *машинами с плавающей запятой*. В них все операции над числами сводятся к операциям над их порядками и мантиссами, записанными в двоичной системе.

Точность вычислений в машинах с фиксированной запятой меньше, чем в машинах с плавающей запятой. Для машин с фиксированной запятой труднее составлять программу работы, так как приходится подбирать масштабные коэффициенты, чтобы все промежуточные результаты были меньше единицы. Но их устройство проще, чем у машин с плавающей запятой, у которых имеется дополнительная аппаратура для хранения порядков введенных в машину чисел и выполнения арифметических операций с ними.

Контрольные вопросы

1. Что такое система счисления?
2. Почему в вычислительной технике используется двоичная система счисления?
3. Приведите пример непозиционной системы счисления.
4. Переведите число $(100)_{10}$ в двоичную систему счисления.
5. Переведите число $(100)_2$ в десятичную и восьмеричную системы счисления.
6. По какому правилу число из шестнадцатеричной системы счисления переводится в двоичную?
7. Сложите двоичные числа 01011 и 10101. Проверьте результат, переведя слагаемые в десятичную систему.
8. Составьте таблицу умножения для восьмеричной системы.
9. Составьте таблицу умножения для шестнадцатеричной системы.

ЛОГИЧЕСКИЕ ОСНОВЫ РАБОТЫ ЭВМ

6.1. Элементарные логические функции

Обработка информации в ЭВМ во многом напоминает процесс мышления. Законы и формы мышления изучает наука, называемая *логикой*. Одним из ее разделов является математическая логика. В нем используются математические, в основном алгебраические, методы, впервые разработанные в трудах английского математика Джорджа Буля (1815 — 1864). Поэтому этот раздел логики называют также *алгеброй логики*, или *булевой алгеброй*. Предметом изучения алгебры логики являются суждения (высказывания), в связи с чем встречается и такое ее название как исчисление высказываний. В исчислении высказываний оперируют только такими суждениями, про которые известно, что они либо истинны, либо ложны (обязательно одно из двух). Каждое конкретное высказывание имеет вполне определенное значение истинности: для истинного высказывания это значение равно единице, для ложного — нулю.

Воспользуемся для обозначения высказываний прописными латинскими буквами и рассмотрим несколько из них: A — свинец — это металл; B — Земля больше Солнца; C — студенты — люди; D — дважды два — четыре; E — соль — сладкая. Запишем значения истинности для этих высказываний: $A = 1$; $B = 0$; $C = 1$; $D = 1$; $E = 0$. Из подобных простых высказываний могут образовываться сложные. Например, «свинец — это металл, и Земля больше Солнца». Или «студенты — это люди, и дважды два — четыре». Значения истинности сложных высказываний зависят от истинности или ложности простых высказываний, т. е. являются их функциями. В нашем случае первое сложное высказывание обозначим $F(A, B)$, а второе $F(C, D)$. Запишем значения истинности этих функций: $F(A, B) = 0$; $F(C, D) = 1$. Функции, которые могут принимать только одно из двух значений (1 или 0), принято называть *логическими (булевыми)*, или *двоичными функциями*. Аналогично и переменные, входящие в логические функции и принимающие значения 1 или 0, называют логическими переменными.

Для представления логических переменных и значений истинности могут служить элементы, имеющие два устойчивых состояния. Одно из этих состояний принимается за 1, а другое — за 0. В двоичной системе счисления, рассмотренной в гл. 5, числа могут рассматриваться как значения функции n переменных (n —

число разрядов), каждая из которых принимает только одно из двух значений (0 или 1). Доказано, что математические операции с двоичными числами могут быть сведены к логическим.

В современных ЭВМ для записи и хранения информации используются электрические (чаще всего электронные и магнитные) элементы, имеющие два устойчивых состояния. Арифметические операции в этих ЭВМ выполняются над двоичными числами. Обработка информации проводится по правилам алгебры логики. Для обозначения состояний элементов, записи чисел, указания истинности и ложности суждений используются одни и те же условные знаки — 1 и 0. Благодаря этому соответствию между, на первый взгляд, разными явлениями и предметами оказывается возможным и практически удобным описывать работу ЭВМ с помощью алгебры логики.

Сложные высказывания строятся из простых с помощью логических связок-операций, которые выражаются словами «не», «и», «или». Этим словам соответствуют основные логические (булевы) функции: НЕ — *инверсия*, И — *конъюнкция*, ИЛИ — *дизъюнкция*. Простые высказывания являются *аргументами*. Составленные из этих простых высказываний сложные являются *функциями* этих аргументов.

Из приведенных простых высказываний (аргументов A , B , C , D и E) можно составить различные сложные высказывания, т. е. сложные функции. Рассмотрим несколько таких сложных высказываний, построенных с использованием трех основных логических функций. Для записи этих основных логических функций и соответствующих им логических операций служат специальные математические знаки (подобно тому как для записи арифметических операций служат знаки «+», «-» и т. п.). При чтении записанных таким образом сложных функций используют привычные слова «не», «и», «или», выполняющие роль логических связок.

Простейшей логической связкой алгебры логики является операция отрицания, или инверсии, выражаемая в русском языке частицей «не». Если, например, имеем простое высказывание B «Земля больше Солнца», то новое сложное высказывание «Земля не больше Солнца» обозначается \bar{B} и читается: «не B ». Легко видеть, что если B — ложное высказывание ($B = 0$), то \bar{B} — истинное ($\bar{B} = 1$), и наоборот, если $B = 1$, то $\bar{B} = 0$. Этот факт лежит в основе определения логической операции НЕ в виде формулы $A + \bar{A} = 1$. Из него следует, что $A = \bar{\bar{A}}$, т. е. двойное отрицание логической функции восстанавливает ее прежнее значение.

В качестве второй логической связки рассмотрим операцию конъюнкции, выражаемую союзом «и». Обозначается конъюнкция знаком « \wedge », который ставится между высказываниями. Если A и B — простые высказывания, то $F(A, B) = A \wedge B$ — сложное высказывание (читается: « A и B »). При этом суть операции конъюн-

юнкции определяется так. Сложное высказывание $A \wedge B$ истинно в том и только в том случае, если оба высказывания A и B истинны, т.е. $A \wedge B = 1 \wedge 1 = 1$; $F(A, B) = 1$. Но если одно из двух высказываний (или оба) ложное, то и сложное высказывание ложно: $A \wedge B = 1 \wedge 0 = 0$; $A \wedge B = 0 \wedge 1 = 0$; $A \wedge B = 0 \wedge 0 = 0$. Это определение вполне соответствует смыслу союза «и», в чем можно убедиться, воспользовавшись высказываниями B , C , D и E , приведенными ранее. Сложное высказывание «студенты — люди, и дважды два — четыре» ($C \wedge D = 1 \wedge 1 = 1$) истинно, а сложные высказывания «студенты — люди, и Земля больше Солнца» ($C \wedge B = 1 \wedge 0 = 0$); «соль — сладкая, и дважды два — четыре» ($E \wedge D = 0 \wedge 1 = 0$); «Земля больше Солнца, и соль — сладкая» ($B \wedge E = 0 \wedge 0 = 0$) являются ложными. Операцию конъюнкции, особенно в приложениях математической логики, часто называют операцией И, или логическим умножением, и обозначают, как в алгебре, знаком « \cdot » или, вообще, его опускают.

Третья операция, которая употребляется в алгебре логики, выражается союзом «или», который в русском языке имеет два различных значения. В одном случае может быть исключающее «или», например в сложном высказывании «монета после падения оказывается лежащей вверх «орлом» или «решкой». В другом высказывании «при стуке в дверь сестра или брат проснется» союз «или» не исключает возможности того, что проснутся оба. Логическая операция, соответствующая *неисключающему* ИЛИ, т.е. допускающему обе возможности, называется дизъюнкцией и обозначается знаком « \vee ». Если A и B — два простых высказывания, то их дизъюнкция $A \vee B$ (читается: « A или B ») есть сложное высказывание, которое ложно тогда и только тогда, когда ложны оба высказывания, т.е. $A \vee B = 0 \vee 0 = 0$, но $A \vee B = 1 \vee 0 = 1$, $A \vee B = 0 \vee 1 = 1$; $A \vee B = 1 \vee 1 = 1$. Часто операцию дизъюнкции называют логическим сложением и обозначают знаком « $+$ ».

Отрицание, конъюнкция и дизъюнкция относятся к элементарным логическим операциям. Функции одного и двух аргументов называются элементарными, если логические выражения этих функций содержат не более одной логической элементарной операции. Элементарной функцией одного аргумента является отрицание (инверсия). При двух аргументах возможны четыре варианта их набора: 0 и 0; 0 и 1; 1 и 0; 1 и 1. Для каждого набора будет определенное значение функции, т.е. количество таких функций при всех возможных наборах двух аргументов будет выражаться четырехразрядным двоичным числом. Всего возможны $2^4 = 16$ различных элементарных функций. Из них кроме конъюнкции и дизъюнкции следует упомянуть функцию Пирса (другие названия — стрелка Пирса, функция Вебба) и функцию Шеффера (другое название — штрих Шеффера). Функция Пирса реализует логическое сложение с отрицанием, т.е. логическое ИЛИ — НЕ, функ-

ция Шеффера — логическое умножение с отрицанием, т. е. логическое И — НЕ.

Четыре функции: дизъюнкция, конъюнкция, функция Пирса, функция Шеффера — относятся к функциям конституенты единицы или нуля. Они обращаются в 0 (или 1) при одном единственном варианте набора аргументов. При трех остальных вариантах эти функции обращаются соответственно в 1 (или 0).

Дизъюнкция двух переменных обращается в 0 лишь в том случае, если оба входных аргумента равны 0. Функция Пирса при таком входном наборе обращается в 1.

Конъюнкция двух переменных обращается в 1 лишь в том случае, если оба входных аргумента равны 1. Функция Шеффера при таком входном наборе обращается в 0.

Таким образом, дизъюнкция и функция Шеффера относятся к функциям *конституенты нуля*, а конъюнкция и функция Пирса — к функциям *конституенты единицы*.

Функции дизъюнкции, конъюнкции, Шеффера и Пирса в общем случае могут быть функциями произвольного числа аргументов. Следовательно, можно определить их значения для любого входного набора, т. е. любого поданного на вход двоичного числа. Подобные функциональные зависимости и используются в вычислительной технике. Поскольку над двоичными числами выполняются операции по правилам арифметики, а над информацией, представленной в виде двоичных чисел, — операции по правилам алгебры логики, то в состав вычислительной машины входит специальное арифметико-логическое устройство (АЛУ).

В вычислительной технике довольно часто применяются еще три элементарных функции: эквивалентности (или равнозначности), импликация и сложения по модулю 2. Подробнее о них рассказано в подразд. 6.2.

6.2. Формы представления логических функций

В вычислительной технике происходит преобразование информации, представленной в виде двоичных кодовых комбинаций. В отдельных частях ЭВМ используются устройства, преобразующие одну комбинацию (входную) в другую (выходную). Преобразование происходит в момент поступления входной комбинации (с некоторой задержкой по времени, которая не существенна) и не зависит от тех комбинаций, которые поступали в предшествующие моменты времени. Такие устройства принято называть *комбинационными*. Поскольку преобразование происходит автоматически, то иногда используется термин «комбинационный автомат». Можно сказать, что комбинационное устройство «не помнит» предыстории поступления сигналов на его входы, оно реагирует только на данную конкретную входную комбинацию. На

Таблица истинности логической функции Пирса

Номер набора	x_1	x_2	$f_3 = x_1 \downarrow x_2$
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	0

отдельных этапах проектирования комбинационных устройств используют различные формы представления логических функций: словесную, табличную, аналитическую и геометрическую (кубическую). Эта последняя форма была кратко рассмотрена только в первых трех изданиях книги.

Словесная, табличная и аналитическая формы. Любая логическая функция может быть представлена в виде *словесного описания*. Например, логическая функция Пирса (*стрелка Пирса*) словесно может быть описана так: значение функции Пирса равно 1 только тогда, когда обе входные переменные x_1 и x_2 равны нулю, при любых других значениях входных переменных значение функции Пирса равно 0.

Эту же функцию можно представить в *табличной форме* — в виде таблицы истинности или карты Карно.

Таблица истинности (табл. 6.1) и карта Карно содержат все четыре возможных входных набора и значения функции, соответствующие каждому из них. Карта Карно для логической функции двух переменных (рис. 6.1) представляет собой квадрат, разделенный на четыре ячейки, по одной на каждый входной набор. Переменная x_1 связана со строками карты, а переменная x_2 — со столбцами. Следовательно, ячейка, расположенная слева вверху, соответствует входному набору (00), а справа внизу — входному набору (11). Представление логической функции на карте Карно производится в соответствии с таблицей истинности. Если функция $f_3 = (\bar{x}_1 \wedge \bar{x}_2) = 1$ при входном наборе (00), то этот факт отражается на карте Карно записью в левую верхнюю ячейку единицы (см. рис. 6.1, а). Остальные ячейки остаются незаполненными. Нулями заполняют ячейки карты Карно, соответствующие входным наборам, при которых $f_3 = 0$ (рис. 6.1, б).

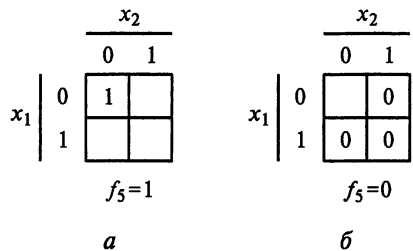


Рис. 6.1. Карта Карно для логической функции двух переменных: а — $f_3 = 1$; б — $f_3 = 0$

		$x_2 x_3$			
		00	01	11	10
x_1	0	000	001	011	010
	1	100	101	111	110

Рис. 6.2. Карта Карно для логической функции трех переменных

		$x_3 x_4$			
		00	01	11	10
$x_1 x_2$	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10

Рис. 6.3. Карта Карно для логической функции четырех переменных

Карта Карно для логической функции трех переменных (рис. 6.2) содержит восемь ячеек, по одной для каждого входного набора, указанного внутри ячейки. Переменная x_1 связана с двумя строками карты, а переменные x_2 и x_3 — с четырьмя столбцами. Таким образом, любые две рядом расположенные ячейки являются соседними, т. е. их координаты отличаются только одной переменной. Кроме того, соседними являются ячейки, стоящие в первом и последнем столбцах карты.

Поскольку для четырех переменных существует 16 входных наборов, карта Карно для логической функции четырех переменных (рис. 6.3) разделена на 16 ячеек. Каждая ячейка пронумерована в соответствии с порядковым номером входного набора.

В случае пяти переменных целесообразно использовать две 16-ячеечные карты, а не одну 32-ячеечную. Каждая из указанных на рис. 6.4 карт Карно связана с одним из значений переменной x_5 (a — при $\bar{x}_5 = 1$; b — при $x_5 = 1$). Для логических функций с числом переменных $n \geq 6$ карты Карно становятся громоздкими и неудобными для практического применения.

Устройство, реализующее действия над двоичными числами, можно рассматривать как логический функциональный преобра-

		$x_3 x_4$			
		00	01	11	10
$x_1 x_2$	00	0	2	6	4
	01	8	10	14	12
	11	24	26	30	28
	10	16	18	22	20

a

		$x_3 x_4$			
		00	01	11	10
$x_1 x_2$	00	1	3	7	5
	01	9	11	15	13
	11	25	27	31	29
	10	17	19	23	21

б

Рис. 6.4. Две карты Карно для логической функции пяти переменных:

a — при $\bar{x}_5 = 1$; b — при $x_5 = 1$

зователь с n входами и m выходами (рис. 6.5), на входы которого подаются исходные двоичные числа, а на выходе получается результат преобразования также в виде двоичного числа. Работа устройства состоит в том, что при поступлении на его вход двоичного числа P (т.е. комбинации нулей и единиц, общее количество которых равно n) на выходе образуется выходное двоичное число Q (т.е. комбинация нулей и единиц, общее количество которых равно m). Если работа устройства полностью определяется только входным двоичным числом, то она может быть определена для всех входных чисел (всего таких двоичных чисел может быть $z = 2^n$) следующим образом: входу P_1 соответствует выход Q_1 , входу P_2 — выход Q_2 , ..., входу P_z — выход Q_z .

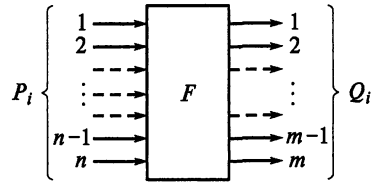


Рис. 6.5. Логический функциональный преобразователь

Значение выходного числа определяется конкретным сочетанием значений всех n разрядов входного двоичного числа, которое называется *двоичным набором*. Каждому набору на входе устройства будет соответствовать 0 или 1 на определенном выходе. Для описания работы такого устройства используется математический аппарат алгебры логики, или булевой алгебры. Функция алгебры логики — это функция, однозначно определяющая соответствие каждого двоичного набора 0 или 1. Таким образом, двоичные наборы являются наборами значений аргументов функции алгебры логики. Количество возможных различных наборов равно максимальному числу, которое может быть изображено с помощью двоичных разрядов, т.е. 2^n .

Любую функцию алгебры логики, зависящую от n аргументов, можно задать в виде таблицы с 2^n строками. В этих строках записывают все возможные двоичные наборы значений аргументов x_1, x_2, \dots, x_n и указывают значения функции $f(x_1, x_2, \dots, x_n)$, которые она принимает при каждом наборе (т.е. 0 или 1). Такая таблица, как уже было отмечено, называется таблицей истинности. Примером таблицы истинности для трех аргументов может быть табл. 6.2.

Таблица 6.2

Таблица истинности функции трех аргументов

Номер набора	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0

Номер набора	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Для полного определения функции не обязательно задавать ее значения во всех наборах, достаточно определить либо все наборы, обращающие ее в 0, либо все наборы, обращающие ее в 1.

Возможен и аналитический способ задания функций алгебры логики. Он предусматривает запись функции в виде логического выражения, показывающего, какие логические операции над аргументами функции должны выполняться и в какой последовательности.

Функции дизъюнкции (f_3) и конъюнкции (f_4) в аналитической форме приведены в подразд. 6.1.

Для обозначения функции Пирса, или стрелки Пирса (f_5), используется символ « \downarrow »: $f_5(x_1, x_2) = x_1 \downarrow x_2$. В аналитической форме $f_5(x_1, x_2) = x_1 \vee x_2$.

Для обозначения функции Шеффера, или штриха Шеффера (f_6), используется символ « $/$ »: $f_6(x_1, x_2) = x_1/x_2$. В аналитической форме $f_6(x_1, x_2) = x_1 \wedge x_2$.

Для обозначения функции эквивалентности, или равнозначности (f_7), используется символ « \sim »: $f_7(x_1, x_2) = x_1 \sim x_2$. В аналитической форме $f_7(x_1, x_2) = (x_1 \wedge x_2) \vee (\bar{x}_1 \wedge \bar{x}_2)$.

Таблица 6.3

Таблица истинности некоторых логических функций

Номер набора	x_1	x_2	Дизъюнкция (f_3)	Конъюнкция (f_4)	Стрелка Пирса (f_5)	Штрих Шеффера (f_6)	Равнозначность (f_7)	Импликация (f_8)	Сложение по модулю 2 (f_9)
0	0	0	0	0	1	1	1	1	0
1	0	1	1	0	0	1	0	1	1
2	1	0	1	0	0	1	0	0	1
3	1	1	1	1	0	0	1	1	0

Для обозначения функции импликации x_1 в x_2 (f_8) используется символ « \rightarrow »: $f_8(x_1, x_2) = x_1 \rightarrow x_2$. В аналитической форме $f_8(x_1, x_2) = \bar{x}_1 \vee x_2$.

Для обозначения функции сложения по модулю 2 (f_9) используется символ « \oplus »: $f_9(x_1, x_2) = x_1 \oplus x_2$. В аналитической форме $f_9(x_1, x_2) = (x_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2)$.

Табл. 6.3 является таблицей истинности для всех вышеперечисленных логических функций двух аргументов. Эти функции пронумерованы, начиная с цифры 3. Обычно цифры 1 и 2 оставлены для нумерации логических функций одного переменного: функции повторения и функции отрицания (инверсии).

6.3. Законы алгебры логики

Функции конъюнкции и дизъюнкции обладают рядом свойств, аналогичных свойствам обычных операций умножения и сложения. Для показа этих свойств удобно использовать аналитическую форму записи. Для указанных функций справедливы следующие законы.

Сочетательный закон:

$$x_1 \wedge (x_2 \vee x_3) = (x_1 \wedge x_2) \wedge x_3; \quad x_1 \vee (x_2 \wedge x_3) = (x_1 \vee x_2) \vee x_3.$$

При выполнении одинаковых логических действий можно изменять их последовательность, для чего ставить скобки иным способом.

Переместительный закон:

$$x_1 \wedge x_2 = x_2 \wedge x_1; \quad x_1 \vee x_2 = x_2 \vee x_1.$$

Этот закон справедлив для произвольного числа переменных. Последовательность одинаковых действий над аргументами можно изменять.

Распределительный закон:

$$x_1 \wedge (x_2 \vee x_3) = (x_1 \wedge x_2) \vee (x_1 \wedge x_3); \quad x_1 \vee (x_2 \wedge x_3) = (x_1 \vee x_2) \wedge (x_1 \vee x_3).$$

Распределительный закон умножения по отношению к сложению имеет такой же вид, как и для алгебраических умножения и сложения, а для сложения по отношению к умножению является специфичным для алгебры логики и не имеет аналога в обычной алгебре. Подобно переместительному закону, распределительный справедлив для произвольного числа аргументов. Благодаря некоторому сходству законов обычной алгебры и алгебры логики логическую функцию дизъюнкция называют логическим сложением, а логическую функцию конъюнкция называют логическим умножением.

Из распределительного закона могут быть получены следующие формулы:

$$x_1 \wedge (\bar{x}_1 \vee x_2) = x_1 \wedge x_2; \quad x_1 \vee (\bar{x}_1 \wedge x_2) = x_1 \vee x_2.$$

Справедливы также следующие тождества:

$$x_1 \vee (x_2 \wedge x_1) = x_1; \quad (x_1 \wedge x_2) \vee (x_1 \vee \bar{x}_2) = x_1.$$

$$x_1 \wedge (x_1 \vee x_2) = x_1; \quad (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2) = x_1.$$

Связь между дизъюнкцией и конъюнкцией реализуется с помощью операции инверсии (отрицания) и определяется законом двойственности (правилом, или формулами, де Моргана):

$$x_1 \vee x_2 = \overline{\bar{x}_1 \wedge \bar{x}_2}; \quad x_1 \wedge x_2 = \overline{\bar{x}_1 \vee \bar{x}_2}.$$

Для дизъюнкции, конъюнкции и инверсии справедливы также следующие тождества:

$$x \vee x = x; \quad x \vee 1 = 1;$$

$$x \wedge x = x; \quad x \wedge 1 = x;$$

$$x \vee \bar{x} = 1; \quad x \vee 0 = x;$$

$$x \wedge \bar{x} = 0; \quad x \wedge 0 = 0.$$

Из приведенных формул следует, что дизъюнкция в ее простейшей форме содержит каждый аргумент не более одного раза, так как члены $x \vee x \vee x \dots$, а также члены $\bar{x} \vee \bar{x} \vee \bar{x} \dots$ можно заменить на один член x или \bar{x} , а сочетание $x \vee \bar{x}$ обращает дизъюнкцию независимо от наличия в ней других членов в единицу. Аналогично конъюнкция в ее простейшей форме также содержит каждый элемент не более одного раза.

Для функции сложения по модулю 2 справедливы законы:

переместительный — $x_1 \oplus x_2 = x_2 \oplus x_1$;

сочетательный — $x_1 \oplus (x_2 \oplus x_3) = (x_1 \oplus x_2) \oplus x_3$;

распределительный относительно конъюнкции —

$$x_1 \wedge (x_2 \oplus x_3) = (x_1 \wedge x_2) \oplus (x_1 \wedge x_3).$$

Для функции сложения по модулю два справедливы следующие тождества:

$$x \oplus x = 0; \quad x \oplus 0 = x;$$

$$x \oplus \bar{x} = 1; \quad x \oplus 1 = \bar{x}.$$

Обратите внимание на два тождества, набранные полужирным шрифтом. В результате логического сложения (дизъюнкции) или сложения по модулю 2 аргумента x и его отрицания \bar{x} получается 1. Поэтому отрицание называют не только инверсией, но и дополнением. Отрицание события x дополняет его до 1, т. е. вероятность того, что какое-либо событие произойдет или это же событие не произойдет, равна 1. Или да, или нет. Третьего не дано.

Для импликации переместительный и сочетательный законы не имеют места, поскольку для нее справедливы следующие соотношения:

$$x_1 \rightarrow x_2 = \bar{x}_2 \rightarrow \bar{x}_1; \quad x_1 \rightarrow x_2 \rightarrow x_1 = x_1.$$

Для импликации выполняются следующие тождества:

$$x \rightarrow x = 1; \quad x \rightarrow 1 = 1; \quad 0 \rightarrow x = 1;$$

$$x \rightarrow \bar{x} = \bar{x}; \quad x \rightarrow 0 = \bar{x}; \quad 1 \rightarrow x = 1.$$

Дизъюнкция и конъюнкция могут быть выражены:

через импликацию — $x_1 \vee x_2 = \bar{x}_1 \rightarrow x_2$; $x_1 \wedge x_2 = x_1 \rightarrow x_2$;

через функцию сложения по модулю два — $x_1 \vee x_2 = x_1 \oplus x_2 \oplus x_1 \wedge x_2$;

$$x_1 \wedge x_2 = x_1 \oplus x_1 \wedge \bar{x}_2.$$

Для функций Пирса и Шеффера справедлив переместительный закон: $x_1 \downarrow x_2 = x_2 \downarrow x_1$; $x_1 | x_2 = x_2 | x_1$.

Сочетательный закон для этих функций не имеет места, в связи с чем действия раскрытия скобок и вынесения за скобки для этих функций специфичны и выполняются по следующим правилам:

$$x_1 | (x_2 \downarrow x_3) = (\bar{x}_1 \downarrow x_2) | (\bar{x}_1 \downarrow x_3);$$

$$x_1 \downarrow (x_2 | x_3) = (\bar{x}_1 | x_2) \downarrow (\bar{x}_1 | x_3);$$

$$(x_1 | x_2) \downarrow (x_3 | x_4) = \overline{\bar{x}_1 \downarrow \bar{x}_2 \downarrow \bar{x}_3 \downarrow \bar{x}_4};$$

$$(x_1 \downarrow x_2) | (x_3 \downarrow x_4) = \overline{\bar{x}_1 | \bar{x}_2 | \bar{x}_3 | \bar{x}_4}.$$

Для функций Пирса и Шеффера справедливы следующие тождества:

$$x \downarrow x = \bar{x}; \quad x \downarrow 0 = \bar{x}; \quad x \downarrow 1 = 0;$$

$$x | x = \bar{x}; \quad x | 0 = 1; \quad \bar{x} | 1 = x.$$

Функции Пирса и Шеффера связаны между собой отношениями, аналогичными формулам де Моргана для дизъюнкции и конъюнкции: $x_1 | x_2 = \bar{x}_1 \downarrow \bar{x}_2$; $x_1 \downarrow x_2 = \bar{x}_1 | \bar{x}_2$.

Приведенные выше свойства элементарных функций используются при анализе и синтезе логических схем различных цифровых устройств. С помощью элементарных функций, как указывалось выше, может быть записана любая функция от произвольного числа аргументов. Наиболее употребительными формами являются формы записи с помощью инверсии, дизъюнкции и конъюнкции.

6.4. Основной базис алгебры логики

Любая логическая функция может быть записана с помощью трех элементарных функций: инверсии (НЕ), дизъюнкции (ИЛИ), конъюнкции (И). Проведя преобразования записанной таким об-

разом формулы, можно ее упростить, т.е. уменьшить число выполняемых при расчете операций. Однако такие преобразования в целях упрощения сами по себе не так уж просты, для их выполнения требуется опыт, надо ими часто заниматься. Поэтому порой при расчете вручную можно просто последовательно подставлять все возможные наборы переменных и определять для них значение функции. Это не очень сильно удлинит расчет. Но в вычислительной технике значения логической функции определяет не человек, а комбинационное устройство. Чтобы это устройство состояло из меньшего числа элементов, а сами элементы были в максимальной степени однотипными, необходимо преобразовать реализуемую логическую функцию в определенную форму. В алгебре логики различают несколько форм, которым соответствуют наиболее рациональные построения схем комбинационных устройств.

Логические функции, представляющие собой дизъюнкции отдельных членов, каждый из которых, в свою очередь, есть некоторая функция, содержащая только конъюнкции и инверсии, называются *логическими функциями дизъюнктивной формы*. Форма представления дизъюнктивной функции, в которой инверсия применяется лишь непосредственно к аргументам, но не к более сложным функциям этих аргументов, называется *дизъюнктивной нормальной формой* (ДНФ) представления функций. Если же каждый член ДНФ от n аргументов содержит все эти n аргументов (с инверсией или без нее), то такая форма представления функции называется *совершенной дизъюнктивной нормальной формой* (СДНФ).

Логические функции, представляющие собой конъюнкцию отдельных членов, каждый из которых есть функция, содержащая только дизъюнкции и инверсии, называются *логическими функциями конъюнктивной формы*. По аналогии с дизъюнктивными формами возможны *конъюнктивные нормальные формы* (КНФ) и *совершенные конъюнктивные нормальные формы* (СКНФ).

Возможность записи функций в дизъюнктивных и конъюнктивных формах определяется исходя из следующего. Рассмотрим произвольную логическую функцию от n аргументов типа конститuentы единицы. Как указывалось ранее, конститuenta единицы полностью определяется заданием набора, обращающего ее в 1. В этом наборе некоторые аргументы могут быть равны 1, а остальные равны 0. Составим конъюнкцию от всех n аргументов, причем те из аргументов, которые в указанном наборе равны 0, возьмем с инверсией, а аргументы, равные 1, — без инверсии. Если все аргументы будут соответствовать заданному набору, то при нем функция обратится в конъюнкцию n единиц и будет равна 1. Для всех остальных наборов хотя бы один член конъюнкции, а значит, и вся конъюнкция обратится в 0. Таким образом, произвольная функция от n аргументов типа конститuentы единицы может быть выражена через конъюнкцию и инверсию.

Например, функция четырех аргументов, которая обращается в 1 при $x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 0$ и в 0 при всех остальных наборах, может быть записана в виде $f(x_1, x_2, x_3, x_4) = \bar{x}_1 \wedge x_2 \wedge x_3 \wedge \bar{x}_4$.

Аналогично произвольную логическую функцию от n аргументов типа конъюнкты нуля можно выразить через дизъюнкцию и инверсию. Например, если функция четырех аргументов при $x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1$ обращается в 0, а при всех остальных наборах равна 1, то она может быть записана в виде $f(x_1, x_2, x_3, x_4) = \bar{x}_1 \vee x_2 \vee x_3 \vee \bar{x}_4$.

Переход от табличного задания функции к записи в виде СДНФ выполняется в такой последовательности:

- в таблице задания функции выбираются все наборы аргументов, при которых функция обращается в 1;
- выписываются конъюнкции, соответствующие этим наборам аргументов. Если аргумент x_i входит в данный набор как 1, то он вписывается без изменения в конъюнкцию, соответствующую данному набору; если же x_i входит в данный набор как 0, то в соответствующую конъюнкцию вписывается его отрицание;
- все полученные конъюнкции соединяются между собой знаками дизъюнкции.

Пример. Функция $f(x_1, x_2, x_3)$ задана следующей таблицей.

Номер набора	x_1	x_2	x_3	$f(x_1, x_2, x_3)$	Номер набора	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	0	4	1	0	0	1
1	0	0	1	0	5	1	0	1	0
2	0	1	0	0	6	1	1	0	1
3	0	1	1	1	7	1	1	1	0

Требуется получить для нее дизъюнктивную совершенную нормальную форму.

Для нахождения СДНФ выбираем из таблицы только те наборы значений аргументов, которые обращают функцию в 1, т.е. третий, четвертый и шестой. Выписываем конъюнкции, соответствующие выбранным наборам: $\bar{x}_1 \wedge x_2 \wedge x_3; x_1 \wedge \bar{x}_2 \wedge \bar{x}_3; x_1 \wedge x_2 \wedge \bar{x}_3$.

Соединяя эти конъюнкции знаками дизъюнкции, окончательно получаем

$$f(x_1, x_2, x_3) = (\bar{x}_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \bar{x}_2 \wedge \bar{x}_3) \vee (x_1 \wedge x_2 \wedge \bar{x}_3).$$

Аналогично выполняется переход от табличного задания функции к записи в виде СКНФ:

- в таблице задания функции выбираются все наборы аргументов, при которых функция обращается в 0;
- вписываются дизъюнкции, соответствующие этим наборам аргументов. Если аргумент x_i входит в данный набор как 0, он вписывается без изменения в дизъюнкцию, соответствующую данному набору; если же x_i входит в данный набор как 1, то в соответствующую дизъюнкцию вписывается его отрицание;
- все полученные дизъюнкции соединяются между собой знаками конъюнкции.

Пример. Написать СКНФ для функции, заданной следующей таблицей.

Номер набора	x_1	x_2	x_3	$f(x_1, x_2, x_3)$	Номер набора	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	0	4	1	0	0	1
1	0	0	1	1	5	1	0	1	1
2	0	1	0	1	6	1	1	0	1
3	0	1	1	0	7	1	1	1	0

Ответ: $f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$.

Выбор той или иной формы аналитической записи определяется видом таблицы заданной функции. Если большинство значений данной функции 0, то выгодно записывать ее в СДНФ, в противном случае более экономную запись дает СКНФ.

Таким образом, любую функцию алгебры логики можно записать в виде СДНФ или СКНФ, т. е. представить с помощью системы из трех элементарных функций: инверсии, дизъюнкции и конъюнкции. Существуют и другие системы функций, с помощью которых может быть выражена произвольная функция. Полная система функций, с помощью которой можно представить любую функцию от произвольного числа аргументов, называется *базисом*. Базис называют *полным*, если любая функция F представима суперпозицией функций, составляющих рассматриваемый базис. Под суперпозицией функций понимают любую последовательность функциональных отношений из перечня функций, составляющих базис. Иными словами, из функций, составляющих базис, можно записать любую сложную функцию.

Базис из дизъюнкции, конъюнкции и инверсии не является минимальным. Поскольку конъюнкцию можно представить через инверсию и дизъюнкцию, то можно получить минимальный базис из дизъюнкции и инверсии. С другой стороны, дизъюнкцию можно представить через инверсию и конъюнкцию, т. е. получить минимальный базис из конъюнкции и инверсии.

Возможны и другие минимальные базисы, например из конъюнкции и инверсии. Каждая из функций Пирса и Шеффера также составляет полную систему и представляет собой минимальный базис. Благодаря тому, что любую, сколь угодно сложную логическую функцию можно реализовать с помощью малого числа типов логических операций (например, только дизъюнкции и инверсии), самые сложные электросхемы могут быть построены из однотипных элементов. Таких элементов потребуется очень много, но при массовом производстве они будут очень дешевы. К тому же наладить массовое производство всего одного или двух типов элементов значительно дешевле, чем большого числа типов. Особенно это важно при производстве больших интегральных схем.

Имеются несколько систем элементарных логических операций, между которыми существует однозначная связь. Большей частью используется система из трех приведенных выше элементарных операций: НЕ, ИЛИ и И. Следует иметь в виду, что эта система избыточна, т. е. для осуществления любых логических операций достаточно иметь систему только из двух операций (либо из НЕ и ИЛИ, либо из НЕ и И). В первой системе операция И через НЕ и ИЛИ выражается так: $A \wedge B = \overline{(\overline{A} \vee \overline{B})}$. Во второй системе операция $A \vee B = \overline{(\overline{A} \wedge \overline{B})}$.

Комбинационные схемы, интерпретирующие поведение булевых функций, обычно реализуют либо нулевую, либо единичную константы, так как только одна комбинация уровней сигналов, интерпретирующих аргументы, определяет закрытое или открытое состояние некоторой электрической цепи.

6.5. Минимизация логических функций

Проблема полноты базиса, рассмотренная в подразд. 6.4, эквивалентна проблеме выбора стандартного (типового) набора логических элементов, из которых будет строиться автомат, выполняющий операции с наборами, составленными из нулей и единиц. Если, например, в качестве базиса выбраны функции отрицания, дизъюнкции и конъюнкции, что соответствует стандартному (типовому) набору логических элементов, состоящему из элементов трех типов, то все функции могут быть представлены в ДСНФ или КСНФ и после этого реализованы на стандартных элементах. Уменьшение числа функций, входящих в базис, соответствует уменьшению числа различных логических элементов, принятых за стандартные.

Однако следует учитывать и другую сторону вопроса. При реализации автомата важно не только число типов стандартных элементов, но и общее число элементов, из которых построен авто-

мат. При этом сложность автомата с точки зрения числа использованных элементов существенно зависит от вида реализуемых им функций и функций, выбранных в качестве базиса. Возникает задача о простейшем представлении реализуемой функции через систему базисных функций.

В ряде случаев запись в СДНФ или СКНФ не является самой простой для выражения заданной функции в аналитической форме и можно упростить логическое выражение, не нарушая значения функции. Методы такого упрощения называются *методами минимизации*. В результате минимизации логические функции могут быть представлены в ДНФ или КНФ с минимальным числом членов и минимальным числом аргументов в каждом члене. Для упрощения выражений функций алгебры логики разработаны как графические, так и алгебраические методы.

Из большого числа различных приемов и методов минимизации логических функций рассмотрим только один. При небольшом числе переменных удобен графический метод упрощения выражений для функций алгебры логики с помощью карт Карно. Карта Карно представляет собой определенную форму таблицы истинности для двух, трех и четырех аргументов (см. рис. 6.1 — 6.4). Каждая клетка соответствует конкретному набору аргументов, причем этот набор определяется присвоением значения 1 аргументам, на пересечении строк и столбцов которых расположена клетка. Число клеток карты равно числу возможных наборов значений аргументов (при n аргументах равно 2^n). В каждую клетку записывается значение функции при соответствующем этой клетке наборе значений аргументов. Например, если функция $f(x_1, x_2, x_3)$ задана таблицей истинности (см. табл. 6.2), то в форме карты Карно эта функция будет представлена так, как показано на рис. 6.6, а. При этом, например, две клетки верхней строки, содержащие 1, соответствуют следующим наборам: первая — $x_1 = 1; x_2 = 1$;

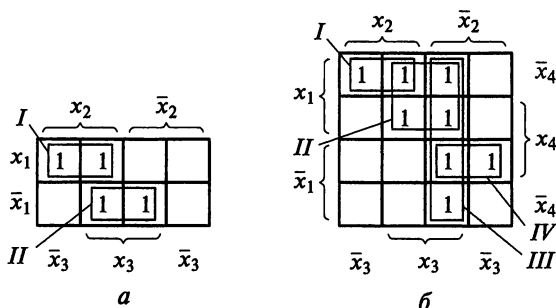


Рис. 6.6. Выделение замкнутых областей на картах Карно:
а — для трех переменных; б — для четырех переменных

$\bar{x}_3 = 1 (x_3 = 0)$; вторая — $x_1 = 1; x_2 = 1; x_3 = 1$. В свою очередь, две аналогичные клетки нижней строки соответствуют наборам:

первая — $\bar{x}_1 = 1; x_2 = 1; x_3 = 1$; вторая — $\bar{x}_1 = 1; \bar{x}_2 = 1; x_3 = 1$.

При записи функции в минимальной форме по карте Карно используют следующие правила.

Все клетки, содержащие 1, объединяют в замкнутые области. При этом каждая область должна представлять собой прямоугольник с числом клеток 2^k , где $k = 0, 1, 2, 3, \dots$. Области могут пересекаться, т. е. одни и те же клетки могут входить в разные области. Затем производят запись минимального выражения в дизъюнктивной нормальной форме (МДНФ). Каждая область в такой записи представляется членом, число аргументов в котором на k меньше общего числа аргументов функции n , т. е. равно $n - k$. Каждый член МДНФ составляется лишь из тех аргументов, которые для соответствующей области имеют значения либо без инверсий, либо только с инверсией.

Таким образом, при охвате клеток замкнутыми областями следует стремиться, чтобы число областей было минимальным (так как при этом будет минимальным число членов в МДНФ), а каждая область содержала возможно большее число клеток, поскольку при этом число аргументов в членах будет минимальным. Так, для функции трех аргументов, представленной на рис. 6.6, а, клетки, содержащие 1, охватываются двумя областями. В каждой области две клетки, и так как $2^k = 2$, то, следовательно, $k = 1$. Поэтому для этих областей $n - k = 3 - 1 = 2$. В результате в МДНФ будут два члена и в каждом из них — два аргумента. Первой области соответствует импликанта $x_1 \wedge x_2$, второй — импликанта $\bar{x}_1 \wedge x_3$. Следовательно, минимальная ДНФ для этой функции будет $f(x_1, x_2, x_3) = x_1 x_2 \vee \bar{x}_1 x_3$.

Пример задания функции четырех аргументов с помощью карты Карно приведен на рис. 6.6, б, где выделены четыре области. Области I и IV имеют по две клетки; для них $n - k = 4 - 1 = 3$, соответствующие им члены будут $x_1 \wedge x_2 \wedge \bar{x}_4$ и $\bar{x}_1 \wedge \bar{x}_2 \wedge x_4$. Области II и III содержат по четыре клетки; для них $n - k = 4 - 2 = 2$, т. е. в МДНФ они будут выражены членами, содержащими по два аргумента: $x_1 \wedge x_3$ и $\bar{x}_2 \wedge x_3$. Таким образом, минимальная форма функции $f(x_1, x_2, x_3, x_4) = (x_1 \wedge x_2 \wedge \bar{x}_4) \vee (x_1 \wedge x_3) \vee (\bar{x}_2 \wedge x_3) \vee (\bar{x}_1 \wedge \bar{x}_2 \wedge x_4)$.

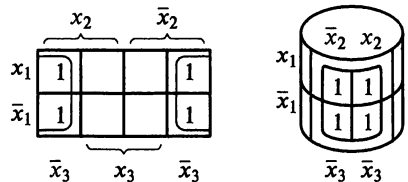


Рис. 6.7. Сворачивание карты Карно в цилиндр

При построении замкнутых областей допускается сворачивать карты в цилиндр как по горизонтальной, так и по вертикальной осям с объединением противоположных граней карты. На рис. 6.7 показано, как сворачивается карта Карно для объединения области из четырех клеток.

Контрольные вопросы

1. Что такое логическая функция?
2. Какие логические функции считаются элементарными?
3. Приведите пример сложного логического высказывания.
4. Почему логическую операцию конъюнкция называют логическим умножением?
5. Почему логическую операцию дизъюнкция называют логическим сложением?
6. Какие существуют способы представления логических функций?
7. Что такое основной базис алгебры логики?
8. Что такое логическая функция конъюнктивной формы?
9. Что такое логическая функция дизъюнктивной формы?

РАЗДЕЛ III

ЭЛЕМЕНТЫ И УСТРОЙСТВА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Глава 7

ТИПОВЫЕ ЭЛЕМЕНТЫ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

7.1. Назначение типовых элементов

В гл. 3 отмечалось, что ЦВМ состоит из трех основных функциональных устройств: процессорного, ввода и вывода, запоминающего. Эти устройства взаимодействуют, т. е. каждое из них выполняет определенные функции и обменивается с другими устройствами информацией. При этом все они вместе образуют систему. Под *системой* будем понимать совокупность устройств различного назначения, взаимодействие которых позволяет получить новое качество, которым не обладало ни одно из этих устройств. Каждое из основных функциональных устройств может состоять из более простых функциональных узлов. Простые функциональные узлы состоят из отдельных элементов. Соединение элементов и их взаимодействие приводит к появлению нового качества. Поэтому простой функциональный узел также является системой. Аналогично и объединение нескольких простых функциональных узлов, каждое из которых выполняет несложную функцию, позволяет получить то или иное основное функциональное устройство, выполняющее уже более сложную функцию.

Поскольку в ЦВМ информация представлена в двоичной форме, т. е. в виде набора (нескольких разрядов) нулей и единиц (двух условных знаков), то элементом будем считать простейшую часть, выполняющую какую-либо операцию с условным знаком, который находится в одном разряде. Какие же функции (т. е. какие преобразования или какую обработку) должны выполнять элементы вычислительной техники? Прежде всего, это логические и арифметические операции. Кроме того, это запоминание и хранение условного знака (нуля или единицы). Несколько логических элементов могут выполнять преобразование сразу в нескольких разрядах, поэтому они способны обрабатывать уже всю двоичную комбинацию (слово или число) целиком. Нескольким логическим элементам под силу более сложные логические операции. Такие элементы представляют собой уже простые функциональ-

ные узлы. К типовым элементам вычислительной техники будем относить и собственно элементы, и простые функциональные узлы.

Таким образом, ЦВМ представляет собой систему, состоящую из основных функциональных устройств. Основное функциональное устройство также представляет собой систему (будем называть ее подсистемой, поскольку она входит в состав более крупной системы), состоящую из типовых элементов вычислительной техники, к которым кроме собственно элементов отнесем построенные из них простые функциональные узлы.

В процессорном устройстве, или процессоре, выполняются арифметические и логические операции. В его состав кроме логических элементов входят преобразователи информации, представленной в виде двоичных комбинаций. Такие преобразователи называются *шифраторами* и *дешифраторами*, а общее для них название — *кодирующие* и *декодирующие элементы*. Для преобразования двоичных комбинаций в аналоговый (непрерывный) сигнал служат цифроаналоговые преобразователи, для обратного преобразования — аналого-цифровые. Обычно такие преобразователи используют в устройствах ввода информации в ЭВМ. С их помощью вводится информация от различных датчиков, измеряющих температуру, давление, расход, скорость и другие различные параметры. Эта информация нужна в системах автоматического управления технологическими и производственными процессами.

Сначала были созданы аналоговые системы автоматического управления и регулирования, датчики для которых имели аналоговые выходные сигналы. С появлением ЦВМ одной из наиболее эффективных областей их применения стало автоматическое управление. Но для того чтобы ЦВМ могли воспринимать («понимать») сигналы от уже существовавших датчиков, потребовались преобразователи аналогового сигнала в цифровой, т.е. в двоичную кодовую комбинацию. Эту задачу и выполняют аналого-цифровые преобразователи. Получив необходимую информацию от датчиков, управляющая ЦВМ обрабатывает ее и вырабатывает регулирующее воздействие для подачи на исполнительные устройства в системе автоматики. Большинство исполнительных устройств имеют в своем составе электродвигатели, подавляющая часть которых питается обычным непрерывным (постоянным или переменным) напряжением. Для того чтобы исполнительные устройства воспринимали выработанное управляющей ЦВМ регулирующее воздействие, и служат цифроаналоговые преобразователи, используемые в устройствах вывода.

В настоящее время аналого-цифровые и цифроаналоговые преобразователи шире всего применяются в цифровой телефонии. Вначале для передачи голосовых сообщений по проводам использовались только аналоговые сигналы. В городах были проложены миллионы проводных линий от телефонных аппаратов к телефон-

ным станциям. Стремление уменьшить число этих линий, обеспечить передачу по одной паре проводов нескольких одновременно ведущихся разными абонентами разговоров привело специалистов телефонной связи к идее преобразования аналогового сигнала в цифровой двоичный код. Независимо от специалистов в области вычислительной техники ими были разработаны соответствующие преобразователи и средства уплотнения каналов связи. А во второй половине XX в. оказалось, что достижения в областях вычислительной техники и цифровой телефонии можно направить на общее дело и получить в итоге всемирную сеть, которая объединяет миллионы ЭВМ с помощью телефонных проводов и иных каналов, связывающих телефонные аппараты с телефонными станциями.

Для уплотнения каналов связи служат *мультиплексоры*, объединяющие для передачи по одной линии информацию от многих источников. После прохождения этой информации по общему каналу связи она должна быть распределена уже по соответствующим потребителям. Для этого применяются *демультиплексоры*.

Полученная в результате обработки в процессоре новая информация должна быть сохранена, записана. Для хранения одного разряда двоичной комбинации служат *триггеры*. Для хранения всей комбинации, т.е. нескольких разрядов, используются *регистры*, состоящие из нескольких триггеров. В регистрах может выполняться и обработка двоичной информации. Поэтому на базе регистров могут быть построены *счетчики*, *регистры сдвига*, *сумматоры*.

В процессе обработки информации часто возникает задача сравнения различных сигналов по величине. Для этого служат *компараторы*.

7.2. Основные логические элементы

Логический элемент — это устройство, реализующее одну из логических операций. Логические элементы, используемые в вычислительной технике и системах автоматики, основаны на использовании самых различных физических явлений и свойств. Наиболее часто применяются электронные устройства в виде интегральных микросхем. Промышленностью выпускаются серии интегральных схем, выполняющих самые разнообразные логические операции. Например, широко применяемые серии К155, К555, К1533 и ряд других включают в себя более 150 микросхем, в том числе около 50 схем логических элементов различного функционального назначения. Любую логическую функцию можно выполнить с помощью логических операций И, ИЛИ, НЕ. Эти операции называются элементарными, а устройства для их реализации называются *элементарными логическими элементами*. На рис. 7.1 показаны условные обозначения логических элементов НЕ, ИЛИ, И.

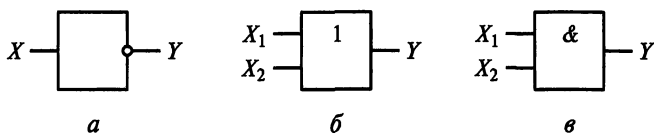


Рис. 7.1. Условные обозначения элементарных логических элементов:
 а — НЕ; б — ИЛИ; в — И

Все логические элементы изображаются в виде прямоугольников с линиями, по которым подводятся входные и отводятся выходные сигналы. Обычно слева располагаются линии входных сигналов, а справа — выходных. В прямоугольнике ставится знак логической операции: & — И, 1 — ИЛИ. Если выход обозначен окружностью, то элемент производит логическое отрицание результата операции, указанной внутри прямоугольника. Логическое отрицание называют инверсией, а выход, обозначенный окружностью, называют *инверсным выходом*. На рис. 7.1 входные сигналы обозначены буквой X , а выходные — Y . Работу элементов НЕ, ИЛИ, И поясняют табл. 7.1 — 7.3, в которых показано соответствие выходного сигнала любой возможной комбинации входных сигналов. Такие таблицы называются таблицами истинности, или таблицами переключений.

На рис. 7.1 у элементов И и ИЛИ по два входа, но они выпускаются и с большим числом входов. Логический элемент И называют также *конъюнктом*, а логический элемент ИЛИ — *дизъюнктом*.

Входные и выходные сигналы могут принимать одно из двух значений: логическая 1 и логический 0. При конкретной реализации эти сигналы представляются различными физическими величинами (например, электрическим напряжением или потенциалом). Знание абсолютной величины сигнала при этом не требуется, достаточно различать более положительную и менее положительную величину. Сигнал может иметь и отрицательную полярность. На рис. 7.2 эти два значения обозначены латинскими буква-

Таблица 7.1

Таблица истинности логического элемента НЕ

X	Y
0	1
1	0

Таблица 7.2

Таблица истинности логического элемента ИЛИ

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

Таблица 7.3

Таблица истинности логического элемента И

X_1	X_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

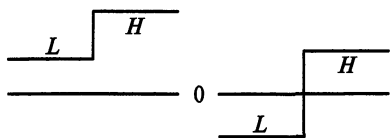


Рис. 7.2. Логические уровни сигналов

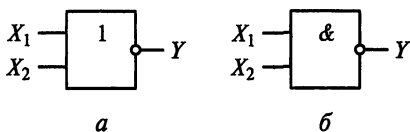


Рис. 7.3. Условные обозначения базовых логических элементов:

a — ИЛИ—НЕ; *б* — И—НЕ

ми *H* (от *англ.* high — высокий) и *L* (от *англ.* low — низкий). Указанные значения называются *логическими уровнями*. Один из них принимается за 1, другой — за 0 (в зависимости от договоренности, соглашения). Различают соглашение положительной логики (при котором логический уровень *H* принимают за 1, а логический уровень *L* — за 0) и соглашение отрицательной логики (при котором *H* принимают за 0, а *L* — за 1). Обычно принимают единое соглашение для всей схемы, или используют указатели полярности сигналов.

Логические элементы И и ИЛИ обладают свойством двойственности — один и тот же элемент в зависимости от принятого соглашения может выполнять функции либо элемента И, либо ИЛИ. Как уже отмечалось, любая сложная логическая функция может быть выполнена с помощью элементарных логических элементов И, ИЛИ, НЕ. Но есть возможность выполнить любую сложную логическую функцию и с помощью некоторого количества совершенно однотипных элементов, реализующих только одну операцию. Например, есть серии микросхем, построенных на основе составной логической схемы И—НЕ, а есть серии, построенные на основе составной логической схемы ИЛИ—НЕ. Условные обозначения логических элементов И—НЕ и ИЛИ—НЕ показаны на рис. 7.3. Применение единого базового элемента для всей серии позволяет использовать единую технологию для всей серии микросхем, увеличить объем выпуска и, следовательно, снизить стоимость каждого элемента.

Таблица 7.4

Таблица истинности логического элемента ИЛИ—НЕ

X_1	X_2	Y
0	0	1
0	1	0
1	0	0
1	1	0

Таблица 7.5

Таблица истинности логического элемента И—НЕ

X_1	X_2	Y
0	0	1
0	1	1
1	0	1
1	1	0

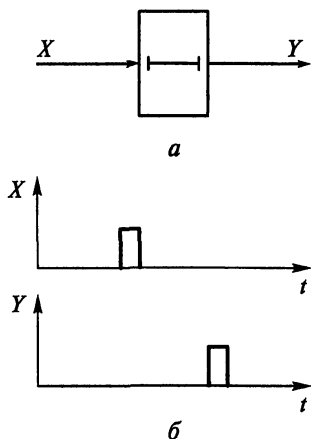


Рис. 7.4. Элемент задержки:
 а — условное обозначение;
 б — графики входного и вы-
 ходного сигналов

В табл. 7.4 и 7.5 приведены соотношения между комбинациями входных сигналов и выходным сигналом для базовых логических элементов (т. е. таблицы истинности).

По виду входных и выходных сигналов логические элементы делятся на потенциальные и импульсные. В потенциальных элементах сигналы 1 и 0 представляются двумя уровнями (см. рис. 7.2), а в импульсных — наличием или отсутствием импульсов (или импульсами разной полярности). Наиболее распространены потенциальные элементы.

Полученную в результате логических операций информацию в виде двоичных кодовых слов необходимо запоминать и хранить. Для этой цели служат устройства памяти (триггеры и регистры).

Благодаря наличию элементов памяти на работу цифрового устройства могут оказывать влияние не только результаты преобразования информации в комбинационных схемах, но и результаты ранее выполненных операций. Такое автоматическое устройство называется уже не комбинационным, а *конечным автоматом*. Если про комбинационный автомат говорят, что он «не помнит», то конечный автомат «помнит». Использовать результат предыдущей операции, выполненной комбинационным устройством, можно и с помощью специального элемента, который называется *элементом задержки*. Такой элемент реализует операцию, описываемую простым уравнением: $Y_t = X_{t-1}$. Это означает, что выходной сигнал схемы в некоторый момент времени t равен входному сигналу, поступившему на эту схему в момент времени $t - 1$. Другими словами, входная величина задерживается в этой схеме на некоторую единицу времени, которую иногда называют *тактом задержки*. Такая схема носит также название *линия задержки*. На рис. 7.4 показан элемент задержки, т. е. одноканальная линия задержки. При необходимости получить задержку на несколько тактов (условных единиц времени) последовательно включают соответствующее количество таких элементов.

7.3. Триггеры

Основным устройством, которое способно запомнить цифровую информацию, является триггер. Он имеет два устойчивых состояния, одно из которых принимается за 1, а другое — за 0. В вычислительной технике наибольшее распространение получи-

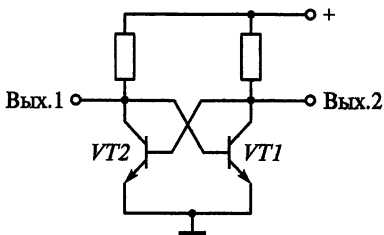


Рис. 7.5. Схема статического триггера

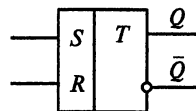


Рис. 7.6. Условное обозначение RS-триггера

ли полупроводниковые триггеры, выпускаемые в виде интегральных микросхем. Они, как правило, являются двухкаскадными усилителями постоянного тока с положительной обратной связью (выход усилителя соединен с его входом).

Схема простейшего статического триггера показана на рис. 7.5. Из двух транзисторов один обязательно открыт, а другой закрыт. Если закрыт транзистор $VT1$, то положительный потенциал с его коллектора подается на базу транзистора $VT2$, и наоборот. Соединение коллектора одного транзистора с базой другого и обеспечивает положительную обратную связь. Несмотря на полную симметрию схемы, такое ее состояние, когда оба транзистора открыты, является неустойчивым и практически невозможным. Даже незначительное случайное увеличение коллекторного тока одного транзистора вызывает уменьшение положительного потенциала на его коллекторе и соответственно на базе другого транзистора. Это приводит к уменьшению коллекторного тока другого транзистора, увеличению потенциала на его коллекторе и соответственно на базе первого транзистора. В итоге первый транзистор еще больше открывается, а второй еще больше закрывается. Этот процесс протекает очень быстро (лавинообразно) и заканчивается тогда, когда первый транзистор полностью открывается (режим насыщения), а второй транзистор полностью закрывается, поскольку на его базу будет подан практически нулевой потенциал.

Перевод триггера из одного устойчивого состояния в другое осуществляется подачей положительных или отрицательных импульсов на коллектор одного или другого транзистора. При этом один из входов принимают устанавливающим триггер в состояние 1 и называют S (от *англ.* set — установить) а другой, устанавливающий (сбрасывающий) триггер в состояние 0, называют входом R (от *англ.* reset — сбросить). Такой триггер называют RS-триггером.

Его условное обозначение показано на рис. 7.6. В сериях микросхем RS-триггеры обычно построены на двух базовых логических элементах, на которых основана вся серия, т. е. либо на двух ИЛИ — НЕ (рис. 7.7, а, б), либо на двух И — НЕ (рис. 7.7, в, г).

Поясним работу RS -триггера на базе элементов ИЛИ — НЕ (см. рис. 7.7, *a*) с помощью диаграммы, показанной на рис. 7.7, *б*. После поступления сигнала 1 на вход S триггер переключается в состояние 1, если он был в состоянии 0, или сохраняет 1 на выходе Q , если он уже находился в этом состоянии. Соответственно при поступлении 1 на вход R триггер переключается в 0 или сохраняет это состояние. Исходное состояние триггера (сразу после включения и при отсутствии сигналов 1 на входах S и R) не определено, оно является случайной величиной. В отличие от схемы, представленной на рис. 7.7, *a*, в схеме, представленной на рис. 7.7, *в*, используется отрицательная логика, т. е. 1 имеет менее положительный потенциал, чем 0.

По способу записи информации различают асинхронные и синхронные триггеры. Состояние (выходной сигнал) асинхронного триггера может измениться в любой момент — тогда, когда придет входной сигнал. В синхронном триггере состояние может меняться только в определенные моменты времени — тогда, когда поступает дополнительный синхронизирующий сигнал. RS -триг-

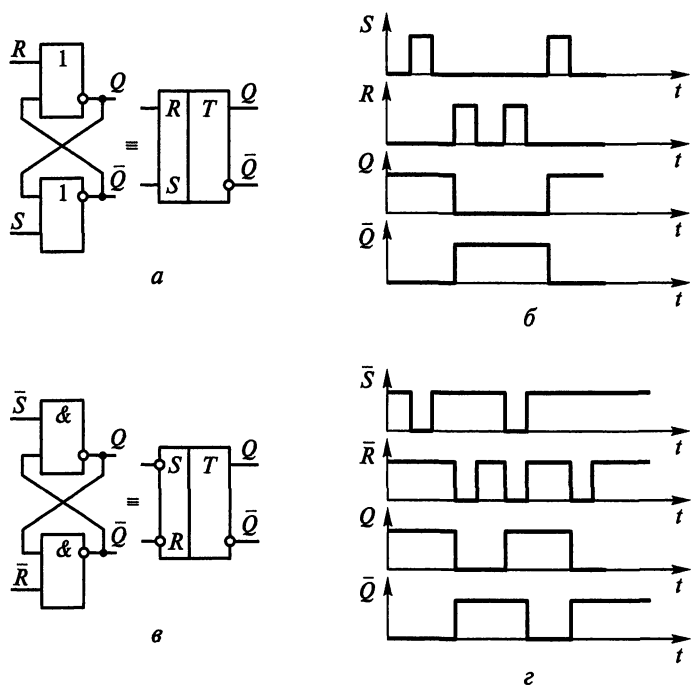


Рис. 7.7. RS -триггеры на базе элементов ИЛИ — НЕ и И — НЕ:

a — схема с положительной логикой и ее условное обозначение; *б* — диаграмма сигналов для схемы *a*; *в* — схема с отрицательной логикой и ее условное обозначение; *г* — диаграмма сигналов для схемы *в*

гер является асинхронным. На его базе может быть построен синхронный D -триггер (рис. 7.8, а).

Сигналы, предназначенные для записи в триггер, поступают на информационный вход D . Сигналы, определяющие момент записи, поступают на вход C . Изменение состояния статического D -триггера возможно только в течение того времени, когда $C = 1$. Если же на вход C поступает сигнал 0, то изменение сигнала на выходе триггера не происходит, он сохраняет предыдущее состояние. На диаграмме сигналов (рис. 7.8, б) видно, что по окончании первого синхроимпульса на информационный вход D поступал уже сигнал 0, однако состояние триггера, соответствующее этому сигналу, возникло только тогда, когда пришел второй синхроимпульс. Аналогичным образом состояние 1 на выходе Q сохранялось от третьего до пятого синхроимпульса, хотя сигнал 1 на входе D сменился на сигнал 0 раньше, чем пришел пятый синхроимпульс. Поскольку такой триггер задерживает выходной сигнал до прихода очередного синхроимпульса, он и получил название « D -триггер» (от *англ.* delay — задержка).

В динамическом синхронном D -триггере информация записывается только в момент перепада напряжения на входе C , т. е. передним фронтом синхроимпульса.

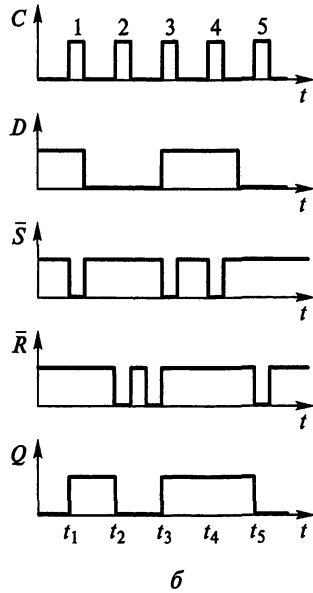
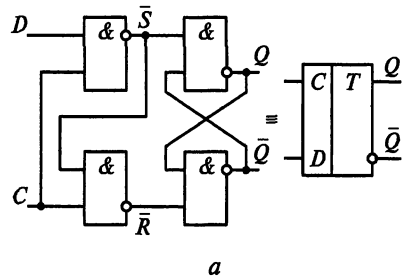


Рис. 7.8. Синхронный D -триггер: а — схема и ее условное обозначение; б — диаграмма сигналов

7.4. Регистры

Регистр представляет собой упорядоченную последовательность (совокупность) триггеров, число которых соответствует числу разрядов в слове. Регистр используется для хранения n -разрядного слова и выполнения логических преобразований над ним. В регистре могут выполняться следующие микрооперации: прием (запись)

информации составляет первый такт. Во втором такте по сигналу 1, подаваемому по шине Π (Прием), двоичное число $x_n \dots x_2 x_1$ всеми разрядами одновременно (параллельно) через конъюнкторы (элементы И) записывается в разряды регистра. Выдача числа в прямом коде осуществляется по сигналу 1, подаваемому по шине $B_{пр}$, а в обратном — по сигналу 1, подаваемому по шине $B_{обр}$.

Ввод и вывод информации в рассматриваемом регистре может осуществляться однофазным и парафазным способами. При однофазном число представляется в прямом или обратном коде; при парафазном — одновременно в прямом и обратном кодах. При парафазном вводе числа его i -й разряд в прямом и обратном кодах (x_i и \bar{x}_i) необходимо подавать на оба входа i -го триггера. Этим исключается необходимость предварительной установки триггера в 0, так как теперь его состояние целиком будет определяться сигналами на S - и R -входах, т.е. конкретной двоичной цифрой в разряде кода. Такая запись числа осуществляется в один такт и производится намного быстрее, чем двухтактная.

Параллельный регистр может быть реализован и на других типах триггеров, имеющих информационные входы.

Условное обозначение параллельного четырехразрядного регистра приведено на рис. 7.10, где $Q1-Q4$ — выходы разрядов регистра, а $D1-D4$ — входы, с которых в регистр одновременно записываются все разряды заносимого слова; C — вход, импульс на котором разрешает запись с входов $D1-D4$.

Последовательный регистр. В таких регистрах двоичное число вводится и выводится последовательно разряд за разрядом. Разряды самого регистра соединены последовательно. Каждый разряд выдает информацию в следующий разряд и одновременно принимает новую информацию из предыдущего. Для этого каждый разряд должен иметь два запоминающих элемента, т.е. сдвоенный или двухступенчатый триггер. В первую ступень передается информация из предыдущего разряда, одновременно вторая ступень передает свою информацию в последующий разряд. Затем информация, принятая первой ступенью, передается во вторую, а первая освобождается для приема новой информации. Двухступенчатый триггер (например, JK -триггер, D -триггер) представляет собой совокупность двух запоминающих элементов, поэтому он один может составлять разряд последовательного регистра. Если в цепи таких триггеров выходы одного соединить с входами другого, то по фронту тактового импульса, подаваемого на вход C , во входную (первую) ступень каждого триггера будет заноситься инфор-

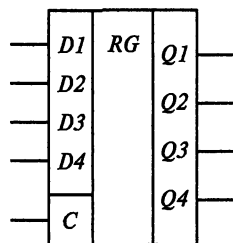


Рис. 7.10. Условное обозначение параллельного четырехразрядного регистра

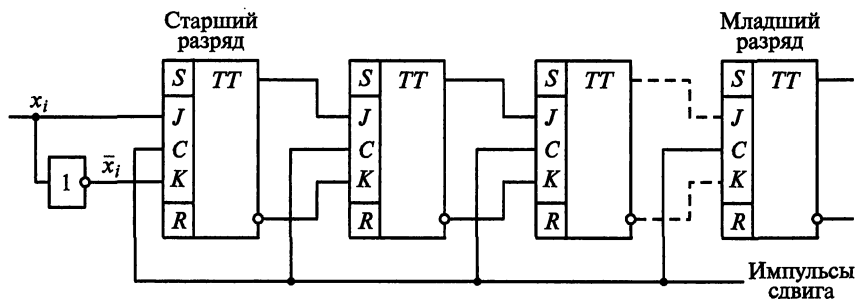


Рис. 7.11. Функциональная схема последовательного регистра

мация из выходной (второй) ступени предыдущего триггера, а по спаду импульса она будет переписываться в выходную ступень. По фронту следующего тактового импульса во входной ступени триггера информация может быть заменена новой (из предыдущего триггера) без опасения, что предыдущая окажется потерянной.

Функциональная схема последовательного регистра приведена на рис. 7.11. Крайний левый триггер предназначен для хранения старшего разряда числа, а крайний правый — для хранения младшего разряда.

Разряды двоичного числа (в виде высоких и низких потенциалов), начиная с младшего, последовательно поступают на входы старшего разряда регистра. Поступление разрядов числа на входы J и K чередуется с поступлением импульсов сдвига на входы C , которыми вводимые разряды продвигаются вдоль регистра, пока младший разряд n -разрядного числа не окажется в младшем разряде регистра.

Для выдачи записанного числа в последовательной форме надо на входы старшего разряда регистра подать $x_i = 0$, $\bar{x}_i = 1$, а на шину импульсов сдвига — n импульсов. Первый импульс выдвинет из младшего разряда регистра младший разряд числа, на его место передвинется второй разряд числа и т. д. В итоге все число сдвинется вдоль регистра на один разряд, а в старший разряд регистра будет записан 0. Второй импульс сдвига выдвинет из регистра второй разряд числа и продвинет 0 из старшего разряда регистра в соседний, более младший, и т. д. После n импульсов сдвига число будет полностью выведено из регистра, в разряды которого окажутся записанными нули.

Регистр, в котором можно осуществить сдвиг числа, называется *сдвигающим (сдвиговым)*, или просто *регистром сдвига*. Сдвигающий регистр может быть *однонаправленным* (для сдвига числа в сторону младшего разряда — правый сдвиг, в сторону старшего разряда — левый сдвиг), а также *реверсивным*, обеспечивающим сдвиг в обе стороны.

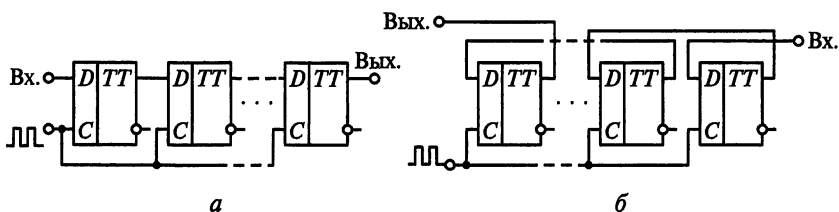


Рис. 7.12. Функциональные схемы сдвиговых регистров на D -триггерах: a — для сдвига вправо; b — для сдвига влево

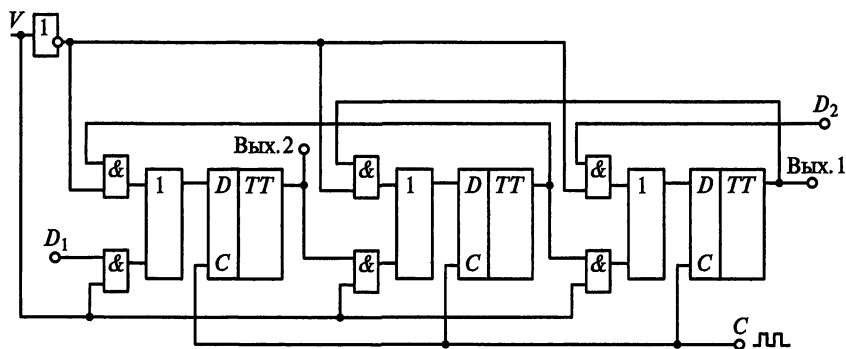


Рис. 7.13. Функциональная схема реверсивного сдвигового регистра

Функциональные схемы сдвиговых регистров на D -триггерах со сдвигом числа вправо и влево показаны соответственно на рис. 7.12, a и 7.12, b . Функциональная схема реверсивного сдвигового регистра изображена на рис. 7.13. Направление сдвига определяется сигналом V на входе. При $V = 1$ верхний ряд конъюнкторов (элементов И) заблокирован и в регистр сдвиговыми импульсами могут вдвигаться разряды числа слева направо с входа D_1 . При $V = 0$ блокируется нижний ряд конъюнкторов и слово может вдвигаться в регистр с входа D_2 справа налево.

7.5. Счетчики

Счетный триггер, или T -триггер, условное обозначение которого показано на рис. 7.14, a , имеет один вход и два выхода. Сигналы на выходах меняются на противоположные при каждом положительном перепаде напряжения на счетном входе T . Счетный триггер может быть создан на базе динамического D -триггера, если его инверсный выход соединить с информационным входом, как показано на рис. 7.14, b .

Рассмотрим работу счетного триггера с помощью диаграммы, приведенной на рис. 7.14, $в$.

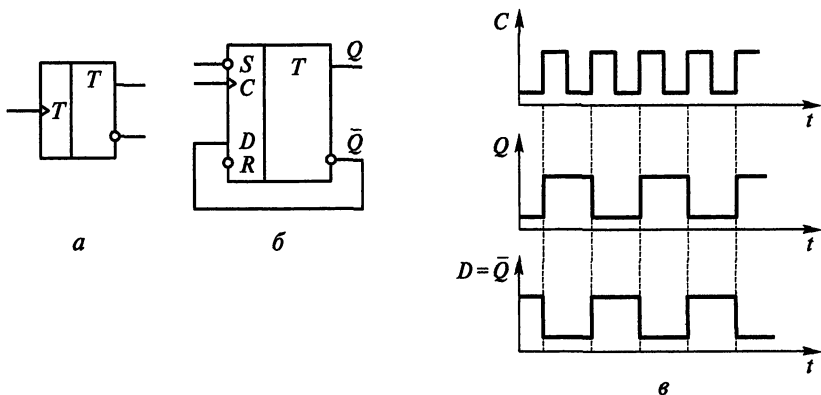


Рис. 7.14. Счетный триггер:

a — условное обозначение; *б* — схема на базе динамического *D*-триггера; *в* — диаграмма сигналов

Пусть в начальный момент времени на прямом выходе был сигнал 0, тогда на инверсном выходе и, следовательно, на входе *D* — сигнал 1. По фронту первого синхроимпульса сигнала 1 с входа переписывается на прямой выход, а на инверсном выходе появляется 0. По фронту второго синхроимпульса этот сигнал 0 переписывается на прямой выход и будет там сохраняться до прихода третьего синхроимпульса и т.д. Обратите внимание, что частота сигналов на выходе вдвое меньше входной частоты синхроимпульсов, поэтому счетный триггер называется *делителем частоты*.

Для хранения информации о многоразрядном кодовом слове используются несколько триггеров, по одному на каждый разряд. В этом случае такая группа триггеров называется регистром.

Для подсчета импульсов применяют регистры, состоящие из *T*-триггеров. На рис. 7.15 показан простой трехразрядный двоичный счетчик импульсов, состоящий из трех *T*-триггеров, которые имеют входы *R* для установки нуля.

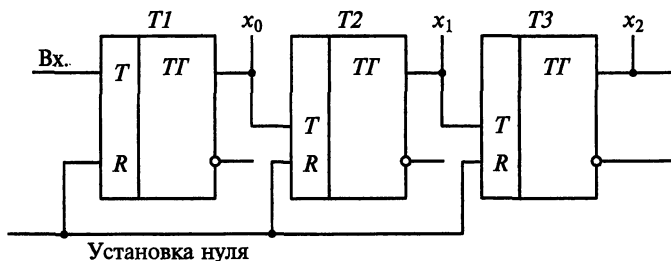


Рис. 7.15. Трехразрядный двоичный счетчик импульсов

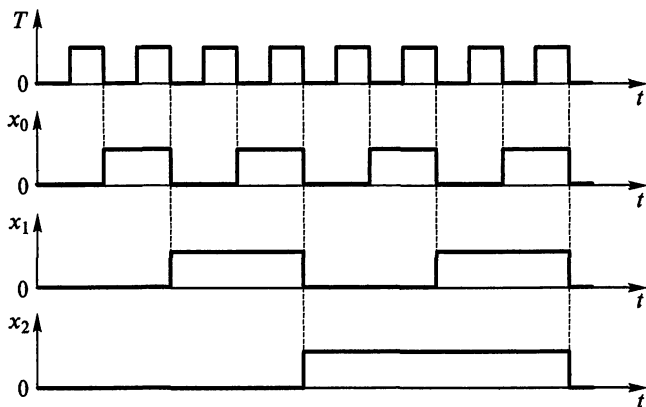


Рис. 7.16. Диаграммы сигналов в счетчике импульсов

лов в таком счетчике приведены на рис. 7.16, а табл. 7.6 иллюстрирует состояние триггеров. В исходном положении все триггеры находятся в состоянии 0. После первого входного импульса триггер $T1$ переходит в состояние 1, после второго входного импульса в состояние 1 переходит триггер $T2$, а $T1$ возвращается в состояние 0 и т.д. Из табл. 7.6 видно, что по состоянию триггеров можно определить, сколько импульсов поступило на вход к данному моменту времени. После восьмого входного импульса все три триггера переходят в состояние 0 и счет повторяется. В общем случае емкость счетчика (т.е. коэффициент пересчета) равна 2^n , где n — число триггеров в счетчике. С помощью обратных связей можно получить коэффициент пересчета меньше указанного значения.

Таблица 7.6

Состояния триггеров счетчика импульсов

Число входных импульсов	Состояние триггера		
	$T3$	$T2$	$T1$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

7.6. Сумматоры

Сумматор представляет собой комбинационное цифровое устройство (КЦУ), предназначенное в основном для суммирования двоичных чисел. Кроме того, с помощью сумматора могут выполняться вычитание, умножение, деление, преобразование чисел в дополнительный код и некоторые другие операции. Обычно сумматор состоит только из логических элементов, а результат операции направляется затем для записи в регистр.

Классификация сумматоров может быть проведена по трем основным признакам:

1) числу входов (полусумматоры, одноразрядные и многоразрядные сумматоры). Многоразрядные сумматоры, в свою очередь, подразделяются на последовательные и параллельные; последние по способу организации межразрядных переносов подразделяются на сумматоры с последовательным и параллельным переносом и с групповой структурой;

2) способу тактирования (синхронные и асинхронные сумматоры);

3) системе счисления (двоичные, двоично-десятичные и др.).

Полусумматорами (рис. 7.17, 7.18) называют КЦУ с двумя входами (a , b) и двумя выходами, на одном из которых вырабатывается сигнал суммы (выход S), а на другом — сигнал переноса (выход P). Табл. 7.7 является таблицей истинности полусумматора.

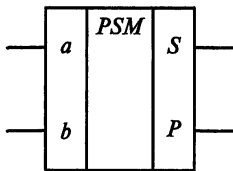


Рис. 7.17. Условное обозначение полусумматора

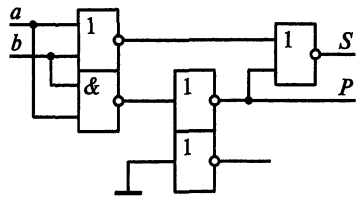


Рис. 7.18. Схема полусумматора на логических элементах

Таблица 7.7

Таблица истинности полусумматора

a	b	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Одноразрядным сумматором (рис. 7.19, 7.20) называют КЦУ с тремя входами и двумя выходами. Кроме двух входов для чисел он имеет третий вход, на который подается сигнал переноса из предыдущего разряда. Одноразрядный сумматор является основным элементом многоразрядных сумматоров. Он выполняет арифметическое сложение одноразрядных двоичных чисел a_i и b_i и перенос P_{i-1} из предыдущего разряда с образованием на выходе суммы S_i и переноса P_i в старший разряд (табл. 7.8).

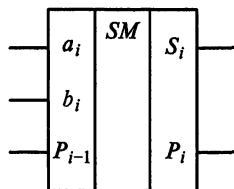


Рис. 7.19. Условное обозначение одноразрядного сумматора

Аналогичным способом могут быть построены логические схемы вычитателей. Как сумматоры, так и вычитатели предназначены для выполнения основных арифметических операций — сложения и вычитания. Имея на входе дополнительные средства для изменения знака второго аргумента, сумматор может прибавлять к первому слагаемому второе с измененным знаком, т.е. вычитать, а вычитатель — вычитать из уменьшаемого вычитаемое с измененным знаком, т.е. прибавлять. Таким образом, в арифметико-логических устройствах (АЛУ) в большинстве случаев используется только один из двух рассматриваемых узлов, традиционно — именно сумматор, хотя по всем показателям вычитатель подобен сумматору.

Операции сложения и вычитания бывают последовательными и параллельными. В данном случае под последовательностью понимается поочередное, разряд за разрядом, сложение (или вычитание) на одноразрядной схеме с задержкой переносов (или зай-

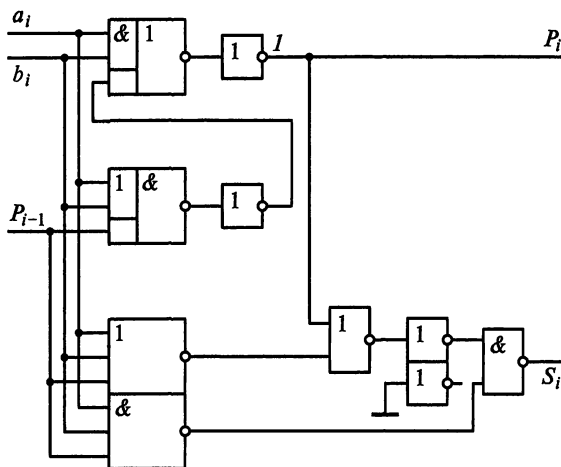


Рис. 7.20. Схема одноразрядного сумматора на логических элементах

Таблица истинности одноразрядного сумматора

a_i	b_i	P_{i-1}	S_i	P_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

мов) для использования их как третьих аргументов в следующем такте, т. е. в разряде.

При параллельных сложениях (или вычитаниях) используются столько одноразрядных сумматоров (или вычитателей), сколько разрядов в исходных числах (точнее — сколько разрядов в самом большом из них числе). Эти одноразрядные сумматоры взаимодействуют между собой по цепям переносов (или займов). Очевидно, что полный параллелизм при этом не достигается, так как переносы и займы распространяются с некоторой, хотя и небольшой, задержкой от младших разрядов к старшим. Имеется в виду схемное распространение займа в отличие от логического, направленного в противоположную сторону. Проблема сокращения времени распространения переносов (или займов) по разрядам — одна из главных при проектировании АЛУ.

Отметим некоторые особенности логики работы сумматоров и вычитателей:

- сумма равна 1, если единичные значения принимает нечетное число аргументов;
- выходной перенос равен 1, если единичные значения принимают больше двух аргументов;
- разность равна 1, если при отсутствии входного займа из 1 вычитается 0 или из 0 вычитается 1; она также равна 1, если аргументы равны при наличии входного займа;
- выходной заем равен 1, если из 0 вычитается 1, а также если аргументы равны при наличии входного займа.

В структуре АЛУ часто присутствует накопительный блок, состоящий из комбинационного сумматора (или вычитателя) и регистра результата. Подобный блок необходим при реализации по-

следовательного алгоритма вычислений, когда вновь поступающий аргумент прибавляется к ранее накопленному результату или из него вычитается, а новый результат вычислений заменяет исходный.

Функцию накопительного блока, сочетающего функции сумматора (или вычитателя) и регистра, может выполнять набор T -триггеров, работающих в режиме инверсии состояния, т. е. сложения по модулю 2. Такой сумматор на основе T -триггеров называется *накопительным*. Он уже является не комбинационным устройством, а конечным автоматом, поскольку обладает памятью.

Операции сложения (или вычитания) с учетом переноса (или займа) выполняются всегда над тремя аргументами, поэтому накопительный сумматор (или вычитатель) должен содержать управляющие коммутационные схемы, чтобы разнести сложение (или вычитание) на два такта. В этом состоит главный его недостаток. В остальном накопительный сумматор (или вычитатель) — самый простой и экономичный. Помимо несложных коммутационных схем он содержит дополнительно только цепи переноса (или займа).

Булевы функции, описывающие работу одноразрядного двоичного сумматора (по табл. 7.8), можно записать в следующем виде:

$$S_i = (\bar{a}_i \wedge \bar{b}_i \wedge P_i) \vee (\bar{a}_i \wedge b_i \wedge \bar{P}_i) \vee (a_i \wedge \bar{b}_i \wedge \bar{P}_i) \vee (a_i \wedge b_i \wedge P_i);$$

$$P_{i+1} = (\bar{a}_i \wedge b_i \wedge P_i) \vee (a_i \wedge \bar{b}_i \wedge P_i) \vee (a_i \wedge b_i \wedge \bar{P}_i) \vee (a_i \wedge b_i \wedge P_i).$$

Используя различные варианты преобразования этих функций, можно реализовать большое число структур одноразрядных двоичных сумматоров (например по схеме, показанной на рис. 7.21).

Для обработки многоразрядных чисел объединяется соответствующее число одноразрядных сумматоров. При этом отдельные разряды обрабатываемых чисел A и B подаются на входы a_i и b_i . На

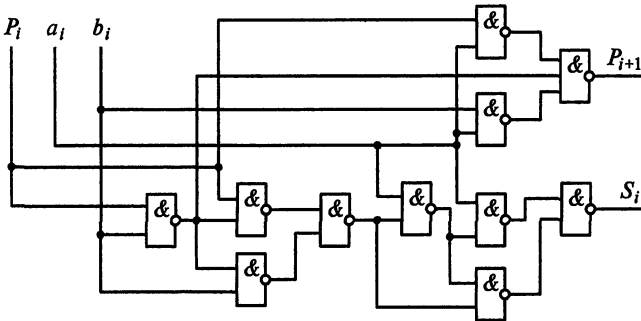


Рис. 7.21. Схема одноразрядного двоичного сумматора

вход P_i подается перенос из предыдущего, более младшего разряда. Формируемый в данном разряде перенос P_{i+1} передается в следующую, более старший разряд. Такая организация процесса формирования переноса, называемая последовательным переносом, снижает быстродействие многоразрядного сумматора, так как получение результата в старшем разряде сумматора обеспечивается только после завершения распространения переноса по всем разрядам. Поэтому иногда организуется параллельный перенос.

7.7. Кодировочные и декодирующие устройства

Шифраторы и дешифраторы. Основными видами преобразования информации являются шифрование и дешифрование (сжатие данных и обратное преобразование). Для реализации таких преобразований служат шифраторы и дешифраторы. Они представляют собой комбинационные устройства, поскольку состоят только из логических элементов, а элементов памяти не имеют. Шифраторы преобразуют код «1 из N » в двоичный код, а дешифраторы — двоичный код в код «1 из N ». Число разрядов двоичного кода обычно меньше N , поэтому операцию шифрования можно считать сжатием данных. Иными словами, шифратор превращает сигнал 1 на одном из нескольких входных зажимов в выходную кодовую комбинацию. Поэтому для шифратора используется также название «кодер», а на его условном обозначении пишут буквы *CD*. Дешифратор превращает комбинацию из нулей и единиц на входе в сигнал 1 только на одном единственном из нескольких выходных зажимов. Поэтому для дешифратора используется также название «декодер», а на его условном обозначении пишут *DC*.

Как и для других комбинационных устройств, для шифраторов и дешифраторов связь между входными и выходными сигналами можно задать с помощью логических функций или таблиц истинности. Для синхронизации выходных сигналов этих преобразователей тактовыми импульсами используют дополнительные входы. В этом случае преобразователи называют стробируемыми. *Стробирование* — это выделение сигнала в определенный момент времени. Наличие входов стробирования расширяет функциональные возможности шифраторов и дешифраторов.

Дешифраторы широко применяются в устройствах управления, вывода информации на цифровые индикаторы, в коммутаторах для распределения сигналов по различным цепям.

Различают полные и неполные дешифраторы. Число выходов у полного дешифратора $N_{\text{вых}} = 2^n$, а у неполного $N_{\text{вых}} < 2^n$, где n — число двоичных разрядов (число входов). На рис. 7.22, а приведено условное графическое обозначение полного стробируемого дешифратора «1 из 8», а табл. 7.9 представляет собой его таблицу истинности.

Существуют два способа стробирования дешифраторов: введением дополнительного входа в каждый элемент (стробирование по выходу — рис. 7.22, б) и блокированием всех элементов через одну из входных цепей (стробирование по входу — рис. 7.22, в).

С помощью дешифраторов в вычислительных устройствах могут быть реализованы различные логические функции, а также преобразование кода одного типа в код другого типа. В качестве примера приведена таблица истинности преобразователя двоично-десятичного кода в код «3 из 5» (табл. 7.10). Такой преобразователь имеет четыре входа (а при необходимости стробирования — пять) и пять выходов (v, w, x, y, z). В любой выходной кодовой комбинации (т.е. при любом входном сигнале) всегда будут три единицы и два нуля. Данный код способен выявить одиночную ошибку при передаче. Если из-за какой-то случайной причины (помехи) устройство, принимающее такой код, выявит наличие большего или меньшего числа

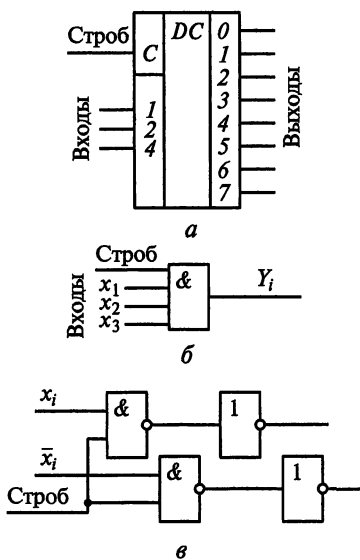


Рис. 7.22. Дешифратор «1 из 8»: а — условное обозначение; б — стробирование по выходу; в — стробирование по входу

Таблица 7.9

Таблица истинности дешифратора «1 из 8»

Входная кодовая комбинация				Сигнал на выходе							
C	x_1	x_2	x_3	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1
0	Любые комбинации нулей и единиц			0	0	0	0	0	0	0	0

единиц, то станет ясно, что вкралась ошибка. Существуют специальные коды, не только выявляющие ошибку, но и определяющие разряд, в котором она имеется.

Шифратор — это комбинационное устройство, преобразующее код «1 из N » в двоичный код. Полный шифратор имеет 2^n входов и n выходов. Одно из основных применений шифратора — ввод данных с клавиатуры, при котором нажатие на клавишу с десятичной цифрой должно приводить к передаче в устройство этой цифры в двоичном коде. При нажатии любой из десяти цифровых клавиш единица появляется только на одном из десяти входов шифратора X_0, X_1, \dots, X_9 . На выходе шифратора должен появиться двоичный код $(y_0 y_1 y_2 y_3)$ введенного десятичного числа. Из таблицы истинности (табл. 7.11) видно, что в этом случае нужен преобразователь с десятью входами и четырьмя выходами, т.е. так называемый шифратор «10—4».

На выходе y_0 единица должна появиться при нажатии любой нечетной клавиши (X_1, X_3, X_5, X_7, X_9), т.е. $y_0 = X_1 \vee X_3 \vee X_5 \vee X_7 \vee X_9$. Состояние остальных выходов определяется следующими логическими функциями:

$$y_1 = X_2 \vee X_3 \vee X_6 \vee X_7; \quad y_2 = X_4 \vee X_5 \vee X_6 \vee X_7; \quad y_3 = X_8 \vee X_9.$$

Следовательно, для реализации указанного шифратора понадобятся четыре логических элемента ИЛИ: пятиходовый, два че-

Таблица 7.10

Таблица истинности преобразователя двоично-десятичного кода в код «3 из 5»

Десятичное число	Входная кодовая комбинация				Выход	Выходная кодовая комбинация				
	x_1	x_2	x_3	x_4		v	w	x	y	z
0	0	0	0	0	Y_0	1	0	0	1	1
1	0	0	0	1	Y_1	1	0	1	0	1
2	0	0	1	0	Y_2	1	1	0	0	1
3	0	0	1	1	Y_3	0	0	1	1	1
4	0	1	0	0	Y_4	0	1	0	1	1
5	0	1	0	1	Y_5	0	1	1	0	1
6	0	1	1	0	Y_6	0	1	1	1	0
7	0	1	1	1	Y_7	1	0	1	1	0
8	1	0	0	0	Y_8	1	1	0	1	0
9	1	0	0	1	Y_9	1	1	1	0	0

Таблица истинности шифратора «10 — 4»

Десятичное число	Сигнал на входе									Выходная кодовая комбинация				
	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	Y_3	Y_2	Y_1	Y_0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	0	0	0	1	0	1
6	0	0	0	0	0	0	1	0	0	0	0	1	1	0
7	0	0	0	0	0	0	0	1	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	1	1	0	0	1

тырехходовых и двухходовых. Условное обозначение такого шифратора показано на рис. 7.23.

Аналого-цифровые преобразователи. Аналого-цифровой преобразователь (АЦП) предназначается для автоматического преобразования (измерения и кодирования) непрерывно изменяющихся во времени (т.е. аналоговых) величин в соответствующие значения числовых кодов. В данном случае под словом «цифра» понимается двоичный код. Когда говорят о цифровой звукозаписывающей и воспроизводящей аппаратуре или о цифровой телефонии, то подразумевают, что непрерывно изменяющийся звуковой сигнал записывается или передается оцифрованным, т.е. в виде двоичных (бинарных) кодов.

В зависимости от способа преобразования АЦП подразделяются на последовательные, параллельные и последовательно-параллельные. На рис. 7.24 показана схема АЦП последовательного типа.

По команде «Пуск» цифровой автомат ЦА вырабатывает последовательность двоичных чисел, которые поступают на вход цифроаналогового преобразователя ЦАП, вырабатывающего напряжение $U_{ЦАП}$, соответствующее каждому входному двоичному сигна-

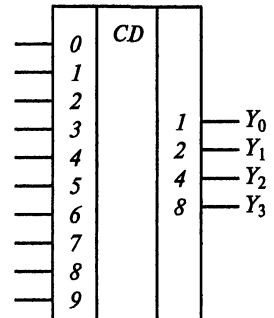


Рис. 7.23. Условное обозначение шифратора

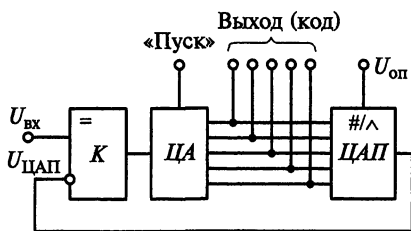


Рис. 7.24. Схема аналого-цифрового преобразователя последовательного типа

лу. Это напряжение (непрерывно растущее, пока работает ЦА) подается на один из входов компаратора K , на другой вход которого поступает входное напряжение $U_{вх}$. Компаратор сравнивает эти два напряжения и выдает сигнал при их равенстве. По этому сигналу ЦА останавливается, а на его выходе фиксируется двоичный код, соответствующий $U_{вх}$. Таким образом, преобразование в последовательном

АЦП происходит в ступенчатом режиме. Выходное значение отдельными шагами (тактами), т.е. последовательно, приближается к измеряемому значению. Поэтому последовательные АЦП на каждое преобразование аналогового сигнала затрачивают много времени. Для повышения их быстродействия используется метод поразрядного уравнивания. Иллюстрирующая этот метод схема показана на рис. 7.25.

Роль цифрового автомата выполняет регистр P_2 с датчиком тактовых импульсов ДТИ. Считывание выходного кода происходит по сигналу схемы готовности данных СГД, который подается при поступлении сигнала от компаратора K о равенстве входного напряжения $U_{вх}$ и напряжения $U_{ЦАП}$. Работа компаратора синхронизирована импульсами ДТИ. Эти же импульсы последовательно переводят разряды регистра P_2 в состояние 1. Перевод начинается со старшего разряда, а младшие остаются в состоянии 0. При этом ЦАП вырабатывает соответствующее напряжение, которое срав-

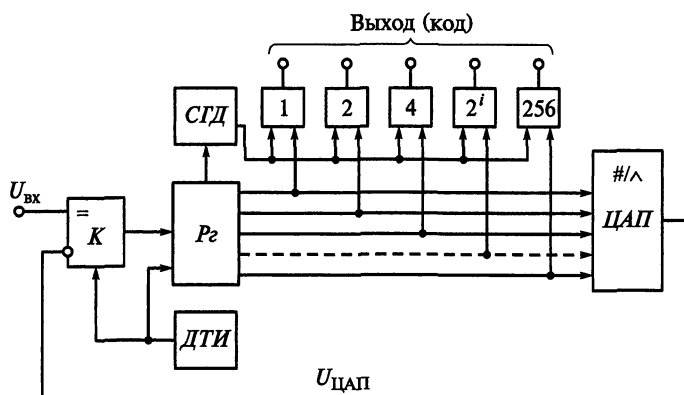


Рис. 7.25. Схема аналого-цифрового преобразователя с поразрядным уравниванием

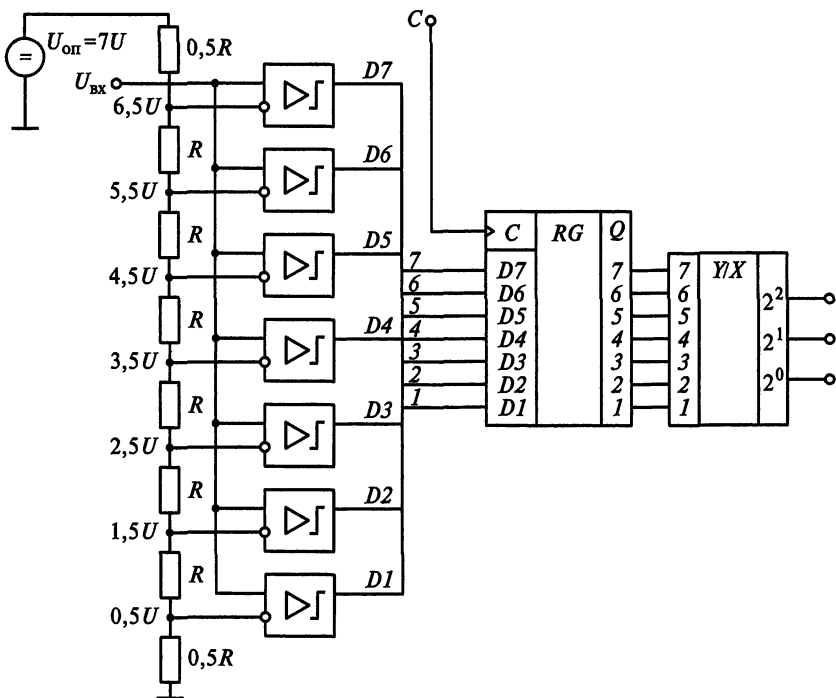


Рис. 7.26. Схема параллельного трехразрядного АЦП

нивается в компараторе K с входным. Если $U_{ЦАП} > U_{вх}$, то по команде компаратора старший разряд сбрасывается в состояние 0; если $U_{ЦАП} < U_{вх}$, то в старшем разряде остается 1. Затем в состоянии 1 переводится следующий по старшинству разряд P_2 и снова производится сравнение напряжений $U_{ЦАП}$ и $U_{вх}$. Цикл повторяется до тех пор, пока не будет зафиксировано равенство указанных напряжений при переводе в состояние 1 какого-то из младших разрядов. После этого $СЦД$ подает сигнал о выдаче выходного кода. Число циклов сравнения в таком АЦП будет равно числу разрядов выходного кода.

Наиболее быстродействующими являются АЦП параллельного типа. Преобразование аналогового сигнала в код в них осуществляется за один шаг, но такие АЦП требуют нескольких компараторов. Входное напряжение одновременно сравнивается во всех компараторах с несколькими опорными напряжениями. Параллельные АЦП имеют большее число элементов, чем последовательные.

Рассмотрим работу параллельного трехразрядного АЦП (рис. 7.26). Тремя двоичными разрядами можно представить восемь чисел — от 0 до 7. Поэтому используются семь компараторов для

сравнения входного напряжения с опорными напряжениями, получаемыми с помощью резисторного делителя. От каждого компаратора поступает сигнал 0, если входное напряжение меньше опорного, и 1 — в противном случае.

Состояния компараторов и соответствующие им двоичные коды представлены в табл. 7.12. Преобразователь кода выдает двоичное трехразрядное число. Время преобразования параллельных АЦП может составлять несколько десятков наносекунд, что в сотни раз быстрее, чем у последовательных АЦП.

Аналого-цифровые преобразователи используются для преобразования звуковых аналоговых сигналов в цифровой телефонии, цифровой записи звукового сопровождения в компьютерных играх, записи речи и музыки в цифровом формате и последующего воспроизведения аудио- и видеoinформации не только на соответствующих цифровых проигрывателях, но и с помощью компьютеров.

Стандартный цифровой канал с пропускной способностью 64 Кбит/с разработан и уже широко используется прежде всего для передачи в реальном масштабе времени речи, т. е. аналоговых сигналов в полосе частот 0,3...3,4 кГц. Чтобы указанные сигналы преобразовать в цифровой поток со скоростью передачи информации по каналу связи 64 Кбит/с, осуществляют три операции: дискретизацию, квантование и кодирование. Такие же преобразования используются и для компьютерной телефонии, поэтому рассмотрим эти операции подробнее. Будем использовать при этом рис. 4.1, кратко описанный в подразд. 4.2.

Таблица 7.12

Зависимость цифрового кода от входного напряжения

Относительное значение входного напряжения $U = U_{вх}/U_{оп}$	Состояние компараторов							Двоичный код-число		
	7	6	5	4	3	2	1	4	2	1
$U < 0,5$	0	0	0	0	0	0	0	0	0	0
$0,5 \leq U < 1,5$	0	0	0	0	0	0	1	0	0	1
$1,5 \leq U < 2,5$	0	0	0	0	0	1	1	0	1	0
$2,5 \leq U < 3,5$	0	0	0	0	1	1	1	0	1	1
$3,5 \leq U < 4,5$	0	0	0	1	1	1	1	1	0	0
$4,5 \leq U < 5,5$	0	0	1	1	1	1	1	1	0	1
$5,5 \leq U < 6,5$	0	1	1	1	1	1	1	1	1	0
$6,5 \leq U$	1	1	1	1	1	1	1	1	1	1

На основании теоремы Котельникова частота дискретизации f_d должна быть не менее чем в 2 раза больше верхней преобразуемой частоты f_b , т.е. $f_d \geq 2f_b$. Для телефонии $f_b = 3,4$ кГц. В цифровых системах передачи информации принята $f_d = 8$ кГц. Следовательно, период дискретизации $T_d = 1/f_d = 125$ мкс.

После дискретизации сигнал имеет форму отдельных выборок (см. рис. 4.1, б), называемых амплитудно-импульсно-модулированным сигналом (АИМ-сигналом).

Сущность квантования состоит в замене произвольных значений u (некоторые из тех, что принимает аналоговый сигнал, показанный на рис. 4.1, а) разрешенными значениями. Если всех возможных значений аналогового сигнала существует бесконечное множество, то число разрешенных значений сравнительно мало, т.е. они образуют конечное множество. Процесс квантования поясняется на рис. 4.1, в. При квантовании весь диапазон изменения u разбивается на некоторое число примыкающих друг к другу областей, границами которых являются уровни квантования P . Разность между двумя соседними уровнями называется *шагом квантования* $\Delta u_{\text{кв}}$. Если шаг квантования постоянен, то квантование называется *равномерным*. После квантования амплитуды импульсов уже не произвольны, а соответствуют ближайшим из разрешенных уровней. Можно сказать, что с математической точки зрения квантование эквивалентно округлению чисел до ближайшего целого. Такое округление всегда связано с появлением погрешности. Разность между квантованными и неквантованными АИМ-сигналами называется *ошибкой*, или *погрешностью квантования* $\delta_{\text{кв}}$. Очевидно, что максимальное значение $\delta_{\text{кв}}$ не превосходит $\Delta u_{\text{кв}}/2$. С уменьшением $\Delta u_{\text{кв}}$ ошибка квантования уменьшается. После передачи таких сигналов по телефонной сети и восстановления их (т.е. превращения обратно в аналоговый сигнал) эта ошибка воспринимается на слух как шум. Поэтому помеху, вызванную квантованием, часто называют *шумом квантования*.

При кодировании каждому из разрешенных уровней присваивается свое кодовое обозначение (кодовая комбинация). Нумерация уровней обычно осуществляется в двоичной системе счисления, использование которой позволяет упростить техническую реализацию кодирующих устройств. Порядок нумерации может быть произвольным, важно только, чтобы число уровней квантования было не больше числа различных кодовых комбинаций. На практике чаще всего встречается система нумерации, основанная на симметричном двоичном коде. При этом символ в крайнем левом (старшем) разряде кодовой комбинации определяет полярность кодируемого импульса (1 соответствует положительной полярности, 0 — отрицательной), а символы в остальных разрядах — его абсолютное значение.

На рис. 4.1, *г* показаны цифровые сигналы, соответствующие исходному аналоговому (непрерывному) сигналу, изображенному на рис. 4.1, *а*. Число уровней квантования этого сигнала было принято равным семи с учетом нулевого уровня (см. рис. 4.1, *в*). Для их нумерации достаточно трехразрядных двоичных чисел. Различных кодовых комбинаций, содержащих три символа, существует только восемь: 000, 001, 010, 011, 100, 101, 110, 111. Кодовая комбинация 000 не используется, поскольку отсутствие импульсов можно трактовать и как информацию о соответствующем уровне, и как кратковременный разрыв цепи передачи информации.

Кодовые комбинации, состоящие из трех разрядов, передаются с интервалом T . Каждый разряд представлен очень коротким импульсом длительностью $\tau_{и}$ (см. рис. 4.1, *г*). Величина импульса (т.е. напряжение) превышает некоторый обусловленный порог в том случае, если этот импульс соответствует значению 1. Обратите внимание, что третий отсчет взят в тот момент времени, когда значение исходного аналогового сигнала равнялось нулю (см. рис. 4.1, *а* и *б*). Такому значению соответствует уровень $P = 0$, кодовая комбинация которого имеет вид 100 (см. рис. 4.1, *в*). Таким образом, даже в случае нулевого значения исходного аналогового сигнала по каналу связи передается сигнал, который невозможно истолковать как неисправность цепи.

Число различных n -разрядных комбинаций $M = 2^n$. Так как каждой кодовой комбинации ставится в соответствие один уровень квантования, то это выражение определяет максимальное число уровней квантования, которое может быть получено при использовании n -разрядных кодовых комбинаций.

При $n = 7$ число различных семиразрядных комбинаций равно 128, ошибка квантования $\delta_{кв} \leq 1/256$, т.е. менее 0,4%. Для передачи речи средствами цифровой телефонии такая ошибка совсем незначительна.

В цифровой телефонии достоинства цифрового сигнала проявились особенно ярко. Сам метод преобразования аналогового сигнала получил название *импульсно-кодовая модуляция* (ИКМ). Аппаратура ИКМ позволяет по одной паре проводов передавать практически одновременно сотни и тысячи телефонных разговоров. При этом необходимо решать проблемы управления таким процессом передачи информации. Каждую семиразрядную комбинацию сопровождает восьмой импульс, используемый для управления. Таким образом, скорость передачи сообщений в стандартном цифровом канале $V = f_{д}n = 8 \cdot 8 = 64$ Кбит/с.

Для повышения точности аналого-цифрового преобразования телефонных сигналов используется неравномерное квантование.

В общем случае точность аналого-цифрового преобразования тем выше, чем меньше размер шагов квантования, поскольку рас-

стояние между соседними уровнями и есть размер шага квантования. На верхнем графике рис. 7.27 показан групповой АИМ-сигнал, по вертикали отмечены восемь уровней квантования P (с нулевого по седьмой). Расстояния между уровнями одинаковые (равномерное квантование) и равны шагу квантования $\Delta u_{\text{кв}}$. Напряжение седьмого уровня соответствует напряжению ограничения $U_{\text{огр}}$. Амплитуда четвертого импульса принимается равной нулю, поскольку исходный сигнал в этот момент меньше половины напряжения первого уровня. Очевидно, что при квантовании сигнала возникают ошибки $\delta_{\text{кв}}$ (затемненные участки на рис. 7.27), величина которых случайна, имеет равномерное распределение и, как уже отмечалось, не превышает половины шага квантования. Сигнал после квантования представляет собой сумму исходного сигнала и сигнала ошибки (сигналы ошибок показаны на нижнем графике).

Для исходных сигналов малой величины относительная величина ошибки очень велика. Например, исходный сигнал, равный половине напряжения первого уровня, после квантования будет передаваться как сигнал 1-го уровня, в результате чего относительная ошибка составит 50%. Поэтому при аналого-цифровом преобразовании телефонного сигнала применяется неравномерное квантование. При таком квантовании уменьшают величины шагов для малых мгновенных значений сигнала за счет увеличения шагов для больших значений сигнала, как это показано на рис. 7.28.

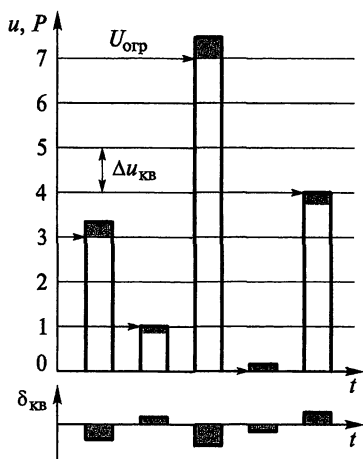


Рис. 7.27. Возникновение ошибок при равномерном квантовании

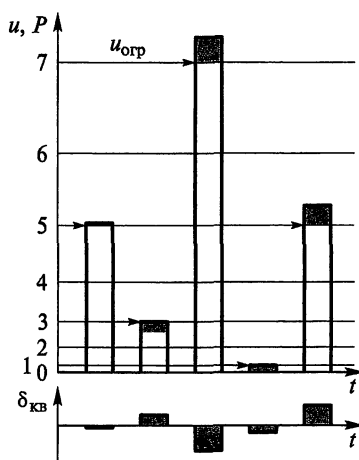


Рис. 7.28. Возникновение ошибок при неравномерном квантовании

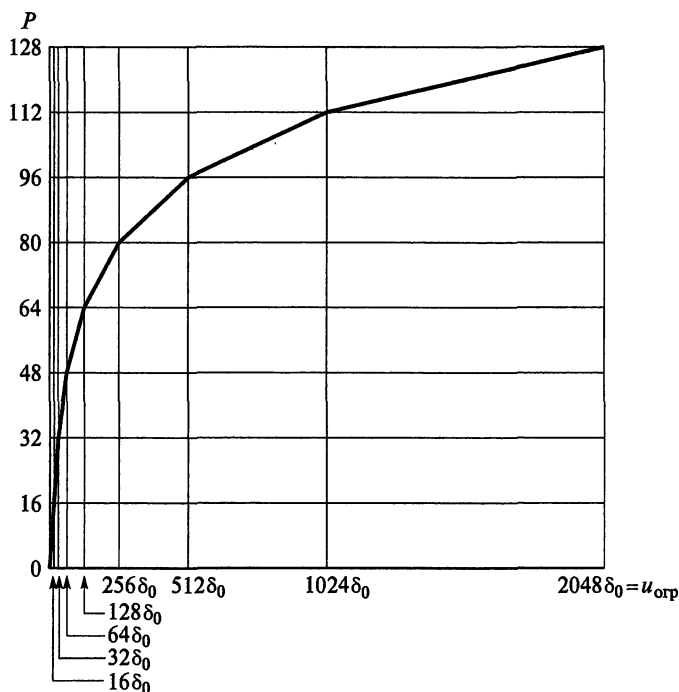


Рис. 7.29. Амплитудная характеристика неравномерного квантования

В цифровых системах передачи (ЦСП) применяются сегментные неравномерные характеристики квантования, поскольку они достаточно просто реализуются на цифровой основе. На рис. 7.29 представлена характеристика типа А87,6/13 (положительная ветвь), применяемая в АЦП ЦСП европейской плезиохронной цифровой иерархии. Характеристика симметрична относительно нуля, положительная и отрицательная ее ветви состоят каждая из восьми сегментов, каждый сегмент поделен на 16 одинаковых шагов (внутри сегмента квантование равномерное). В нулевом и первом сегментах шаг минимален и равен $\Delta u_{\text{кв}}$, а в каждом следующем сегменте величина шага удваивается по отношению к предыдущему. Коды шагов внутри любого сегмента одинаковы и соответствуют натуральному несимметричному коду (1-й шаг имеет код 0000, 2-й — 0001, ..., 16-й — 1111). Согласно этой характеристике 8-разрядное кодовое слово, соответствующее мгновенному значению сигнала, имеет структуру $PXYZABCD$, где P — старший разряд, указывающий на полярность сигнала (1 — положительная, 0 — отрицательная), XYZ — код сегмента, $ABCD$ — код шага внутри сегмента. Кодовое слово подвергается еще одной операции — инверсии четных разрядов — с тем, чтобы исключить пакеты нулевых символов, существенно усложняющих выделение из кодо-

вой комбинации сигнала тактовой частоты, необходимого для синхронизации генераторного оборудования приемной части аппаратуры. Так, вместо весьма часто встречающейся комбинации 00000000 (в телефонии она соответствует молчанию собеседников) будет передана комбинация 01010101, вероятность появления которой в исходном сигнале относительно невелика. На приемном конце инверсия четных разрядов снимается.

Верхняя граница верхнего сегмента соответствует максимальному напряжению сигнала — напряжению ограничения $u_{огр}$. Минимальный шаг квантования принят равным $\delta_0 = u_{огр}/2048$, а шаг квантования в сегментах $2 - 7\delta_i = 2^{i-1}\delta_0$, где i — номер сегмента.

В цифровой телефонии используется понятие отношение «сигнал — шум квантования» (ОСШК). Величину ОСШК измеряют в децибелах (дБ). В системах ИКМ с равномерным квантованием кодовое пространство используется весьма неэффективно: для малого сигнала ОСШК составляет 26 дБ, а для сигнала с максимальной амплитудой — 56 дБ.

При неравномерном квантовании ОСШК одинаково для всех уровней сигналов. При неодинаковых размерах шагов квантования между кодовыми комбинациями и соответствующими им значениями амплитуд импульсов существует нелинейное соотношение.

Нелинейное преобразование аналогового сигнала в цифровой при неравномерном квантовании называется *компрессированием*, а обратное нелинейное преобразование на приемной стороне, при котором аналоговый сигнал восстанавливается из цифрового, называется *экспандированием*. Процесс, при котором сначала осуществляется компрессирование, а затем экспандирование сигнала, носит название *командирования*.

По мере распространения компьютеров возникла необходимость быстрой передачи информации между ними. Сложнее всего оказалось связать компьютеры, расположенные на большом расстоянии друг от друга. Прокладка специальных линий связи представлялась очень дорогой. В то же время уже существовала телефонная сеть, охватывающая весь земной шар. Однако телефонные каналы позволяют передавать только аналоговый сигнал, компьютер же работает только с дискретным (цифровым) сигналом.

Поэтому были разработаны специальные устройства, предназначенные для преобразования дискретных (цифровых) сигналов в аналоговую форму, передачи полученного сигнала по телефонной линии и приема аналогового сигнала из телефонной линии с последующим преобразованием его в цифровую форму для подачи в компьютер (рис. 7.30). Такое устройство получило название «модем» (см. подразд. 3.2). На рис. 7.30 показана общая схема соединения компьютеров при помощи модема.

Цифроаналоговые преобразователи. Цифроаналоговый преобразователь (ЦАП) предназначается для автоматического преобразо-

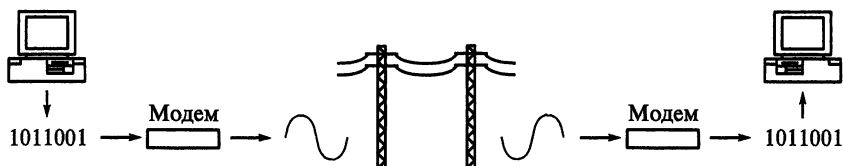


Рис. 7.30. Схема соединения компьютеров через модемы

вания (декодирования) входных величин, представленных числовыми кодами, в соответствующие им значения непрерывно изменяющихся во времени (т. е. аналоговых) величин. Иными словами, ЦАП выполняет обратное по сравнению с АЦП преобразование. Выходные физические величины АЦП чаще всего представляют собой электрические напряжения и токи, но могут быть также временными интервалами, угловыми перемещениями и т. п. В системе автоматики с ЭВМ удобнее обрабатывать (преобразовывать и передавать) цифровой сигнал, но человеку (оператору) привычнее и удобнее воспринимать аналоговые сигналы, соответствующие значениям числовых кодов. С помощью АЦП информация вводится в ЭВМ, а с помощью ЦАП она выводится из ЭВМ для воздействия на управляемый объект и восприятия человеком.

В схемах ЦАП обычно используется представление двоичного числа, состоящего из нескольких разрядов, в виде суммы степеней числа 2. Каждый разряд (если в нем записана единица) преобразуется в аналоговый сигнал, пропорциональный числу 2 в степени, равной номеру разряда, уменьшенному на единицу.

На рис. 7.31 показана простая схема ЦАП, основу которой составляет резистивная матрица — набор резисторов, которые подключаются ко входу операционного усилителя ключами, управляемыми соответствующими разрядами двоичного числа. В качестве ключей могут быть использованы триоды (например МДП-транзисторы). Если в данном разряде записана 1, то ключ замкнут, если 0 — разомкнут.

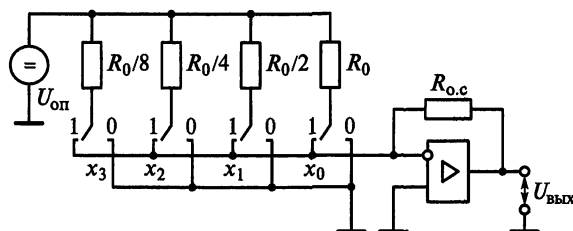


Рис. 7.31. Схема цифроаналогового преобразователя на базе резистивной матрицы

Операционный усилитель — это аналоговое устройство (в отличие от всех других рассматриваемых в данной главе устройств, которые являются цифровыми, или дискретными). Необходимость в нем обусловлена тем, что в ЦАП выходной сигнал является аналоговым. И входной, и выходной сигналы операционного усилителя представляют собой напряжения постоянного (в смысле неизменной полярности) тока. Отношение выходного напряжения к входному называется *коэффициентом передачи*. Операционный усилитель является основным элементом для построения аналоговых вычислительных машин. Подсоединяя к нему определенным образом резисторы и конденсаторы, можно обеспечить различные функциональные зависимости выходного сигнала от входного. При использовании операционного усилителя в схеме ЦАП необходимо, чтобы такая зависимость была строго пропорциональной.

Коэффициент передачи операционного усилителя равен отношению сопротивления резистора $R_{o.c}$ в цепи обратной связи к сопротивлению резистора на входе усилителя, которое, как видно из рис. 7.31, для каждого разряда имеет свое значение. Коэффициенты передачи $K = -U_{\text{вых}}/U_{\text{оп}}$ по каждому разряду преобразуемого двоичного числа (если в этом разряде записана 1) соответственно равны:

$$K_0 = R_{o.c}/R_0; K_1 = 2R_{o.c}/R_0; K_2 = 4R_{o.c}/R_0; K_3 = 8R_{o.c}/R_0.$$

Выходное напряжение ЦАП

$$U_{\text{вых}} = -U_{\text{оп}}(K_3 + K_2 + K_1 + K_0) = -U_{\text{оп}}(R_{o.c}/R_0) (8x_3 + 4x_2 + 2x_1 + x_0),$$

где x принимает значение 1 или 0 в зависимости от того, что записано в данном разряде двоичного числа.

Таким образом, четырехразрядное двоичное число преобразуется в напряжение $U_{\text{вых}}$, которое может принимать 16 возможных значений от 0 до $15\Delta u_{\text{кв}}$, где $\Delta u_{\text{кв}}$ — шаг квантования.

Для уменьшения погрешности квантования необходимо увеличивать число двоичных разрядов ЦАП. При изготовлении ин-

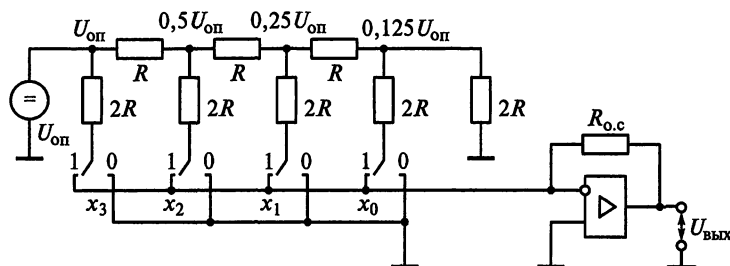


Рис. 7.32. Схема цифроаналогового преобразователя на базе резистивной матрицы $R-2R$

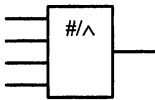


Рис. 7.33. Условное обозначение ЦАП

тегральных микросхем ЦАП по данной схеме очень трудно сделать высокоточные резисторы с сопротивлениями, отличающимися друг от друга в десятки и сотни раз. Кроме того, нагрузка источника опорного напряжения $U_{\text{оп}}$ изменяется в зависимости от состояния ключей, поэтому необходимо применять источник с малым внутренним сопротивлением.

Схема ЦАП, показанная на рис. 7.32, свободна от указанных недостатков. В ней весовые коэффициенты каждого разряда задаются последовательным делением опорного напряжения с помощью резистивной матрицы типа $R-2R$, представляющей собой многоразрядный делитель напряжения. В данной схеме ЦАП используются двухпозиционные ключи, которые подсоединяют резисторы $2R$ либо ко входу операционного усилителя (при 1 в данном разряде), либо к общему нулевому проводу. Входное сопротивление резистивной матрицы при этом не зависит от положения ключей. Коэффициент передачи между соседними узловыми точками матрицы составляет 0,5. Выходное напряжение

$$U_{\text{вых}} = -U_{\text{оп}} (R/16R)(x_1 + 2x_2 + 4x_3 + 8x_4).$$

Наибольшее влияние на погрешность ЦАП оказывают отклонения сопротивлений резисторов от их номинальных значений, а также то, что у реального ключа сопротивление в закрытом состоянии не равно бесконечности, а в открытом — не равно нулю. Выпускаемые резистивные матрицы имеют относительную погрешность около сотых долей процента, т. е. являются очень точными.

Условное обозначение ЦАП показано на рис. 7.33.

7.8. Компараторы

Для сравнения двух сигналов используются компараторы (от *англ.* compare — сравнить). Они позволяют дать ответы на вопросы, равны или не равны два сигнала; если не равны, то какой из этих сигналов больше. Для сравнения двух двоичных чисел нужен цифровой компаратор. В табл. 7.13 указана связь между сигналами на выходах и входах компаратора при сравнении одноразрядных чисел a и b , которые могут быть равны 1 или 0. На соответствующем выходе появляется логическая 1, когда в указанном соотношении находятся сигналы на входах. Так, если $a = 1$, $b = 1$ (числа одинаковы), то функция, характеризующая равенство чисел, $F_{a=b} = 1$, а функции, характеризующие их неравенство, $F_{a>b} = 0$ и $F_{a<b} = 0$. Аналогично заполняются другие строки таблицы.

Входные и выходные сигналы цифрового компаратора

Входы		Выходы		
a	b	$F_{a>b}$	$F_{a=b}$	$F_{a<b}$
1	1	0	1	0
1	0	1	0	0
0	1	0	0	1
0	0	0	1	0

По известным правилам на основании табл. 7.13 можно записать следующие логические функции, характеризующие соотношение одноразрядных чисел:

$$F_{a>b} = a \wedge \bar{b}; \quad F_{a=b} = (a \wedge b) \vee (\bar{a} \wedge \bar{b}); \quad F_{a<b} = \bar{a} \wedge b.$$

Если значения a и b таковы, что правая часть функции равна 1, то соотношение, указанное в левой части, выполняется. Если правая часть функции равна 0, то соотношение между a и b противоположно указанному.

Схема одноразрядного компаратора, реализующая приведенные функции, изображена на рис. 7.34.

Остановимся подробнее на функции равенства чисел $F_{a=b}$, или функции «Равнозначность». По смыслу она противоположна функции $F_{a \neq b}$ «Неравнозначность» (исключающее ИЛИ). Поскольку $F_{a \neq b} = a_1 \bar{b}_1 + \bar{a}_1 b_1 = a_1 \oplus b_1$, то $F_{a=b} = \bar{F}_{a \neq b} = \overline{a_1 \bar{b}_1 + \bar{a}_1 b_1} = \overline{a_1 \oplus b_1}$.

Следовательно, проверку равенства пары одноименных разрядов двух чисел можно осуществить, используя логический элемент «Равнозначность» (рис. 7.35, а) или «Неравнозначность», дополненный инвертором (рис. 7.35, б).

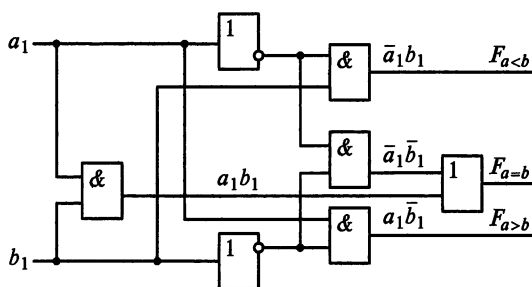


Рис. 7.34. Схема цифрового одноразрядного компаратора

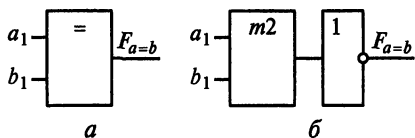


Рис. 7.35. Проверка равенства разрядов:

a — на логическом элементе «Равнозначность»; b — на логическом элементе «Неравнозначность»

Числа A и B равны, если их одноименные разряды содержат одинаковые цифры ($a_0 = b_0$ И $a_1 = b_1$ И ... И $a_{n-1} = b_{n-1}$), т.е. функция, характеризующая соотношение чисел, должна быть конъюнкцией функций, характеризующих соотношение цифр в их одноименных разрядах:

$$F_{A=B} = F_{a_0=b_0} \wedge F_{a_1=b_1} \wedge \dots \wedge F_{a_{n-1}=b_{n-1}}.$$

Когда цифры в одноименных разрядах чисел A и B одинаковы, на выходах всех элементов «Равнозначность» (рис. 7.36, a) логические 1 и $F_{A=B} = 1$. Если хотя бы в одной паре разрядов находятся разные цифры, то на выходе соответствующего элемента «Равнозначность» будет логический 0 и $F_{A=B} = 0$, что указывает на неравенство чисел A и B .

Если цифры в одноименных разрядах чисел A и B одинаковые, то на выходах всех элементов «Неравнозначность» (рис. 7.36, b) логические 0, поэтому на выходе дизъюнктора — 0, а на выходе инвертора — 1, т.е. $F_{A=B} = 1$.

На рис. 7.37 приведены условные обозначения компараторов и показан способ наращивания их разрядности. Каждый из изображенных компараторов предназначен для сравнения двух четырехразрядных чисел и имеет выходы $A > B$, $A = B$ и $A < B$. Аналогичные входы служат для наращивания разрядности компараторов. Результат сравнения младших разрядов отражается на выходах компаратора $K1$: на одном из них появляется 1, на других — 0. Компаратор $K2$ воспринимает этот результат, с учетом которого формируется

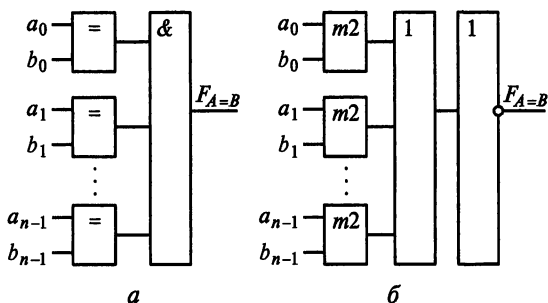


Рис. 7.36. Поразрядное сравнение:

a — на логических элементах «Равнозначность»; b — на логических элементах «Неравнозначность»

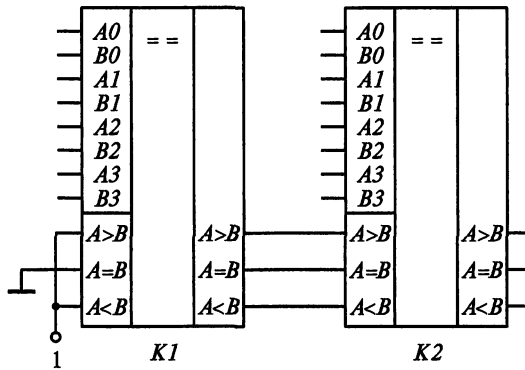


Рис. 7.37. Соединение цифровых компараторов для наращивания их разрядности

окончательный результат сравнения. Подобным образом можно осуществлять дальнейшее наращивание разрядности. Указанные потенциалы на входах компаратора *K1* младших разрядов обеспечивают правильное функционирование многокаскадного компаратора на данных микросхемах.

Для сравнения двух аналоговых сигналов по величине используется компаратор на базе операционного усилителя (ОУ) (рис. 7.38). В данной схеме (в качестве примера) выполняется сравнение двух разнополярных напряжений — постоянного $U_{оп}$ (его называют опорным напряжением) и плавно изменяющегося с постоянной скоростью U_x . Графики сравниваемых напряжений и выходного напряжения компаратора показаны на рис. 7.39. До момента времени t_1 напряжение U_x по абсолютной величине меньше $U_{оп}$ и выходное напряжение компаратора отрицательное (напомним, что ОУ инвертирует сигнал). В момент времени t_1 напряжение U_x достигает порогового значения $U_{вх.пор}$, равного по абсолютной величине $U_{оп}$, т.е. на вход ОУ поступает результирующий сигнал отрицательной полярности. Напряжение на выходе компаратора изменяется скачком, сигнализируя о том, что сравниваемые напряжения стали одинаковыми по абсолютной величине. Сопротивление резистора баланса (см. рис. 7.38) определяется по формуле $R_6 = R_1 R_2 / (R_1 + R_2)$.

В рассмотренной схеме используется операционный усилитель без обратной связи, следовательно, его коэффициент уси-

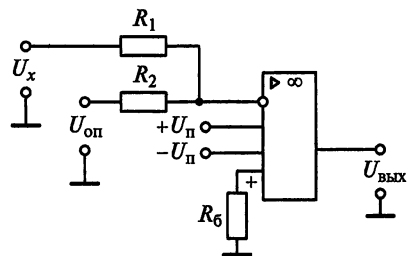


Рис. 7.38. Схема аналогового компаратора

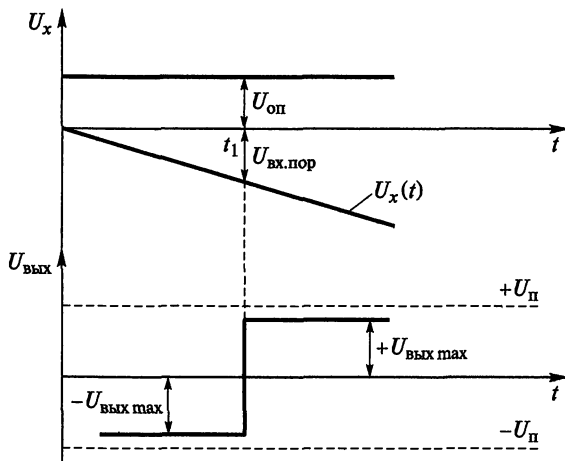


Рис. 7.39. Диаграмма напряжений аналогового компаратора

ления очень велик. Это способствует увеличению скорости переключения компаратора. Выходное напряжение компаратора $U_{\text{вых max}}$ близко к его напряжению питания $U_{\text{п}}$.

7.9. Типы коммутаторов. Мультиплексоры и демультиплексоры

Устройство, переключающее электрические цепи, называется коммутатором. В вычислительной технике применяются в основном коммутаторы двух типов: осуществляющие подключение с нескольких входов на один выход и, наоборот, с одного входа на несколько выходов.

Коммутатор типа «несколько входов — один выход» (рис. 7.40, а) дает возможность подключить канал Y к разным источникам информации ($D1, D2, D3$). Выбор присоединяемого источника (одного из информационных входов коммутатора) осуществляется сигналом на адресном входе. Информация будет поступать из того канала, на элемент И которого подается разрешение в виде логической 1 с одного из адресных входов $A1, A2, A3$. Рассмотренный коммутатор выполнен на одной микросхеме, содержащей в корпусе три элемента И, выходы которых соединены с входами элемента ИЛИ.

Аналогичную задачу решает мультиплексор — коммутатор, в котором выбор входа по его номеру (адресу) осуществляется двоичным кодом.

На рис. 7.40, б приведен коммутатор типа «один вход — несколько выходов», позволяющий подключать канал источника цифровой информации D к разным каналам на выходе ($Y1, Y2$,

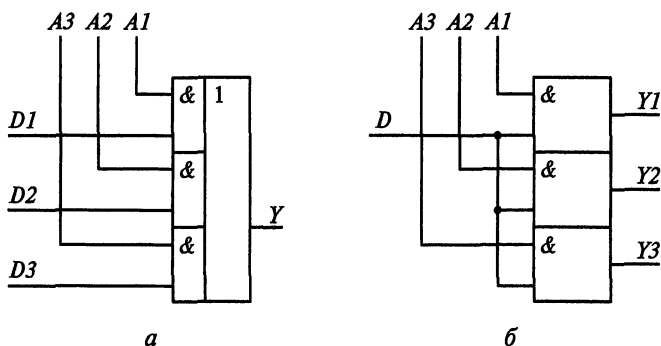


Рис. 7.40. Типы коммутаторов:

a — несколько входов — один выход; *б* — один вход — несколько выходов

УЗ). Выбор выходного канала осуществляется подачей логической 1 с одного из адресных входов $A1, A2, A3$, активизирующей соответствующий конъюнктор. Коммутатор выполнен на одной микросхеме, содержащей в одном корпусе несколько элементов И.

Аналогичную задачу решает демультиплексор. В отличие от коммутатора выбор выхода демультиплексора осуществляется кодом, подаваемым на все адресные входы.

Мультиплексор (рис. 7.41) в вычислительных устройствах используется, например, для записи в регистр кодов, поступающих из разных запоминающих устройств или устройств ввода. В цифровой телефонии он широко применяется для передачи множества телефонных разговоров по одному каналу связи, в системах автоматического управления — для подачи выходных сигналов от нескольких источников (например, однотипных датчиков) к одному приемному устройству (например, показывающему прибору). Подсоединение к источникам сигналов производится последовательно (в так называемых системах обходящего контроля) или адресно — по выбору оператора.

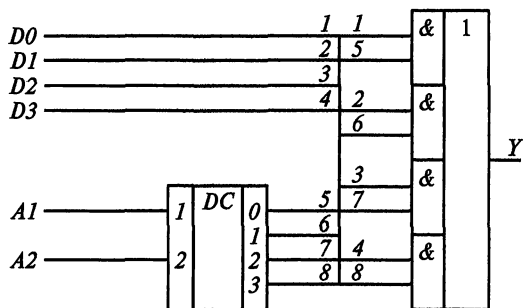


Рис. 7.41. Мультиплексор

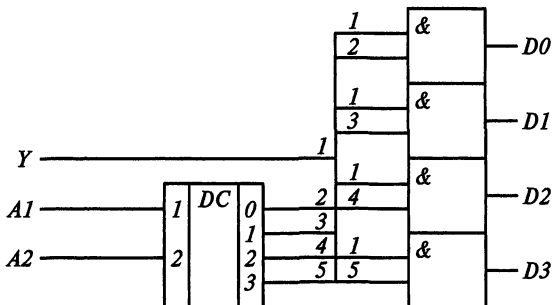


Рис. 7.42. Демультимплексор

Показанный на рис. 7.41 мультиплексор позволяет подключать к выходу Y один из четырех информационных входов $D0$, $D1$, $D2$, $D3$.

Выбор информационного входа осуществляется подачей на два адресных входа $A1$ и $A2$ соответствующего кода: 00, 01, 10, 11. Например, при подаче на адресные входы сигнала 10 (т.е. десятичная двойка) на выходе 2 дешифратора DC появляется 1, которая по входу 7 поступает на двухходовый логический элемент И. На другой вход этого элемента поступает информационный сигнал по каналу $D2$.

Значит, именно этот второй информационный вход будет подключен к выходу Y мультиплексора. С помощью мультиплексора может быть организована передача сигналов по одному и тому же каналу (выходу Y) от нескольких источников информации, но не одновременно.

При подаче на адресные входы $A1$ и $A2$ кода 00 информация передается от входа $D0$, при подаче кода 01 — от входа $D1$ и т.д. Следовательно, можно сказать, что выход Y представляет собой канал с временным разделением сигналов. Это особенно важно тогда, когда один и тот же физический канал связи используется для передачи большого числа разных сигналов, например при обмене информацией в сети Интернет или при автоматическом управлении космическими аппаратами.

После получения информации по такому единственному каналу связи Y ее необходимо разделить между соответствующими приемниками.

Эту задачу решает демультимплексор (рис. 7.42). Выбор соответствующего информационного выхода осуществляется с помощью адресного входа. Как и в схеме мультиплексора, используется дешифратор DC .

При подаче на адресные входы $A1$ и $A2$ сигнала 10 на выходе 2 появляется 1 и входной сигнал Y проходит на информационный выход $D2$.

7.10. Типовые элементы аналоговых вычислительных машин

Вычислительная машина, которая обрабатывает информацию, представленную в аналоговой (непрерывной) форме, называется *аналоговой вычислительной машиной*. С помощью АВМ воспроизводятся (моделируются) самые разные процессы: движение механических объектов, преобразование энергии разного вида (механической, тепловой, электрической); управление технологическими и производственными объектами. Несмотря на различную физическую природу все эти процессы описываются одними и теми же дифференциальными уравнениями, которым подчиняется и поведение АВМ. Поэтому АВМ часто называют еще и моделирующими установками, или аналоговыми моделями.

Выполнить интегрирование, дифференцирование, получить иные функциональные преобразования непрерывного сигнала можно с помощью схем, построенных на базе операционного усилителя с различными обратными связями.

Операционный усилитель используется и при аналоговом моделировании систем автоматики. Выпускаются ОУ в виде полупроводниковых интегральных микросхем, широкому применению которых способствует их низкая стоимость, близкая к стоимости отдельных транзисторов.

Операционный усилитель (рис. 7.43) представляет собой усилитель постоянного тока с большим коэффициентом усиления (до нескольких миллионов) по напряжению, большим входным (несколько мегаом) и малым выходным (не более нескольких сотен ом) сопротивлениями. Почти всегда он используется с внешней глубокой отрицательной обратной связью, которая и определяет его результирующие характеристики, т. е. ту математическую операцию, которую выполняет ОУ. Одна из основных схем включения ОУ показана на рис. 7.44. Она обеспечивает усиление входного сигнала (напряжения $U_{\text{вх}}$) с коэффициентом $K = -R_{\text{о.с.}}/R$, т. е. выходной сигнал

$$U_{\text{вых}} = KU_{\text{вх}} = -(R_{\text{о.с.}}/R)U_{\text{вх}}.$$

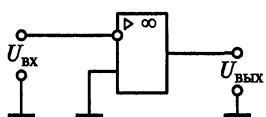


Рис. 7.43. Условное обозначение операционного усилителя

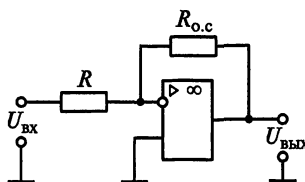


Рис. 7.44. Схема усиления входного сигнала

Знак «минус» в данном случае показывает, что фаза выходного напряжения отличается от фазы входного на 180° . Таким образом, ОУ изменяет полярность входного сигнала, т.е. инвертирует сигнал.

Изменяя соотношение между сопротивлениями резисторов в прямой цепи (R) и в цепи обратной связи ($R_{o.c}$), можно получить любой требуемый коэффициент усиления. Следовательно, рассмотренная схема включения выполняет операцию умножения (или деления) входного сигнала на постоянную величину.

В схеме, показанной на рис. 7.45, выходное напряжение

$$U_{\text{ВЫХ}} = -(1/RC) \int U_{\text{ВХ}} dt.$$

Такая схема включения ОУ выполняет операцию интегрирования, поэтому ее часто называют интегратором. Интеграторы очень широко применяются при решении на АВМ дифференциальных уравнений. Если выходное напряжение интегратора представляет собой некоторую функцию времени, то входное напряжение при этом представляет собой производную этой функции по времени.

В схеме, показанной на рис. 7.46, выходное напряжение

$$U_{\text{ВЫХ}} = -R_{o.c}C(dU_{\text{ВХ}}/dt).$$

Такая схема включения ОУ выполняет операцию дифференцирования.

В сумматорах на базе ОУ на вход ОУ поступает несколько сигналов, а на выходе получается их сумма, умноженная на коэффициент усиления ОУ. На рис. 7.47 показана схема сумматора для трех сигналов (U_1 , U_2 , U_3), каждый из которых приложен между соответствующим входным зажимом и общим зажимом (масса или «земля»). Выходной сигнал такого сумматора

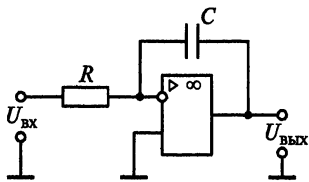


Рис. 7.45. Схема интегрирующего элемента на базе операционного усилителя

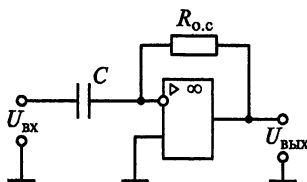


Рис. 7.46. Схема дифференцирующего элемента на базе операционного усилителя

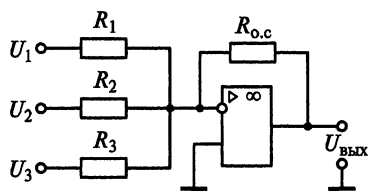


Рис. 7.47. Схема сумматора на базе операционного усилителя

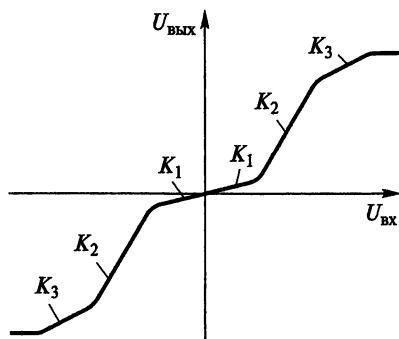


Рис. 7.48. Статическая характеристика нелинейного преобразователя

$$U_{\text{ВЫХ}} = -((R_{0.c} / R_1)U_1 + (R_{0.c} / R_2)U_2 + (R_{0.c} / R_3)U_3).$$

Для сравнения двух сигналов по величине используется схема аналогового компаратора, также построенная на базе ОУ (см. подразд. 7.8, рис. 7.38 и 7.39).

Таким образом, различные схемы включения ОУ обеспечивают выполнение самых разных математических операций, причем не только арифметических. Каждая из этих схем называется *решающим устройством* АВМ. В состав АВМ входит большое число таких решающих устройств, а также резисторов и конденсаторов с различными значениями сопротивлений и емкостей, что позволяет получить любой требуемый коэффициент преобразования при выполнении той или иной математической операции. Кроме линейных решающих устройств имеются и нелинейные решающие устройства, выполняющие преобразования входных сигналов с переменными коэффициентами. Например, в состав аналогового вычислительного комплекса типа АВК-6 входит универсальный нелинейный преобразователь. Его статическая характеристика (рис. 7.48) состоит из нескольких участков, каждый из которых имеет свой коэффициент преобразования (K_1 , K_2 или K_3). Значения этих коэффициентов, размеры участков характеристики, в пределах которых они действуют, можно изменять (регулировать) в широких пределах.

Электронные АВМ являются самым распространенным типом АВМ. В то же время существует большое число сравнительно несложных, но очень важных практических задач, для решения которых вполне достаточно АВМ механического, пневматического и электромеханического типов. Такие машины просты в обслуживании, контроль за их работой, как правило, может быть лишь периодическим (например, раз в месяц) и не требует высокой квалификации работника.

В механических АВМ широко применяются такие счетно-решающие механизмы (СРМ), как суммирующие (конические дифференциалы) и функциональные преобразователи (кулачковые механизмы, механические передачи с некруглыми зубчатыми колесами, обеспечивающие нелинейные зависимости выходных сигналов от входных, и приборы с неравномерными шкалами).

Основными элементами пневматических АВМ являются дроссели (пневматические сопротивления), пневматические емкости и мембраны.

К элементам электромеханических АВМ относятся потенциометры, сельсины, вращающиеся трансформаторы и тахогенераторы.

Контрольные вопросы

1. Перечислите типовые элементы вычислительной техники.
2. Какие логические элементы относятся к основным?
3. Что такое триггер и для чего он используется в вычислительной технике?
4. Как устроен регистр и каково его основное назначение?
5. Зачем нужен регистр сдвига?
6. Что такое счетчик?
7. Что такое сумматор?
8. Зачем нужны шифраторы и дешифраторы, в чем разница между ними?
9. Зачем нужны АЦП и ЦАП, в чем разница между ними?
10. Что такое компаратор?
11. Для чего нужны мультиплексор и демультиплексор, в чем разница между ними?
12. Что такое операционный усилитель?
13. Какие математические операции можно выполнять на операционном усилителе?

ОСНОВЫ МИКРОПРОЦЕССОРНЫХ СИСТЕМ**8.1. Назначение процессоров и микропроцессоров**

Процессор — это главная часть цифровой ЭВМ, осуществляющая сложную переработку информации. В него входит также устройство управления ЭВМ. Процессор не только обрабатывает информацию и управляет данным процессом, но и обеспечивает при этом взаимодействие с устройствами памяти ввода и вывода.

В ЭВМ первых поколений, построенных на дискретных (т.е. отдельных) элементах (электронных лампах, полупроводниковых триодах), процессор представлял собой большое устройство, состоявшее из нескольких электронных плат с размещенными на них навесными компонентами радиоэлектроники. Кроме ламп и транзисторов на этих платах находились диоды, резисторы, конденсаторы. Все соединения между отдельными компонентами осуществлялись с помощью пайки и проводов, а позднее — печатным монтажом.

Успехи микроэлектронных технологий позволили в одном элементе объединять несколько транзисторов, диодов, резисторов и соединений между ними. Таким образом, появились так называемые интегральные схемы (ИС). С годами степень интеграции (т.е. число элементов в одной ИС) возрастала, появились большие интегральные схемы (БИС), а затем и сверхбольшие интегральные схемы (СБИС). Основой ИС является кристалл полупроводника, на котором формируются полупроводниковые переходы, выполняющие роль транзисторов и диодов. На том же кристалле создаются микрообласти с добавлением примесей, осуществляющие функции резисторов и конденсаторов; выполняются также электрические соединения между ними. Если в первых ИС на одном кристалле размещалось до десятка транзисторов, то в современных — десятки миллионов элементов. Использование СБИС позволяет значительно повысить эффективность цифровых систем: увеличить их производительность и надежность, уменьшить габаритные размеры, массу и потребляемую мощность. Современные технологии изготовления СБИС очень сложны и требуют дорогостоящего оборудования.

Создание завода по производству СБИС обходится в миллиарды долларов. Но стоимость цифровой техники, построенной на СБИС, неуклонно снижается. Объясняется это следующим обстоятельством. Интегральная схема, содержащая большое число эле-

ментов, является универсальной, т.е. находит применение в самых разных устройствах. Следовательно, ее можно выпускать огромными тиражами — миллионами штук, а при массовом выпуске экономически оправдано использование высокопроизводительных автоматических и робототехнических линий и участков производства.

Применение СБИС оказало большое влияние на принципы построения цифровых систем, их архитектуру, логическую структуру, математическое обеспечение. Появился новый подход к проектированию таких систем — на основе программируемой логики. Этот подход предполагает использование при построении систем одной (или очень малого количества) стандартной универсальной СБИС, управляемой программно.

Специализация системы осуществляется программой, которая управляет стандартной универсальной СБИС. В 1970-х годах появилась СБИС, которая в значительной степени была способна выполнять функции процессора. Такая интегральная схема получила название *микропроцессор* (МП).

Если к микропроцессору добавляется память (запоминающее устройство) и устройство ввода — вывода, то такая система может выполнять функции ЭВМ. Созданные на основе микропроцессора вычислительные машины стали называться микроЭВМ. Именно благодаря появлению микропроцессоров удалось сделать доступные для многих ЭВМ, получившие название «персональный компьютер».

Итак, микропроцессор — это выполненное по интегральной технологии цифровое устройство, обрабатывающее информацию в соответствии с программой и управляющее вводом и выводом информации. Наибольшее распространение получили микропроцессоры, выполненные на одном кристалле, или однокристалльные МП. О них в дальнейшем и пойдет речь.

Микропроцессор (рис. 8.1) представляет собой неразъемный конструктивный элемент, подсоединяемый к другим элементам

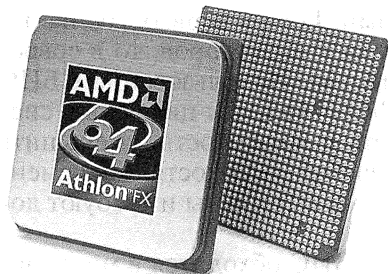


Рис. 8.1. Внешний вид современного микропроцессора

вычислительной машины с помощью выводов. Корпус МП сделан обычно из пластмассы или керамики. Число выводов может быть разным: 28, 40, 64 и больше. Первые МП имели выводы с двух сторон корпуса, по одному ряду с каждой стороны. Современные МП имеют выводы на нижней плоскости с четырех сторон, по несколько рядов с каждой стороны (например, у МП Athlon восемь рядов).

С ростом числа компонентов в одном МП (счет, как уже отмечалось, идет на миллионы) увеличивается и число выводов. В современных МП число выводов (ножек) близко тысяче. По соображениям удобства на число выводов стараются наложить ограничения.

Непрерывное совершенствование интегральных технологий приводит к изменениям в структуре микропроцессора.

8.2. Архитектура и структура микропроцессора

Архитектура микропроцессора — это совокупность сведений о составе его компонентов, организации обработки в нем информации и обмена информацией с внешними устройствами ЭВМ, а также о функциональных возможностях микропроцессора, выполняющего команды программы.

Структура микропроцессора — это сведения только о составе его компонентов, соединениях между ними, обеспечивающих их взаимодействие. Таким образом, архитектура является более общим понятием, включающим в себя кроме структуры еще и представление о функциональном взаимодействии компонентов этой структуры между собой и с внешней средой.

Основой любого МП (рис. 8.2) является арифметико-логическое устройство *АЛУ*, выполняющее обработку информации — арифметические и логические действия над исходными данными в соответствии с командами. Сами данные (исходные, промежуточные и конечный результат) находятся в регистрах данных *РД*, а команды — в регистре команд *РК*. Управление всеми процессами по вводу и выводу информации, взаимодействию между *АЛУ*, *РД* и *РК* осуществляет многофункциональное устройство управления *УУ*. Данные, команды и управляющие сигналы передаются по внутренней шине *ВШ*.

Рассмотрим в общем виде взаимодействие приведенных на рис. 8.2 структурных элементов МП в процессе обработки информации. Порядок действий задается устройством управления *УУ*, которому для обеспечения синхронного взаимодействия всех элементов необходимы постоянно поступающие тактовые импульсы. Эти импульсы вырабатывает тактовый генератор. При поступлении каждого тактового импульса происходит или перемещение информации из одного регистра в другой, или преобразование информации в том или ином регистре, или подключение к шине, или какие-то иные дей-

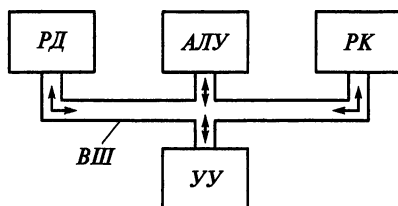


Рис. 8.2. Типовая структурная схема микропроцессора

ствия. Каждая операция выполняется в определенной последовательности за несколько циклов синхронизации, число которых зависит от сложности операции. Содержание операций определяется командами, т.е. информацией, которую МП принимает в регистр команд *РК*. Информация о данных, над которыми выполняются операции, МП принимает в регистр данных *РД*. И информация о командах, и информация о данных представляет собой двоичную кодовую комбинацию, состоящую из некоторого определенного для данного МП числа разрядов. В общем случае эта двоичная комбинация называется *словом*. Слово — это информация, которой оперирует МП. В первых микропроцессорах слово состояло только из четырех разрядов. По мере совершенствования микропроцессоров увеличивалась длина используемых в нем слов и, соответственно, число разрядов в регистрах, где эти слова записываются. Поэтому бывают микропроцессоры 8-разрядные (или 8-битные), 16-, 32- и 64-разрядные. Чем больше разрядов, тем большими функциональными возможностями обладает МП и тем выше его производительность.

В 8-разрядном РК могут находиться 256 различных чисел: от 0000 0000 до 1111 1111 в двоичном представлении, или от 00 до FF в шестнадцатеричном. Следовательно, 8-разрядный МП в общем случае имеет 256 различных команд.

Способность выполнять множество различных операций делает МП мощным и гибким инструментом. Но эта способность не является принципиально новой для цифровых устройств, поскольку многие из них уже при среднем уровне интеграции допускают реализацию ограниченного числа команд. Покажем на простом примере, что одно и то же вычислительное устройство может выполнять разные операции в зависимости от команды, т.е. кодовой комбинации, подаваемой на его входы. Рассмотрим двоичный реверсивный счетчик с начальной установкой, показанный на рис. 8.3.

На входы $+1$, -1 реверсивного счетчика *СТ* подаются счетные импульсы соответственно в режимах сложения и вычитания. Через входы предварительной записи $D0 - D2$ в счетчик может

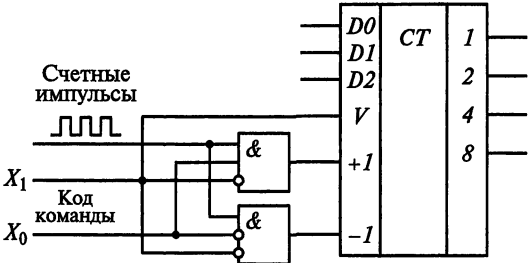


Рис. 8.3. Схема реверсивного счетчика, выполняющего четыре команды

быть записано число, т.е. выполнена начальная установка. Такая запись осуществляется при поступлении импульса на вход разрешения V . Выводы 1, 2, 4, 8 — выходы разрядов счетчика, цифры указывают на вес каждого из них.

Можно считать, что такой счетчик вместе с логическими элементами И, некоторые входы которых являются инверсными, представляет собой устройство, выполняющее три операции в зависимости от значений двоичных сигналов, управляющих направлением счета и начальной установкой. Например, сигнал 00 — счет на вычитание; 01 — разрешение ввода начального состояния; 10 — счет на прибавление; 11 — разрешение ввода начального состояния.

Таким образом, 1 в младшем разряде разрешает ввод начального состояния независимо от символа во втором разряде. При 0 в младшем разряде счетные импульсы распределяются между входами $+1$ и -1 следующим образом: 1 во втором разряде обеспечивает счет на прибавление, а 0 в этом разряде — счет на вычитание. Последовательность значений этой двухразрядной (двухбитовой) команды в сочетании с тактовыми импульсами определяет порядок работы схемы по подсчету входных сигналов. Эта простая схема может выполнять команды, но не способна выбирать их из памяти. При этом в реверсивный счетчик CT вводится не сам двухразрядный код команды, а результат его преобразования двумя логическими элементами. Следовательно, эти элементы являются дешифраторами кода команды.

На рис. 8.4 показана более подробная структурная схема МП, на которой по сравнению с рис. 8.2 детальнее отмечено назначение регистров. В регистр команд PK записывается код команды, который преобразуется в управляющие сигналы дешифратором команд $ДшК$. Исходные данные и промежуточные результаты записываются в регистры общего назначения $РОН$, один из которых более тесно взаимодействует с АЛУ. В нем обычно и размеща-

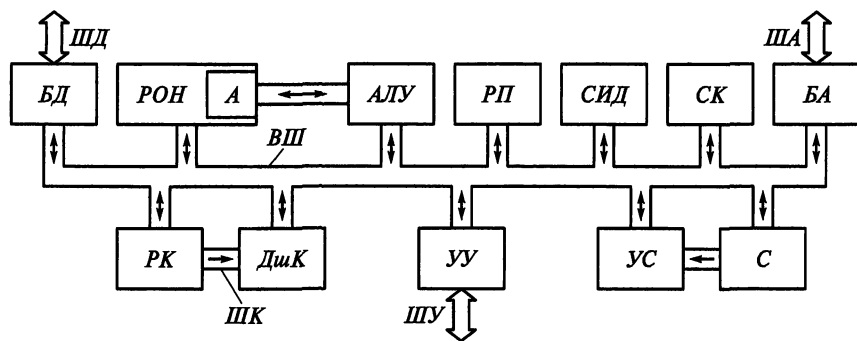


Рис. 8.4. Детальная структурная схема микропроцессора

ются результаты каждой арифметической операции. Этот регистр называется *аккумулятором*, или регистром *А*. В процессе вычислений и обработки информации используется так называемый *регистр флагов*, или регистр признаков *РП*. Под словом «флаг» понимается сигнал о том, что в процессе вычислений или обработки информации результат получил некий характерный признак. К числу таких признаков относятся следующие: получился равный нулю результат (флаг нуля); получился отрицательный результат (флаг знака); количество цифр 1 в результате есть четное число (флаг четности); наличие переноса из старшего разряда (флаг переноса) и др. Регистр флагов состоит из нескольких независимо управляемых триггеров, каждый из которых при переходе из состояния 0 в состояние 1 сигнализирует о появлении соответствующего признака.

Загрузка некоторых регистров МП, обработка содержащихся в них слов, пересылка слов в другие регистры могут происходить по командам программы. Такие регистры называются пользовательскими. Для временного хранения информации используются *буферные* регистры (данных — *БД* и адресов — *БА*), они программисту недоступны. Для повышения производительности МП используют отдельные шины для передачи информации — шину данных *ШД*, адресов *ША*, команд *ШК*. При этом шина адресов может быть объединена с шиной данных и иметь большее число разрядов, чем шина команд. Для передачи управляющих сигналов служит шина управления *ШУ*, число разрядов которой обычно не больше, чем в шине команд. По этой двунаправленной шине в МП передаются сигналы о состоянии внешних устройств, запросы на прерывание от внешних устройств к МП и разрешения прерывания от МП к внешним устройствам, команды записи слова во внешнюю память и чтения слова из памяти.

Если в 8-разрядном МП числа команд (256) вполне достаточно, то такого же числа адресов мало. Для преодоления этого недостатка адресные регистры 8-разрядного МП имеют 16-разрядный формат. Для записи адреса применяется *двойное слово*, т. е. используются два регистра, загрузка которых производится по внутренней 8-разрядной шине *ВШ* в два приема. С адресной шиной кроме регистра адресов *БА* обычно связаны еще два регистра: счетчик команд *СК* и указатель стека *УС*.

Счетчик команд указывает в МП адреса (т. е. номера ячеек), по которым в памяти находятся команды. После считывания каждой команды автоматически происходит прибавление единицы к адресу (номеру). Эта операция называется *инкрементом*. При начальной установке в счетчик команд заносится определенное значение, откуда и может быть считана первая команда программы. Обычно этот исходный адрес относится к постоянному запоминающему устройству (ПЗУ).

Стек *S* представляет собой группу регистров в МП, куда временно может записываться некоторая информация (слова), которая будет использована позже. Регистр *УС* служит для указания адреса той ячейки в запоминающем устройстве с произвольной выборкой, в которую записано последнее по времени поступления слово. Именно оно и будет считано первым при обращении к стеку. Такой принцип функционирования стека называется «последним записан — первым считан». При написании программ, содержащих команды записи и чтения стека (эти команды обычно называется *занесением в стек* и *извлечением из стека*), необходимо обращать внимание на то, чтобы число занесений равнялось числу извлечений. Под действием некоторых команд и сигналов содержимое программного счетчика заносится («заталкивается») в стек, чтобы сохранить адрес очередной команды в последовательности, естественный ход которой потребовалось нарушить. Команды, при выполнении которых имеют место операции со стекком, называются *командами обращения к подпрограммам*. Их использование позволяет переходить к широко используемым (так называемым стандартным) программам из различных частей основной программы.

Сигналы, вызывающие операции со стекком, называются *сигналами прерываний*. В процессе обращения к подпрограмме или обработки прерываний в счетчик команд *СК* заносится новый адрес команды. Затем выполняется программа, начинающаяся с этого адреса и заканчивающаяся так называемой командой возврата из подпрограммы, посредством которой хранившееся в стеке значение программного счетчика извлекается («выталкивается») из стека. После этого возобновляется выполнение программы в прерванной последовательности. При каждом занесении в стек содержимое указателя стека *УС* автоматически увеличивается на единицу (инкрементируется), а при каждом извлечении из стека — уменьшается на единицу (декрементируется). Поскольку инкрементирование и декрементирование выполняются достаточно часто, в МП имеется схема инкремент — декремент *СИД*.

Если в 8-разрядном МП происходит занесение в стек или извлечение из него 16-разрядных слов, то каждая из таких операций выполняется в два приема, поскольку используются два 8-разрядных регистра. Когда необходимо выполнить передачу данных между МП и ячейками внешней памяти, адреса которых отсутствуют и в счетчике команд, и в указателе стека, применяется регистр адреса. Это расширяет возможности МП по запоминанию и извлечению данных из памяти.

В основе архитектуры любого процессора можно выделить несколько характерных частей: кэш команд и данных, предпроцессор и постпроцессор, называемый также блоком исполнения команд. Английское слово *cash* означает наличные, т. е. деньги,

оплата которыми производится очень быстро, значительно быстрее, чем оплата банковским чеком, для получения денег по которому надо обращаться в банк. В вычислительной технике слово «кэш» обозначает сверхоперативное запоминающее устройство, обращение к которому происходит очень быстро. Объем кэш-памяти ограничен, поэтому иногда различают кэш 1-го уровня и кэш 2-го уровня (у последнего объем больше, но быстродействие меньше).

Процесс обработки данных состоит из нескольких характерных этапов. Прежде всего команды и данные забираются из кэша (который разделен на кэш данных и кэш команд). Такая процедура называется *выборкой*. Выбранные из кэша команды декодируются в понятные для данного процессора машинные команды. Эти декодированные команды поступают на исполнительные блоки процессора, где происходит их выполнение. Результат записывается в оперативную память. Процесс выборки команд-инструкций из кэша, их декодирование и продвижение к исполнительным блокам осуществляется в предпроцессоре, а процесс выполнения декодированных команд — в постпроцессоре. Таким образом, даже в самом простейшем случае команда проходит, как минимум четыре стадии обработки: выборка из кэша, декодирование, выполнение, запись результатов. Данные стадии принято называть *конвейером обработки команд* (рис. 8.5). В рассмотренном случае конвейер является четырехступенчатым. Каждую из ступеней команда должна проходить ровно за один такт. Следовательно, при четырехступенчатом конвейере на выполнение одной команды отводится ровно четыре такта.

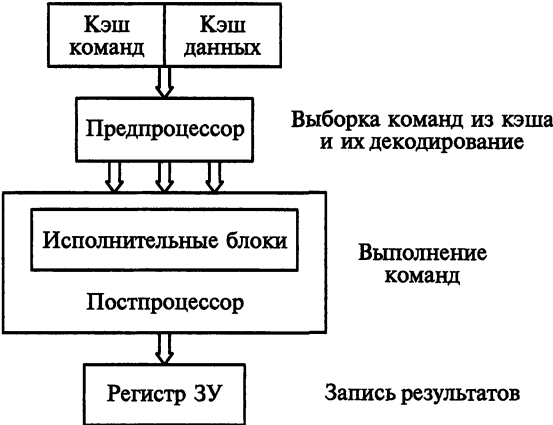


Рис. 8.5. Конвейер обработки команд

В реальных процессорах конвейер обработки команд может быть более сложным и включать в себя большее число ступеней. Например, конвейер процессора Пентиум 4 содержит 20 ступеней. Однако идеология построения процессора остается неизменной. Причина увеличения длины конвейера заключается в том, что многие команды являются довольно сложными и не могут быть выполнены за один такт процессора, особенно при высоких тактовых частотах. Поэтому каждая из четырех стадий обработки команд (выборка, декодирование, выполнение, запись) может состоять из нескольких ступеней конвейера. Длина конвейера является одной из наиболее значимых характеристик любого процессора.

Всякий процессор в конечном счете должен быть сконструирован таким образом, чтобы за минимальное время выполнять максимальное число команд. Именно число выполняемых за единицу времени команд определяет производительность процессора.

Существуют два принципиально различных подхода к повышению производительности процессора (не считая, конечно, увеличения тактовой частоты). Первый заключается в том, чтобы за счет уменьшения длины конвейера увеличивать число исполнительных блоков. По существу при таком подходе реализуется множество параллельных коротких конвейеров. При этом предпроцессор работает по классической схеме: осуществляет выборку команд, их декодирование и посылку на множество исполнительных блоков, которые входят в АЛУ. Данный подход позволяет реализовать параллелизм на уровне команд, когда несколько из них выполняются одновременно в различных исполнительных блоках процессора. Важно, что число ступеней конвейера внутри постпроцессора (исполнительного блока) невелико, поэтому команды выполняются за небольшое число тактов.

Второй подход заключается в параллельном выполнении нескольких команд. Для реализации параллелизма на уровне команд необходимо, чтобы поступающие на исполнительные блоки команды можно было выполнять параллельно (одновременно). Если, например, для выполнения следующей по порядку команды требуется знать результат выполнения предыдущей (взаимозависимые команды), то параллельное выполнение невозможно. Поэтому предпроцессор прежде всего проверяет взаимозависимость команд и располагает их не в порядке поступления, а так, чтобы их можно было выполнять параллельно. На последних ступенях конвейера команды выстраиваются в исходном порядке.

Таким образом, при коротком конвейере на каждой ступени процессор способен выполнять большее количество работы, однако при прохождении команды через каждую ступень конвейера требуется больше времени, что ограничивает повышение тактовой частоты процессора.

8.3. Характеристики и классификация процессоров и микропроцессоров

Поскольку процессор является основным устройством ЭВМ и именно в нем выполняются все вычисления и обработка информации, то его основные характеристики определяют эффективность использования ЭВМ в целом. К важнейшим характеристикам процессора, определяющим его вычислительные свойства, относятся:

- разрядность;
- емкость адресуемой памяти;
- длина конвейера;
- назначение (универсальный или специализированный);
- число внутренних регистров;
- способ управления;
- число уровней прерывания;
- тип стековой памяти;
- состав резидентного и программного обеспечения.

Эти же характеристики определяют и вычислительные свойства микропроцессора. Но для оценки области использования и особенностей разработки вычислительной техники на основе МП важными являются также характеристики микропроцессора как интегральной схемы. Основными из них являются:

- быстродействие;
- потребляемая мощность;
- масса и габаритные размеры;
- число источников питания;
- надежность;
- эксплуатационная стойкость;
- стоимость.

Классификация МП по наиболее существенным из перечисленных характеристик служит основой для выбора эффективной области применения того или иного типа МП.

По назначению МП подразделяются на универсальные и специализированные.

К универсальным относятся МП, имеющие широкое применение в различных областях при выполнении самых разных задач. В персональных компьютерах используются именно универсальные МП.

Специализированные МП предназначены для конкретных применений, их характеристики наиболее соответствуют определенному кругу задач. Например, в ранних моделях компьютеров применялись в основном универсальные МП (модели фирмы Intel 8088, 80286, 80386), в которых не была предусмотрена специальная команда для обработки чисел с плавающей запятой. При необходимости работы с такими числами МП выполнял

каждую операцию очень медленно — за несколько десятков тактов. Поэтому на материнской плате было предусмотрено место для установки дополнительного специализированного МП, так называемого математического сопроцессора (модели фирмы Intel 8087, 80287, 80387).

Наличие дополнительного специализированного МП позволило уменьшить время выполнения некоторых операций (например, извлечение корня или вычисление тригонометрических функций) в десятки и сотни раз. Однако для большого числа пользователей, которым подобные вычисления не требуются, вполне достаточно только основного МП.

По разрядности МП подразделяют на МП с фиксированной и изменяемой разрядностью слова (модульные). Постоянное совершенствование микроэлектронных технологий позволяет непрерывно увеличивать разрядность МП. Для многокристалльных секционированных МП характерно то, что операционная часть, содержащая АЛУ и регистры, разделена (секционирована) на равные части, представляющие собой 2-, 4- или 8-разрядные слои со своими адресными и информационными шинами. Управление выполнением операций осуществляется от отдельного кристалла микропрограммного управления, причем микропрограммное слово поступает на все секции. В зависимости от требуемой разрядности может быть выбрано число секций, обеспечивающее 16, 32 и более разрядов.

Число внутренних регистров служит одним из показателей вычислительных возможностей МП. Этот показатель также непрерывно возрастает: 2 — в самых простых МП, 8 и 16 — в достаточно распространенных, 64 и более — в МП типа Пентиум и других новых моделях. Число регистров МП фактически характеризует объем сверхоперативной памяти МП с малым временем обращения.

Современные МП, как уже отмечалось в подразд. 8.2, имеют кэш-память (или кэш) 1-го и 2-го уровней. Кэш 1-го уровня — это память с минимальным временем обращения. Его объем невелик (например, 16 Кбайт), тогда как объем кэш 2-го уровня достигает нескольких мегабайт.

Быстродействие МП характеризуется тактовой частотой, которая в новейших моделях составляет тысячи мегагерц.

Производительность МП является его интегральной характеристикой, которая зависит от тактовой частоты работы процессора, его разрядности, а также от особенностей архитектуры (наличие кэш-памяти и др.).

Производительность МП нельзя вычислить, она определяется в процессе тестирования по скорости выполнения МП определенных операций в какой-либо программной среде.

По способу управления МП подразделяются на микро- и макропрограммируемые. Микропрограммное управление позволяет

пользователю установить свой собственный набор команд, который будет наилучшим образом соответствовать решению конкретных задач.

Обычно в микропроцессорных секциях с наращиваемой разрядностью применяется именно такой способ управления. Макропрограммное управление использует набор неизменных команд, определяемых схемой МП, поэтому такое управление называют также жестким аппаратным.

Число необходимых источников питания определяет сложность монтажа вычислительного устройства с МП и влияет на габаритные размеры, надежность и стоимость этого устройства. Обычно требуются два-три источника питания, но при некоторых технологиях изготовления удается обойтись одним.

Новинкой последних лет является появление двух- и четырехядерных процессоров. В одном корпусе (или даже на одном кристалле кремния) размещены два ядра процессора. Это позволяет параллельно (т.е. одновременно) выполнять несколько задач. Таким образом, достигается повышение производительности. Прежде производительность повышали за счет увеличения частоты. Можно рассматривать двухядерный процессор как два процессора, но в некоторых типах таких процессоров оба ядра имеют общую кэш-память первого уровня, распределяя ее ресурсы по необходимости.

Контрольные вопросы

1. Каково назначение процессора в ЭВМ?
2. Что такое микропроцессор?
3. Для чего предназначены буферные регистры?
4. Какие сигналы поступают по шинам процессора?
5. Перечислите основные характеристики микропроцессора.
6. Какие показатели характеризуют интегральные схемы?
7. Что такое кэш-память?

АРИФМЕТИКО-ЛОГИЧЕСКИЕ УСТРОЙСТВА ПРОЦЕССОРА

9.1. Назначение и состав арифметико-логических устройств

Арифметические и логические операции над числами (операндами, словами) выполняются в главной части процессора — арифметико-логическом устройстве.

Все арифметические действия с двумя числами (сложение, вычитание, умножение, деление) сводятся в АЛУ к операции сложения или вычитания. Поэтому в состав АЛУ обязательно входит сумматор или вычитатель (большой разницы между ними нет). Два числа (операнды) находятся в двух регистрах с соответствующими логическими схемами, их взаимодействием с сумматором руководит устройство управления. Результат выполненной операции может направляться по указанному в команде адресу, но обычно остается в специальном регистре — аккумуляторе. Процессор содержит несколько регистров, но наиболее близок к АЛУ именно аккумулятор.

На рис. 9.1 показана структурная схема АЛУ и его связи с устройством управления процессора УУП и запоминающим устройством процессора ЗУП. Аккумулятор на схеме не изображен, его роль может выполнять один из двух регистров, например $Р_2$. Сумматор $С_m$ предназначен для суммирования чисел, регистры $Р_1$ и $Р_2$ — для хранения слагаемых, или уменьшаемого и вычитаемого, или множимого и множителя, или делимого и делителя (в зависимости от выполняемой операции). Устройство управления вычислениями УУВ координирует работу АЛУ, управляет последовательностью действий, необходимых при выполнении конкретной операции. С ЗУП АЛУ связано шинами чтения данных ШЧД и записи данных ШЗД, с УУП — шиной управления ШУ, по которой в УУВ

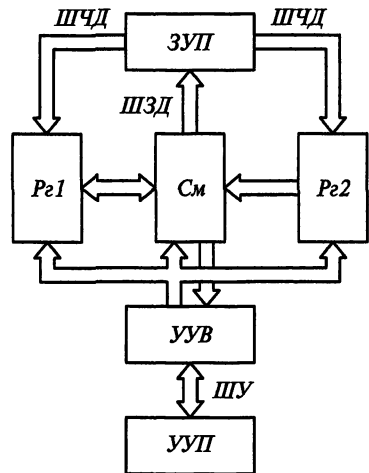


Рис. 9.1. Структурная схема арифметико-логического устройства

поступают тактовые импульсы, а из *УУВ* — сигнал об окончании вычислений.

Работает АЛУ следующим образом. Из *УУП* код арифметической или логической операции поступает в *УУВ*, где формируются сигналы, соответствующие данной операции. Затем из *ЗУП* выбирается первый операнд (по адресу, указанному в команде), который по *ШЧД* поступает на *P₂₁*. Второй операнд, выбранный из *ЗУП* по второму адресу, указанному в команде, поступает также по *ШЧД* в *P₂₂*. После приема обоих операндов начинается выполнение операции, в ходе которой используются сигналы переноса, делаются необходимые сдвиги. Результат операции формируется в *См*. Сигналы, характеризующие признаки результата в *См*, называются *признаками*. Эти сигналы являются также составной частью понятия «состояние процессора» (подробнее см. подразд. 11.4).

По окончании формирования результата вырабатывается сигнал признака конца операции, в соответствии с которым результат из *См* поступает через *ШЗД* в *ЗУП* по адресу, указанному в команде. Кроме формирования результата в АЛУ могут вырабатываться различные сигналы, обозначающие признаки результата (признак переполнения, признак отрицательного результата и т. д.). Эти признаки поступают в *УУП* и влияют на дальнейший ход вычислительного процесса.

Основные характеристики АЛУ и его структура зависят от принятой системы счисления, способа реализации вычислительного процесса, формы представления чисел, способа представления отрицательных чисел, разрядности чисел, типа схемы АЛУ, состава операций, используемой методики вычислений и требуемого быстродействия.

В зависимости от принятой системы счисления различают АЛУ с двоичной (используется чаще всего), десятичной или двоично-десятичной системой (иногда говорят — арифметикой).

В зависимости от способа реализации вычислительного процесса различают АЛУ последовательного и параллельного действия.

В АЛУ последовательного действия каждый операнд вводится последовательно, разряд за разрядом.

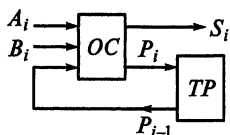


Рис. 9.2. Схема, поясняющая принцип действия АЛУ последовательного действия

Операции производятся также последовательно, поразрядно. Числа представляются в виде временной последовательности сигналов и имеют один общий выход, причем каждому разряду отводится определенная временная позиция относительно заданного начала отсчета. Такое АЛУ преобразовывает временные последовательности, соответствующие обоим слагаемым, во временную после-

довательность, соответствующую сумме, выдаваемой по специальной цепи, начиная от младших разрядов и кончая старшими разрядами и знаком. Эту функцию обычно выполняет двоичный одноразрядный сумматор. Если его дополнить схемой хранения переносов, то он может играть роль последовательного сумматора в АЛУ последовательного действия. На рис. 9.2, поясняющем принцип действия такого АЛУ, показаны одноразрядный сумматор OC и триггер переноса TP . При суммировании цифр A_i и B_i очередного i -го разряда в OC из TP поступает сигнал переноса P_{i-1} , полученный при суммировании цифр предыдущего $(i - 1)$ -го разряда. По быстродействию АЛУ последовательного действия уступают АЛУ параллельного действия, но зато содержат меньше элементов, т. е. являются более дешевыми.

В АЛУ параллельного действия (рис. 9.3) все n разрядов каждого операнда поступают одновременно по n каналам. Действия над числами производятся также одновременно во всех n разрядах. Подобное АЛУ можно представить как n одноразрядных сумматоров, соединенных таким образом, что выход сигнала переноса P_i предыдущего одноразрядного сумматора OC_i является входом последующего одноразрядного сумматора OC_{i+1} . В каждом из таких сумматоров складываются цифры соответствующего разряда чисел A и B .

В зависимости от формы представления чисел различают АЛУ, оперирующие числами с фиксированной запятой, с плавающей запятой и целыми. При одинаковой разрядности в АЛУ с плавающей запятой имеется больший диапазон представления чисел, чем в АЛУ с фиксированной запятой. При обработке чисел с плавающей запятой возможны два способа выполнения операций: последовательный и параллельный. При последовательном способе в одном и том же устройстве сначала вычисляется порядок результата, а затем его мантисса. При параллельном способе порядок и мантисса результата вычисляются в разных устройствах.

С разрядностью АЛУ связаны точность и скорость вычислений. Чем больше разрядность, тем выше точность, но меньше быстродействие. Разрядность АЛУ может быть постоянной и переменной.

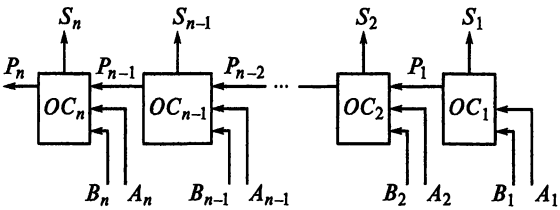


Рис. 9.3. Схема, поясняющая принцип действия АЛУ параллельного действия

В современных ЭВМ, как правило, можно обрабатывать числа различного размера (переменной длины). Существуют два способа указания переменной длины числа. При первом в команде указывается число разрядов операндов, которое может изменяться от 1 до некоторого заданного максимума. При втором способе в памяти отводятся специальные (так называемые маркерные) разряды, которые служат для указания того, что предыдущий разряд был последним для числа, а со следующего разряда начинается другое число.

Структура АЛУ зависит от состава операций, выполняемых ЭВМ, и от принятой методики вычислений, т. е. от выбора алгоритмов операций. Особенно сильное влияние оказывает методика выполнения умножения и деления. При умножении необходимы по меньшей мере три регистра для записи множимого, множителя и сумм частичных произведений. Умножение двоичных чисел в АЛУ может быть сведено к последовательности сложений и сдвигов. Наибольший практический интерес представляет следующий алгоритм: умножение начинается с младших разрядов множителя, множитель сдвигается вправо, сумма частичных произведений также сдвигается вправо, множимое не сдвигается. Этот алгоритм может быть расчленен на следующие этапы:

1) все регистры (P_21 , P_22 , C_m — см. рис. 9.1) устанавливаются в нулевое состояние, затем множитель записывается в P_21 , множимое — в P_22 (сумма частичных произведений будет записываться в C_m);

2) анализируется младший разряд множителя. Если он имеет значение 1, то к сумме частичных произведений прибавляется множимое; если он имеет значение 0, то выполняется следующий по порядку этап;

3) производится сдвиг множителя и суммы частичных произведений на один разряд вправо, младшие разряды частичного произведения попадают в освободившиеся старшие разряды P_21 ;

4) этапы 2 и 3 повторяются столько раз, сколько разрядов в множителе.

Регистров P_21 , P_22 , C_m достаточно и для выполнения операции деления. Она тоже реализуется с помощью многократных сдвигов и сложений (вычитаний). Знаки сомножителей в процессе умножения не используются. Для определения знака результата выполняется сложение знаков сомножителей по модулю 2.

В зависимости от типа схемы АЛУ разделяются на комбинационные и накапливающие. В комбинационных АЛУ результат на выходе появляется только одновременно с входными сигналами. С исчезновением входных сигналов результат также пропадает, поскольку в таких АЛУ нет элементов памяти. Комбинационные схемы имеют обычно потенциальные (не импульсные) связи между элементами. В накапливающих АЛУ операнды поступают последо-

вательно друг за другом, результат операции остается в сумматоре и после исчезновения входных сигналов.

При выполнении логических операций используются обычно те же регистры, что и для арифметических операций. Последовательность выполнения микроопераций, передачу информации между отдельными блоками внутри АЛУ и процессора, связь АЛУ с другими устройствами процессора и ЭВМ обеспечивают УУВ и УУП (см. рис. 9.1). В современных ЭВМ явно выражена тенденция к увеличению и усложнению функций АЛУ. Поэтому состав АЛУ значительно расширяется: увеличивается число регистров и сумматоров, повышается их разрядность и разрядность шин, используются схемы и алгоритмы, обеспечивающие одновременность выполнения нескольких операций.

9.2. Комбинационные схемы

В АЛУ используются комбинационные схемы, а для сохранения результатов, полученных после обработки информации в этих схемах, применяются регистры. Рассмотрим особенности работы комбинационной схемы. Ее выходные сигналы Y в некоторый момент времени t_i зависят только от совокупности (комбинации) сигналов X , присутствующих на ее входах в тот же момент времени t_i , и не зависят от входных сигналов, поступающих в предшествующие моменты времени. Иными словами, комбинационная схема «не помнит» предыстории поступления сигналов на ее входы. Эти сигналы задаются в виде двоичных цифр, поэтому такую схему называют также комбинационным цифровым устройством.

Правила функционирования КЦУ, как уже отмечалось в подразд. 6.2, могут быть заданы различными способами: словесно, таблицами истинности, булевыми выражениями. Для реализации КЦУ выбирают определенные логические элементы из заданного набора и соединяют их таким образом, чтобы обеспечивалась зависимость цифровых выходных сигналов от входных в соответствии с заданными правилами функционирования.

Так как логические элементы, входящие в состав КЦУ, переключаются с задержкой t_3 , то при изменении в некоторый момент времени комбинации входных сигналов выходные сигналы устройства (если они изменяются в результате этого) примут установившиеся значения только после того, как закончатся переходные процессы в соответствующих логических элементах. В КЦУ отдельные логические элементы включены последовательно, поэтому длительность переходных процессов зависит от числа логических элементов в такой последовательной цепочке. Обычно для оценки общей задержки цепочки просто суммируют задержки отдельных логических элементов. При оценке быстродействия КЦУ

необходимо выявить ту цепочку логических элементов между его входами и выходами, которая будет задавать наибольшую задержку, и сложить между собой задержки логических элементов этой цепочки. Обычно она содержит наибольшее число включенных последовательно логических элементов. Но могут быть исключения, связанные, например, с наличием в более короткой цепочке отдельных инерционных логических элементов с большим t_3 . Поэтому в общем случае необходимо проанализировать все цепочки логических элементов от входов до выходов КЦУ и выявить такую, которая дает наибольшую задержку.

Поскольку при прохождении сигнала через отдельные части КЦУ времена задержек в этих частях неодинаковы, то возникают своеобразные состязания в скорости переключения логических элементов. По этой причине на выходе некоторых элементов могут возникать ложные сигналы — помехи. Время задержки может быть разным даже для однотипных элементов.

Под *состязанием* понимают неоднозначное протекание переходного процесса, вызванное случайными отклонениями значений задержек в элементах, разными путями прохождения сигнала (состязание цепей), случайным разбросом моментов изменения переменных на нескольких входах при переходе в несоседнее состояние (состязание входов), разбросом значений времени перехода элементов памяти из одного устойчивого состояния в другое (состязание, или гонки, элементов памяти). В общем случае могут иметь место состязания, вызванные сочетаниями указанных причин.

В качестве примера рассмотрим состязание цепей, которое может возникнуть в КЦУ при изменении одной из входных переменных. Комбинационное устройство, представленное на рис. 9.4, *а*, состоит из трех элементов, реализующих логические функции I_1 , I_2 и ИЛИ. Преобразуем исходную схему с учетом задержек. Логи-

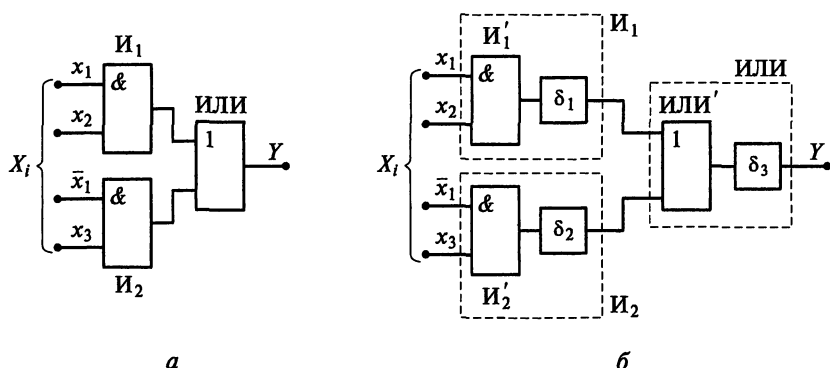


Рис. 9.4. Влияние задержек на работу комбинационного устройства:
 а — исходная схема; б — схема с учетом задержек

ческие элементы с нулевой задержкой обозначим I_1 , I_2 и ИЛИ', а соответствующие им задержки — δ_1 , δ_2 , δ_3 (рис. 9.4, б). Рассматриваемое КЦУ имеет четыре входа с входными переменными x_1, x_2, x_3, \bar{x}_1 , каждая из которых может принимать значение 0 или 1. Входной сигнал КЦУ представляет собой набор сигналов (символов) на всех его входах, а выходной сигнал КЦУ — набор сигналов на всех его выходах (в данном случае на единственном выходе Y). Следовательно, входной набор переменных (алфавит) $X_i = X_1, X_2, \dots, X_8$, а выходной $Y = 1, 0$. В табл. 9.1 указаны сигналы, возникающие на выходе каждого из элементов при отсутствии задержек (идеальный случай). Пусть в момент времени t на вход автомата был подан сигнал (символ) X_1 . Тогда выходным сигналом (символом), как следует из табл. 9.1, будет 1. В идеальном случае при поступлении в момент времени $t_0 > t$ сигнала X_5 , т. е. при переходе x_1 из 1 в 0, выходной сигнал должен остаться неизменным и быть 1. Однако если $\delta_2 > \delta_1$, то на выходе в течение времени $\delta_2 - \delta_1$ будет 0, что видно из рис. 9.5. Следовательно, наличие задержек в элементах автомата может привести к ошибочным переходам, что необходимо учитывать при проектировании АЛУ.

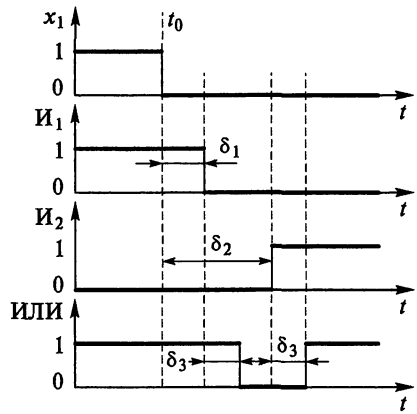


Рис. 9.5. Диаграмма сигналов в комбинационном устройстве

Таблица 9.1

Входные и выходные сигналы КЦУ

Входные сигналы	Значения входных переменных				Сигналы на выходе элементов		
	x_1	x_2	x_3	\bar{x}_1	I_1	I_2	ИЛИ (Y)
X_1	1	1	1	0	1	0	1
X_2	1	1	0	0	1	0	1
X_3	1	0	1	0	0	0	0
X_4	1	0	0	0	0	0	0
X_5	0	1	1	1	0	1	1
X_6	0	1	0	1	0	0	0
X_7	0	0	1	1	0	1	1
X_8	0	0	0	1	0	0	0

9.3. Конечные автоматы

Обработка информации в АЛУ происходит с использованием регистров, т.е. элементов памяти. Поэтому анализировать работу АЛУ и проектировать его необходимо с учетом теории конечных автоматов. *Конечный автомат* отличается от комбинационной схемы наличием памяти. В конечных автоматах выходные сигналы Y в момент времени t_i определяются не только комбинациями входных сигналов X , подаваемыми в тот же момент времени t_i , но и сигналами, поступившими на входы в предшествующие моменты времени. В составе АЛУ обязательно присутствуют элементы памяти, внутреннее состояние которых отражает предысторию поступления последовательности входных сигналов. Конечные автоматы называют также *последовательностными автоматами* (в отличие от комбинационных).

Пусть имеется некоторая система с дискретными переменными, которую назовем автоматом A . Автомат A может иметь k входов, т.е. k входных переменных x_1, x_2, \dots, x_k , каждая из которых может принимать α дискретных значений, и c выходов, т.е. c выходных переменных y_1, y_2, \dots, y_c , каждая из которых может принимать β дискретных значений. Значения выходных переменных являются реакциями автомата на входные переменные. В некоторый момент времени каждая из входных и выходных переменных принимает только одно из дискретных значений. Следовательно, входы автомата могут находиться в одном из $n = \alpha^k$ дискретных состояний. Обозначим через $X = \{x_1, x_2, \dots, x_k\}$ множество состояний входа автомата, которое называется также *входным алфавитом* (элементы этого множества называют *входными символами*). Аналогично выходы могут находиться в одном из $m = \beta^c$ дискретных состояний и образуют множество состояний $Y = \{y_1, y_2, \dots, y_c\}$, или *выходной алфавит* (элементы этого множества называются *выходными символами*). Такое представление входных и выходных переменных автомата позволяет рассматривать его как автомат с одним входом и одним выходом. Будем считать, что у автомата имеется некоторое число r внутренних переменных q_1, q_2, \dots, q_r , каждая из которых принимает дискретные значения, т.е. внутренние состояния автомата представляются множеством состояний $Q = \{q_1, q_2, \dots, q_r\}$, или *внутренним алфавитом* (элементы этого множества называют *состояниями*).

Обычно конечный автомат определяется как совокупность пяти множеств:

$$A = (Q, X, Y, F, G),$$

где $F = \{f_1, f_2, \dots, f_s\}$ — множество отображений $\{X\} \times \{Q_i\}$ в Q , или функция переходов, $j = 1, 2, \dots, k; i = 1, 2, \dots, r$; $G = \{g_1, g_2, \dots, g_p\}$ — множество отображений $\{X\} \times \{Q_i\}$ в Y , или функция выходов.

Запись $\{X_j\} \times \{Q_i\}$ означает, что элементы множеств X и Q объединяются во всевозможные пары (наборы) вида (X_j, Q_i) .

В детерминированном автомате функции переходов и выходов являются всюду определенными (однозначными) функциями.

Когда для данной системы установлены входной и выходной алфавиты и множество состояний, ее словесное описание может быть формализовано с помощью таблицы, графа или матрицы. Таблица, граф и матрица — это различные формы представления функций F и G конечного автомата, который моделирует данную систему. Такое представление необходимо для проведения точного анализа или синтеза конечного автомата. Поскольку одной формой представления автомата целесообразно пользоваться при одних обстоятельствах, другой — при других, познакомимся с двумя из указанных форм.

Автомат может быть задан двумя конечными таблицами, соответствующими функциям F и G и называемыми таблицами соответственно переходов и выходов. Их строки соответствуют символам входного алфавита, а столбцы — состояниям. На пересечении строки и столбца таблицы переходов указывается состояние, в которое переходит автомат из данного состояния под действием данного входного символа. В таблице выходов на пересечении строки и столбца указывается значение, т. е. символ, который появляется на выходе.

Примером задания автомата A с алфавитами $X = \{a, b, c\}$, $Q = \{Q_1, Q_2, Q_3, Q_4\}$ и $Y = \{0, 1\}$ являются табл. 9.2 и 9.3. Часто обе такие таблицы объединяют в одну таблицу переходов. При этом на пересечении строки и столбца помещают состояние $f(X_j, Q_i)$, в которое переходит автомат из Q_i под действием входного символа X_j , и значение $g(X_j, Q_i)$, которое при этом возникает на выходе. Например, на пересечении строки a и столбца Q_2 будут располагаться символы Q_3 и 1. Табличному способу задания автомата свойственны точность и краткость по сравнению со словесным описанием.

При рассмотрении задания автомата с помощью графа будем пользоваться терминологией теории графов. Граф переходов пред-

Таблица 9.2

Таблица переходов автомата

X_j	Q_i			
	Q_1	Q_2	Q_3	Q_4
a	Q_2	Q_3	Q_3	Q_2
b	Q_1	Q_1	Q_3	Q_1
c	Q_1	Q_1	Q_3	Q_4

Таблица 9.3

Таблица выходов автомата

X_j	Y			
	Q_1	Q_2	Q_3	Q_4
a	1	1	0	0
b	1	0	0	1
c	0	1	1	0

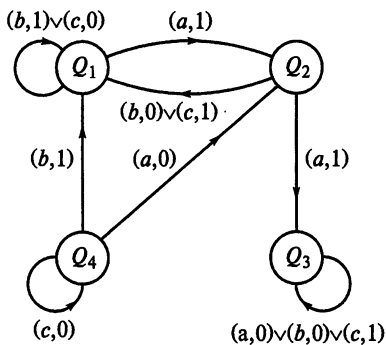


Рис. 9.6. Граф переходов конечного автомата

ставляет собой структуру, состоящую из вершин, изображаемых в виде кружков, и ориентированных дуг, которые изображаются в виде линий между парами вершин и снабжены стрелками, указывающими направление от одной вершины к другой. Граф переходов, описывающий автомат с r состояниями, содержит r вершин, причем каждая из них соответствует одному состоянию автомата и снабжается соответствующим этому состоянию обозначением. Дуги проводятся и обозначаются по следующему правилу. Из вершины Q_i

проводится дуга со стрелкой, направленной от этой вершины к вершине Q_k , если автомат, находящийся в состоянии Q_i , под действием входного символа может быть переведен в состояние Q_k . Пусть X_1, X_2, \dots, X_a — все входные символы, переводящие автомат из Q_i в Q_k , а Y_1, Y_2, \dots, Y_a — выходные символы, соответствующие этим переходам. Тогда дуге, ведущей из Q_i в Q_k , присваивается обозначение $(X_1, Y_1) \vee (X_2, Y_2) \vee \dots \vee (X_a, Y_a)$. Дуга, соединяющая Q_i с Q_i , называется петлей при вершине Q_i . В случае петли входной символ не изменяет внутреннее состояние автомата, но на его выходе возникает соответствующий символ.

Указанные правила устанавливают соответствие между графом переходов и таблицей переходов для одного и того же автомата. Поэтому, зная представление в одной форме, можно получить его и в другой. На рис. 9.6 приведен граф автомата A , заданного табл. 9.2 и 9.3.

Графы обладают большей наглядностью, чем таблицы. Из приведенного графа видно, что состояние Q_3 является тупиковым (оказавшись в нем, автомат не может выйти из него под действием входных символов), а Q_4 — невозвратным. Из состояния Q_4 может быть осуществлен переход в состояние Q_1 или Q_2 , но после выхода из состояния Q_4 в него нельзя попасть ни из какого другого. Заметить подобные свойства по таблице более сложно. Преимущество графов переходов состоит и в том, что они облегчают определение реакции автомата на входную последовательность любой длины. При заданных начальном состоянии Q_i и входной последовательности $X_{j_1}, X_{j_2}, \dots, X_{j_l}$ реакция автомата легко определяется прослеживанием (в направлении стрелок) непрерывной последовательности l дуг, которая начинается в вершине Q_i и k -я дуга которой ($k = 1, 2, \dots, l$) соответствует паре (X_{j_k}, Y_{pk}) . Выходная последовательность, которую при этом выдает автомат, будет

$Y_{p1}, Y_{p2}, \dots, Y_{pl}$. Состояние, в которое перешел автомат, определяется по обозначению вершины, в которой заканчивается последовательность из l дуг. Например, реакция автомата A , представленного графом рис. 9.6, при начальном состоянии $Q_0 = Q_4$ для входной последовательности a, c, b, a, b, a, a, c будет $0, 1, 1, 1, 0, 1, 1, 1$ и автомат в итоге перейдет в состояние Q_3 .

Всякий автомат $A = (Q, X, Y, F, G)$ задается графом в алфавитах $Q, X \times Y$, но не всякий граф в этих алфавитах задает автомат, т.е. не всякий граф является автоматным графом. Условия автоматности графа следующие:

не существует двух дуг с одинаковыми входными символами в обозначении, выходящих из одной и той же вершины (условие однозначности, или детерминизма);

для всякой вершины Q и для любого X есть такая дуга, имеющая в обозначении символ X , которая выходит из Q (условие полноты).

Граф, приведенный на рис. 9.6, удовлетворяет указанным условиям и, следовательно, является автоматным.

Различают синхронные и асинхронные автоматы. Для *синхронного* автомата характерно наличие генератора тактовых, или синхронизирующих, импульсов. Входные импульсы могут воздействовать на автомат лишь при наличии тактового импульса от синхронизирующего генератора.

Автоматы, которые проектируются для работы без синхронизирующих тактовых импульсов, называются *асинхронными*. Они используются тогда, когда входные символы поступают на автомат в случайные моменты времени. Выполняя те же функции, асинхронные автоматы оказываются более быстродействующими, чем синхронные. В реальном автомате каждый элемент, входящий в его состав, вносит некоторую задержку, или запаздывание, равное времени, которое должно пройти, для того чтобы изменение входной переменной элемента вызвало изменение его выходной переменной. Если в синхронных автоматах этими задержками можно пренебречь, то в асинхронных автоматах они оказывают существенное влияние на протекающие процессы.

Следовательно, для асинхронных автоматов имеет значение проблема состязательности, рассмотренная в подразд. 9.2.

9.4. Работа арифметико-логического устройства

Арифметико-логическое устройство является основой не только микропроцессора, но и любого другого устройства с программируемой логикой, которое предназначено для обработки информации. Современные ЭВМ используют МП (и соответственно АЛУ) с очень большими значениями разрядности, быстродействия, числа регистров и т.п. Рассмотреть работу АЛУ легче при более простых

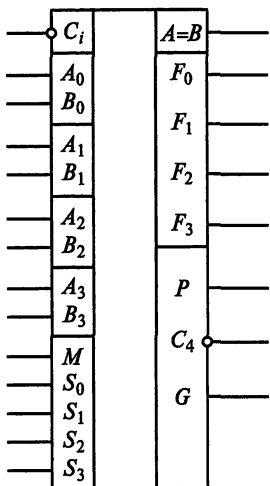


Рис. 9.7. Условное обозначение микросхемы АЛУ

схемах. При анализе архитектуры МП будем считать, что его АЛУ по своим функциональным возможностям, разрядности и набору входов и выходов соответствует стандартному 4-разрядному АЛУ, условное обозначение которого приведено на рис. 9.7. Это АЛУ содержит две входные (A и B) и одну выходную (F) 4-разрядные шины. Информация, поступающая по входным шинам, обрабатывается в соответствии с набором управляющих сигналов, подаваемых на управляющие входы АЛУ M , S_3 , S_2 , S_1 , S_0 . Результат обработки появляется на выходной шине. Вход C_i служит для приема сигнала переноса, а на выходе переноса C_4 образуется сигнал старшего разряда при выполнении арифметических действий. С выхода $A = B$ подается сигнал о равенстве входных двоичных слов A и B при поразрядном сравнении.

Этот выход позволяет использовать данную микросхему АЛУ в качестве компаратора. Дополнительные выходы G (образования ускоренного переноса) и P (распространения ускоренного переноса) используются при организации многоразрядных АЛУ. Реализуемые в АЛУ операции представлены в табл. 9.4, в которой приняты следующие обозначения: \vee — операция дизъюнкции; \wedge — операция конъюнкции; \oplus — операция неравнозначности; $+$ — операция сложения; $-$ — операция вычитания.

Другим важным узлом МП является набор регистров общего назначения (РОН), часто называемый сверхоперативным запоминающим устройством (СОЗУ). В РОН хранятся информационные слова, подлежащие обработке в АЛУ, результаты обработки информации в АЛУ и управляющие слова. Обращение к РОН — адресное. Эти регистры допускают считывание и запись информации, поэтому содержат входную и выходную шины, адресную шину и управляющий вход, информация на котором задает режим работы (запись, хранение или чтение информации).

Рассмотрим упрощенную структурную схему обрабатывающей части МП (рис. 9.8). При этом не будем указывать управляющие входы, что позволит обратить основное внимание на функциональные характеристики системы при передаче и обработке данных. Содержимое любого регистра общего назначения $РОН$ может быть передано на буферный регистр $БР$ и регистр сдвига $РС$. Арифметико-логическое устройство $АЛУ$ может выполнять указанные в табл. 9.4 логические и арифметические операции над содержимым

Операции, выполняемые АЛУ

Код операции				Логические операции ($M=1$)	Арифметические и арифметико-логические операции ($M=0$)	
S_3	S_2	S_1	S_0		$C_i=0$	$C_i=1$
0	0	0	0	$F = \bar{A}$	$F = A$	$F = A + 1$
0	0	0	1	$F = \overline{A \vee B}$	$F = A \vee B$	$F = (A \vee B) + 1$
0	0	1	0	$F = \bar{A} \wedge B$	$F = A \vee \bar{B}$	$F = (A \vee \bar{B}) + 1$
0	0	1	1	$F = 0$	$F = -1$	$F = 0$
0	1	0	0	$F = \overline{A \wedge B}$	$F = A + (A \wedge \bar{B})$	$F = A + (A \wedge B) + 1$
0	1	0	1	$F = \bar{B}$	$F = (A \vee B) + (A \vee \bar{B})$	$F = (A \vee B) + (A \vee \bar{B}) + 1$
0	1	1	0	$F = A \oplus B$	$F = A - B - 1$	$F = A - B$
0	1	1	1	$F = A \wedge \bar{B}$	$F = (A \wedge \bar{B}) - 1$	$F = A \wedge \bar{B}$
1	0	0	0	$F = \bar{A} \vee B$	$F = A + (A \wedge B)$	$F = A + (A \wedge B) + 1$
1	0	0	1	$F = \overline{A \oplus B}$	$F = A + B$	$F = A + B + 1$
1	0	1	0	$F = B$	$F = (A \vee \bar{B}) + (A \wedge B)$	$F = (A \vee \bar{B}) + (A \wedge B) + 1$
1	0	1	1	$F = A \wedge B$	$F = (A \wedge B) - 1$	$F = A \wedge B$
1	1	0	0	$F = 1$	$F = A + A$ (т.е. сдвиг на один разряд влево)	$F = A + A + 1$
1	1	0	1	$F = A \vee \bar{B}$	$F = (A \vee B) + A$	$F = (A \vee B) + A + 1$
1	1	1	0	$F = A \vee B$	$F = (A \vee \bar{B}) + A$	$F = (A \vee \bar{B}) + A + 1$
1	1	1	1	$F = A$	$F = A - 1$	$F = A$

обоих регистров; результат может быть записан в любой из *РОН*. При подаче соответствующих управляющих сигналов возможны, например, следующие операции: передача данных из одного *РОН* в другой путем пересылки выбранного из первого *РОН* слова транзитом через *БР* и *АЛУ* во второй *РОН*; увеличение или уменьшение на единицу содержимого любого *РОН* путем изменения в *АЛУ* выбранного из *РОН* значения на единицу и засылки полученного результата в тот же регистр; сдвиг содержимого любого *РОН* путем передачи выбранного числа в *РС*, сдвига этого числа и записи через *АЛУ* в тот же *РОН*.

Очевидно, что для выполнения этих и других операций в *АЛУ*, *РОН* и вспомогательные регистры *БР* и *РС* должны подаваться

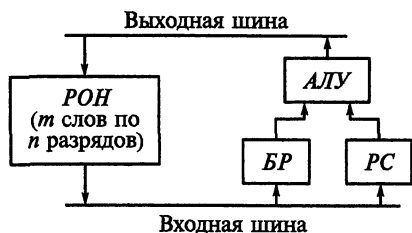


Рис. 9.8. Упрощенная структурная схема обрабатывающей части микропроцессора

определенные управляющие сигналы, причем важное значение имеет распределение этих сигналов во времени. Например, для передачи данных из одного РОН в другой требуются два такта:

- такт 1 — адресация РОН, выборка содержимого РОН, прием выбранного слова в БР,
- такт 2 — адресация РОН, запись в РОН информации, переданной на вход РОН через АЛУ.

Тактовые сигналы могут поступать от тактового генератора. Максимально возможная частота, а значит, и время выполнения операций, будут определяться задержкой сигналов на различных компонентах схемы.

В современных микропроцессорах АЛУ является только частью микросхемы сверхбольшой интеграции. Но среди микросхем просто большой интеграции имеются специальные микросхемы специализированного назначения, которые так и называются — арифметико-логические устройства. Примерами таких микросхем являются К155ИПЗ и 564ИПЗ.

Микросхема К155ИПЗ (рис. 9.9) предназначена для действий с двумя 4-разрядными двоичными словами: $A = A_3A_2A_1A_0$ и $B = B_3B_2B_1B_0$. Конкретный вид операции, выполняемой микросхемой, задается 5-разрядным кодом, подаваемым на входы M, S_3, S_2, S_1, S_0 . Всего это АЛУ способно выполнить $2^5 = 32$ операции: 16 логических (И, ИЛИ, И—НЕ, ИЛИ—НЕ, исключающее ИЛИ и др.) и 16 арифметических и арифметико-логических (сложение, вычитание, удвоение, сравнение чисел и др.). Операции сложения и вычитания проводятся с ускоренным переносом из разряда в разряд. Кроме того, имеется вход приема сигнала переноса C_j .

На выходах F_3, F_2, F_1, F_0 формируются результаты логических преобразований и арифметических действий. На выходе переноса C_4 образуется сигнал старшего (пятого) разряда при выполнении арифметических операций. Дополнительные выходы P и G используются только при организации многоразрядных АЛУ в случае их сочетания с блоком ускоренного переноса.

Слова A и B , подлежащие обработке, могут быть представлены в положительной или отрицательной логике, но таблицы истинности для каждого варианта логики отличаются друг от друга.

Старший разряд кода выбора операции (вход M) определяет характер действий, выполняемых АЛУ. При $M = 1$ АЛУ производит логические операции поразрядно над каждой парой бит слов A и B .

Рассмотрим работу АЛУ на примере приведенной на рис. 9.10 схемы, которая состоит из трех ступеней:

I — ступень выполнения логических операций и формирования полусумм;

II — ступень генерации переносов;

III — ступень формирования полных сумм.

На входы I ступени через входные мультиплексоры поступает входная информация A_i, B_i . На входы $S_0 - S_3$ подаются коды выполняемой операции.

На выходе I ступени формируются функции полусуммы $\Sigma_{1/2}$, генерации G_i и передачи P_i переноса C_i . Функция полусуммы предназначена для формирования результата суммирования с учетом переноса из соседнего младшего разряда на III ступени АЛУ. Функция G_i определяет условие генерации переноса в данном разряде ($G_i = A_i \wedge B_i$). Функция P_i определяет условие передачи переноса из соседнего младшего разряда в следующий, более старший ($P_i = A_i \vee B_i$). Перенос из n -го разряда в процессе двоичного сложения определяется выражением

$$C_n = G_n \vee G_{n-1}P_{n-2} \vee G_{n-2}P_{n-1} \vee \dots \vee G_0P_1P_2P_3 \dots P_n, \quad (9.1)$$

где 0, 1, ..., n — номера разрядов.

В соответствии с этим уравнением строятся схемы, определяющие перенос для каждого разряда АЛУ. Управляющий вход S_4 задает тип выполняемых операций: при $S_4 = 0$ выполняются логи-

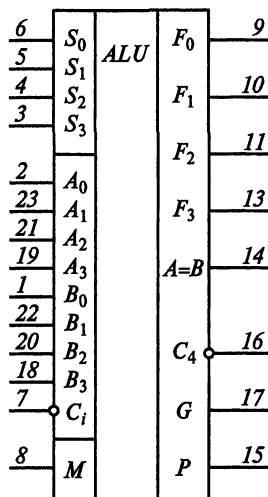


Рис. 9.9. Микросхема АЛУ К155ИП3

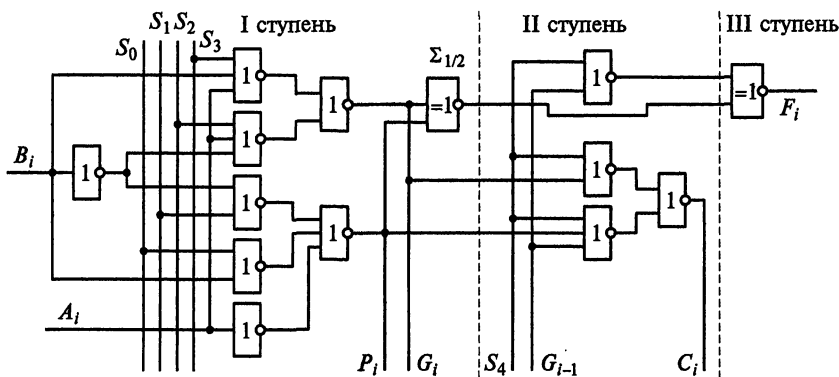


Рис. 9.10. Принципиальная схема АЛУ (вычислительная часть)

ческие операции, при $S_4 = 1$ — арифметические. При выполнении логических операций переносы не распространяются. На III ступени формируется полная сумма из полусумм, формируемых на I ступени, и переносов, формируемых на II ступени.

В секционных модулях операционного устройства АЛУ имеет ограниченную разрядность, равную 2, 4, 8, 12, 16. Объединение нескольких подобных модулей в операционном устройстве требует организации связей по переносам АЛУ. Быстродействие расширенного по разрядности АЛУ в большей степени зависит от способа организации переноса внутри разрядно-секционированного модуля и между модулями. Внутри модуля могут быть предложены следующие способы переноса: последовательный сквозной, параллельный, групповой.

При последовательном сквозном переносе во II ступени АЛУ для каждого i -го разряда формируется сигнал переноса на основании функций генерации и передачи переноса и сигнала переноса в предыдущем младшем разряде в соответствии с выражением $C_i = PC_{i-1} \vee G_i$. При этом время распространения переноса $t = nt_n$, где n — число разрядов в модуле; t_n — время распространения переноса через один разряд.

При параллельном переносе во II ступени формируются параллельно все сигналы переноса для каждого разряда на основании функций генерации и передачи переноса в предыдущих разрядах в соответствии с (9.1). Такая организация сигнала переноса обеспечивает лучшее быстродействие, но требует аппаратных затрат при большом числе разрядов в модуле.

При групповом переносе вся схема разделяется на равные по числу разрядов группы. В каждой k -й группе организуется параллельный перенос (с учетом переноса C_{k-1} из предыдущей группы) и формируется перенос C_k из данной группы в следующую по старшинству $(k+1)$ -ю группу. Это более гибкая схема переноса с учетом факторов быстродействия и аппаратных затрат.

Между модулями обычно используют два способа переноса — последовательный сквозной и групповой. При последовательном сквозном переносе между модулями реализуют параллельный или групповой перенос внутри модулей. При групповом переносе в каждом модуле формируются функции генерации и передачи переноса из модуля. Соответствующие им сигналы поступают во внешний узел ускоренного параллельного переноса, как показано на рис. 9.11.

На основании этих функций в узле ускоренного переноса формируются входные переносы для каждого модуля. Последовательный сквозной перенос между модулями применим при малом числе связанных модулей. При числе модулей больше четырех целесообразен групповой способ с внешним узлом параллельного переноса.

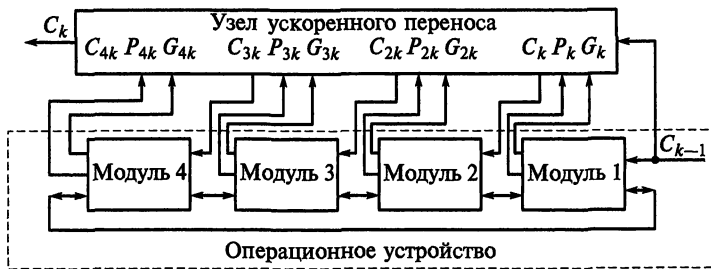


Рис. 9.11. Схема организации ускоренного переноса

Структура операционного устройства зависит от числа внешних магистралей и организации обмена информацией по ним, а также от организации внутренних магистралей и порядка обмена информацией между блоками операционного устройства.

Операционные устройства, реализованные в составе однокристалльных МП БИС, отличаются фиксированной разрядностью и системой команд. Ограничения по числу информационных магистралей и внешних контактов приводят к необходимости организовать последовательный вид передачи информации, при котором по одной информационной магистрали последовательно осуществляется обмен информацией между всеми внутренними узлами БИС и внешними магистралями.

Контрольные вопросы

1. Каково назначение АЛУ?
2. Что такое комбинационная схема?
3. Что такое конечный автомат?
4. Назовите возможные формы представления конечного автомата.
5. В чем заключается работа АЛУ?
6. Как осуществляется выбор той или иной операции АЛУ?
7. Как выполняется групповой перенос?

УПРАВЛЕНИЕ ПРОЦЕССОМ ОБРАБОТКИ ИНФОРМАЦИИ

10.1. Назначение устройства управления

Процессор является главной частью цифровой ЭВМ не только потому, что именно в нем выполняется сам процесс обработки информации, но и потому, что он управляет в ходе этого процесса работой других частей машины — в первую очередь, устройствами ввода и вывода и запоминающими. Самому процессору также требуется управление.

Процессор является конечным автоматом и в общем случае его можно представить в виде некоторого цифрового устройства, состоящего из двух частей: операционной и управляющей (рис. 10.1). В операционной части совершаются элементарные действия по обработке информации:

- запись слов в регистры;
- передача слов из регистра в регистр;
- сдвиг содержимого регистров влево или вправо;
- определение состояния регистров;
- инвертирование содержимого регистров;
- логические операции при поразрядном сравнении содержимого регистров и т. д.

Все эти операции производятся под воздействием сигналов управляющей части процессора и синхронизируются тактовыми сигналами. Элементарные преобразования информации, выполняемые в течение одного такта сигналов синхронизации, называются *микрооперациями*. В течение одного такта сигналов синхронизации могут

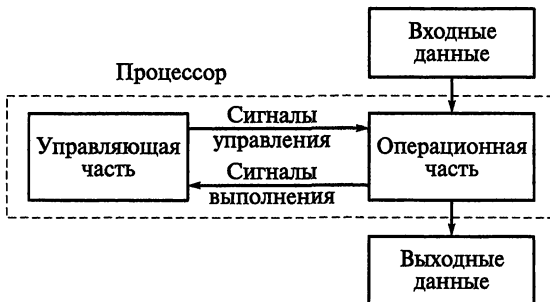


Рис. 10.1. Схема взаимодействия операционной и управляющей частей процессора

выполняться несколько микроопераций (в разных элементах операционной части). Совокупность микроопераций, выполняемых одновременно в течение одного такта, называется *микрокомандой*, а весь набор различных микрокоманд — *микропрограммой*. Поскольку управляющее цифровое устройство определяет всю последовательность выполнения микрокоманд, т. е. микропрограмму, оно называется *микропрограммным автоматом*. Для выработки управляющих сигналов при выполнении некоторых микрокоманд такому автомату могут потребоваться сигналы о состоянии операционной части или внешние сигналы.

Для построения микропрограммного автомата используются принципы схемной (аппаратной) и программируемой логики, а методы управления процессором подразделяются на *аппаратные* и *программные*.

На верхнем уровне системы управления (решение задач и функционирование операционной системы) используется программный метод. На нижнем уровне (например, ввод сигналов с клавиатуры, кодирование аналоговых сигналов) часто используется аппаратный метод.

Для фиксации алгоритмов управления существуют два основных способа:

- 1) с помощью управляющих автоматов;
- 2) с помощью управляющих кодов в запоминающем устройстве.

Оба способа в конечном счете обеспечивают образование необходимой последовательности микрокомандных сигналов, т. е. микропрограмм. В этом смысле указанные способы равнозначны, однако конструктивные отличия обуславливают различные области их применения. Первый способ, называемый *аппаратным*, применяется, как правило, во всех ЭВМ. Он наиболее эффективен для управления несложными, но часто встречающимися операциями. Второй способ, называемый обычно *микропрограммным*, получает все большее распространение и особенно эффективен для управления сложными операциями типа встроенных процедур. Обычно оба этих способа сочетаются, а распределение управляющих воздействий при каждом из них выбирают таким образом, чтобы обеспечить высокое быстродействие и оптимальную организацию вычислительного процесса.

10.2. Аппаратное управление

Аппаратное управление осуществляется последовательностным цифровым устройством (ПЦУ), построенным на принципах схемной логики. В соответствии с управляющими сигналами ПЦУ выполняются микропрограммы, поэтому его можно назвать микропрограммным автоматом. Для каждой операции в ПЦУ имеется свой набор логических схем, вырабатывающих определенные уп-

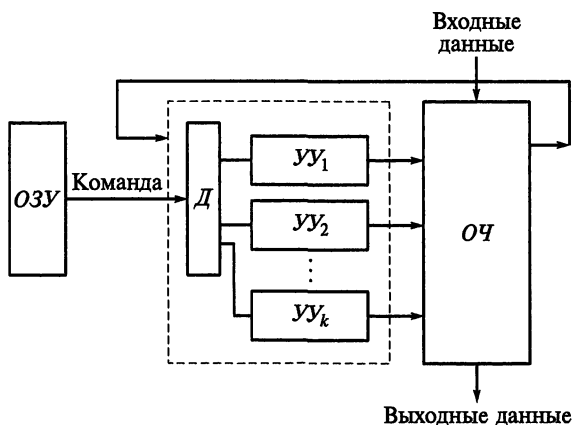


Рис. 10.2. Схема процессора с жесткой логикой

равляющие сигналы для выполнения микроопераций в заданные моменты времени. При указанном построении управляющего ПЦУ микрооперации осуществляются за счет однажды соединенных между собой логических схем, поэтому такие устройства называются *процессорами с жесткой логикой управления*. Это означает, что для процессора существует определенный фиксированный набор команд по числу операций и неизменная структура связей между узлами, обеспечивающая заданную последовательность выполнения каждой операции (рис. 10.2). В управляющей части процессора имеется ряд узлов для выполнения различных операций на одном и том же оборудовании операционной части *ОЧ* процессора. Каждой операции из фиксированного для такого процессора набора (на рис. 10.2 всего k операций) соответствует определенная команда, которая поступает из оперативного запоминающего устройства *ОЗУ*. В управляющей части процессора код этой команды с помощью дешифратора команд *Д* преобразуется в сигнал на определенном выходе, который включает в работу соответствующий узел управления $УУ_i$ (всего их k — столько же, сколько и операций) процессом выполнения данной операции.

На основе рассмотренной схемы может быть создан и микропроцессор, т. е. процессор в виде интегральной схемы. Реализация принципа схемной логики в МП означает, что после его изготовления практически невозможно изменить набор команд и операций. Это приводит к узкой специализации изделия и снижает серийность выпуска, а следовательно, стоимость такого МП будет большой.

В состав управляющего устройства входят следующие основные узлы:

- регистр команд;
- счетчик команд;

- дешифратор операций;
- адресный сумматор;
- индексные регистры;
- шины адресов, команд и данных.

Регистр команд обеспечивает хранение кода команды. Часть разрядов регистра команд (регистр кода операции) предназначена для хранения кода выполняемой операции, остальные разряды (для хранения кодов адресов операндов) связаны с регистром адреса запоминающего устройства. Они могут быть связаны также со счетчиком команд и другими устройствами ЦВМ в зависимости от ее структуры. Счетчик команд обеспечивает хранение кодов адресов команд, поступающих из ОЗУ на регистр команд, и осуществляет управление переходом к выполнению следующей команды в соответствии с программой вычислений. С регистром кода операции связан дешифратор операций, число выходных шин которого равно числу операций ЦВМ. Каждой операции соответствует своя временная последовательность управляющих сигналов, реализующая необходимую для выполнения данной операции последовательность микроопераций.

На рис. 10.3 показана логическая структурная схема однокристалльного МП. Микропроцессор имеет магистральную внутреннюю структуру. Это означает, что все его блоки и узлы соединены между собой с помощью общей информационной магистрали. Эта магистраль, представляющая собой большое число проводов, по которым передаются необходимые сигналы, называется магистральной шиной. Она состоит из шины входных данных $ШД_1$, шины выходных данных $ШД_2$, шины внутренних информационных данных $ШД_{вн}$ и шины управляющих сигналов $ШУ_{вн}$. Внешние подсоединения МП образуют информационные магистрали: однонаправленную шину адресов $ША$; двунаправлен-

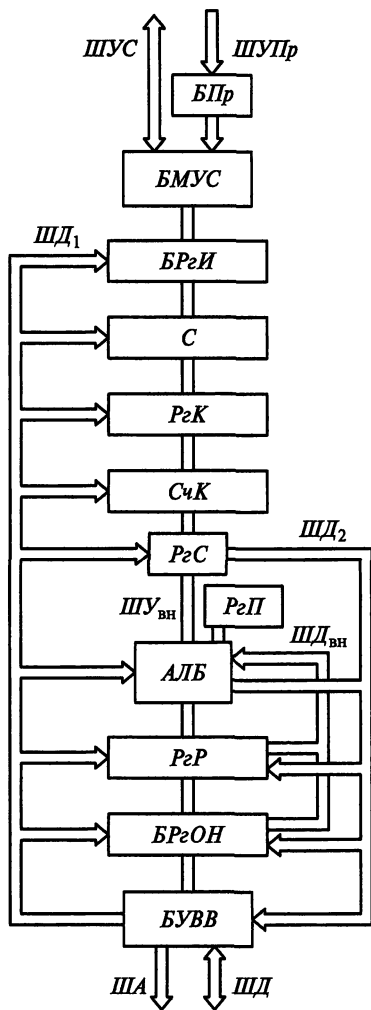


Рис. 10.3. Логическая структурная схема однокристалльного микропроцессора

ную шину данных *ШД*; двунаправленную шину сигналов системного управления *ШУС*; однонаправленную шину сигналов запроса на прерывание процессора *ШУПр*.

Основные блоки МП:

- арифметико-логический *АЛБ*;
- управления вводом и выводом информации *БУВВ* с внутренними буферными регистрами для временного хранения вводимых и выводимых кодов;
- прерываний процессора *БПр*;
- местного управления и синхронизации *БМУС*.

Арифметико-логический блок *АЛБ* связан с регистром триггеров признаков *Р₂П*, число триггеров в котором может изменяться в широких пределах (в типичном случае *Р₂П* содержит четыре разряда для хранения сигналов переполнения, знака, переноса и нулевого результата).

Для системного управления кроме сигналов блока *БМУС* используются также сигналы регистра кода состояния микропроцессора *Р₂С*.

В МП может быть один или несколько регистров результата *Р₂Р* (накапливающих регистров или аккумуляторов). Блок универсальных регистров общего назначения *БР₂ОН*, или регистровое сверхоперативное запоминающее устройство, расширяет возможности МП по хранению данных и управляющей информации. Счетчик команд *С₄К*, регистр команд *Р₂К*, блок индексных регистров *БР₂И* позволяют организовать программное управление, переходы в программе и индексирование команд. При этом сама индексация может происходить в *АЛБ*. С помощью схемы инкремента—декремента (схемы увеличения или уменьшения содержимого индексного регистра на единицу) содержимое индексных регистров может изменяться на «проходе», т. е. при пересылке значения индекса из индексного регистра в *АЛБ* и обратно. Это экономит время работы *АЛБ* при индексации команд.

Для работы с подпрограммами может быть организован стек *С*. Если он расположен в *ОЗУ*, то достаточно иметь только регистр—указатель стека. В качестве последнего, а также счетчика команд *С₄К* и индексных регистров могут быть использованы универсальные регистры блока *БР₂ОН*, что позволяет программным способом гибко использовать регистровые ресурсы процессора и уменьшать общее число элементов и связей в кристалле МП за счет исключения таких специализированных ресурсов МП, как регистр—указатель стека, *БР₂И*, *С₄К* и др.

Для ускорения обработки прерываний и перехода к подпрограммам стек может быть использован в специальных регистрах внутри кристалла МП. Но при этом глубина стека ограничена.

К управляющей части МП следует отнести: *БУВВ*; *БПр*; *БМУС* с *Р₂С*; *С₄К*; *Р₂К*; *БР₂И*.

10.3. Программное управление

При программном управлении некоторые функции управляющей части процессора реализуются аппаратным путем. В частности, в большинстве ЭВМ используется естественный порядок следования команд, при котором применяется счетчик команд. В начале работы по данной программе в счетчик засылается адрес первой команды этой программы, а при исполнении каждой очередной команды содержимое счетчика возрастает на единицу. Исполнение команд условного или безусловного перехода вызывает замену содержимого счетчика. В него засылается первый адрес новой программной последовательности. Суть принципа программируемой логики заключается в том, что при выполнении новой операции (отличающейся от предыдущей) используются прежние логические элементы, но изменяются функциональные связи между ними.

При реализации этого принципа кодовые комбинации управляющих сигналов представляются в виде кодов микрокоманд, которые сохраняются в управляющей памяти УП (рис. 10.4, а). При выполнении некоторой операции из УП выбирается микрокоманда и выдается в виде совокупности управляющих сигналов в операционную часть ОЧ. Для каждой операции в УП имеется своя микропрограмма. В соответствии с содержимым счетчика команд из оперативной памяти выбирается команда и по ее коду определяется соответствующая ей микропрограмма в УП. Микрокоманды найденной микропрограммы последовательно считываются и подаются в ОЧ. В результате реализуется операция, определяемая данной командой. Такой способ реализации операций получил, как уже указывалось в подразд. 10.1, название «микропрограммный», а процессор с устройством управления, основанном на этом принципе, называется *процессором с программируемой логикой*.

Микрокоманда (МК) содержит три поля (рис. 10.4, б): адреса, условных переходов, управляющих сигналов. По содержанию поля адреса определяется адрес следующей МК. В поле условных переходов указывается наличие безусловного или условного перехода, а в случае

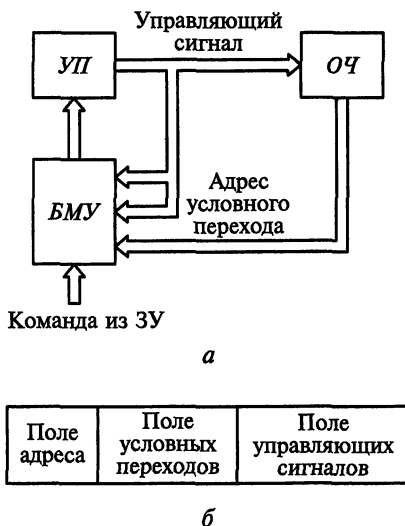


Рис. 10.4. Процессор с программируемой логикой:
а — структура устройств; б — структура команд

условного перехода отмечаются условия определения адреса очередной МК. Поле управляющих сигналов МК служит для организации функционирования *ОЧ* (см. рис. 10.4, *а*). Поступающая из ЗУ команда используется для определения с помощью блока микропрограммного управления *БМУ* адреса первой МК той микропрограммы, которая реализует заданную командой операцию. Адреса последующих МК определяются *БМУ* следующим образом. Поле адреса МК содержит адрес очередной МК. В случае условного перехода один из разрядов поля условных переходов отводится для указания вида перехода (например, 0 — безусловный переход, 1 — условный переход). Для каждого условия отводится разряд, определяющий участие данного условия в определении адреса. В зависимости от условия образуются два различающихся младшим разрядом адреса и очередная МК считывается из одной или другой ячейки *УП*. В результате получается разветвление на два направления. Таким образом, МК может быть разбита на две. Микрокоманда микропрограммного управления определяет функционирование *БМУ* при определении адреса очередной МК по полю адреса и полю условных переходов.

Использование принципа программируемой логики при построении управляющего устройства может привести к снижению быстродействия МП из-за увеличения числа тактов выполнения микропрограммы. Достоинство такой организации управления заключается в возможности гибкого изменения набора команд в МП с помощью изменения совокупности микропрограмм, реализующих эти команды.

Программа записывается в ЗУ в виде последовательности команд. Каждая команда определяет вид операции, исполняемой в данном цикле работы, адреса операндов, участвующих в операции, место расположения результата операции, адрес расположения следующей команды. При малой разрядности МП трудно задать такую обширную информацию с помощью только одного слова. Проблемы выбора формата команд и кодирования полей команд МП имеют особое значение. Гибкость МП и его эффективность определяются числом команд и полнотой системы команд, средствами и способами адресации, возможностями организации разветвленных вычислительных процессов. С увеличением разрядности команды растут и возможности МП. Ограниченная разрядность команды создает существенные трудности в размещении информации о ходе операции и методе адресации данных. Для преодоления этих трудностей в систему команд вводятся операции с удвоенной разрядностью, а также команды с переменной разрядностью.

Кроме поля кода операции и кодов адресов операндов команда должна содержать поле признаков с указаниями способов адресации. Способы адресации определяют механизм формирования

прямого адреса памяти по полю адреса и полю признаков адресации. Гибкость системы команд в значительной мере определяется разнообразием способов адресации. Команды различают по функциональному назначению, числу адресов, способу кодирования команд, длине команды, способу адресации.

По функциональному назначению различают команды передачи данных, обработки данных, передачи управления и дополнительные команды. Команды передачи данных включают в себя подгруппы команд передачи кодов между регистрами МП, пересылки кодов между МП и ОЗУ, передачи кодов между МП и внешними устройствами. Команды обработки данных подразделяются на арифметические, логические и команды сдвига. Команды передачи управления используются для изменения естественного порядка следования команд и организации циклических участков в программах. Среди них выделяются команды безусловного и условного переходов. Дополнительные команды используются для задания останова программы, начальной установки аппаратных средств, реализации ожидания.

По числу адресов различают нуль-адресные, одноадресные, двухадресные и многоадресные команды.

По способу кодирования различают команды с фиксированным и расширяющимся полем кода операций.

По длине различают команды длиной в одно, два и три слова.

Способ адресации в значительной мере влияет на эффективность обработки информации в МП. Для преодоления ограничений из-за малой разрядности кодов команд используются различные способы адресации, которые позволяют задать полный адрес памяти меньшим числом бит, вычислять адреса во время обработки, вычислять адреса данных относительно позиции команды таким образом, чтобы можно было загружать программу в любую область памяти без изменений адресов в программе.

Способы адресации можно разделить на две группы. К первой группе относятся способы, в которых исполнительный адрес определяется одним значением кода в команде. Таковыми являются прямая регистровая, косвенная регистровая, непосредственная, автоинкрементная и автодекрементная адресации.

В случае прямой адресации код адреса в команде является исполнительным адресом обращения к памяти. При *прямой регистровой* адресации обрабатываемое слово (операнд) содержится в одном из регистров МП. При *косвенной регистровой* адресации косвенный адрес извлекается из внутреннего регистра МП. При *непосредственной* адресации операнд задается в команде.

Автоинкрементная адресация основана на вычислении исполнительного адреса так же, как и при косвенной регистровой ад-

ресации, с последующим увеличением содержимого регистра на некоторую константу.

При *автодекрементной* адресации сначала из содержимого регистра вычитается константа, затем полученный результат используется в качестве исполнительного адреса. Совместное применение автоинкрементной и автодекрементной адресаций обеспечивает использование любого регистра в качестве стека.

Ко второй группе относятся такие способы адресации, в которых используется содержимое адресной части команды и нескольких регистров для формирования исполнительного адреса. Такими являются страничная, индексная и относительная адресации.

При *страничной* адресации память разбивается на ряд страниц одинаковой длины. Адресация страниц осуществляется с помощью регистра страниц, а адресация ячеек памяти внутри страниц — с помощью адреса в команде. Номера всех страниц могут находиться в таблице страниц, которая представляет собой нулевую страницу.

Индексная адресация используется при обращении к массивам слов и таблиц. Для образования исполнительного адреса к адресной части команды прибавляется смещение (индекс) из регистра, называемого индексным. Содержимое индексного регистра можно изменять; это позволяет изменять исполнительный адрес без модификации адресной части команды.

При *относительной* адресации исполнительный адрес образуется сложением базового адреса с адресом команды. В качестве базового адреса используется содержимое программного счетчика. Такая адресация позволяет создавать свободно перемещаемые в памяти программы.

10.4. Алгоритм управления

Алгоритмы играют важную роль в вычислительной технике. По существу ЭВМ не решает какие бы то ни было задачи, она только выполняет алгоритм, предложенный ей в виде программы. Первоначальное понятие алгоритма возникло в математике при поисках общих принципов решения одинаковых по типу математических задач.

Понятие алгоритма можно сформулировать следующим образом. *Алгоритм* — это точное предписание, представляющее собой набор указаний (шагов) о выполнении в определенном порядке некоторой системы операций, позволяющее решать совокупность задач определенного класса.

Любому алгоритму присущи три основных свойства: определенность, результативность и массовость. Под определенностью подразумевается однозначность указаний алгоритма. Многократ-

ное применение этих указаний для однотипных исходных данных приводит к идентичным по своему характеру результатам.

Результативность означает, что при использовании алгоритма в условиях, для которых он разработан, искомый результат возникает после конечного числа сравнительно несложных шагов.

Свойство массовости заключается в том, что значимость алгоритма тем выше, чем шире возможность его применения для решения любой задачи, относящейся к задачам определенного класса.

Рассмотрим в качестве примера алгоритм перевода целого десятичного числа N в двоичное (см. подразд. 5.2). Этот алгоритм можно записать в виде последовательности действий (шагов).

Шаг 1. Разделить десятичное число N на два. Полученный при этом остаток представляет собой цифру первого самого младшего разряда двоичного числа.

Шаг 2. Разделить частное от предыдущего деления на два. Вновь полученный при этом остаток — это цифра следующего разряда двоичного числа.

Шаг 3. Если вновь полученное частное больше или равно двум, то перейти к шагу 2, если оно меньше двух — вычисление закончено.

Здесь шаги 2 и 3 обуславливают возникновение цикла, который повторяется, пока выполняется приведенное в шаге 3 условие о том, что полученное частное больше или равно двум. Невыполнение этого условия определяет выход из цикла и переход к продолжению программы.

Запись алгоритма на языке машины представляет собой программу решения задачи. Для относительно сложных задач переходить сразу от алгоритма к программе на языке машины затруднительно. При таком непосредственном переходе теряется связь между отдельными частями программы, становятся трудноразличимыми принципиальные и несущественные детали программы. В этих условиях легко возникают ошибки, отыскание и ликвидация которых весьма сложны и требуют значительного времени.

На первых этапах разработки программы целесообразно применять схемы этих программ. Схема программы — это укрупненное описание программы, части которой изображаются в виде отдельных блоков, представляющих собой соответствующие геометрические фигуры. В состав этих фигур входят прямоугольники, ромбы, овалы, круги и т. п. Условные графические обозначения (символы) в схемах алгоритмов и программ отображают основные операции обработки данных и программирования для вычислительных машин. Эти обозначения следует выполнять по ГОСТ 19.003—80, который соответствует международному стандарту. Внутри каждого блока программист описывает содержание той части программы, которой соответствует этот блок. Связи между блоками показыва-

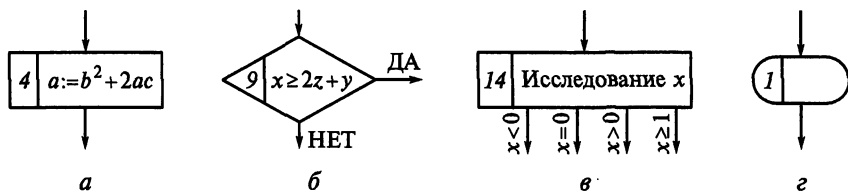


Рис. 10.5. Условные обозначения, используемые в схеме алгоритма:

a — вычислительный блок; *b* — логический блок с двумя выходами; *v* — логический блок с тремя и более выходами; *z* — вспомогательный блок

ют линиями, означающими передачу управления. Условия этой передачи отображаются соответствующими записями.

По целевому назначению блоки можно условно подразделить на три группы.

Вычислительный блок (рис. 10.5, *a*) предусматривает непосредственное вычисление функций или выполнение арифметических операций по обработке информации. Блоки этого типа обычно имеют вид прямоугольника, в левой части которого записывается его порядковый номер.

Логический блок соответствует участку программы, на котором вычислительный процесс разветвляется в зависимости от выполнения или невыполнения записанного условия. При исследовании в логическом блоке двух возможных вариантов такой блок изображают в виде ромба, одна из вершин которого используется для входа, а две другие — для выхода из блока (рис. 10.5, *b*). При исследовании более чем двух вариантов число выходов соответственно увеличивается и блок в этом случае изображают в виде прямоугольника (рис. 10.5, *v*).

Вспомогательный блок (рис. 10.5, *z*) позволяет представить операции по подготовке других блоков к выполнению вычислительного процесса. К ним относятся, например, операции по засылке данных, переадресации и т.п. К числу вспомогательных блоков можно отнести и блоки, представляющие операции ввода и вывода информации, начала и конца программы и т.п.

Схема программы представляет собой вспомогательный материал, используемый программистом на промежуточном этапе разработки программы. Программист сам определяет, какие этапы вычислительного процесса включать в тот или иной блок, какие этапы вычисления описывать в блоках детально, а какие в укрупненном масштабе. Для задач, значительных по своим объемам и сложности, целесообразно сначала разработать укрупненную схему всего вычислительного процесса, а затем для ее отдельных блоков выполнить детальные схемы.

При разработке программы часто возникает ситуация, когда в зависимости от полученных промежуточных результатов необхо-

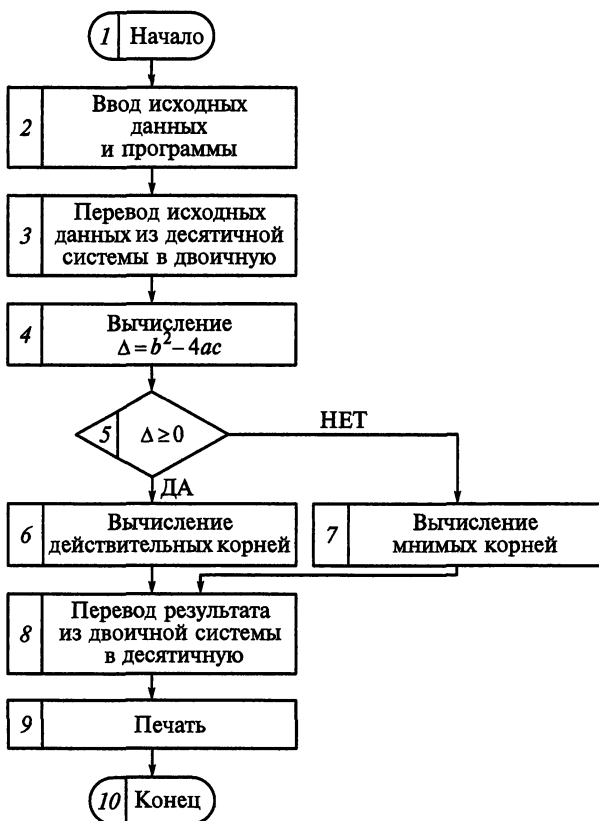


Рис. 10.6. Схема алгоритма решения квадратного уравнения

можно вести дальнейшие вычисления либо по одним, либо по другим формулам. В таких случаях вычислительный процесс может следовать по разным направлениям, или разным ветвям. При каждом разветвлении дальнейший ход вычисления определяется на основе проверки соответствующего условия, например $a \geq 0$; $a < b$; $a \neq b$ и т. п. Выполнение проверяемого условия обуславливает одно направление программы, невыполнение — другое.

Рассмотрим пример построения блок-схемы разветвляющегося вычислительного процесса применительно к алгоритму решения квадратного уравнения $ax^2 + bx + c = 0$.

Ветвь, по которой будет осуществляться вычислительный процесс, устанавливается на основании проверки знака дискриминанта, т. е. подкоренной величины $\Delta = b^2 - 4ac$. При $\Delta \geq 0$ уравнение характеризуется двумя действительными корнями, которые вычисляются по формуле $x_{1,2} = (-b \pm \sqrt{b^2 - 4ac}) / 2a$. При $\Delta < 0$ урав-

нение обладает двумя комплексными сопряженными корнями $x_{1,2} = A \pm jB$, где $A = -b / 2a$; $B = \sqrt{4ac - b^2} / 2a$.

Схема алгоритма решения квадратного уравнения приведена на рис. 10.6. В этой схеме блок 5 — логический блок, проверяющий выполнение условия $\Delta \geq 0$. Если это условие выполняется, то происходит переход к блоку 6, которым определяется одна ветвь процесса. При невыполнении этого условия, т. е. при $\Delta < 0$, осуществляется переход к блоку 7, что соответствует другой ветви процесса. После вычислений, предусмотренных любым из этих блоков, программа продолжается с блока 8.

Переход к одному или другому блоку на основании проверки выполнения логического условия называется *условным*. В рассматриваемой схеме условным является переход от блока 5 к блоку 6 или 7. Переход от одного блока к другому без проверки каких-либо условий называется *безусловным*. К таким переходам относятся, например, переходы от блока 4 к блоку 5, от блоков 6 и 7 к блоку 8.

Контрольные вопросы

1. Какие функции выполняет устройство управления?
2. В чем разница между аппаратным и программным управлением?
3. Какие существуют способы адресации?
4. Как различают команды по функциональному назначению?
5. Дайте словесный алгоритм выполнения домашнего задания по какому-нибудь предмету, т. е. опишите последовательность действий.
6. Что такое схема алгоритма?

РАБОТА МИКРОПРОЦЕССОРА

11.1. Система команд микропроцессора

Принципиальным достоинством микропроцессора является его программируемость. Это означает, что, подавая на вход МП команды, можно обеспечить нужную последовательность операций, т.е. реализацию определенного алгоритма. Алгоритм решаемой задачи может быть сколь угодно сложным, необходимо лишь, чтобы он был разбит на шаги в соответствии с системой команд МП. Наличие или отсутствие какой-либо команды или группы команд может существенно повлиять на выбор МП для конкретного применения.

Классификация команд МП представлена на рис. 11.1. По числу ячеек памяти, необходимых для размещения одной команды, различают команды длиной в одно, два или три слова. Команды длиной в два и три слова требуют для выборки соответственно два и три цикла обращения к памяти. В принципе возможны команды и большей длины, но практически они не используются.

Во многих случаях, в частности при сравнении МП со сходной архитектурой, оказывается полезной классификация команд в соответствии с архитектурными характеристиками МП. По этому признаку можно выделить следующие команды:

- изменения содержимого ячеек памяти;
- изменения содержимого аккумулятора;
- изменения содержимого индексного регистра;
- выполнения операций со стеком;
- выполнения операций в АЛУ;

КОП	Номер регистра
-----	----------------

a

КОП	Вид адресации	Номер регистра	Адрес
-----	---------------	----------------	-------

б

КОП	Номер $Pz1$	Номер $Pz2$
-----	-------------	-------------

в

КОП	Номер регистра	Константа
-----	----------------	-----------

г

Рис. 11.1. Форматы команд микропроцессора:

a — изменения состояния регистра; *б* — выполнения операций над содержимым регистров; *в* — обращения к памяти; *г* — непосредственной адресации

- передачи управления;
- изменения содержимого регистра состояния (или регистра признаков);

- ввода — вывода.

С функциональной точки зрения команды разделяются на три большие группы:

- 1) передачи данных;
- 2) управления данными;
- 3) обработки данных.

Рассмотрим подробнее основные команды, применяемые в МП, пользуясь классификацией по функциональным признакам. Названия команд будем обозначать русскими словами, указывающими на смысл выполняемых операций.

Команды передачи данных. Команды этой группы обеспечивают простую пересылку информации без выполнения каких-либо операций обработки. Они подразделяются на команды, связанные с обращением к памяти, команды, связанные с обращением к регистрам, и команды ввода — вывода. Обычно программы решения задач, исходные данные и результаты операций хранятся в оперативной запоминающем устройстве (ОЗУ). Каждому элементу информации в ОЗУ отводится определенное место, которое имеет адрес.

В большинстве ЭВМ минимальной единицей адресуемой информации является байт, в связи с чем объем ОЗУ исчисляется в байтах. Достаточно часто для операндов одного байта мало, поэтому размер ячейки памяти выбирают большим, например четыре байта. Для эффективного использования всего адресуемого пространства ОЗУ МП автоматически выбирает из памяти информацию различного размера: слова (4 байт), двойные слова (8 байт), полуслова (2 байт), 1 байт, а в некоторых случаях даже отдельные биты. Используемый в командах размер информации может задаваться специальным словом в команде или в неявном виде кодом операции. При этом часть младших разрядов адреса определяет номер байта.

К командам, связанным с обращением к памяти, относятся:

- **ЗАГРУЗИТЬ (ПРОЧИТАТЬ)**, по которой содержимое одной из ячеек памяти засылается в регистр;
- **ЗАПОМНИТЬ (ЗАПИСАТЬ)**, по которой содержимое регистра засылается в ячейку памяти.

В этих командах, связанных с пересылкой байта или слова, должны указываться номер конкретного регистра, адрес ячейки памяти и, если необходимо, номер модуля ЗУ.

Команды, связанные с обращением к регистрам, должны указывать номер источника информации и номер регистра результата. В эту подгруппу команд передачи данных входят команды:

• **ПЕРЕСЛАТЬ**, по которой содержимое одного регистра пересылается в другой;

• **ЗАГРУЗИТЬ НЕПОСРЕДСТВЕННО**, по которой в регистр записывается константа, указанная в коде команды. Эта команда часто используется для сброса регистра в 0 или установки его в 1.

К командам ввода—вывода относятся:

• **ВВОД**, по которой содержимое устройства ввода засылается во внутренний регистр МП;

• **ВЫВОД**, по которой содержимое внутреннего регистра МП (обычно аккумулятора) пересылается в устройство вывода.

Эти команды должны следовать за командами, подготавливающими операции ввода—вывода, например за командой переключения устройства ввода—вывода (УВВ) в режим приема информации.

Команды управления. Эти команды, часто называемые командами перехода, позволяют выполнять различные действия в соответствии со значением внешних сигналов или выработанных внутри системы условий. При этом нарушается естественный порядок следования команд, при котором после выполнения очередной команды выбирается команда, расположенная в следующей по порядку ячейке памяти. Все команды управления подразделяются на команды безусловного и условного перехода.

К командам безусловного перехода относятся:

• **БЕЗУСЛОВНЫЙ ПЕРЕХОД (БП)**, по которой в счетчик команд записывается содержимое адресного поля команды БП, т.е. обеспечивается переход в программе по адресу, указанному в команде;

• **ПРОПУСТИТЬ**, по которой пропускается следующая команда программы;

• **БЕЗУСЛОВНЫЙ ПЕРЕХОД С ВОЗВРАТОМ** (переход к подпрограмме), по которой в программный счетчик записывается новое содержимое (адрес первой команды подпрограммы), но в отличие от команды БП в памяти сохраняется состояние счетчика команд и некоторых других регистров. После выполнения подпрограммы по ее последней команде ВОЗВРАТ восстанавливается содержимое счетчика команд и всех регистров.

Команды условного перехода требуют проверки состояния какого-либо разряда регистра, флагового триггера или другого параметра. От результата проверки зависит, будет выполняться переход или нет. Обычно переход выполняется, если результат проверки соответствует указанному в команде условию. В эту подгруппу команд управления входят следующие команды:

• **УСЛОВНЫЙ ПЕРЕХОД (УП)** по адресу. В коде команды УП обязательно указывается проверяемое условие, в качестве которого в МП используются нулевое или ненулевое значение результата, положительный или отрицательный знак результата, нали-

чие или отсутствие сигналов переноса, переполнения и др. При выполнении условия в программный счетчик записывается содержимое адресного поля команды УП, т.е. обеспечивается переход в программе по адресу, указанному в команде. При невыполнении условия управление передается следующей команде программы;

- **УСЛОВНЫЙ ПЕРЕХОД С ВОЗВРАТОМ**, которая отличается от команды **БЕЗУСЛОВНЫЙ ПЕРЕХОД С ВОЗВРАТОМ** тем, что переход к подпрограмме происходит только при выполнении указанного условия;

- **УСЛОВНО ПРОПУСТИТЬ**, по которой проверяется определенное условие и, если оно выполнено, пропускается следующая команда программы;

- **ПРОПУСТИТЬ ПО ФЛАГУ**, используемая в основном для организации взаимодействия МП и УВВ. По этой команде проверяется состояние флагового триггера;

- **СРАВНИТЬ И ПРОПУСТИТЬ**, по которой сравнивается содержимое двух регистров, в случае равенства пропускается следующая команда;

- **ЦИКЛ**, по которой содержимое определенного регистра или ячейки памяти уменьшается (или увеличивается) на единицу и, если оно станет равным нулю, пропускается следующая команда.

Команды перехода не выполняют операции с операндами и предназначены лишь для изменения порядка следования команд. Поэтому вместо сведений о местоположении операндов и результата в них содержатся сведения о местоположении очередной команды.

Обычно в систему команд МП включается еще несколько вспомогательных команд, которые позволяют управлять состоянием регистров или триггеров, влияющих на выполнение условных переходов. К этим командам относятся: **УСТАНОВИТЬ ФЛАГ**; **СБРОСИТЬ ФЛАГ**; **УСТАНОВИТЬ СТАРШИЙ РАЗРЯД АККУМУЛЯТОРА**; **СБРОСИТЬ СТАРШИЙ РАЗРЯД АККУМУЛЯТОРА** и др.

Команды обработки данных. Данные команды подразделяются на арифметические и логические. К арифметическим командам относятся:

- **СЛОЖИТЬ** содержимое двух регистров или регистра и ячейки памяти;

- **ВЫЧЕСТЬ** из содержимого ячейки памяти или регистра содержимое регистра;

- **УВЕЛИЧИТЬ НА 1** (или **ИНКРЕМЕНТ**) содержимое ячейки памяти или регистра, в частности, указателя стека, индексного регистра, аккумулятора;

- **УМЕНЬШИТЬ НА 1** (или **ДЕКРЕМЕНТ**) содержимое ячейки памяти или регистра;

- **СЛОЖИТЬ С УЧЕТОМ ПЕРЕНОСА**, по которой выполняется сложение с учетом состояния триггера переноса, что позволяет легко организовывать обработку чисел большой длины;

- **ВЫЧЕСТЬ С УЧЕТОМ ЗАЕМА**;

- **СДВИГ** содержимого ячейки памяти или регистра (обычно на один разряд).

В подгруппу логических команд входят команды:

- **И** (или **ЛОГИЧЕСКИ УМНОЖИТЬ**), по которой выполняется операция конъюнкции между содержимым двух регистров или ячейки памяти и регистра;

- **ИЛИ** (или **ЛОГИЧЕСКИ СЛОЖИТЬ**), по которой выполняется операция дизъюнкции между содержимым двух регистров или ячейки памяти и регистра;

- **НЕРАВНОЗНАЧНОСТЬ**, по которой производится поразрядное сравнение содержимого двух регистров или ячейки памяти и регистра;

- **ИНВЕРСИЯ** содержимого ячейки памяти или регистра.

Команда МП состоит в общем случае из набора сведений о коде операции, адресов операндов и регистров, участвующих в операции, способов адресации. Для размещения этих сведений выделяется несколько полей, т.е. регистров, состоящих из определенного числа разрядов. Структура таких регистров и правила размещения в них команды называется *форматом* команды. Команды различных типов могут иметь разный формат. Каждый формат определяется кодом операции (КОП), который размещается в первом по порядку поле.

В группу команд изменения состояния регистра входят команды: **УВЕЛИЧИТЬ НА 1**; **УМЕНЬШИТЬ НА 1**; **ИНВЕРСИЯ**; **СБРОСИТЬ В 0**; **УСТАНОВИТЬ В 1**; **СДВИГ** и некоторые другие. Формат команд этой группы кроме поля КОП предусматривает поле, в котором указывается номер регистра (рис. 11.1, а).

В командах выполнения операций над содержимым регистров должны указываться (явно или неявно) номера трех регистров: двух регистров, содержащих операнды, и регистра результата. В МП наиболее распространен формат команд, включающий в себя номера только двух регистров, которые можно обозначить R_1 и R_2 . Один из этих регистров используется для хранения одного из операндов и результата операции. В указанную группу входят следующие команды: **ПЕРЕСЛАТЬ** содержимое R_1 в R_2 ; **СЛОЖИТЬ** содержимое регистров R_1 и R_2 ; **ВЫЧЕСТЬ** содержимое R_1 из R_2 ; **СРАВНИТЬ** содержимое R_1 и R_2 и некоторые другие. Формат команд этой группы должен предусматривать три поля: КОП и номера двух регистров (рис. 11.1, б).

В группу команд, связанных с обращением к памяти, входят команды **ЗАГРУЗИТЬ**, **ЗАПОМНИТЬ** и некоторые

другие. Для этих команд необходимы поля: КОП, вид адресации, номер регистра, адрес ячейки памяти (рис. 11.1, в).

Команды с непосредственной адресацией содержат константу, которая может размещаться в регистре или использоваться в качестве операнда. Для команд этой группы предусмотрены три поля: КОП, номер регистра, константа (рис. 11.1, г).

Число разрядов в регистрах, выделяемых под каждое поле, зависит от конкретного МП, т.е. от его числа регистров общего назначения и их разрядности. Аппаратные ограничения по числу разрядов можно преодолеть формированием двойных и даже тройных слов, что расширяет размеры полей.

11.2. Процедура выполнения команд

Пусть при работе микроЭВМ с клавиатурой и монитором последовательно выполняются три действия (рис. 11.2): нажатие пользователем клавиши *A* клавиатуры; размещение буквы *A* (в кодированной форме) в памяти; воспроизведение буквы *A* на экране монитора. Такая последовательность действий (ввод — размещение — вывод) является типичной. Она многократно повторяется не только при создании с помощью ЭВМ текстовых документов, но и при наборе программ на языках программирования. Более подробная схема на рис. 11.3 позволяет понять, как именно выполняются команды.

Прежде всего рассмотрим содержимое регистров программной памяти с адресами от 100 до 105. В эти регистры предварительно были загружены три команды:

- 1) ВВОД — ввести данные, поступающие из порта 1;

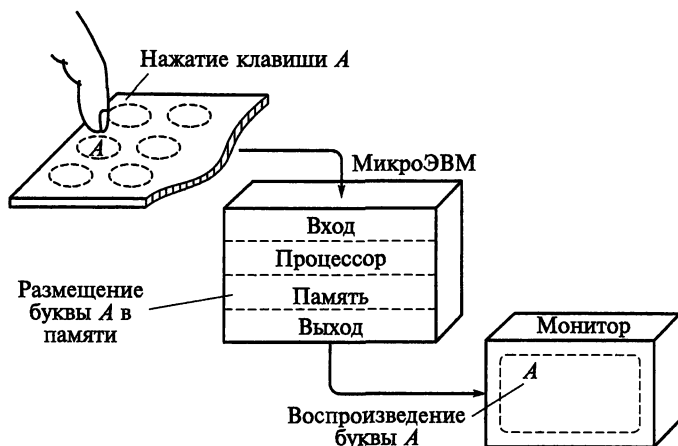


Рис. 11.2. Действия при работе микроЭВМ

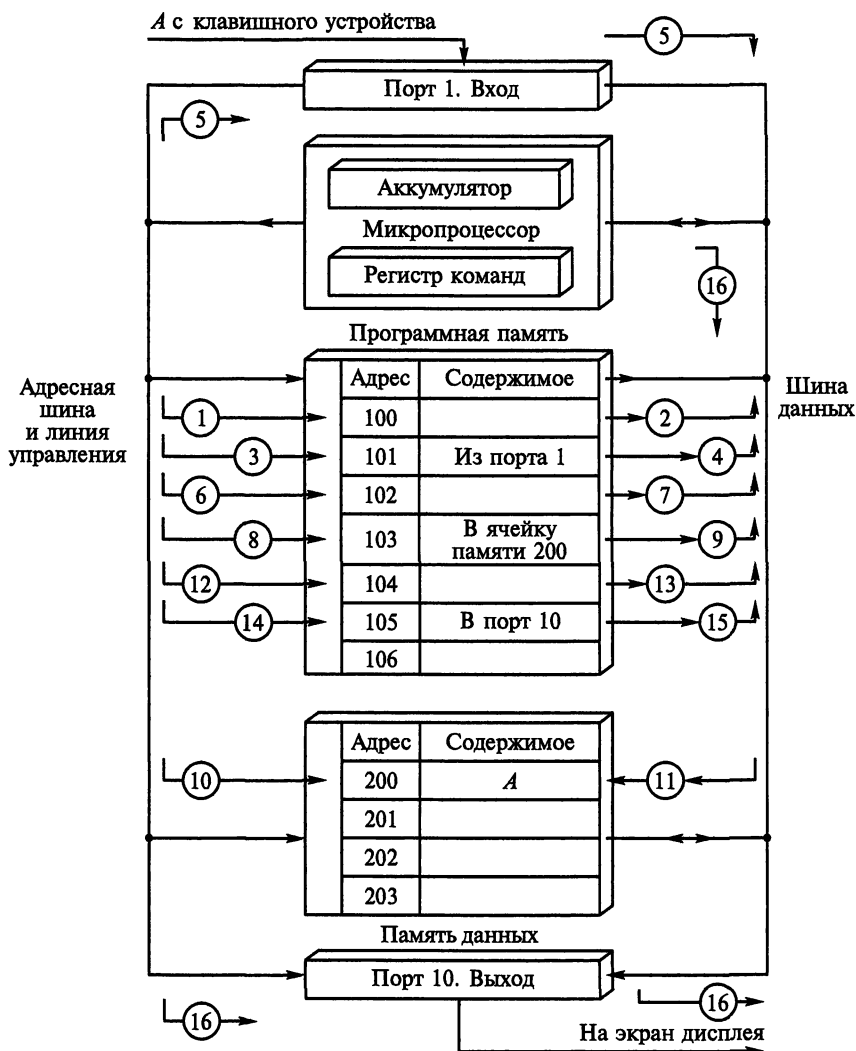


Рис. 11.3. Процедура выполнения команд

2) **ЗАПОМНИТЬ** данные, поступающие из порта 1, т.е. записать их в ячейке памяти данных с адресом 200;

3) **ВЫВОД** — вывести данные через порт вывода 10.

Программа, содержащая три указанные команды, размещается в шести ячейках. Это обусловлено тем, что каждая из этих трех простых команд поделена на две части. Например, первая часть первой команды говорит, что надо выполнить операцию **ВВОД** (ввести данные), а вторая часть указывает, откуда подлежащие

вводу данные поступают (из порта 1). Первая часть — это и есть код операции, а вторая часть — операнд. Код операции ВВОД содержится в ячейке памяти с адресом 100, код операции ЗАПОМНИТЬ — в ячейке 102, код операции ВЫВОД — в ячейке 104.

В МП на рис. 11.3 показаны только аккумулятор и регистр команд. Поскольку никаких арифметических действий эта простая программа не предусматривает, то и нет нужды показывать сумматор. Напомним, что именно МП является центром всех преобразований данных и операций.

Проследим шаг за шагом (их номера указаны на рис. 11.4 в кружках) всю процедуру выполнения трех команд.

Шаг 1. МП подает на адресную шину адрес ячейки памяти 100 и активизирует ввод считывания из программной памяти (считывание данных означает копирование информации из ячейки памяти).

Шаг 2. Программная память выставляет первую команду (ВВОД) на шину данных, а МП принимает эту закодированную информацию. Она помещается в регистр команд МП. Этой команде нужен операнд.

Шаг 3. МП подает на адресную шину адрес ячейки памяти 101. Линия управления готовит вход считывания из программной памяти.

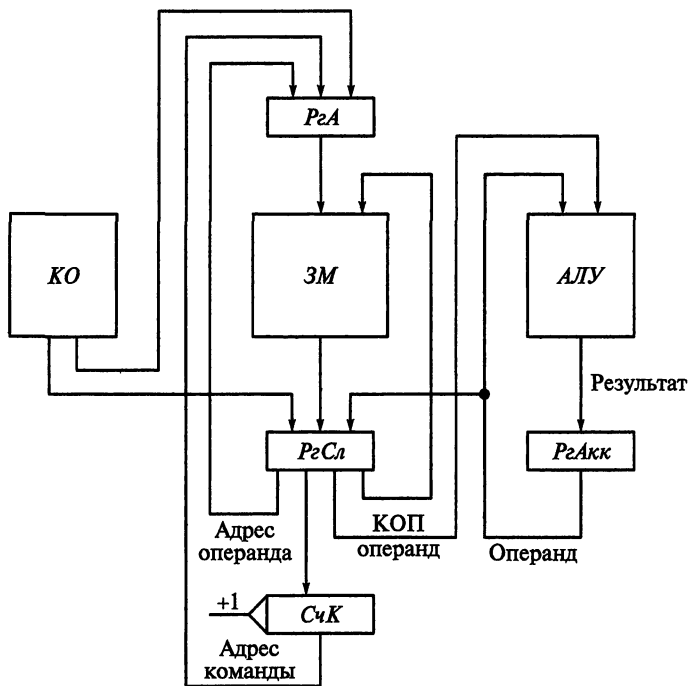


Рис. 11.4. Структурная схема простейшей ЭВМ

Шаг 4. Программная память помещает операнд (изъятый из порта 1) на шину данных. Этот операнд находился в ячейке памяти 101. Кодированное послание (адрес порта 1) берется с шины данных и помещается в регистр команд. МП декодирует полную команду ВВОД, как «Ввести данные, поступающие из порта 1».

Шаг 5. МП побуждает открыть порт 1 посредством адресной шины и линии управления устройством ввода. Кодированная форма *A* передается в аккумулятор, где и размещается.

Первая команда выполнена. Необходимо отметить, что МП все время действует в последовательности извлечение — декодирование — выполнение, т. е. сначала извлекает из программной памяти команду, затем расшифровывает (декодирует) ее и, наконец, выполняет.

Шаг 6. МП подает на адресную шину адрес ячейки памяти 102 и активизирует вход считывания посредством управляющих линий.

Шаг 7. Код команды «ЗАПОМНИТЬ данные» подается на шину данных. МП принимает его, помещает в регистр команд, декодирует и определяет, что нужен операнд.

Шаг 8. МП подает на шину данных адрес ячейки памяти 103 и активизирует вход считывания из программной памяти.

Шаг 9. Код операнда «В ячейку памяти 200» подается из памяти (программы) на шину данных. МП принимает его и помещает в регистр команд.

Таким образом, команда «ЗАПОМНИТЬ данные, расположенные в ячейке памяти 200» полностью извлечена и декодирована. Теперь начинается процесс ее выполнения.

Шаг 10. МП подает на адресную шину адрес ячейки памяти 200 и активизирует вход записи в память (запись означает, что данные вводятся в память).

Шаг 11. МП выдает помещенную в аккумулятор информации (кодированная форма *A*) на шину данных. Это *A* записывается в ячейку памяти 200.

Вторая команда выполнена.

Шаг 12. МП подает на адресную шину адрес ячейки памяти 104 и активизирует вход считывания из памяти.

Шаг 13. Команда «ВЫВОД данных» подается на шину данных. МП принимает ее, помещает в регистр команд, декодирует и устанавливает, что нужен операнд.

Шаг 14. МП подает на адресную шину адрес ячейки памяти 105 и активизирует вход считывания из памяти.

Шаг 15. Код операнда «В порт 10» подается из памяти на шину данных. МП принимает его и помещает в регистр команд.

Шаг 16. МП декодирует полную команду «ВЫВОД данных в порт 10», активизирует порт 10 посредством адресной шины и управляющей линии устройства вывода. Код *A* (находившийся в

аккумуляторе) подается на шину данных и передается портом 10 на видеотерминал.

Большинство микроЭВМ передают информацию описанным способом. Самые существенные различия сосредоточены в элементах ввода и вывода. Может потребоваться больше шагов для осуществления рассмотренных операций. При большом числе разрядов в ячейках памяти и шинах число шагов меньше, поскольку команда извлекается из памяти и передается в регистр команд МП не по частям, а целиком.

Текущий процесс в ЭВМ состоит в последовательном и поэтапном выполнении команд. Этапы могут быть разными, в зависимости от заданной операции, типов операндов, числа адресов и способа их адресации. Но суть работы процессора можно проследить на примере выполнения простейшей вычислительной машинной типовой команды, содержащей двухместную (с двумя операндами) операцию. Будем считать, что простейшая ЭВМ (см. рис. 11.4) состоит из канала обмена КО, процессора и памяти. Процессор содержит арифметико-логическое устройство АЛУ, счетчик команд СчК и регистр-аккумулятор Р₂Акк. Память помимо запоминающей матрицы ЗМ и блоков управления и коммутации (на схеме не показаны) содержит регистр адреса Р₂А и регистр слова Р₂Сл.

На данной ЭВМ выполняются одноадресные команды. Процедура выполнения типовой команды состоит из следующих этапов:

- передача адреса команды (содержимого СчК) в Р₂А;
- чтение команды из памяти с записью в Р₂Сл;
- передача из Р₂Сл первой части команды (кода операции) в АЛУ и второй части команды (адреса операции) в Р₂А;
- чтение операнда из памяти с записью в Р₂Сл и одновременно передача первого операнда из Р₂Акк в АЛУ;
- передача второго операнда из Р₂Сл в АЛУ;
- выполнение операции в АЛУ в соответствии с кодом операции;
- передача результата операции из АЛУ в Р₂Акк и одновременно продвижение СчК на один шаг.

Рассмотренные этапы команды выполняются под воздействием управляющих сигналов, вырабатываемых общим микропрограммным блоком управления. Подобное управление называется *синхронным центральным*. Каждому этапу при этом отводится определенное машинное время, и начало любого из них не зависит от степени выполнения предыдущих этапов (считается, что все они выполнены). Синхронное центральное управление характерно для простейших ЭВМ, в которых все этапы команд выполняются примерно за одинаковое машинное время и нет особого разнообразия между алгоритмами выполнения различных команд.

Однако команды даже простейшей ЭВМ не однотипны, их выполнение занимает разное время. Например, далеко не всегда результат операции в Р₂Акк используется в качестве операнда оче-

редной команды, поэтому в программу перед выполнением операций приходится включать дополнительные команды загрузки из памяти *РзАкк*, а после выполнения операций — дополнительные команды записи в память из *РзАкк*. Необходима также команда перехода, для чего введена дополнительная передача из *РзСл* в *СчК*.

В рассмотренной простейшей ЭВМ используется только прямая адресация оперативной памяти. Косвенная и относительная адресации, обращения к промежуточной сверхоперативной памяти, расширение адресности команд в еще большей степени нарушают однотипность команд. Поэтому в высокопроизводительных ЭВМ используется *асинхронное* управление, при котором переход от одного этапа выполнения команд к другому совершается по сигналам готовности. Это означает, что в ЭВМ отсутствует единый такт работы ее устройств. Выходная информация из устройства передается во входной регистр следующего устройства только тогда, когда передающее устройство полностью сформирует информацию на выходных регистрах, а принимающее устройство освободит входные регистры для приема информации. Асинхронное управление предполагает автономную работу устройств под «своим» (местным) управлением. Передаче между устройствами предшествует проверка признака готовности устройств к передаче и приему информации. Следовательно, необходимы дополнительные аппаратурные и временные затраты, которые компенсируются возможностью эффективного комплектования вычислительных систем разнородными по производительности и временным параметрам устройствами и организацией их параллельной работы в совокупности с шинным и (или) сетевым обменами между устройствами.

11.3. Система прерывания

В подразд. 11.2 рассматривалась процедура выполнения команд, при которой происходил обмен данными между устройствами ввода — вывода, микропроцессором и памятью. В процессе выполнения основной программы (вычислительной или обрабатывающей информацию) МП приходится время от времени отображать ход ее выполнения на мониторе или получать новые указания с клавиатуры. Следовательно, обмен данными между УВВ и МП происходит весьма часто. Для обеспечения такого обмена в ходе выполнения основной программы служит система прерываний.

Система прерываний позволяет УВВ обмениваться данными с МП. Для этого МП должен иметь вход, периодически опрашиваемый управляющей частью МП (обычно после выполнения каждой команды). Сигнал прерывания на этом входе заставляет МП отложить выполнение основной программы и перейти в режим

обслуживания прерывания. Организация системы прерываний требует решения следующих задач:

- идентификация (т. е. определение, опознание) устройства, от которого поступил запрос на прерывание;
- сохранение в памяти состояния активных регистров МП, так как оно может потребоваться после возобновления выполнения основной программы;
- восстановление состояния этих регистров после окончания обслуживания прерывания.

Если в МП поступает несколько сигналов прерывания от различных УВВ, то необходимо провести анализ возникшей ситуации и определить, в какой последовательности должны быть выполнены поступившие запросы на прерывание.

Различают два способа реализации перечисленных функций: программный и аппаратный.

Программная реализация системы прерываний отличается простотой и характеризуется минимальным числом сигналов для идентификации источника прерывания. Как правило, все запросы на прерывание поступают по общей шине и устанавливают в состояние 1 триггер прерывания. Единичное состояние этого триггера запрещает другие прерывания и приводит к выполнению одного или нескольких шагов, в течение которых содержимое программного счетчика (ПС), а в некоторых МП — и других регистров, засылается для сохранения в определенные ячейки памяти; ПС после этого устанавливается в состояние, соответствующее адресу первой команды программы обслуживания прерываний.

При выполнении программы обслуживания прерываний прежде всего должно быть записано в память содержимое программно-доступных регистров, если такое сохранение данных не реализовано аппаратным способом. Затем эта программа должна определить, какое именно УВВ вызвало прерывание, и передать управление программе обработки запроса от этого устройства. После выполнения данной программы должно быть восстановлено содержимое регистров и ПС, сброшен в 0 триггер прерывания и разрешено поступление новых запросов на прерывание.

Приоритеты в рассматриваемом варианте устанавливаются в ходе проверки состояния флаговых триггеров УВВ. Для такой проверки обычно используется последовательность команд условного перехода, анализирующих состояние флаговых триггеров от УВВ с высшим до УВВ с низшим приоритетом. При обслуживании прерывания по запросу от некоторых УВВ в ряде случаев может разрешаться поступление в любой момент новых запросов на прерывание, которые необходимо обслужить. При такой системе прерываний в микроЭВМ должна быть организована магазинная память в целях сохранения состояния регистров и ПС для каждого уровня прерывания. Максимально допустимое число последова-

тельных прерываний программ называется *глубиной прерывания* и определяется емкостью магазинной памяти.

Если глубина прерывания равна 1, то прерывающая программа выполняется до конца даже при поступлении запросов на прерывание в период ее выполнения.

Основным недостатком программной реализации системы прерываний является большое время реакции на запросы, т. е. большой промежуток времени между моментами поступления запроса и начала его обслуживания. Поэтому такая реализация системы прерываний использовалась в основном в МП первых поколений.

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.

В качестве иллюстрации аппаратного способа реализации учета приоритетов рассмотрим схему на рис. 11.5. Запрос на прерывание может поступить от трех УВВ, причем высший приоритет имеет УВВ1, а низший — УВВ3. При одновременном поступлении запросов от УВВ1 и любого другого триггер Т1 переходит в состояние 1, сигнализирующее о том, что должен быть обслужен запрос именно от УВВ1. От инверсного выхода этого триггера сигнал 0 поступает на элементы И, стоящие перед триггерами Т2 и Т3, препятствуя тем самым переводу этих триггеров в состояние 1. При отсутствии запроса от УВВ1 и одновременном поступлении запросов от УВВ2 и УВВ3 в состояние 1 переводится только триггер Т2, а сигнал 0 с его инверсного выхода препятствует переводу в состояние 1 триггера Т3. Таким образом, триггер Т2 сигнализирует о необходимости обслуживания запроса на прерывание от УВВ2.

Запросы на прерывание в большинстве МП подразделяются на немаскируемые и маскируемые. *Немаскируемые* запросы на прерывание реализуются аппаратно вне программного контроля и программно не управляются. Прерывания по таким запросам имеют высший приоритет, т. е. выполняются раньше прерываний по всем другим запросам. Примерами не-

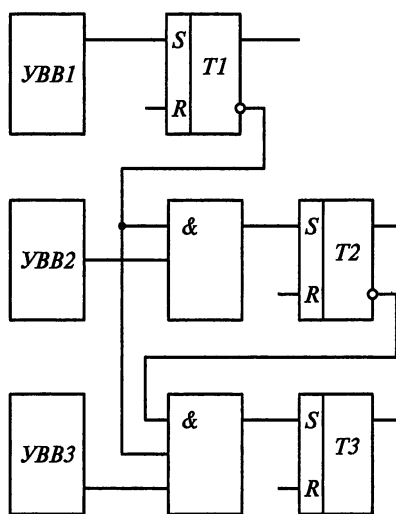


Рис. 11.5. Схема аппаратной реализации прерываний с учетом приоритетов

маскируемых запросов могут служить запросы на прерывание от схем питания, схем контроля правильности передачи данных. *Маскируемые* запросы на прерывание поступают в виде команд программы и обеспечивают возможность гибкого управления вычислительным процессом.

Сообщения, поступающие от различных устройств ввода, могут иметь разную степень важности. Например, в системе могут быть быстродействующие УВ, которые сохраняют информацию только в течение короткого интервала времени и поэтому требуют немедленного внимания. В этом случае желательно установить приоритеты и разрешить УВ с большим приоритетом прерывать подпрограмму обслуживания УВ с меньшим приоритетом. Обслуживание УВ с меньшим приоритетом возобновится после повторного разрешения системы прерываний. При наличии нескольких запросов запрос от УВ с наибольшим приоритетом должен быть воспринят раньше других.

Логическая схема управления приоритетами содержит регистр маски, который для ЭВМ считается портом ввода — вывода. Эта схема определяет, какое прерывающее УВ имеет наибольший приоритет, и сравнивает его с маской, чтобы определить, есть ли запрос прерывания и нужно ли посылать в МП вектор прерывания, соответствующий запросу с наибольшим приоритетом.

Часто вводимые данные не сразу передаются в рабочую область памяти для обработки, а сначала временно содержатся в другой ее области. Такое заранее выделенное пространство называется *буферной областью*. Вывод данных осуществляется аналогичным образом: результаты обработки основной программой поступают в буферную область, а затем передаются оттуда в УВ.

Чтобы совместить обработку и сбор данных, необходимы два буфера. Когда один буфер заполнен, МП обрабатывает данные из него, а вводимые данные накапливаются в другом буфере. Для работы в режиме реального времени обработка данных из одного буфера должна заканчиваться раньше заполнения другого буфера. При заполнении второго буфера они меняются ролями, что можно осуществить посредством модификации указателей буферов в подпрограммах обслуживания ввода — вывода и обработки.

11.4. Понятие о состоянии процессора

Сигналы состояния процессора содержат информацию о внутреннем функционировании МП и иногда могут использоваться в качестве сигналов управления устройствами микропроцессорной системы. Они вместе с сигналами контрольной аппаратуры позволяют оператору ЭВМ получить более полное представление о работе системы. Сигналы этой группы, которые в пределах конкретной системы не используются, могут быть в определенных ситуа-

циях некорректными, не приводя к неправильному функционированию системы в целом. Например, выходная цепь схемы выработки какого-то сигнала может выйти из строя и соответствующий сигнал состояния будет иметь при этом постоянный низкий уровень, но МП по-прежнему будет выполнять свои функции.

В конкретном МП обычно используется несколько сигналов состояния, информирующих о различных ситуациях, возникающих в процессе его работы. Каждый сигнал является индикатором соответствующей ситуации, т. е. он несет информацию о том, возникла данная ситуация или нет. Следовательно, каждый индикатор содержит только один бит информации. Несколько индикаторов образуют регистр состояния, который можно также назвать регистром индикаторов. Например, элементарный МП содержит в своем регистре состояния только индикаторы нуля и переноса, но в современных МП обычно используется большее число индикаторов.

На рис. 11.6 показан в качестве примера регистр состояния (или индикаторов), имеющий восемь одноразрядных индикаторов b_0 — b_7 , в пяти из которых содержится информация, относящаяся к состоянию МП. Эта информация используется условными ветвлениями программы, вызовами подпрограмм и возвратами подпрограмм.

Индикатор переноса CY (разряд b_0) устанавливается в 1 или сбрасывается в 0 в результате выполнения арифметических операций. Его состояние проверяется командами программы. Переполнение 8 бит при сложении устанавливает CY в 1. При выполнении вычитания наличие 1 в индикаторе CY указывает, что вычитаемое больше уменьшаемого.

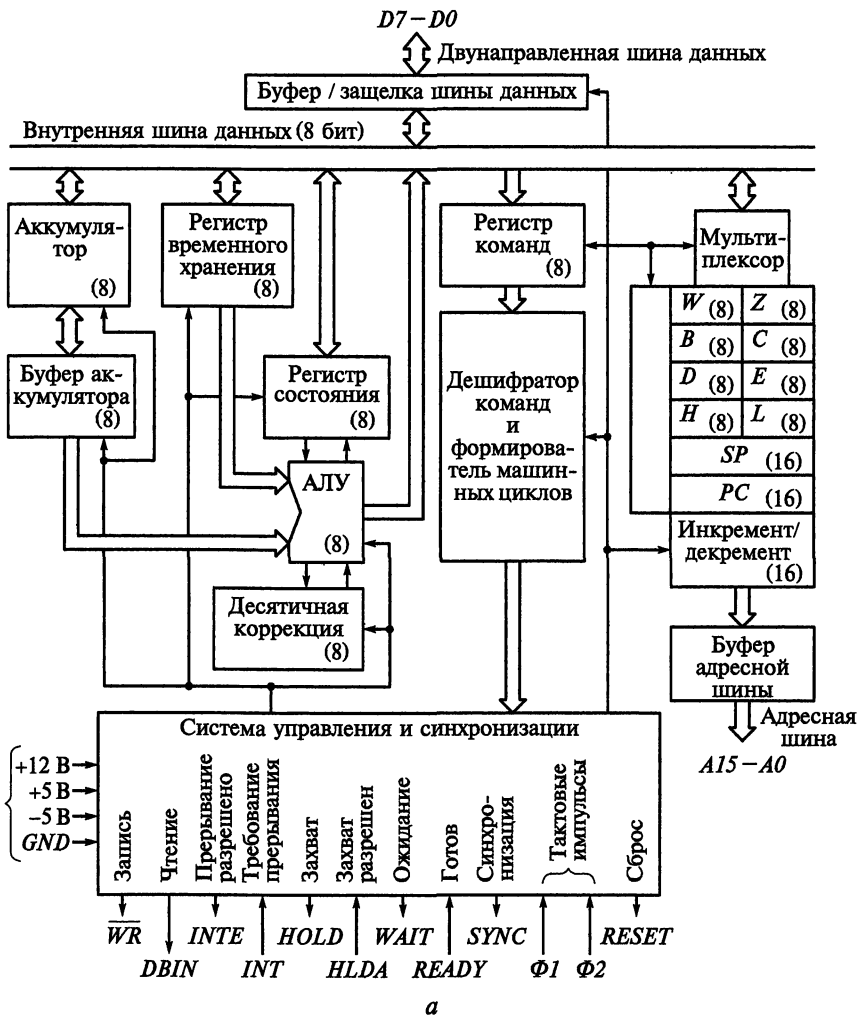
Индикатор нуля Z (разряд b_6) устанавливается в 1, когда результатом некоторых операций является 0; в противном случае он сбрасывается в 0.

Индикатор знака S (разряд b_7) устанавливается в 1 или сбрасывается в 0 в зависимости от состояния знакового бита после выполнения арифметических или логических команд. Эти команды используют самый старший бит данных для представления знака числа, содержащегося в аккумуляторе. Нахождение индикатора в состоянии 1 соответствует отрицательной величине, в состоянии 0 — положительной.

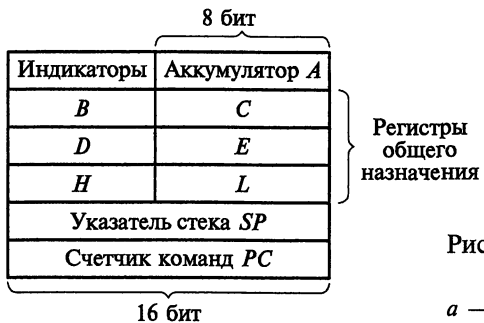
Индикатор вспомогательного переноса AC (разряд b_4) показывает переполнение или перенос в третьем разряде аккумулятора таким же образом, как индикатор переноса показывает переполнение или перенос в седьмом разряде. Этот индикатор используется в ходе выполнения операций двоично-десятичной арифметики.

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
S	Z		AC		P		CY

Рис. 11.6. Регистр состояния МП Intel 8805



a



b

Рис. 11.7. Микропроцессор фирмы Intel с 8-битной шиной:

a — функциональная схема; б — внутренние регистры, доступные при программировании

Индикатор паритета, или четности, P (разряд b_2) проверяет число единиц в аккумуляторе. Если это число четное (четный паритет), то он устанавливается в 1; если нечетное (нечетный паритет) — сбрасывается в 0. Например, если выполнение команды СЛОЖИТЬ дает в аккумуляторе результат 0011 0011, то в индикаторе P будет установлена 1, так как число единиц (4) четное. Если в аккумуляторе результат 1010 1110, то индикатор P сбрасывается в 0, потому что число единиц (5) нечетное.

Так как индикаторы содержат только единственный бит, их иногда называют не индикаторами, а битами. В описаниях работы МП часто встречаются выражения «бит переноса», «бит нуля», «бит вспомогательного переноса», «бит четности, или паритета (в регистре состояния)».

Все рассмотренные одноразрядные индикаторы в совокупности называются *флаговым регистром*. По мере совершенствования МП во флаговый регистр добавляются и другие разряды, информирующие о состоянии МП и особенностях кодовой информации, записанной в регистры общего назначения, прежде всего в регистр-аккумулятор.

Среди многочисленных выводов МП (а их у современных МП может быть несколько сотен) имеются такие, которые информируют подключаемые устройства: о получении запроса на прерывание, выполнении в данный момент времени записи или считывания из памяти, записи (считывания) из УВВ и других состояниях МП.

У МП фирмы Intel с 8-битной шиной (рис. 11.7) регистры B и C , D и E , H и L можно объединять попарно, они являются универсальными. Указатель стека SP , счетчик команд PC и регистр состояния (флаговый) являются специальными регистрами. В паре регистров H и L может быть записан 16-разрядный адрес. Такой МП имеет 40 выводов. Восемь выводов шины данных $D7 - D0$ являются двунаправленными (данные можно вводить и выводить); 16 выводов адресов $A15 - A0$ соединены с адресной шиной; четыре вывода служат для питания (в том числе вывод GND , который является общим для всех источников питания и соединяется с «землей»). Для управления и синхронизации работы всех элементов МП используются 12 выводов, назначение которых написано на рис. 11.7, а по-русски внутри блока системы управления и синхронизации, а снаружи этого блока указано их обозначение латинскими буквами. Выводы $\Phi 1$ и $\Phi 2$ являются входами генератора тактовых импульсов.

11.5. Микроконтроллеры

Все более широкое распространение в промышленности и быту получают различные устройства с микропроцессорным управлением. Схемы управления такими устройствами включают в себя

сам МП и элементы, с помощью которых вводится информация о состоянии управляемого объекта и выводятся сигналы для воздействия на управляемый объект. Такие схемы получили название *микроконтроллеры*.

Микроконтроллеры применяются и в системах автоматического управления технологическими процессами, где они выполняют задачи: программного регулирования параметров процесса; управления пуском, остановкой, реверсом и блокировкой оборудования; сигнализации, индикации и контроля технологических параметров.

Микроконтроллеры располагаются непосредственно у технологических и производственных объектов и осуществляют управление этими объектами или контроль за ними. Указанные функции выполняются с помощью программирования МП или путем построения микропроцессорных подсистем.

Микроконтроллеры, реализующие алгоритмы логического управления, предназначены для замены релейных и логических схем автоматики. Иногда их называют *программируемыми логическими микроконтроллерами*. Микроконтроллеры, реализующие алгоритмы автоматического регулирования процессов, заменяют аналоговые и аналого-цифровые регуляторы. Такие микроконтроллеры называются *программируемыми регулирующими*. Применяются микроконтроллеры и для реализации специальных алгоритмов управления бытовыми приборами, контрольно-измерительными приборами, транспортными механизмами и т.д. Появление микроконтроллеров свидетельствует о тенденции к замене обычных электронных схем с фиксированной структурой на схемы с программируемой структурой.

Программируемые микроконтроллеры (ПМК) характеризуются высокими надежностью и производительностью, универсальностью применения, возможностью работы в окружающей среде с вредными воздействиями (запыленность, вибрация, электромагнитные помехи и т.п.).

Рассмотрим работу логического ПМК (рис. 11.8).

В запоминающее устройство программ ЗУП записывается программа управления в виде последовательности строк логических функций. В запоминающее устройство данных ЗУД записываются состояния одноразрядных входов, выходов и внутренних элементов.

Сигналы от датчиков через устройство связи с объектом УСО поступают на входной регистр

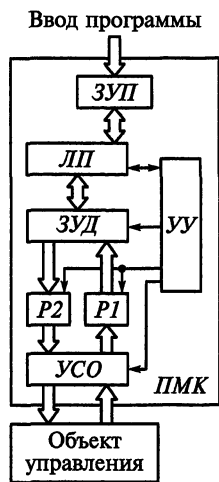


Рис. 11.8. Структурная схема логического ПМК

P1, а значения управляющих сигналов хранятся в выходном регистре *P2* и через *УСО* воздействуют на объект управления. В *УСО* осуществляются масштабирование и нормализация сигналов. Работа логического ПМК проходит в циклическом режиме.

Логический процессор *ЛП* в соответствии с программой моделирования конкретной логической схемы, записанной в *ЗУП*, опрашивает все входы, производит сравнение состояний входов и выходов и в соответствии с полученными результатами включает или выключает те или иные исполнительные органы.

Как правило, *ЛП* строится в виде однобайтового последовательного логического процессора обработки входных данных. Он последовательно, строка за строкой, опрашивает *ЗУП*, производит последовательное вычисление логических функций и заносит результаты в *ЗУД*. После опроса памяти устройство управления *УУ* обеспечивает обмен данными между входными и выходными регистрами и *ЗУД*. Процесс опроса *ЗУП* и обмена данными периодически повторяется.

Время цикла опроса характеризует производительность контроллера. Логические ПМК реализуют относительно простые логические функции управления.

Рабочие циклы, повторяющиеся в процессе управления объектом, состоят из последовательности фаз: получение данных о состоянии объекта; обработка данных в целях получения управляющих сигналов; выдача управляющих сигналов на исполнительные устройства.

Структурная схема регулирующего ПМК (рис. 11.9) внешне напоминает схему микроЭВМ с устройством связи с объектом (*УСО*) и средствами ввода — вывода аналоговых и цифровых сигналов.

Отличается регулирующей ПМК от микроЭВМ общего пользования хранением алгоритмов регулирования в постоянной памяти *ПЗУ*, а также отсутствием стандартных средств ввода — вывода и отладки программ.

Мультиплексор *МПК* и аналого-цифровой преобразователь *АЦП* обеспечивают прием аналоговых сигналов. Подключение датчиков мультиплексором осуществляется под управлением микропроцессорного вычислителя *МПВ*. Управляющие сигналы на выходах формируются с помощью цифроаналогового преобразователя *ЦАП*.

Дискретно-цифровой преобразователь *ДЦП* и цифродискретный преобразователь *ЦДП* выполняют функции согласования уровней логических сигналов с уровнями дискретных сигналов. Центральный процессор *ЦП* представляет собой микропроцессор. Программы регулирования хранятся в *ПЗУ*. Оперативное запоминающее устройство *ОЗУ* используется для временного хранения промежуточных результатов расчета режимов работы и коэффициентов настройки, а также изменяемых параметров регулирующего

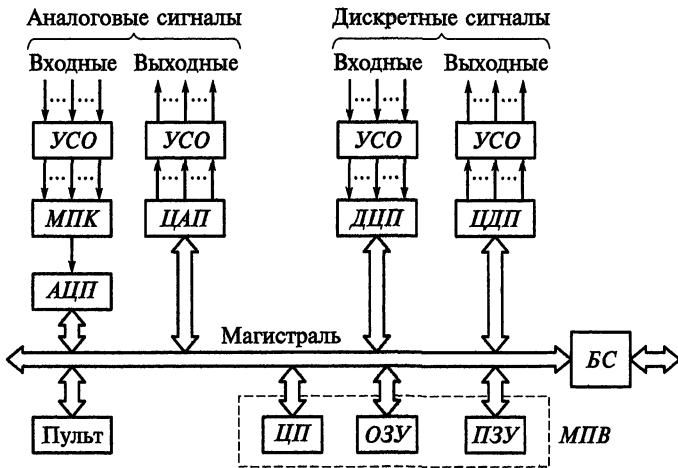


Рис. 11.9. Структурная схема регулирующего ПМК

ПМК. С помощью блока сопряжения **БС** можно осуществлять наращивание регулирующего ПМК, т. е. подключать дополнительные **УСО** и преобразователи. Тем самым достигается возможность обрабатывать информацию от сотен и тысяч датчиков, что улучшает качество работы системы автоматического управления.

Контрольные вопросы

1. Что такое команда?
2. Как происходит выполнение команд?
3. Зачем нужны прерывания?
4. Что такое состояние процессора?
5. Зачем нужны микроконтроллеры?
6. Какие сигналы поступают от объекта управления в микроконтроллер?

ЗАПОМИНАЮЩИЕ УСТРОЙСТВА**12.1. Виды и характеристики запоминающих устройств**

Запоминающие устройства классифицируют по ряду признаков.

По длительности хранения информации ЗУ подразделяются на *долговременные* и *кратковременные*. В свою очередь, долговременные ЗУ, или ЗУ с долговременным хранением информации, подразделяются на *постоянные* ЗУ (ПЗУ) и *полупостоянные* ЗУ (ППЗУ). Характерной чертой ПЗУ и ППЗУ является сохранение информации при отключении источников питания. В ПЗУ возможна лишь однократная запись информации, производимая либо в процессе производства, либо в результате программирования. В ППЗУ возможно многократное изменение хранимой информации при эксплуатации.

Кратковременные ЗУ используются для хранения оперативной, часто меняющейся информации. В этих ЗУ отключение источников питания приводит, как правило, к потере хранимой информации. При сокращении длительности цикла записи ППЗУ могут быть использованы и для хранения оперативной информации. В большинстве случаев ППЗУ могут применяться и в качестве ПЗУ.

По адресации эти устройства подразделяют на ЗУ с *произвольной*, *последовательной* и *ассоциативной* выборкой. В ЗУ с произвольной выборкой (или доступом) время обращения не зависит от адреса ячейки в устройстве. В ЗУ с последовательной выборкой для нахождения числа по определенному адресу необходимо последовательно просмотреть все ячейки, предшествующие заданной, т.е. время обращения зависит от адреса. Для поиска определенной информационной единицы в таком ЗУ необходимо сначала отыскать соответствующий массив, а затем информационную единицу в этом массиве.

Запись в устройство памяти, т.е. в ЗУ, а также чтение из ЗУ называют еще *обращением к памяти*. Важнейшими характеристиками ЗУ являются его информационная емкость и быстродействие.

Емкость памяти определяет число слов, которые могут храниться в памяти, и выражается в битах или байтах. Например, если память ЭВМ составляет 64 Кбайт, это означает, что она может хранить 64×1024 восьмиразрядных слов (1 Кбайт = 1024 байт).

Быстродействие ЗУ определяется продолжительностью обращения к нему. При записи слова время обращения к ЗУ t_0 складыв-

вается из времени поиска слова $t_{\text{п}}$, времени стирания ранее записанного слова $t_{\text{ст}}$ (при необходимости) и времени записи нового слова $t_{\text{зп}}$, т.е. $t_0 = t_{\text{п}} + t_{\text{ст}} + t_{\text{зп}}$. При чтении время обращения к ЗУ складывается из времени поиска слова $t_{\text{п}}$, времени чтения слова $t_{\text{чт}}$ и времени его восстановления в памяти $t_{\text{вос}}$ (при необходимости), т.е. $t_0 = t_{\text{п}} + t_{\text{чт}} + t_{\text{вос}}$.

Работа ЗУ основана на различных физических явлениях. Наибольшее распространение в вычислительной технике получили *полупроводниковые, магнитные и оптические ЗУ*.

По назначению ЗУ подразделяются на *сверхоперативные (СОЗУ), оперативные (ОЗУ), внешние и внутренние*. Внешние ЗУ (ВЗУ) предназначаются для хранения больших массивов информации. Все математическое программное обеспечение вычислительных систем находится именно в ВЗУ. Информационная емкость ВЗУ в настоящее время составляет терабайты.

При необходимости использования той или иной программы в ходе требуемого процесса обработки эта программа предварительно передается из ВЗУ в ОЗУ. В качестве внешних ЗУ используют накопители на магнитных дисках (НМД) и на оптических дисках (НОД). При создании архивов программ и данных, их резервных копий обычно применяют накопители на магнитных лентах (НМЛ).

Для промежуточного хранения данных и программ, которыми обмениваются устройства, работающие с разной скоростью и используемые неравномерно, применяются буферные ЗУ (БЗУ). Наиболее часто они бывают необходимы в многоканальных устройствах обмена.

На рис. 12.1 в виде обобщенной схемы приведена иерархия ЗУ. В конкретных условиях число уровней иерархии, а также число

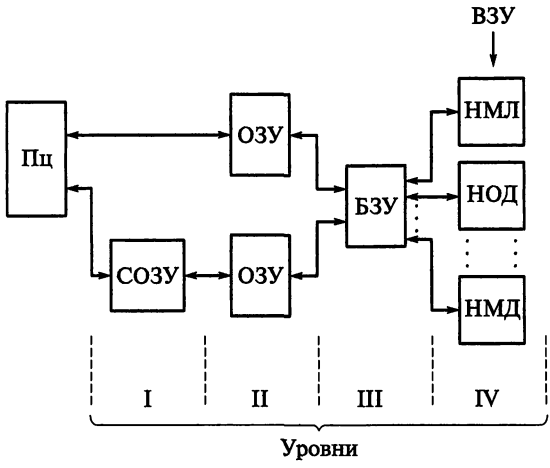


Рис. 12.1. Иерархия запоминающих устройств

блоков СОЗУ, ОЗУ, БЗУ и ВЗУ может быть разным. На I уровне используются сравнительно небольшие по информационному объему, но быстродействующие СОЗУ, тесно связанные с процессором (Пц). К ЗУ II уровня относятся ОЗУ с произвольной выборкой, обладающие большей емкостью, чем ЗУ I уровня, но имеющие более низкое быстродействие. К ЗУ III уровня относятся БЗУ с произвольной выборкой, емкость которых занимает промежуточное положение между емкостями СОЗУ и ОЗУ. Быстродействие БЗУ должно быть высоким; часто обращение к БЗУ организовано по неявным адресам. К IV уровню иерархии относятся ВЗУ с плавающим временем выборки, но с большой информационной емкостью, например НМД, НОД или НМЛ.

12.2. Оперативные запоминающие устройства

Оперативные запоминающие устройства предназначаются для хранения данных и программ текущих вычислений, а также данных и программ, к которым следует перейти, если в ходе вычислительного процесса возникнет прерывание. Быстродействие ОЗУ соизмеримо со средним временем выполнения операций. С процессорными устройствами ОЗУ связано непосредственно или через сверхоперативные ЗУ (СОЗУ). В большинстве случаев в качестве запоминающих элементов используются триггерные схемы в полупроводниковом исполнении, объединенные в запоминающие матрицы.

Быстродействие СОЗУ соизмеримо с быстродействием устройств, выполняющих самые быстрые операции над операндами. Информационная емкость СОЗУ обычно невелика. Такие устройства необходимы для того, чтобы обращение к памяти за операндами не приводило к увеличению времени выполнения операций. С целью сокращения длины линий связи СОЗУ обычно не делают самостоятельными устройствами, а вводят их в состав процессора.

В современных МП имеются СОЗУ, сформированные на том же кристалле, что и все остальные элементы МП. Расстояния от триггеров таких СОЗУ до АЛУ настолько малы, что информация передается за доли наносекунд. В новейших МП самая быстродействующая память получила название кэш 1-го уровня (см. подразд. 8.2), менее быстрое СОЗУ — кэш 2-го уровня. Емкость кэш-памяти 1-го уровня в современных ЭВМ составляет 32 Кбайт, а кэш-памяти 2-го уровня — 512 Кбайт.

Рассмотрим кратко принцип действия полупроводниковой оперативной памяти с произвольным обращением. Основу ЗУ на полупроводниковых транзисторах составляют запоминающие матрицы (ЗМ), состоящие из запоминающих элементов (ЗЭ) в виде триггерных схем. Под словом «матрица» в данном случае понимается схема, содержащая большое число однотипных элементов,

расположение которых напоминает матрицу, т. е. сетку или таблицу, имеющую строки и столбцы. Каждый элемент такой схемы является как бы ячейкой данной матрицы-таблицы. Он определяется номерами строки и столбца. Современные технологии в области микроэлектроники позволяют в одной микросхеме разместить сотни тысяч ЗЭ, т. е. в полупроводниковом ЗУ можно хранить сотни тысяч бит информации.

Организация ЗМ определяется способом выборки (опроса) ЗЭ при чтении и записи. Различают ЗУ с однокоординатной (пословной) и двухкоординатной выборками.

В матрице с однокоординатной выборкой одна строка образует слово из m разрядов. На схеме (рис. 12.2) символами A_0, A_1, \dots, A_{n-1} обозначены адресные шины, а символами P_0, P_1, \dots, P_{m-1} — разрядные. Адресные шины связаны с каждым ЗЭ одного слова, в то время как разрядные шины — с ЗЭ одноименного разряда всех слов. Для записи слова по выбранному адресу A_i на разрядные шины $P_0 - P_{m-1}$ подается комбинация электрических сигналов 1 и 0. Эти сигналы попадают на каждый из запоминающих элементов i -й строки $ЗЭ_{i0}, ЗЭ_{i1}, \dots, ЗЭ_{i(m-1)}$. При наличии в адресной шине сигнала выборки i -го слова состояние каждого из ЗЭ в этом слове может быть определено по сигналам на выходных разрядных шинах, которые подключены к разрядным усилителям считывания.

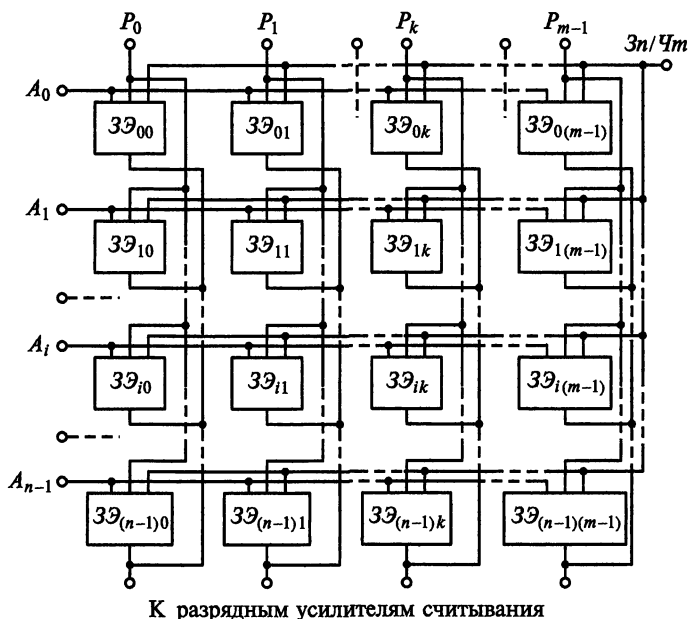


Рис. 12.2. Структурная схема матрицы ЗУ с однокоординатной выборкой

Выбор той или иной адресной шины производится дешифратором адреса (на рис. 12.2 не показан), на вход которого поступает двоичный код номера A_i — номера ячейки, в которую должно быть записано или из которой должно быть считано слово. Эта ячейка имеет только одну координату — номер строки матрицы запоминающих элементов. Микросхема ОЗУ (рис. 12.3) имеет 16 запоминающих ячеек, каждая из которых может выбираться комбинацией двоичных переменных на четырех адресных входах $A0$ — $A3$ ($2^4 = 16$). Ячейка имеет четыре разряда (бита). Запись в нее осуществляется через входы $D0$ — $D3$ при наличии на входе запись — чтение (W/R) логической 1 ($W/R = 1$), а считывание производится с выходов $Q0$ — $Q3$ при $W/R = 0$. Инициализация (ввод в действие, выбор) микросхемы осуществляется логической 1, подаваемой на вход CS . При $CS = 0$ микросхема блокируется.

В двухкоординатной матрице $n \times n$ (рис. 12.4) ЗЭ выбирается с помощью двух адресных шин X_i, Y_j . Например, при наличии сигнала выборки на адресных шинах X_1, Y_1 будет выбран только ЗЭ $_{n+2}$. Его состояние будет зафиксировано общей для всех ЗЭ разрядной шиной P . Чтобы записать бит информации в выбранный ЗЭ, по разрядной шине необходимо подать соответствующий сигнал (0 или 1). К усилителю считывания сигнал от ЗЭ поступает через элемент И при наличии сигнала 1 на входе $3n/Чт$. Данная матрица позволяет хранить n^2 одноразрядных слов. Число матриц в ЗУ должно соответствовать числу разрядов хранимых слов.

Число выходов дешифратора и адресных линий внутри микросхемы равно числу ячеек в накопителе (см. рис. 12.2). Оно может

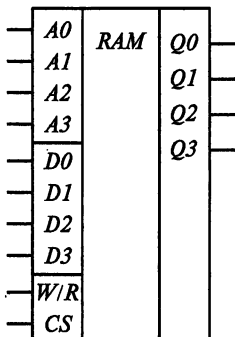


Рис. 12.3. Условное изображение микросхемы ОЗУ

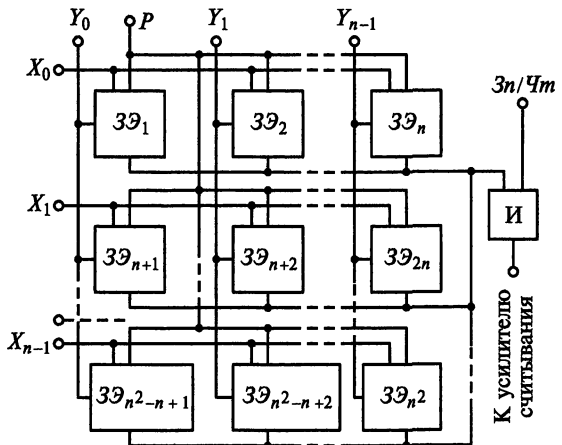


Рис. 12.4. Структурная схема матрицы ЗУ с двухкоординатной выборкой

быть весьма значительным, что технологически ограничивает емкость памяти в пределах кристалла. Так, для матрицы 32×32 (т. е. по 32 ячейки в каждой строке и в каждом столбце) это число равно 1024. Его можно значительно уменьшить, если использовать двухкоординатную адресацию. В этом случае каждый 3Э матрицы выбирается дешифратором строки и дешифратором столбца (см. рис. 12.4). Выбранным оказывается элемент, находящийся на пересечении возбужденных линий, одна из которых принадлежит дешифратору строки, а другая — дешифратору столбца.

Каждая такая двухкоординатная матрица хранит одноименные разряды всех записываемых слов (одна матрица — первые разряды, другая — вторые и т. д.), а все k -разрядное слово хранится в k матрицах. Информационные выводы D и Q на рис. 12.4 не показаны.

12.3. Принцип магнитной записи

Магнитная запись долгое время являлась самым распространенным способом записи информации. Ее физической основой является свойство ферромагнитных материалов сохранять состояние остаточной намагниченности.

Рассмотрим процесс намагничивания кольцевого ферритового сердечника (рис. 12.5, а), который представляет собой запоминающий элемент с двумя обмотками — записи и считывания. В зависимости от направления тока, протекающего через входную обмотку (обмотку записи), сердечник намагничивается по или против часовой стрелки (рис. 12.5, б).

Возможность записи информации на ферритовом сердечнике основана на гистерезисе процесса перемагничивания (рис. 12.6). Условимся одно из направлений намагниченности, например против часовой стрелки, называть отрицательным ($-B_r$) и припишем ему значение логического 0. Тогда другое направление магнитной индукции, по часовой стрелке, будем считать положительным ($+B_r$) и припишем ему значение логической единицы 1. Таким образом, запись информации в сердечнике может быть осуществлена пропусканием тока того или иного направления с амплитудой

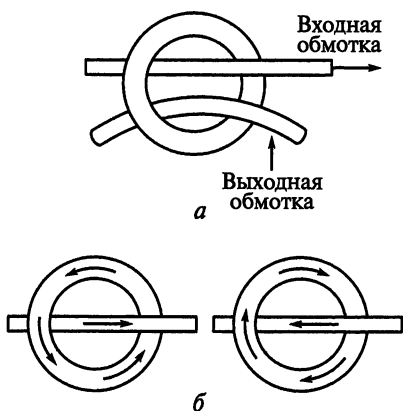


Рис. 12.5. Ферритовый сердечник с обмотками (а) и направление его намагниченности в зависимости от направления тока во входной обмотке (б)

$I_{зп}$, достаточной для создания напряженности магнитного поля H_m , обеспечивающей надежное перемагничивание сердечника.

Для записи 1 во входную обмотку подают ток $I_{зп1}$, который значительно больше тока записи 0 $I_{зп0}$. Для считывания информации во входную обмотку подают импульс тока $I_{сч}$, достаточный для создания напряженности магнитного поля $-H_m$. Тогда, если в сердечнике была записана 1, магнитная индукция в нем изменится от $+B_r$ до $-B_r$. Если же был записан 0 (точка $-B_r$ на рис. 12.6), то изменения индукции практически не будет.

Изменение индукции означает изменение магнитного потока, следовательно, в выходной обмотке сердечника (обмотке считывания) возникает ЭДС, пропорциональная величине этого изменения. Различие между сигналами считывания 0 и 1 заключается в разнице амплитуд выходного сигнала. Малая величина сигнала означает, что сердечник находился в состоянии 0, а большая — в состоянии 1. После считывания сердечник всегда оказывается в состоянии 0, т.е. исходная информация в нем стирается. Если она должна быть сохранена, то необходимо принимать специальные меры для ее восстановления.

Магнитная запись широко применяется в ЗУ ЭВМ, а также в многочисленных устройствах для бытовой и профессиональной звукозаписи, видеозаписи, записи сигналов связи и иных цифровых устройствах для хранения большого объема цифровой информации. В этой главе рассматриваются магнитные ЗУ только такого назначения, т.е. без устройств для записи и воспроизведения аналогового сигнала.

При записи (или считывании) информации в магнитном ЗУ происходит взаимодействие магнитного носителя информации и магнитной головки. Записывающая магнитная головка — это специальный электромагнит, по обмотке которого пропускают им-

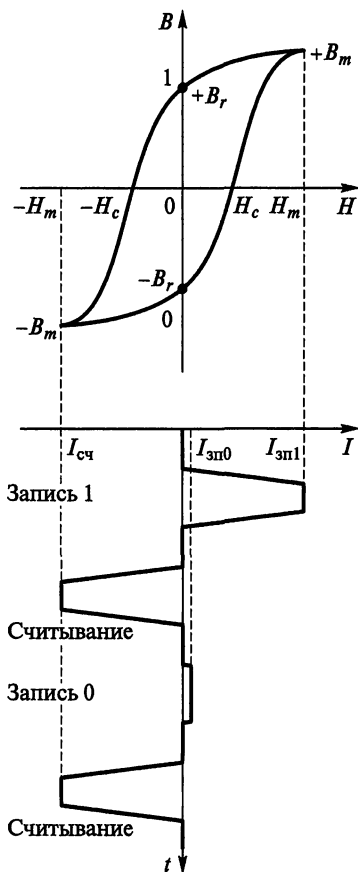


Рис. 12.6. Кривая намагничивания ферромагнетика и временные диаграммы процессов записи и считывания

пульсный ток. Для уменьшения потерь на вихревые токи магнитопровод собирают из тонких пластин магнитомягких материалов или выполняют из феррита. Магнитопровод имеет воздушный зазор, через который замыкаются силовые линии магнитного поля. Под действием этого поля намагничиваются отдельные участки магнитного носителя информации. Для считывания (воспроизведения) данной информации используются магнитные считывающие головки, аналогичные по конструкции записывающим. И запись, и считывание происходят при относительном перемещении носителя и головок.

Взаимное расположение головок и носителя информации в процессе записи — считывания показано на рис. 12.7. При поступательном движении со скоростью v носителя информации 1 относительно записывающей головки 2 отдельные его участки в зависимости от тока записи $i(t)$ (т. е. от направления магнитного поля) приобретают различные значения остаточной индукции ($+B_r$ или $-B_r$).

До записи 1 или 0 ориентация в носителе элементарных магнитов диполей хаотичная, усредненное значение индукции на всех участках носителя близко к нулю. После записи ориентация диполей становится упорядоченной. Диполь с остаточной индукцией $+B_r$ соответствует, например, двоичной 1, а диполь с остаточной индукцией $-B_r$ — двоичному 0. При прохождении носителя мимо считывающей головки 3 силовые линии магнитного поля диполя замыкаются через ее сердечник и создают в нем изменяющийся по величине магнитный поток того или иного направления. В результате в обмотке головки наводится ЭДС $e(t)$, являющаяся сигналом считывания. Считывание происходит без разрушения информации и может выполняться многократно.

Характерной особенностью магнитной записи является то, что она не нуждается в какой-либо промежуточной обработке и может быть воспроизведена немедленно. Запись легко может быть стерта с помощью стирающей головки, такой же, как и записывающая, или с помощью самой записывающей головки при по-

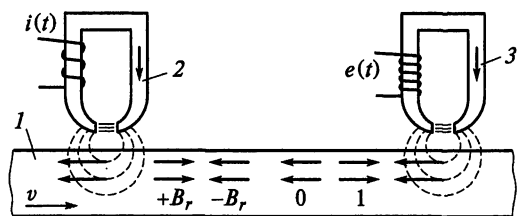


Рис. 12.7. Устройство магнитной записи:

1 — носитель информации; 2 — записывающая головка; 3 — считывающая головка

даче специального сигнала, размагничивающего участки носителя. Для стирания обычно через обмотку головки пропускают ток высокой частоты. Высокочастотное поле многократно меняет ориентацию диполей, в результате чего она вновь становится хаотической и остаточная индукция исчезает. Поскольку информация на магнитных носителях сохраняется неограниченное время и может при необходимости многократно изменяться, магнитные носители информации относят к полупостоянным ЗУ.

В одной из первых в нашей стране ЭВМ «Сетунь» (1959 г.) использовалась троичная система представления чисел. Эта ЭВМ была выполнена на пороговых логических элементах типа быстродействующих магнитных усилителей. Во всех современных ЭВМ используется двоичная система.

При магнитной записи цифровой информации в целях увеличения емкости запоминающих устройств стремятся как можно полнее использовать рабочую поверхность носителя. Степень ее использования определяется плотностью записи информации, т.е. числом двоичных знаков (один знак — это один бит), размещающихся на единице площади носителя. Плотность записи зависит от характеристик магнитного носителя, конструкции головки, величины зазора между носителем и головкой и некоторых других факторов.

Теоретический предел плотности записи информации на магнитных носителях составляет около ста миллиардов (10^{11}) бит на 1 мм^2 . Реально достигнутая плотность записи примерно в миллион раз меньше.

В зависимости от формы и вида используемого носителя информации различают накопители на сердечниках, магнитных лентах, дисках, барабанах, картах.

Из отдельных ферритовых сердечников можно составить ЗМ, подобную рассмотренной в подразд. 12.2 полупроводниковой ЗМ. Каждый сердечник будет выполнять роль триггерного ЗЭ (см. рис. 12.2), но если в полупроводниковой ЗМ информация сохраняется лишь при наличии питания, то в ферритовой ЗМ она сможет храниться практически любое время.

Рассмотрим принцип совпадения токов, используемый в двухкоординатной матрице магнитных сердечников. На рис. 12.8 показана ячейка памяти, выполненная несколько иначе, чем на рис. 12.5. Входная и выходная обмотки сердечника заменены тремя шинами — X , Y и шиной считывания, пропущенными через ферритовое кольцо. При записи информации по шинам X и Y пропускают импульсы тока с амплитудой, равной половине максимального значения тока, необходимого для перемагничивания сердечника: $I_X = I_Y = I_m/2$. Таким образом, перемагничивание сердечника может произойти лишь при одновременном действии совпадающих по направлению токов I_X и I_Y . Создаваемые ими поля складываются

ся, и в сердечнике возникает суммарное магнитное поле, напряженность которого оказывается достаточной для перемагничивания сердечника.

Из таких ячеек памяти набирается ЗМ, одна из возможных схем которой показана на рис. 12.9. Рассмотрим, каким образом в этой ЗМ может быть осуществлен выбор любого элемента для записи или считывания информации. Допустим, что начальное состояние элементов — 0, что соответствует отрицательной намагниченности сердечников.

Для записи 1 в каком-либо элементе, например обозначенном на рис. 12.9 буквой *A*, нужно подать импульсы токов $I_m/2$ по шинам X_2 и Y_3 . При этом все сердечники, лежащие на этих шинах, окажутся под действием тока $I_m/2$, но это ни к чему не приведет, так как половины максимального значения тока недостаточно для изменения состояния сердечника. Только сердечник *A*, который находится на пересечении шин, окажется под действием полного тока и перемагнитится.

Чтобы произвести считывание информации, записанной в сердечнике, по соответствующим шинам подаются импульсы $I_m/2$ обратной полярности (в нашем случае — по шинам X_2 и Y_3). Если в сердечнике была записана 1, то при подаче сигналов обратной полярности он перемагнитится в состояние 0. При этом в его выходной обмотке возникает ЭДС, которая создаст импульс считывания. Если в сердечнике был записан 0, то перемагничивания не произойдет и сигнала на выходе не будет.

Запоминающее устройство на кольцевых ферритовых сердечниках работает довольно быстро — время перемагничивания может составлять доли микросекунд. Энергия, затрачиваемая на пе-

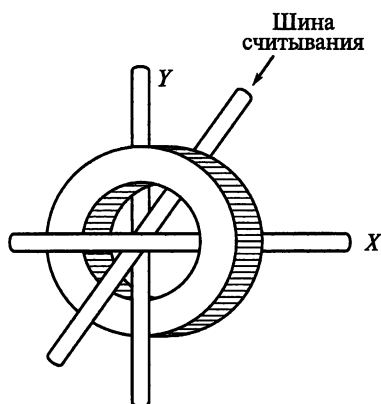


Рис. 12.8. Ячейка памяти с тремя шинами

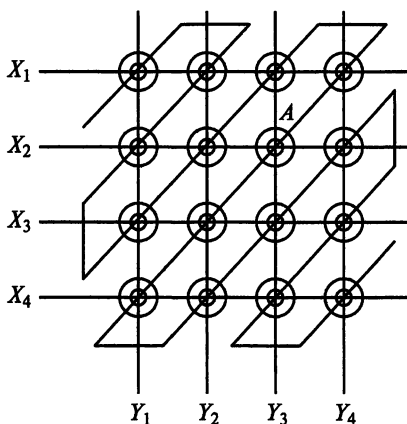


Рис. 12.9. Схема запоминающей матрицы на ферритах

ремагничивание, выделяется в материале сердечника и приводит к его нагреванию. Поскольку теплоотвод осуществляется через поверхностные слои материала, уменьшение размеров сердечника улучшает возможность теплоотвода, так как увеличивает отношение объема поверхностного слоя материала к его полному объему.

Уменьшение размеров сердечников повышает быстродействие ЗУ, снижает потребляемую мощность, уменьшает токи управления, позволяет получить более компактную конструкцию. Первые ферритовые сердечники, использовавшиеся в ЗУ, имели наружный диаметр около 2 мм. Совершенствование технологии изготовления ферритов привело к уменьшению их внешнего диаметра до 0,2 мм, а внутреннего — до 0,1 мм.

Изготовление сердечников малого размера и прошивка их проводами сопряжены с серьезными трудностями. Уменьшение толщины стенок делает сердечники хрупкими, а разброс их параметров существенно возрастает. Прошивка сердечников проводами — трудоемкий процесс, плохо поддающийся автоматизации. Преодоление трудностей изготовления ЗУ на магнитных сердечниках достигается применением методов интегральной технологии. При этом наряду с уменьшением размеров элементов памяти может быть увеличена емкость блока памяти и повышено быстродействие ЗУ.

Практика показывает, что обычно для каждого типа магнитных микроэлектронных устройств имеется область применения, где он дает наиболее оптимальные результаты. Поэтому, как правило, находят применение почти все типы магнитных микроэлектронных устройств.

На принципах, сходных с рассмотренными выше, основана работа ЗУ на многоотверстных ферритовых пластинах, тонких магнитных пленках (плоских и цилиндрических), слоистых ферритовых пластинах.

Развитие интегральных методов технологии привело к созданию структур, в которых система проводников помещена в саму среду из феррита. Такие структуры представляют собой многослойные ферритовые материалы с системой проводников, расположенных между слоями. На рис. 12.10 показано ЗУ на слоистых ферритах. На одну тонкую ферритовую пластину наносятся параллельные проводники, образующие систему строк (координата X), а на другую — параллельные проводники, образу-

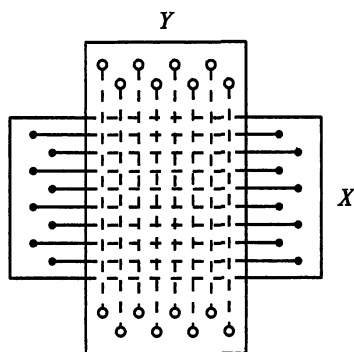


Рис. 12.10. Запоминающее устройство на слоистых ферритах

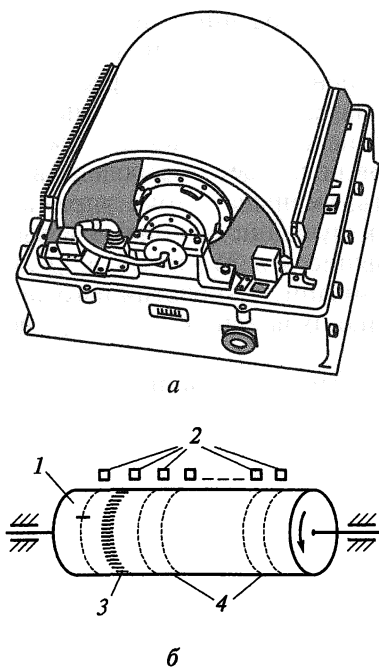


Рис. 12.11. Магнитный барабан:
 а — внешний вид; б — принцип за-
 писи считывания; 1 — носитель ин-
 формации; 2 — записывающие и
 считывающие головки; 3 — метки
 позиций записи; 4 — дорожки (тре-
 ки) записи информации

нитных сердечниках и магнитных барабанах (рис. 12.11). Использование новых методов записи и считывания информации, таких как магнитооптические, позволяет значительно улучшить характеристики ЗУ на магнитных носителях.

12.4. Магнитная лента

Магнитная лента представляет собой гибкую пластмассовую пленку, на поверхность которой нанесен тонкий слой магнитного материала. Этот материал имеет петлю гистерезиса, близкую к прямоугольной, и отличается высокой однородностью параметров. Значение коэрцитивной силы H_c выбирается довольно большим для того, чтобы исключить возможность стирания информации в результате воздействия внешних полей. Ширина ленты стандартная — от 6,5 до 35 мм, толщина — десятки микрометров. Лен-

ющие систему столбцов (координата Y). Между этими пластинами помещается третья тонкая ферритовая пластина. Она обеспечивает изоляцию между двумя системами проводников и служит носителем информации. В каждой точке пересечения проводников хранится 1 бит информации.

Применение тонких магнитных пленок позволяет существенно уменьшить размеры магнитных ЗУ и сократить время записи и считывания информации. В тонких магнитных пленках обеспечивается наивысшая скорость процесса перемагничивания — процесс длится всего около 10^{-9} с.

Хотя ферритовые ЗУ считаются устаревшими, не следует забывать, что по сравнению с другими ЗУ, использующими магнитную запись, они являются более надежными, поскольку не имеют вращающихся частей.

Долгое время в вычислительной технике основным принципом работы ЗУ большой емкости была именно магнитная запись. В ЭВМ, выпускавшихся до 1980-х годов, использовались ЗУ на ферромаг-

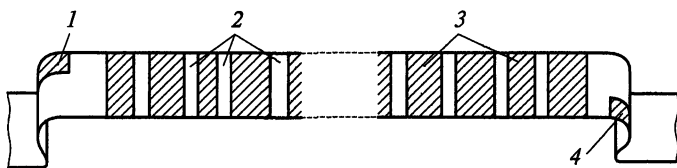


Рис. 12.12. Магнитная лента:

1 — метка начала ленты; 2 — междузонные расстояния; 3 — зоны записи; 4 — метка конца ленты

та сматывается в рулоны и хранится в специальных кассетах, которые используются по мере надобности. Длина ленты в рулонах может быть до 1 000 м, плотность записи — до 64 бит на 1 мм. При считывании или записи магнитная лента протягивается перед блоком магнитных головок, как в обычном магнитофоне, но только с более высокой скоростью — 2... 5 м/с. Обычно лента находится в неподвижном состоянии и приводится в движение лишь во время считывания или записи информации. Число дорожек на ленте определяется ее шириной. В накопителях на магнитной ленте применяют главным образом параллельно-последовательный способ размещения информации. Информацию, записанную на магнитную ленту, размещают на определенных участках поверхности ленты — *зонах записи* 3 (рис. 12.12). Между зонами имеются промежутки — междузонные расстояния 2. При пусках и остановках ленты мимо головок проходит часть ленты, которая меньше, чем эти расстояния. Специальные магнитные метки 1 и 4 обозначают соответственно начало и конец ленты.

Информационная емкость накопителей на магнитной ленте практически неограничена. Ограничением применения таких накопителей может быть лишь время обращения к ЗУ с накопителем на магнитной ленте $t_{м.л} = t_{п} + t_{п.з} + t_{з/с}$, где $t_{п}$ — время пуска ленты; $t_{п.з}$ — время поиска зоны ленты, где находится информация; $t_{з/с}$ — время протекания процесса записи или считывания. Время пуска $t_{п}$ и особенно время поиска $t_{п.з}$ очень велики, при большой длине ленты $t_{п.з}$ может достигать нескольких минут. Сама же скорость считывания и записи, когда магнитная головка находится в требуемой зоне магнитной ленты, достигает сотен килобит в секунду. Информацию на ленте можно стирать и перезаписывать, одна и та же лента может использоваться многократно.

На магнитной ленте обычно сохраняют резервные копии программ и данных, а также многочисленные архивные информационные материалы. Для записи такой информации на кассеты с магнитной лентой служат специальные устройства, которые называются *стримерами*. Стример, как и обычный кассетный магнитофон, имеет лентопротяжной механизм и магнитные голов-

ки, обеспечивающие цифровую запись и считывание. Информационная емкость кассет стримеров достигает десятков гигабайт.

12.5. Гибкие диски

У магнитных гибких дисков информация размещается на концентрических дорожках. Ее запись и считывание осуществляются с помощью магнитных головок.

Гибкие диски (флопи-диски, или дискеты) изготавливают из лавсана, покрытого магнитным слоем. Материал магнитного слоя определяет плотность записи. Оксид железа позволяет записать до 60 дорожек на 1 дюйм (2,54 см). Если магнитный слой выполнен из кобальта, то обеспечивается более высокая плотность — до 135 дорожек на дюйм. Однако при этом требуется вдвое более сильное магнитное поле — напряженностью около 50 А/мм. Соответственно возрастает и линейная плотность, т.е. число бит информации на единицу длины дорожки. В широко распространенной так называемой трехдюймовой дискете (рис. 12.13), диаметр которой на самом деле 3,5 дюйма, или 89 мм, линейная плотность составляет 17 434 бит на 1 дюйм, т.е. около 700 бит на 1 мм.

Информация на диске распределена по секторам: в каждом секторе по 512 байт на дорожке. Например, в пятидюймовом диске двойной плотности 9 секторов, а при высокой плотности — 15. По мере совершенствования ЗУ на гибких магнитных дисках увеличивается число дорожек и уменьшается их ширина. Так, в первых дисках число дорожек составляло 40 на сторону при ширине дорожки 0,33 мм, а в дисках двойной плотности это число возросло вдвое за счет уменьшения ширины до 0,16 мм. Дорожки нумеруются от внешней кромки к центру. Ближайшие к краю диска дорожки заняты служебной информацией, необходимой для взаимодействия ЗУ с устройствами воспроизведения (считывания).

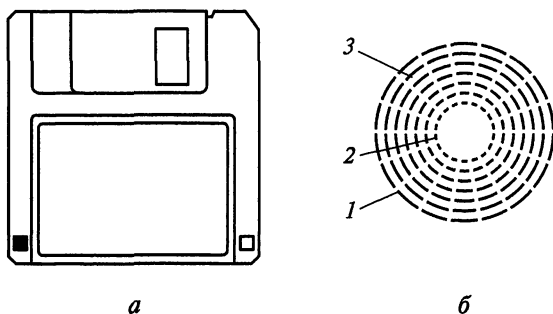


Рис. 12.13. Трехдюймовая дискета:

а — внешний вид; *б* — размещение информации по секторам и дорожкам; 1 — 0-я дорожка; 2 — 79-я дорожка; 3 — сектор 512 байт

Например, в гибких дисках для персональных компьютеров ближайшая к краю дорожка, считающаяся нулевой, зарезервирована для операционной системы, а первый сектор этой дорожки содержит загрузочную запись, которая необходима для начала работы.

Пятидюймовые диски действительно гибкие. Они помещаются в бумажный конверт, который не защищает диск от изгиба и повреждения записи. В трехдюймовых дискетах имеется достаточно жесткий пластмассовый корпус, причем доступ к находящемуся внутри него диску защищен металлической задвижкой, которая открывается, когда дискета вставляется в дисковод, и вновь закрывается при вынимании дискеты.

Перед тем как использовать диск для записи, он должен быть отформатирован. В ходе форматирования магнитная головка записывает в определенных местах диска метки дорожек и секторов. После форматирования диска трехдюймовой дискеты поверхность его оказывается поделенной на 18 секторов, а посылку для записи используются обе стороны, то общая информационная емкость такой дискеты составляет $512 \text{ байт} \times 18 \times 80 \times 2 = 1\,474\,560 \text{ байт} = 1\,474\,560/1\,024 \text{ Кбайт}$. Плотность записи (как радиальная, так и линейная) в такой дискете выше, чем в пятидюймовых дисках. Благодаря лучшей защищенности и большей информационной емкости трехдюймовые дискеты вытесняют (а в персональных компьютерах уже вытеснили) пятидюймовые диски.

12.6. Жесткие диски

Принцип записи информации на жесткий диск и ее воспроизведения тот же, что и в гибких дисках. Сам термин «жесткий диск» объясняется его конструкцией. Диск представляет собой жесткую круглую пластину из металла (обычно алюминия). В большинстве ЗУ эти диски не могут заменяться, поэтому такие ЗУ иногда называют накопителями на фиксированных дисках.

Все накопители информации на жестких дисках обычно выполнены в виде неразборного блока, включающего в себя сам диск (одну или несколько пластин), магнитные головки для считывания и записи, привод головок с электродвигателем, электродвигатель со шпинделем для вращения диска, вентилятор с приводным электродвигателем, электронную плату.

Основными элементами накопителя являются вращающиеся с большой скоростью диски и магнитные головки, плавающие над и под дисками. Диски и головки с их приводом заключены в герметичную камеру, называемую сборкой жесткого диска, или *винчестером* (рис. 12.14).

Типовой жесткий диск содержит одну или несколько круглых пластин толщиной чуть более 3 мм. Эти пластины покрыты оксид-

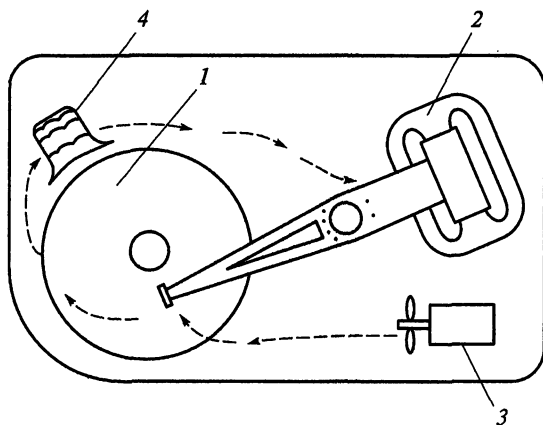


Рис. 12.14. Сборка жесткого диска:

1 — пластина (носитель информации); 2 — магнитные головки с приводом; 3 — вентилятор; 4 — воздушный фильтр

ным носителем или носителем в виде тонкой пленки. Толщина оксидного покрытия составляет примерно 75 мкм. Пленка еще тоньше (5 мкм) и имеет более высокую плотность записи. В компьютерах используются двусторонние и односторонние диски диаметром 89 мм. При раскрутке диска между магнитной головкой и поверхностью диска возникает давление воздуха, которое отжимает головку от диска на 25...50 мкм (для пластин с напыленным носителем — на 15 мкм). Грязь и мелкие пылинки при этом совершенно недопустимы. Собирают ЗУ на жестких дисках в помещениях 100-го класса чистоты, при котором в одном литре воздуха должно быть не более 30 частиц размером 0,5 мкм. (Человек выдыхает примерно 500 таких частиц в минуту.) Внутри герметичной камеры имеются воздушные фильтры, на которых оседают частицы, отделившиеся при соприкосновении головок с пластинами.

Привод всех пластин осуществляется от одного общего электродвигателя постоянного тока. Вал электродвигателя непосредственно связан с валом, на котором закреплены дисковые пластины (т.е. без ременной или зубчатой передачи). Во многих накопителях частота вращения составляет 3600 мин^{-1} , но бывает и вдвое больше. Частота вращения должна поддерживаться с высокой стабильностью. Для этого используется система автоматического регулирования с обратной связью по скорости вращения. Кроме требования регулируемости к электродвигателям привода дисков предъявляются требования малошумности и отсутствия помех. Электродвигатель может быть расположен вне герметичной сборки или внутри ее. Например, фирма Махтог выпускает накопители с встроенным электродвигателем, имеющим вращающую-

юся наружную часть, на которой закреплены восемь дисковых пластин. Инерционность привода при этом возрастает, время разгона достигает 15 с. Для снятия заряда статического электричества вертикальный вал двигателя упирается в подпятник через медно-графитовый контакт.

Для привода магнитных головок в накопителях на жестких дисках используются два типа электродвигателей: шаговый и линейный. Привод от шагового двигателя представляет собой цилиндр диаметром 2...3 см или плоский сравнительно тонкий диск диаметром 8...10 см. Обычно шаговый двигатель расположен вне герметичной сборки и связан с блоком головок посредством двойной стальной ленты (безлюфтовая передача) или зубчатой передачей (точность позиционирования на дорожке при этом невысока).

Недостатком шагового двигателя является ошибка позиционирования из-за нагрева. Дело в том, что при изменении температуры шаг (т. е. минимальное перемещение) остается прежним, а вот размеры диска и, следовательно, расположение дорожек меняются.

Этого недостатка лишен линейный электродвигатель (его иногда называют соленоидным приводом), снабженный системой автоматического позиционирования. По принципу действия линейный двигатель аналогичен магнитоэлектрическому механизму: катушка с током (соленоид) взаимодействует с магнитным полем постоянного магнита. У линейного двигателя нет фиксированных положений; перемещение его подвижной части (соленоида) прекращается тогда, когда электромагнитная сила уравновешивается противодействующей силой. Линейный двигатель может занять любое положение, поэтому для установки магнитной головки на дорожку применяется так называемая специальная свободная поверхность. Одна сторона какой-то из пластин не используется для записи информации, а на нее записываются специальные индексные метки, которые указывают позиции, соответствующие дорожкам (трекам) на всех остальных пластинах винчестера. Для более точной установки в системе автоматического позиционирования используется настройка на максимальный уровень сигнала.

На рис. 12.15 показан блок магнитных головок 1 жестких дисков, которые перемещаются линейным двигателем, состоящим из подвижной обмотки 2 и создающего магнитное поле постоянного магнита 3, размещенных на магнитопроводе 4.

В приводе головок обычно предусмотрен режим автопарковки: при снятии питания блок головок передвигается на посадочную поверхность посредством пружин. Режим автопарковки предотвращает абразивный износ жестких дисков, поскольку исключается механическое касание головок и пластин (в зоне записи информации). Однако посадочная поверхность все-таки

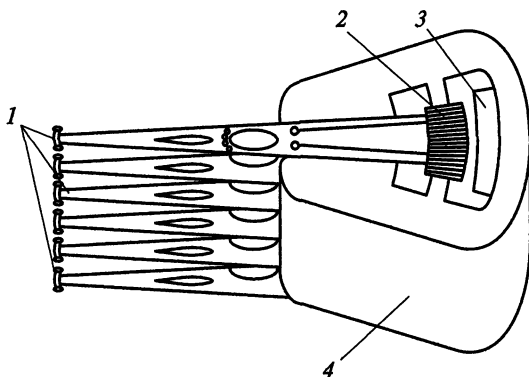


Рис. 12.15. Блок магнитных головок с линейным приводом:

1 — магнитные головки; 2 — обмотка; 3 — постоянный магнит; 4 — магнитопривод

истирается. Для удаления появляющейся пыли используются миниатюрный воздушный фильтр и вентилятор. Линейный привод надежнее шагового, но он примерно в полтора раза дороже. Среднее время наработки на отказ накопителей на жестких дисках составляет десятки тысяч часов, а время доступа — десятки микросекунд.

Информационная емкость ЗУ на жестких дисках может достигать десятков гигабайт. Для удобства размещения информации на жестком диске он может быть условно разбит на несколько так называемых логических дисков.

12.7. Магнитооптические и оптические диски

К магнитооптическим относятся ЗУ, у которых запись или считывание выполняется с помощью светового луча, а хранение и представление информации — с помощью магнитных носителей.

В магнитооптических дисках используют пленочные ферромагнитные материалы, запись на которые осуществляется локальным нагревом участка среды до температуры, при которой может произойти перемагничивание. Наиболее распространена запись с нагревом до температуры, близкой к так называемой точке Кюри. Известно, что при нагреве выше точки Кюри материал переходит из ферромагнитного состояния в парамагнитное. При последующем охлаждении и обратном переходе перемагничивание участка в нужном направлении может осуществляться сравнительно слабым магнитным полем.

Информационный носитель для магнитооптического дискового ЗУ выполняется в виде двухслойной системы — прозрачной подложки и нанесенной на нее магнитной пленки. В качестве ма-

териала носителя наиболее широко используют марганец-висмутую пленку, обладающую способностью намагничиваться перпендикулярно плоскости пленки. Запись осуществляется воздействием лазерного луча в магнитном поле. Если пленку поместить в однородное магнитное поле, недостаточное для изменения магнитного состояния, то после нагрева ее участка выше точки Кюри, этот участок приобретает намагниченность, определяемую внешним полем. Запись информации производится с высокой скоростью — за несколько наносекунд.

Считывание информации в магнитооптическом ЗУ происходит путем анализа плоскости поляризации света при его отражении или прохождении через пленку. Поворот плоскости поляризации при магнитооптическом считывании определяется анализатором, интенсивность света на выходе которого регистрируется фотодатчиком. Направление вращения плоскости поляризации зависит от направления вектора намагниченности магнитной пленки (таким образом различают 1 и 0).

Информацию можно записывать и на многослойных пленках, компоненты которых обладают различными значениями точки Кюри. Если нагреть локальную область такой пленки до температуры, которая превышает температуру, соответствующую точке Кюри одного из слоев, коэрцитивная сила системы слоев уменьшится до более низкого уровня. Это даст возможность с помощью внешнего магнитного поля, которое превышает этот уровень, записать информацию в определенную точку, не воздействуя на остальные.

Запись информации на оптический диск (рис. 12.16) осуществляется с помощью луча лазера 1, который с помощью оптической системы, состоящей из полупрозрачного зеркала 2 и объектива 4, фокусируется на поверхность вращающегося диска 3 и оставляет на этой поверхности метки микронных размеров, которые при считывании фиксируются фотоприемником 5. Обычно применяется полупроводниковый лазер, излучающий в ближней инфракрасной области спектра. Мощность излучения такова, что метка образует за десятки наносекунд. Благодаря этому достигается не

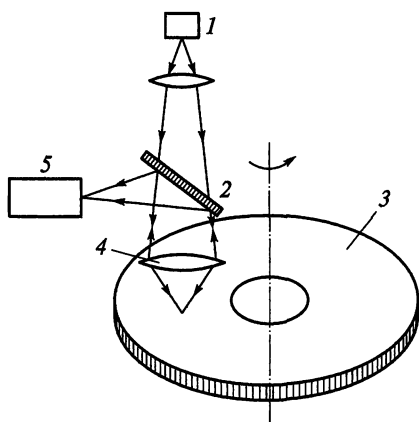


Рис. 12.16. Устройство записи информации на оптический диск и ее считывания:

1 — лазер; 2 — полупрозрачное зеркало; 3 — оптический диск; 4 — объектив; 5 — фотоприемник

только высокая скорость записи, но и отсутствие «размазывания» метки вдоль дорожки при вращении диска. При считывании информации на диск направляется луч того же лазера, но уменьшенной мощности (метка при этом не изменяется). Этот луч «ощупывает» вращающийся диск и по-разному отражается от меток и промежутков между ними. Отраженный луч с помощью оптической системы направляется на фотоприемник, в котором он и преобразуется в электрические сигналы.

Для оптических дисков используются различные материалы и методы нанесения на них меток. Наиболее простой и распространенный метод записи — прожигание микроотверстий в тонкой пленке металла. В месте фокусировки луча пленка металла (обычно теллура) локально разогревается до температуры плавления, и в ней образуется сквозное отверстие. На пленке толщиной около 0,05 мкм удается получать отверстия диаметром до 0,5 мкм. Наличие в данном месте диска отверстия — это и есть запись 1, отсутствие — 0. Оптические диски характеризуются высокой плотностью записи. Теоретически площадь выжигаемой метки составляет примерно квадрат длины волны лазера, т.е. при инфракрасном лазере можно добиться плотности более ста миллионов бит на 1 см². На практике достигается плотность в тысячи раз меньшая, но вполне достаточная для размещения на одном диске нескольких гигабит информации.

Применяемые в качестве накопителей информации для персональных компьютеров лазерные диски называют также компакт-дисками, или CD. Поскольку в ЗУ могут использоваться несколько десятков оптических дисков, общая емкость накопителя может составлять сотни гигабит. Возможность многократной записи для оптических дисков пока не реализована, зато по диску-оригиналу можно тиражировать в больших количествах копии с полностью идентичной информацией.

Диски DVD (Digital Video Disk — цифровой видеодиск) имеют такие же размеры, как и CD, аналогично изготавливаются, записывают и воспроизводят информацию. Внешне они неразличимы. Главное отличие DVD состоит в том, что на них можно записать значительно больше информации и воспроизвести ее с существенно большей скоростью. Основные параметры дисков DVD и CD приведены в табл. 12.1.

Информация записывается на спиральную дорожку, рельеф которой после записи представляет собой чередование плоских участков пластмассовой подложки и микроскопических углублений в ней с одинаковой глубиной и шириной, но с различной длиной. Эти углубления называются *питами* (от *англ.* pit — углубление, ямка). Последовательность питов с различными длиной и расстоянием между ними и представляет собой запись информации в двоичном коде.

Основные параметры дисков DVD и CD

Параметр	DVD	CD
Диаметр, см	12	12
Толщина подложки, мм	0,6	Менее 1,2
Толщина диска, мм	1,2	1,2
Длина волны лазера, нм	635 или 650 (красный)	780 (инфракрасный)
Диаметр фокального пятна лазера, мкм	0,4	0,83
Шаг витков дорожки, мкм	0,74	1,6
Минимальная/максимальная длина питов, мкм	0,4/2,13	0,833/3,560
Скорость считывания, м/с	3,5	1,3
Цифровой поток, Мбит/с	10	1,4
Информационная емкость диска, Гбайт	4,7...17	0,65

При изготовлении DVD (рис. 12.17, *a*) питы 1 формируются на пластмассовой подложке 2. На ту сторону подложки, где они находятся, наносят отражательный слой 3 из алюминия. В простейшем одностороннем DVD на отражательный слой наносят прозрачный холостой слой 4 также из пластмассы.

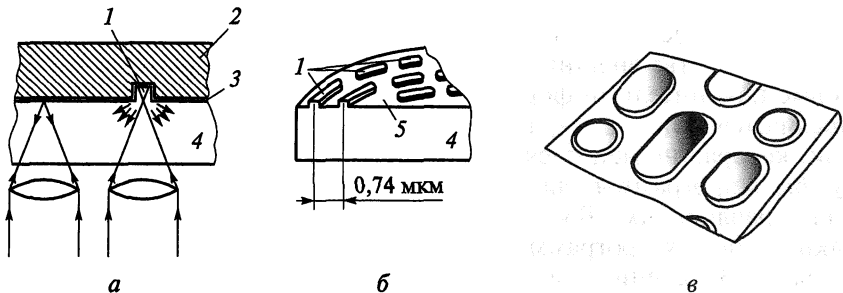


Рис. 12.17. Участок диска DVD:

a — разрез; *b* — вид базовой плоскости; *в* — сильно увеличенный фрагмент; 1 — питы; 2 — пластмассовая подложка; 3 — отражательный слой; 4 — прозрачный холостой слой; 5 — базовая плоскость

Таким образом, слой с рельефными дорожками, несущими информацию, оказывается не на поверхности, а внутри диска. От одной внешней поверхности он защищен холостым слоем, а от другой — подложкой. Если смотреть на дорожку сверху (рис. 12.17, б), то питы оказываются дискретными выступами на базовой плоскости 5. Питы при большом увеличении показаны на рис. 12.17, в (вид с противоположной стороны по сравнению с рис. 12.17, б).

Для считывания информации используется остросфокусированный луч лазера, который направлен снизу через прозрачный холостой слой 4, как показано на рис. 12.17, а. При попадании луча на плоскую базовую поверхность подложки отраженный луч лазера почти полностью возвращается в объектив лазерной головки. При попадании луча лазера в питу происходит его рассеивание и в объектив возвращается значительно меньший по интенсивности луч. Воспринимаемый светоприемником отраженный луч преобразуется в электрический сигнал в двоичном коде.

Воспроизведение информации не приводит к износу диска, а небольшие потертости, царапины, пылинки на его поверхности мало влияют на качество диска, поскольку эти повреждения могут находиться только на относительно большом удалении (0,6 мм) от места, которое находится в фокусе оптической системы. Поэтому лазерные диски значительно качественнее ленточных ЗУ и превосходят все остальные ЗУ по объему сохраняемой информации. Кроме простых дисков DVD с однократной нестираемой записью, сделанной на заводе-изготовителе, имеются диски типов DVD-R (однократную запись может выполнять пользователь), а также DVD-RW и DVD-RAM, на которые можно многократно записывать информацию, воспроизводить и стирать ее.

12.8. Полупроводниковые энергонезависимые запоминающие устройства

В последнее время наряду с жесткими дисками и компакт-дисками в качестве внешних ЗУ стали применяться и полупроводниковые накопители информации. Обычно полупроводниковые ЗУ предназначаются для хранения постоянной информации, т. е. такой, которая не меняется в ходе решения задач на ЭВМ. Это могут быть программы для выполнения типовых задач, характерных для управляющих ЭВМ специального назначения. После отладки таких типовых программ на универсальных ЭВМ они записываются в ПЗУ специализированных ЭВМ. К последним относятся, например, бортовые ЭВМ, управляющие движением различных объектов от морских судов до самолетов и космических аппаратов. Информация записывается раз и навсегда, при этом используются методы создания контактных перемычек в нужных местах матрицы полупроводниковых элементов или, наоборот, удаления

таких перемычек их выжиганием. Получается ЗУ, которое более надежно, чем дисковое, поскольку не имеет движущихся частей. В то же время информация в таком ЗУ сохраняется и после отключения питания, что невозможно, например, при использовании полупроводниковых триггерных регистров.

В последнее время появились и полупроводниковые накопители информации, позволяющие многократно стирать и записывать информацию, а также одновременно сохранять ее при отсутствии электропитания, т. е. энергонезависимые накопители. Речь идет о так называемой *флэш-памяти*.

Флэш-память, созданная в конце 1980-х годов фирмой Intel является представителем класса перепрограммируемых постоянных ЗУ с электрическим стиранием. Однако стирание в ней осуществляется сразу из целой области ячеек или всей микросхемы. Это обеспечивает более быструю запись информации или, как иначе называют эту процедуру, программирование ЗУ. Флэш-память строится на одностранзисторных элементах памяти, что обеспечивает плотность хранения информации даже несколько выше, чем в динамической оперативной памяти. Существуют различные технологии построения базовых элементов флэш-памяти, разработанные ее основными производителями. Эти технологии отличаются числом используемых слоев, методами стирания и записи данных, а также структурной организацией. Наиболее широко известны NOR и NAND типы флэш-памяти, запоминающие транзисторы в которых подключены к разрядным шинам соответственно параллельно и последовательно. Первый тип имеет относительно большие размеры ячеек и быстрый произвольный доступ (около 70 нс), что позволяет выполнять программы непосредственно из этой памяти. Второй тип имеет меньшие размеры ячеек и быстрый последовательный доступ (обеспечивается скорость передачи до 16 Мбайт/с), что более пригодно для построения устройств блочного типа.

Способность сохранять информацию при выключенном питании, малые размеры и высокая надежность привели к широкому распространению флэш-памяти. К недостаткам данного вида памяти можно отнести сравнительно низкую скорость передачи данных и относительно высокую стоимость устройств с большой емкостью. Элементы флэш-памяти организованы в матрицы, как и в других видах полупроводниковой памяти. Разрядность данных для микросхем составляет 1 или 2 байт. Операция чтения из флэш-памяти выполняется как в обычных ЗУ с произвольным доступом (оперативных ЗУ или кэш). Однако запись имеет некоторые особенности, свойственные постоянным ЗУ. Перед записью данных в ЗУ ячейки, в которые будет производиться запись, должны быть очищены (стерты). Стирание заключается в переводе элементов памяти в состояние 1, что возможно сразу для целого блока ячеек

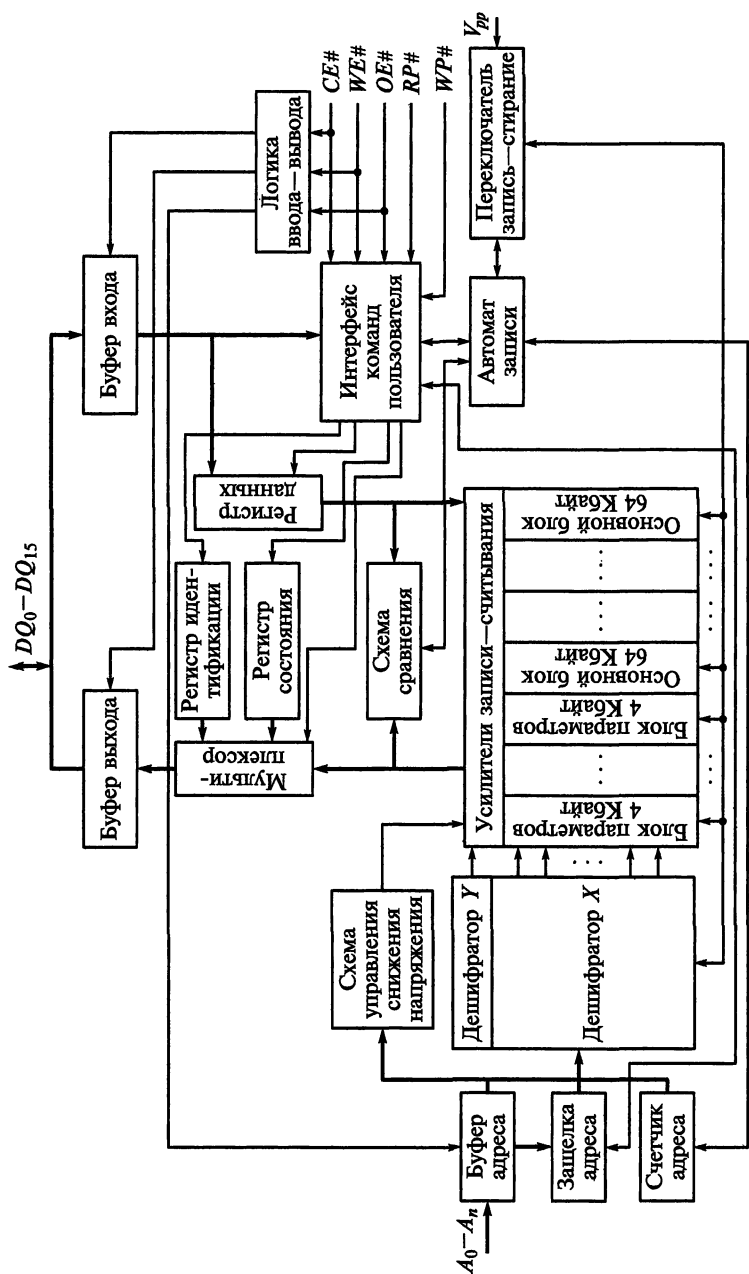


Рис. 12.18. Структурная схема флэш-памяти с асимметричной архитектурой

(в первых микросхемах предусматривалось стирание только из всей матрицы сразу). Выборочное стирание невозможно.

Разбиение адресного пространства микросхемы флэш-памяти на блоки обычно бывает двух видов: симметричное и асимметричное. В первом случае, называемом также Flash File, все блоки имеют одинаковый размер, например 64 или 128 Кбайт. В случае асимметричной архитектуры, называемой также Boot Block, один из блоков, на которые разбито адресное пространство микросхемы, дополнительно разбивается на меньшие блоки. Например, один из 128 блоков по 64 Кбайт каждый выделяется как загрузочный и разбивается на восемь блоков параметров по 8 Кбайт. Этот загрузочный блок может размещаться либо в начале, либо в конце адресного пространства микросхемы. Структурная схема флэш-памяти с асимметричной архитектурой показана на рис. 12.18. В этой схеме управляющий сигнал $WP\#$ (Write Protect) используется для исключения возможности случайной записи по командам программы, а сигнал $RP\#$ (Reset/Deer Power Down) применяется для управления записью, закрывая все блоки для записи при единичном уровне. Остальные управляющие сигналы аналогичны одноименным сигналам в других типах памяти. На вход V_{pp} подается напряжение, необходимое для ускорения операций стирания и записи данных.

Флэш-память используется для различных целей. Непосредственно в самой ЭВМ эту память применяют для хранения базовой системы ввода — вывода. Достаточно широко она используется для построения так называемых твердотельных дисков, эмулирующих работу внешних винчестеров.

Твердотельный диск типа флэш-памяти (рис. 12.19) имеет габаритные размеры примерно $70 \times 20 \times 10$ мм, подключается обычно к шине USB и состоит из собственно флэш-памяти, эмулятора контроллера дисководов и контроллера шины USB. При подключении к ПК в систему (допускаются «горячее» подключение и временное отключение) устройство с точки зрения пользователя ведет себя как обычный (съёмный) жесткий диск. Производительность его меньше, чем у современного жесткого диска; скорость

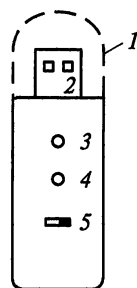


Рис. 12.19. Внешний вид твердотельного диска типа флэш-памяти:

1 — защитный колпачок (снят); 2 — USB-разъем; 3 — индикатор подключения; 4 — индикатор работы (чтение — запись); 5 — переключатель защиты от записи

передачи при записи и чтении составляет менее 1 Мбайт/с. Уже созданы диски, емкость которых превышает 1 Гбайт.

Флэш-память находит также применение в разных модификациях карт памяти, которые используются не только в компьютерах разных классов, но и в цифровых видео- и фотокамерах, плеерах, телефонах, музыкальных центрах и другой медиатехнике.

Контрольные вопросы

1. Каково назначение запоминающих устройств?
2. Какие показатели характеризуют запоминающее устройство?
3. В чем разница между ОЗУ и ПЗУ?
4. На каком физическом явлении основана магнитная запись?
5. Как устроен гибкий диск?
6. Как устроен жесткий диск?
7. Поясните принцип работы оптических дисков.
8. В чем разница между CD и DVD?
8. Как подключается флэш-память к компьютеру?

ОРГАНИЗАЦИЯ ИНТЕРФЕЙСОВ В ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ

13.1. Назначение и характеристики интерфейса

Вся обработка информации (включая вычисления), осуществляемая ЭВМ, происходит в процессоре, который взаимодействует с другими устройствами. Некоторые из этих устройств входят в состав ЭВМ, а некоторые не входят, т.е. являются внешними, или периферийными. Для обмена информацией между этими устройствами необходимы специальные схемы, программы и правила, которые получили название *средства интерфейса*. Интерфейс в буквальном переводе с английского означает «внешнее лицо», т.е. то представление, которое получает внешний наблюдатель. Прежде всего таким наблюдателем является оператор ЭВМ, вводящий исходные данные, программы, команды на выполнение задач. Он же получает информацию о ходе и итогах выполнения задач, результатах вычислений и обработки информации. Устройства, с помощью которых эти сведения вводятся и выводятся, называются устройствами ввода — вывода. Они также входят в понятие «интерфейс». Но к интерфейсу относится и то, что воспринимается не только человеком-оператором. Все источники и потребители информации, циркулирующей внутри ЭВМ, а также между ЭВМ и соединенными с ней устройствами, тоже функционируют посредством интерфейса. Поэтому интерфейс является достаточно широким понятием. Это комплекс аппаратных и программных средств, обеспечивающих обмен информацией в вычислительной технике.

Работа компьютера сопровождается интенсивным обменом информацией между МП, памятью (различными ЗУ) и УВВ. В процессе выполнения программы МП принимает команды из ЗУ команд, расшифровывает их, при исполнении команд, включающих чтение и запись, обращается к ЗУ данных, а при исполнении команд ввода — вывода — к УВВ. Эффективность решения задачи оператором на компьютере в значительной степени определяется организацией этого обмена и структурой связей между МП, памятью и УВВ.

Система шин, вспомогательной аппаратуры и алгоритмов, реализованных на этом оборудовании, предназначенная для организации обмена между МП, памятью и УВВ, и называется интерфейсом. В функции интерфейса входят дешифрование адреса устройства, синхронизация обмена информацией, согласование форматов слов, дешифрование кода команды, связанной с обра-

щением к памяти или УВВ, электрическое согласование сигналов и некоторые другие операции.

Сложность задач, возлагаемых на интерфейс, а также недостаточная мощность буферных схем, входящих в состав БИС МП, привели к распределению средств интерфейса между различными устройствами:

- устройством управления памятью и вводом — выводом, входящим в состав МП;
- непосредственно интерфейсным устройством, являющимся промежуточным звеном между МП, с одной стороны, и памятью и УВВ, с другой;
- специализированными устройствами управления (контроллерами) УВВ, предназначенными для реализации алгоритмов управления.

Организация обмена между МП и памятью или УВВ в простейших случаях возможна на основе средств, содержащихся только в МП. Недостающие средства в таких случаях реализуются программно. Более сложные ЗУ и УВВ соединяются с МП обязательно через дополнительные интерфейсные устройства.

Существуют три способа организации связи между МП и УВВ:

- 1) программно-управляемая передача данных;
- 2) использование прерываний;
- 3) прямой доступ к памяти.

Прежде чем подробно проанализировать эти способы, остановимся на организации связи между МП и ЗУ данных и команд.

Рассмотрим некоторые особенности организации связи МП с памятью. Под памятью будем понимать внешние по отношению к МП запоминающие устройства (в отличие от внутренних РОН, которые могут рассматриваться как СОЗУ) с произвольным доступом, выполняющие две функции:

- 1) хранение команд, под управлением которых работает МП;
- 2) хранение данных для обработки в соответствии с управляющим алгоритмом, т. е. исходных данных, промежуточных и окончательных результатов обработки.

Данные обычно хранятся в оперативном ЗУ (ОЗУ), а команды — в постоянном ЗУ (ПЗУ) или полупостоянном ЗУ (ППЗУ). Внешние выводы отдельных больших интегральных схем ЗУ, через которые МП подключается к памяти, представлены для ОЗУ на рис. 13.1, а, для ПЗУ — на рис. 13.1, б. В ОЗУ показаны две шины данных — входная и выходная. Однако многие ОЗУ имеют одну двунаправленную шину данных, направление передачи информации по которой определяется значением подаваемого на вход U управляющего сигнала $Zn/Чт$. В режиме записи информации ($Zn/Чт = 1$) эта шина работает как входная, а в режиме чтения ($Zn/Чт = 0$) — как выходная.

Интегральные схемы ЗУ имеют также группу управляющих входов выборки кристалла (BK), которые могут быть прямыми или

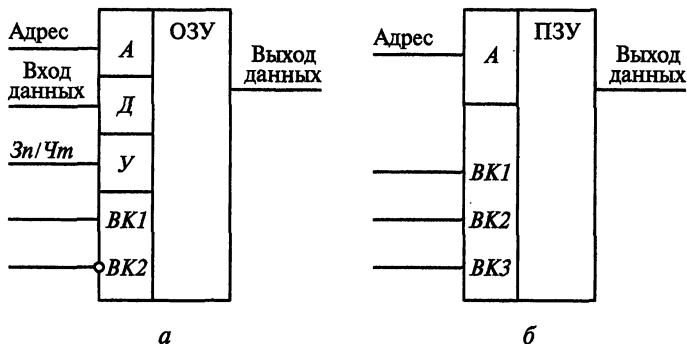


Рис. 13.1. Организация связи (интерфейса) микропроцессора с памятью:
а — оперативной; *б* — постоянной

инверсными. Эти входы являются входами схемы И, разрешающей обращение к БИС ЗУ, поэтому ОЗУ, показанное на рис. 13.1, *а*, может записывать или выдавать информацию при $ВК1 = 1$ и $ВК2 = 0$, а ПЗУ, показанное на рис. 13.1, *б*, — выдавать информацию лишь при $ВК1 = ВК2 = ВК3 = 1$. Выводы *ВК* предназначены для разрешения обращения к данной БИС при наличии нескольких из них. Различные варианты БИС ЗУ различаются числом входов *ВК* и наличием или отсутствием инвертирующих схем по отдельным входам *ВК*.

Многомодульная организация является типовым способом организации памяти микроЭВМ, поскольку емкость БИС (модуля) ЗУ, как правило, меньше необходимой емкости памяти. Пример многомодульной организации памяти, содержащей *k* модулей ЗУ (МЗУ), с двунаправленной шиной данных представлен на рис. 13.2. Для обмена информацией между МП и памятью требуются адресация памяти, подача сигналов *Зн/Чт* и синхронизация работы памяти и МП.

Интерфейс ввода—вывода служит буфером между системной шиной и внешними устройствами. Используя интерфейс ввода—вывода, можно проектировать внешние устройства независимо от структуры шины, совместно с которой он будет использоваться. Для подключения одного и того же ЗУ к двум шинам с различной структурой нужно изменить только интерфейс.

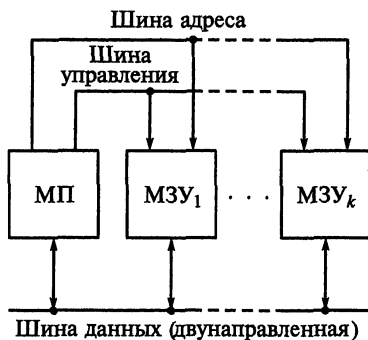


Рис. 13.2. Интерфейс с двунаправленной шиной (многомодульная организация памяти)

В общем случае процессор ЭВМ получает данные от нескольких источников (клавиатуры, датчики, устройства хранения данных и программ, другие процессоры и ЭВМ). Этот же процессор передает информацию на мониторы (дисплеи), прочие индикаторные устройства, ЗУ, вычислительные блоки, управляющие элементы системы. В тех устройствах, которые предназначены для оператора, информация выводится в удобной для человека форме: буквенно-цифровой текст и таблицы появляются на экране или бумаге; графики, схемы, рисунки, чертежи — на бумаге; звуковые сигналы, речь или музыка звучат из динамиков. В остальных устройствах информация проходит в форме электрических сигналов, причем не только в виде двоичных кодовых комбинаций, но и в виде аналогового (непрерывного) сигнала.

Передача данных в процессор и из него может выполняться параллельно или последовательно. При параллельном способе для каждого передаваемого разряда двоичного сообщения имеется отдельная физическая линия (провод), а при последовательном способе все разряды передаются по единственной линии один за другим, по очереди.

При обоих способах передачи данных процессор соединяется с внешними по отношению к нему устройствами. Возникают пара источник — приемник и линия связи между ними, характеристики которой могут неблагоприятно влиять и на источник информации, и на приемник. Согласованная работа всех составляющих является задачей интерфейса.

13.2. Параллельный интерфейс

При параллельном интерфейсе одновременно по нескольким параллельным проводам передается несколько бит информации аналогично тому, как по многоразрядной шине происходит обмен данными между процессором и памятью. Поэтому параллельный интерфейс может быть 8-, 16-битным и т.д. Поскольку для передачи каждого бита используется один разряд в двоичной комбинации, то можно говорить «8-, 16-разрядный интерфейс». При параллельной передаче нескольких разрядов достигается более высокая скорость передачи информации, чем при последовательной. Для характеристики интерфейса по допустимой скорости передачи используется понятие «пропускная способность» интерфейса. Она достигает нескольких десятков мегабайт в секунду. Для увеличения пропускной способности необходимо увеличивать или число разрядов (бит), передаваемых параллельно, или тактовую частоту передачи.

Интерфейс микропроцессорных шин адреса и данных представляет собой специальные аппаратные средства, в которые (или из которых) можно пересылать данные с помощью команд чте-

ния (или записи) памяти и чтения (или записи) сигналов УВВ. Обычно для задания режима работы интерфейсного устройства используются команды, задающие направление передачи информации и другие параметры пересылки. Эти команды управляют работой соответствующих регистров, входящих в состав аппаратуры параллельного ввода — вывода. Указанные регистры называются *портами*. Подсоединение к порту выполняется через соответствующий штепсельный разъем. В компьютере имеется несколько параллельных портов, через которые подключаются периферийные устройства, обозначаемые, например, LPT1, LPT2. Параллельные интерфейсные устройства сходного назначения имеют разные названия: «устройство параллельного ввода — вывода», «периферийно-интерфейсный адаптер», «универсальный интерфейс адаптер».

На рис. 13.3 приведена структурная схема параллельного интерфейсного устройства, состоящего из единственного порта и управляющего регистра. Данные, загружаемые в управляющий регистр, определяют назначение каждого внешнего контакта порта, т. е. используется этот контакт для ввода или вывода. Подача специального входного сигнала указывает направление передачи: между процессором и портом или между процессором и управляющим регистром.

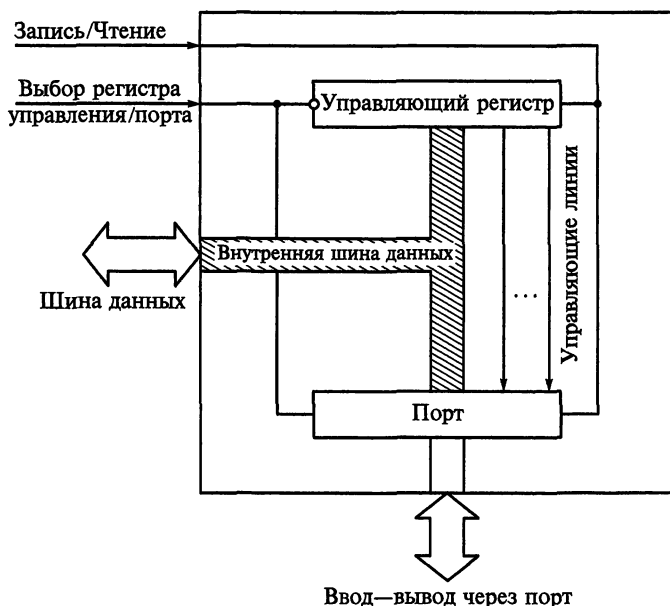


Рис. 13.3. Структурная схема параллельного интерфейсного устройства ввода — вывода

Выбор устройства передачи осуществляется выходными сигналами дешифратора, подключенного к адресной шине внутри системного блока или к шине адресов ввода—вывода, причем одновременно не может быть выбрано более одного устройства. Подобные устройства могут применяться для считывания состояния переключателей и для вывода информации на элементы с двумя состояниями, такие как световые индикаторы, реле, пускатели.

При более сложных пересылках данных в схеме ввода—вывода обычно используются сигналы, характерные для режимов работы с квитированием: сигнал готовности данных и сигнал-квитанция, служащий для подтверждения приема. На рис. 13.4 показано применение этих сигналов для управления передачей (пересылкой) данных между двумя системами (передатчиком и приемником, которые могут меняться ролями). Передатчик изменяет значения данных на выходных линиях и после небольшой задержки, в течение которой происходит установка этих значений, информирует приемник с помощью сигнала готовности данных *ДГ* о том, что данные для пересылки готовы. Приемник воспринимает данные и сообщает об этом посредством короткого сигнала-квитанции *КВ*

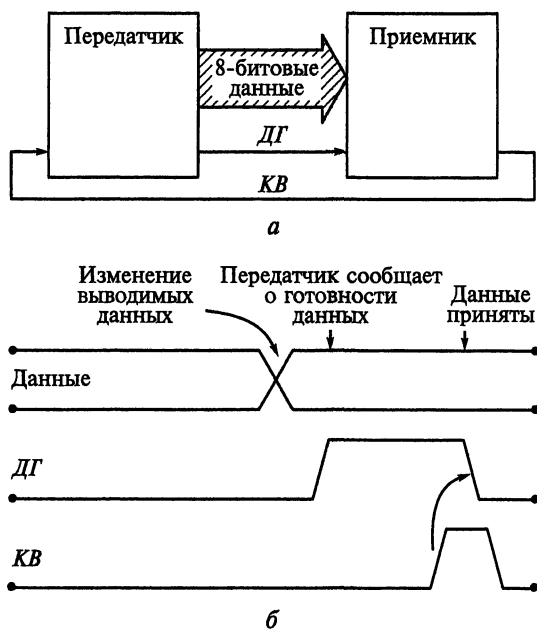


Рис. 13.4. Квитирование (управление передачей данных) при параллельных пересылках:

а — схема пересылки; *б* — временная диаграмма

положительной полярности. Сигнал *KВ* может быть использован в передатчике для сброса сигнала *ДГ*, а в некоторых случаях и для выработки сигнала прерывания, сообщающего процессору о возможности вывода очередного элемента данных.

Параллельный интерфейс с квити́рованием впервые был применен фирмой Hewlett Packard для совместной работы ЭВМ с различными электроизмерительными приборами, поэтому его иногда называют интерфейсной шиной НРІВ. Каждое устройство, подсоединенное к шине, имеет свой адрес. По этому адресу контроллер, управляющий работой шины, выбирает устройства, между которыми проводится пересылка данных. Одно из них назначается передатчиком, а другое (или несколько других) — приемником. Кроме того, определяются устройства, которые не будут участвовать в пересылке. Скорость пересылки определяется быстродействием самого медленного из устройств, подключенных к шине на время выполнения данной конкретной пересылки. Поэтому определение устройств, не участвующих в данной пересылке, позволяет избежать снижения ее скорости из-за участия медленных устройств, которым эти передаваемые данные не нужны. Например, принтер по сравнению с внешними дисковыми ЗУ принимает данные значительно медленнее. Если целью пересылки данных является запись в ЗУ, то незачем передавать их с такой скоростью, чтобы их успевал принимать и принтер.

13.3. Последовательный интерфейс

Последовательный интерфейс не обеспечивает столь высокой скорости передачи информации, как параллельный, тем не менее, он достаточно часто применяется в вычислительной технике. Объясняется это тем, что с помощью параллельного интерфейса экономически эффективно передавать информацию лишь на короткие расстояния. При значительном удалении источника от приемника возрастает стоимость кабелей и буферных устройств, что существенно повышает стоимость всей системы в целом.

Последовательный интерфейс позволяет передавать информацию, содержащую много бит, по одной линии (т. е. по одной паре проводов). Рассмотрим принцип работы последовательного интерфейса, пользуясь структурной схемой, показанной на рис. 13.5. Сначала источник данных загружает информацию в сдвиговый регистр с параллельной записью, запускает генератор тактовых импульсов и счетчик. Каждым тактовым сигналом данные в регистре сдвигаются на одну позицию вправо и поступают при этом на линию данных. Приемник состоит из еще одного сдвигового регистра, счетчика и логической схемы, которые управляются теми же тактовыми сигналами. После того как счетчик фиксирует поступление необходимого числа тактовых импульсов, он может

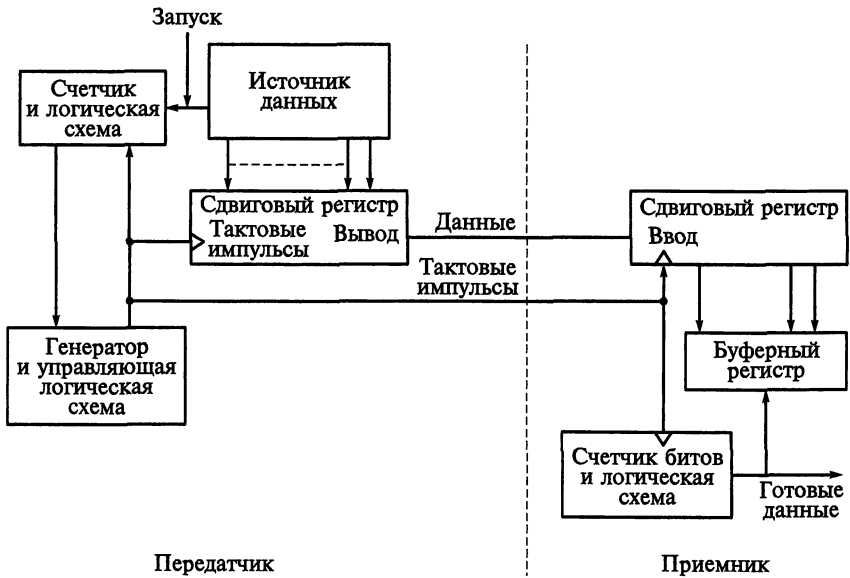


Рис. 13.5. Структурная схема синхронной последовательной передачи данных

инициировать параллельную передачу данных из сдвигового регистра в буферный. Этот метод последовательной передачи данных является одним из видов синхронной передачи, при которой вместе с данными необходимо передавать тактовые сигналы. Для такой последовательной передачи не требуется несколько параллельных линий передачи данных — достаточно одной, но еще одна линия нужна для передачи тактовых сигналов.

При асинхронной последовательной передаче не требуется и отдельная линия для тактовых сигналов. Структурная схема такой передачи показана на рис. 13.6. Оба сдвиговых регистра, и на стороне источника, и на стороне приемника, управляются тактовыми импульсами, но на каждой стороне имеется собственный генератор тактовых импульсов. Оба генератора должны вырабатывать импульсы одинаковой частоты. Для того чтобы информировать приемник о начале передачи, по линии посылается так называемый стартовый бит. Затем передается блок данных определенного размера (например, 8 бит, начиная с младшего). Далее информация из сдвигового регистра приемника передается в его буферный регистр (на это выделяется определенное время), после чего можно передавать следующий блок данных, снова предварительно пошлав стартовый бит. Для асинхронной последовательной передачи устанавливается определенная стандартная частота. Поскольку при передаче одного блока может возникнуть ошибка в

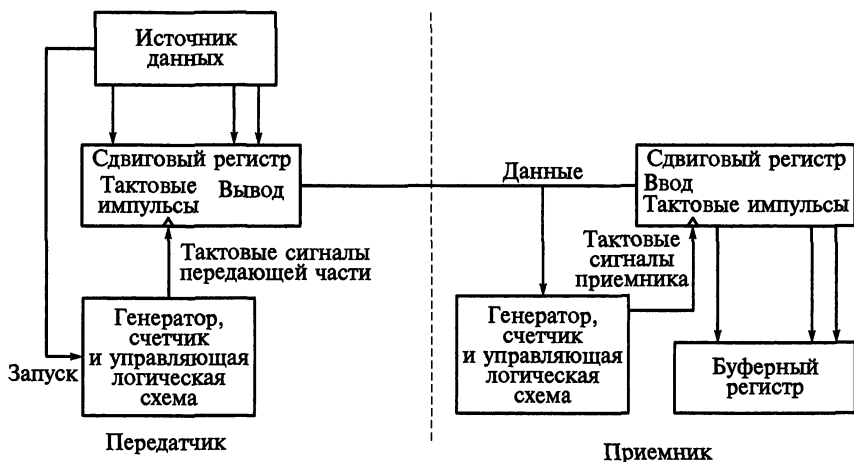


Рис. 13.6. Структурная схема асинхронной последовательной передачи данных

передаче какого-либо бита (т.е. вместо 1 будет принят 0 или наоборот), добавляется еще один бит — контрольный, или так называемый бит четности. Значение его (0 или 1) выбирается таким образом, чтобы вместе с ним общее число единиц в блоке было четным. Например, если в блоке передана кодовая комбинация 01011000, то значение бита четности должно быть 1, так как в этом случае общее число единиц (4) будет четное. Логическая схема проводит контроль четности, и в случае выявления ошибки передача данного блока повторяется. Для определения уровней сигналов и режимов последовательной передачи используются определенные стандарты. Широко применяется разработанный Международным консультативным комитетом по телеграфии и телефонии (МККТТ) стандарт V24. Ему соответствует аналогичный американский стандарт RS232. Эти стандарты позволяют, например, объединять компьютеры в глобальные сети с помощью существующих кабелей телефонных сетей. При этом необходимо использование дополнительных устройств (модемов, мультиплексоров и демультимплексоров). Подробнее компьютерные сети рассмотрены в гл. 19.

При передаче данных понятия логической 1 и логического 0 являются условными, поэтому в разных стандартах и разных наборах микросхем могут быть приняты разные условия для сигналов. Пусть принято условие, что при отсутствии передачи данных в линии имеется сигнал 1. Поскольку генератор тактовых импульсов в это время не включен, то этот сигнал говорит о том, что линия готова для передачи, но самой передачи пока нет. Для информации о начале передачи сигнал в линии изменяется на 0 в тот же момент времени, когда проходит первый из тактовых им-

пульсов. Затем начинают передаваться биты данных, по одному за каждый период тактовых импульсов, начиная от младшего разряда к старшему. Для контроля правильности передачи после старшего бита данных, как уже отмечалось, передается бит четности.

13.4. Интерфейсы современных персональных компьютеров

Число разнообразных устройств, которые могут работать совместно с персональным компьютером, непрерывно растет. К ним относятся дополнительные внешние запоминающие устройства (жесткие диски, компакт-диски, DVD-устройства), сканеры, цифровые фото- и видеокамеры, различные средства оргтехники и т.д. Компьютер может управлять работой различных бытовых приборов, системами охранной сигнализации, обеспечения комфортных условий в помещении, помогать в планировании и выполнении распорядка дня.

Все большее распространение получают информационно-коммуникационные технологии, дистанционное обучение, электронная коммерция, телемедицина, системы компьютерного обеспечения типа smart home, smart ship (умный дом, умный корабль) и др. С помощью компьютера можно существенно разнообразить досуг. Подключение к компьютерным сетям обеспечивает поступление деловой и текущей информации, различных видео- и аудиоматериалов.

Все это требует развития средств взаимодействия компьютера с большим числом цифровых устройств. Для обеспечения такого взаимодействия нужны современные внешние интерфейсы. Рассмотрим некоторые из них.

Широкую популярность завоевал интерфейс USB (Universal Serial Bus — универсальная последовательная шина), или шина USB. Стандарт этого интерфейса был утвержден в 1996 г. по инициативе целого ряда фирм, производящих компьютеры, другие средства и элементы вычислительной техники. Основная цель, поставленная перед разработчиками этого стандарта, заключалась в том, чтобы максимально облегчить пользователям компьютеров подключение новых периферийных устройств, их настройку и работу с ними. Такой подход получил название plug-and-play, т.е. включил и играй (без каких-либо дополнительных забот). Это означает, что должны быть предусмотрены подключение устройства к работающему компьютеру, автоматическое распознавание этого устройства сразу после подключения, подбор и установка соответствующих управляющих программ — драйверов. Перечисленные операции компьютер должен выполнить сам, не требуя от пользователя каких-либо специфических сведений.

Стандарт на интерфейс USB продолжает развиваться, и уже появилась вторая его версия. Для интерфейса USB 2.0 максимальное число подключенных устройств (включая размножители) достигло 127. Обмен информацией может происходить с двумя скоростями. Высокая скорость обмена составляет 60 Мбайт/с (увеличилась в 40 раз по сравнению с первой версией). Максимальная длина кабеля для такой скорости — 5 м. Скорость обмена с медленными устройствами — 1,5 Мбайт/с при длине кабеля 3 м. В случае установки дополнительного оборудования (концентратора) можно подключать устройства, находящиеся и на большем удалении. Допускается подключение устройств с различными скоростями обмена. Возможность использования только двух скоростей обмена данными ограничивает применение шины USB, но существенно уменьшает число линий интерфейса и упрощает аппаратную реализацию.

Подключаемые периферийные устройства обеспечиваются напряжением питания 5 В при потребляемом токе 0,5 А. Этого достаточно для питания клавиатуры, мыши, джойстиков и некоторых других устройств.

Интерфейс USB особенно удобен для подключения дополнительных внешних ЗУ. Он позволяет иметь в качестве сменного накопителя информации не трехдюймовую дискету емкостью 1,44 Мбайт, а накопителя в сотни раз большей емкости. На практике лучше всего использовать флэш-память (см. подразд. 12.8), поскольку для нее не требуется никакого дисководов.

Этот интерфейс наиболее подходит для часто подключаемых и отключаемых приборов, таких как цифровые камеры.

Подключение всех периферийных устройств к компьютеру с использованием интерфейса USB осуществляется через специальную плату внутри системного блока, которая носит название *хост*. В английском языке слово *host* имеет два значения: хозяин таверны и множество. Технический термин «хост» следует понимать как устройство, по-хозяйски управляющее многими другими устройствами.

Для того чтобы к одному порту интерфейса USB можно было подключать более одного устройства, применяются *хабы*. В английском языке *hub* — втулка (а от нее идет много спиц, т. е. связей). Корневой хаб находится внутри компьютера вместе с хостом, к которому он непосредственно подключен.

При описании интерфейса USB используют специальный термин «функция», под которым понимается логически законченное устройство, выполняющее какую-либо определенную функцию. Топология интерфейса USB представляет собой набор из семи уровней (рис. 13.7). На самом верхнем находятся хост и корневой хаб, а на самом нижнем — только функции. На каждом уровне может быть выполнено подключение и хабов, и функций, т. е. устройств.

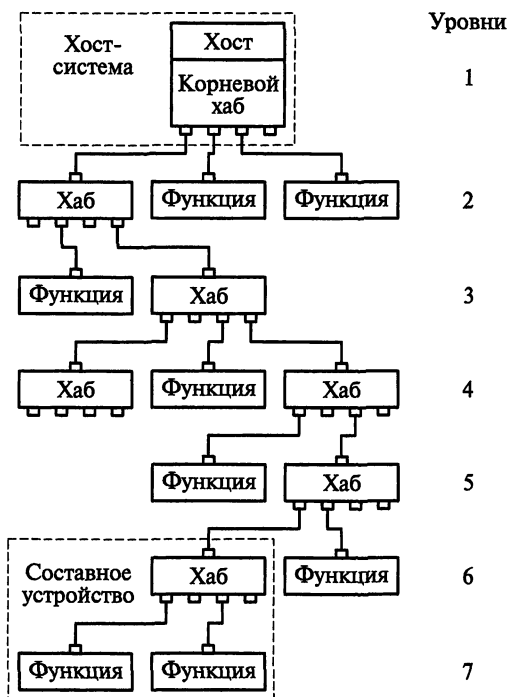


Рис. 13.7. Топология интерфейса USB

Все передачи данных по интерфейсу управляются хостом, а сами данные передаются в виде *пакетов*, т.е. наборов некоторого числа нулей и единиц, несущих определенную информацию. Используются несколько видов пакетов. Пакет-признак описывает тип и направление передачи данных, адрес устройства и порядковый номер конечной точки. В зависимости от типа этого пакета данные передаются от компьютера к устройству или от устройства к компьютеру.

Пакет с данными содержит передаваемые данные. Пакет согласования предназначен для сообщения о результатах пересылки данных. Все эти пакеты (вместе с напряжением питания) передаются по четырехпроводному кабелю, заканчивающемуся стандартными разъемами (рис. 13.8).

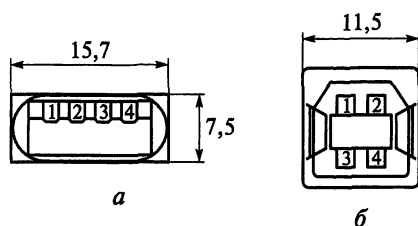


Рис. 13.8. Разъемы для подключения к шине USB:

a — тип А (для подключения к компьютеру или хабу); *б* — тип В (для подключения к периферийному устройству)

Высокоскоростной последовательный интерфейс FIREWIRE имеет характеристики, близкие к характеристикам интерфейса

USB, появился раньше его, но менее распространен, поскольку более дорог. Он также обеспечивает питанием подключаемые устройства. При этом потребляемый ими ток в три раза больше, чем у USB — 1,5 А, однако общее число подключаемых устройств вдвое меньше. На практике редко требуется подключение более десятка периферийных устройств, но и в этом случае проблемой является большое число кабелей. Поэтому очень важно появление беспроводных интерфейсов для подключения дополнительных устройств.

Беспроводной интерфейс BLUETOOTH использует высокочастотный (2,4 ГГц) радиосигнал, что позволяет подсоединять устройства, находящиеся на расстоянии до 100 м от компьютера.

Беспроводной интерфейс IrDA использует оптический инфракрасный канал (сокращение «Ir» в названии этого интерфейса означает *infrared* — инфракрасный) для подключения устройств, находящихся на расстоянии в несколько десятков метров. При таких расстояниях это оказывается дешевле, чем использование радиосигнала.

Контрольные вопросы

1. Поясните понятие «интерфейс».
2. Что такое параллельный интерфейс?
3. Что такое последовательный интерфейс?
4. Для чего выполняется квити́рование?
5. В чем разница между синхронной и асинхронной передачами данных?
6. Какие достоинства у интерфейса USB?
7. На каких принципах основаны беспроводные интерфейсы?
8. С помощью какого прибора можно к одному порту USB подключить несколько устройств?

14.1. Основные типы устройств ввода — вывода

Устройства ввода — вывода предназначены для ввода исходных данных и программ в ЭВМ и для вывода результатов обработки информации. Кроме того, УВВ выполняют необходимые при этом преобразования данных из одной формы представления в другую. Поэтому при вводе информации от устройств с непрерывным (аналоговым) сигналом используются аналого-цифровые преобразователи, которые были рассмотрены в гл. 7. Соответственно при выводе результатов на аналоговое устройство используются цифроаналоговые преобразователи, также описанные в гл. 7.

Для непосредственного ввода данных в компьютер чаще всего используется клавиатура (рис. 14.1). С ее помощью вводится символьная информация, состоящая из букв и цифр, а также знаков препинания и некоторых специальных знаков. При нажатии клавиш происходит замыкание контактов. Клавиш, т. е. контактных элементов, более 100, и надежность (вероятность безотказной работы) клавиатуры в очень большой степени зависит от конструкции каждого из них.

Человек в быту и на производстве очень часто использует самые различные кнопки для включения электрических приборов. Разработано много разных кнопок и собрано много данных об их

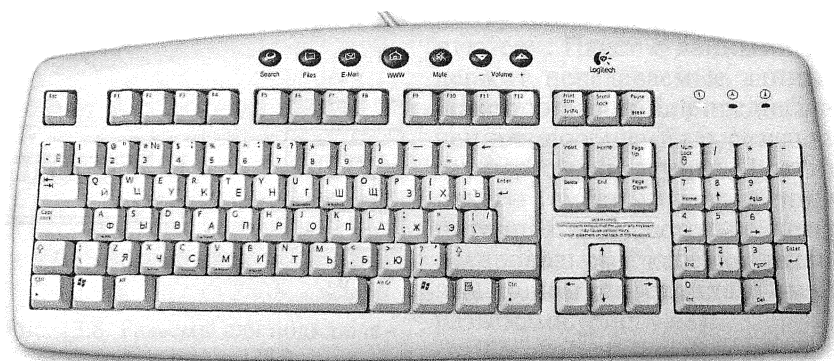


Рис. 14.1. Клавиатура персонального компьютера

надежности, которая в значительной степени определяется материалом контактов. Для кнопок с серебряными контактами гарантируется 4 млн срабатываний. Если бы такие довольно дорогие кнопки использовались в клавиатуре, с которой вводятся данные (цифровые и текстовые), то клавиатура отказывала бы очень часто. Возьмем для примера среднюю скорость набора данных 200 знаков в минуту (кстати, квалифицированная машинистка набирает текст со скоростью в полтора раза большей). Некоторые буквенные клавиши используются чаще, чем другие. Для цифровых данных можно считать равновероятным появление любой из десяти цифр. Примем для простоты, что вероятность нажатия одной и той же клавиши равна 0,1. Это означает, что за час работы один и тот же контактный элемент срабатывает $200 \times 60 \times 0,1 = 1\,200$ раз. Поделив 4 млн на 1 200, получим примерно 3 тыс. часов наработки на отказ одной клавиши. А ведь их, как уже говорилось, более сотни. Значит, такая клавиатура выходила бы из строя каждые два месяца. Поэтому в клавиатуре компьютера необходимо применять контактные элементы очень высокой надежности. Пока для этой цели используются разные конструкции. В одной, например, контакты 1 (рис. 14.2), изготовленные методом печатного монтажа, замыкаются проводящей шайбой 3. Шайба закреплена на пластиковом мостике 2, который прогибается (деформируется) при нажатии пальцем на клавишу 4, а после прекращения нажатия возвращается в исходное положение. В другой конструкции посеребренные шайбы размещаются на пленке, расположенной поверх печатной платы клавиатуры. Перспективно применение бесконтактных клавиш на герконах (герметизированных контактах из упругого ферромагнитного материала, которые замыкаются под влиянием магнитного поля), и особенно сенсорных контактов, работа которых основана на изменении емкости конденсатора при прикосновении пальца.

Часть клавиш клавиатуры предназначена для перемещения курсора на экране монитора. Для более эффективного выполнения этого действия используется специальный манипулятор, который получил название «мышь» (рис. 14.3). При перемещении мыши ладонью по поверхности рабочего стола (а лучше — по поверхности специального коврика) вращается шарик 1, который через ролик 6 передает вращение на диск 4 с прорезями. Поворот этого диска и преобразуется в

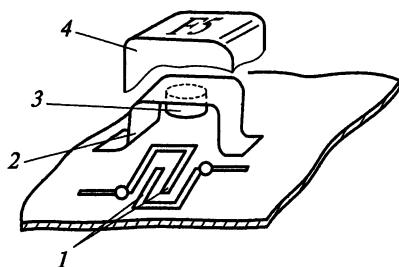


Рис. 14.2. Контактный элемент клавиши:

1 — контакты; 2 — пластиковый мостик; 3 — шайба; 4 — клавиша

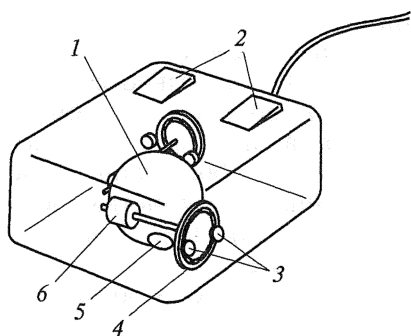


Рис. 14.3. Манипулятор типа мышь:

1 — шарик; 2 — кнопки управления;
3 — фотоприемник; 4 — диск; 5 — осветитель; 6 — ролик

электрический сигнал с помощью фотодатчика, состоящего из осветителя 5 и фотоприемника 3. Мышь имеет также две или три кнопки управления 2. Манипуляторы такого назначения выпускаются разными фирмами и могут иметь разное конструктивное исполнение.

Все большее распространение получают устройства ввода с помощью сенсорных устройств, чувствительных к прикосновению пальца пользователя. Например, справочную информацию на вокзале, в магазине, в библиотеке клиент может получить, прикасаясь пальцем к экрану монитора, на котором отмечены различные разделы справочной системы. Этот монитор имеет специальный *сенсорный экран*.

Для портативных компьютеров, на которых приходится работать в условиях ограниченного пространства, когда применение мыши оказывается неудобным, можно использовать *сенсорную панель* (рис. 14.4, а).

Принцип действия сенсорных устройств ввода поясняет рис. 14.4, б. Металлические проводники 1, разделенные тонкой изолирующей прокладкой из лавсановой пленки 2, образуют сетку, которая представляет собой набор очень большого числа маленьких конденсаторов. Так как человеческое тело является хорошим проводником, то при приближении руки к поверхности сенсорной панели (или экрана) происходит изменение электрического поля, а следовательно, и емкости этих конденсаторов. Измеряя изменение емкости каждого конденсатора в сетке, можно точно определить координаты пальца на поверхности панели (экрана) и даже приблизительно оценить давление, оказываемое на панель.

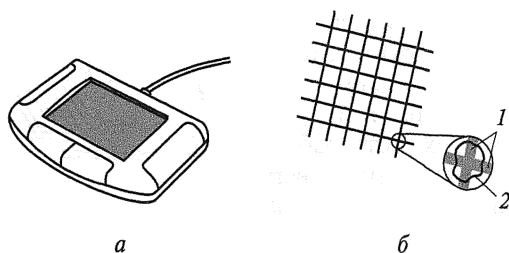


Рис. 14.4. Сенсорная панель:

а — внешний вид; б — принцип действия

Таким образом, прижав палец к сенсорной панели и передвигая его по ее поверхности, пользователь может перемещать курсор так же, как и с помощью мыши. Как и мышь, сенсорная панель может иметь кнопки для выбора действия после установки курсора, но возможно сделать выбор непосредственным нажатием пальца на сенсорную панель, поскольку емкость конденсатора зависит и от давления пальца.

Разработаны сенсорные панели повышенной чувствительности, позволяющие вводить информацию специальной ручкой или даже ногтем. Эта панель дает возможность ввода данных привычным для человека способом — записывая их рукой. Такую панель можно также использовать для ввода графических изображений и даже для подписывания созданных на компьютере документов.

Вводить данные в компьютер от руки можно с помощью устройства, которое называется *световое перо*. Пером это устройство названо условно, так как никакого воздействия на экран оно не оказывает, а само воспринимает его световое излучение. Световое перо (рис. 14.5) имеет цилиндрический корпус 3, внутри которого размещен светочувствительный элемент — фотодиод 5. На заостренном наконечнике 4 пера есть отверстие, в котором закреплена линза, фокусирующая попадающий на нее свет и направляющая его на светочувствительный элемент, посылающий сигнал на транзисторный усилитель 2 через пружинный контакт 6. Для исключения воздействия окружающего света перо включается лишь после прижатия его конца к поверхности экрана. Кабель 1 подсоединяет световое перо к компьютеру. При совмещении кончика пера с отображаемым на экране монитора графическим элементом возникает сигнальный импульс. В этот момент может быть прочитан адрес ячейки, где записан код отмеченного пером элемента, и могут быть определены координаты точки экрана, в которой находится наконечник пера. В некоторых конструкциях пера связь с экраном осуществляется с помощью пучка оптических волокон, а светочувствительный элемент и усилители располагаются в отдельном корпусе. При таком устройстве размеры и масса пера уменьшаются.

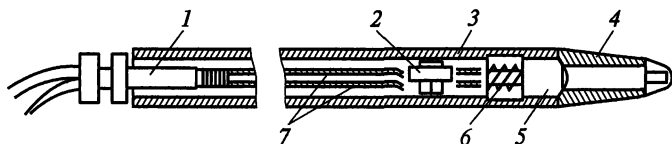


Рис. 14.5. Световое перо:

1 — соединительный кабель; 2 — усилитель; 3 — корпус; 4 — наконечник; 5 — фотодиод; 6 — пружинный контакт; 7 — провода

Указав пером на какой-либо элемент и определив таким образом для схемы управления его расположение в памяти, оператор нажатием функциональной клавиши выдает команду на соответствующее изменение этого элемента: стирание, сдвиг, изменение конфигурации, замену и т.д. Очевидно, что сигнал от светового пера может быть получен только при касании им точки экрана, где имеется светящееся изображение, так что определить какую-либо точку в темном месте экрана с помощью пера невозможно. Для устранения этого недостатка необходим специальный режим так называемого негативного изображения, когда высвечиваются все точки формата кадра, кроме тех, через которые проведены графические образы.

Для непосредственного ввода текстовой информации используются автоматические читающие устройства разной сложности. Специализированные устройства разработаны для прочтения почтового индекса на конвертах, считывания штрихового кода на отдельных изделиях и бланках документов, меток различных опросных листов и т. п.

Наиболее универсальным средством ввода текстовой и графической информации является *сканер*, который преобразует видеоинформацию на бумаге (текст, рисунки, слайды, фотографии) в электрический сигнал. С помощью специальной программы компьютер анализирует этот сигнал, определяет соответствие отдельных его элементов определенным символам (т. е. буквам, цифрам, знакам препинания) и воспринимает как текстовый файл, который может быть обработан, отредактирован и записан на диск.

В системах автоматического проектирования для ввода контурных изображений большого формата (чертежей, карт, схем) используются графические планшеты.

Несмотря на обилие различных моделей классификацию сканеров можно провести по нескольким признакам:

- способу формирования (кодирования) изображения;
- типу кинематического механизма (способу перемещения преобразователя свет — сигнал и оригинала относительно друг друга);
- типу сканируемого изображения;
- степени прозрачности оригинала, особенностям аппаратного и программного обеспечения.

По способу формирования изображения различают линейные и матричные сканеры, по типу кинематического механизма — ручные и настольные, по типу сканируемого изображения — черно-белые, полутоновые и цветные, по степени прозрачности оригинала — отражающие и прозрачные.

Технология считывания данных в современных устройствах оцифровывания изображений основана на использовании светочувствительных датчиков двух типов:

- 1) приборов с зарядовой связью (ПЗС);
- 2) фотоэлектронных умножителей (ФЭУ).

Неотъемлемой частью любого сканера являются АЦП, предназначенные для преобразования непрерывно изменяющихся значений напряжения, получаемых с помощью ПЗС или ФЭУ, в числа, соответствующие оттенкам цвета или градациям серого. Качество сканирования напрямую связано с разрядностью используемого в сканере АЦП. В черно-белых (двухуровневых) сканерах аналогичное преобразование выполняет компаратор, сравнивая зафиксированное значение напряжения с опорным.

Прибор с зарядовой связью — это твердотельный электронный компонент, состоящий из множества крошечных датчиков, которые преобразуют падающий на них свет в пропорциональный его интенсивности электрический заряд. В основу ПЗС положена чувствительность проводимости $p-n$ -перехода обыкновенного полупроводникового диода к степени его освещенности. На $p-n$ -переходе создается заряд, который уменьшается со скоростью, зависящей от освещенности. Чем меньше заряд, тем больший ток проходит через диод.

В зависимости от типа сканера ПЗС могут иметь различную конфигурацию. При линейном способе считывания информации микродатчики ПЗС размещаются на кристалле в одну линию (для трехпроходного сканирования) или в три линии (для однопроходного сканирования). Такая конфигурация позволяет устройству производить выборку всей ширины исходного аналогового изображения и записывать его как полную строку. Данный способ формирования изображения (рис. 14.6) обычно используется в доступ-

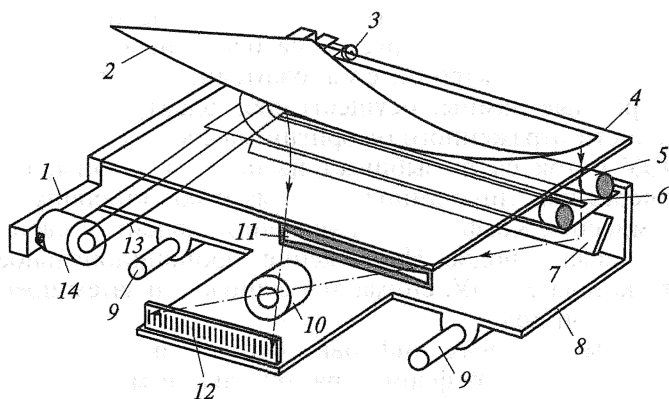


Рис. 14.6. Способ формирования изображения в планшетных сканерах:
 1 — основание; 2 — оригинал; 3 — шкив; 4 — стекло; 5 — источник света; 6 — диафрагма; 7 — зеркало; 8 — каретка; 9 — направляющие; 10 — объектив; 11 — диафрагма; 12 — линейка ПЗС; 13 — тростик; 14 — шаговый двигатель

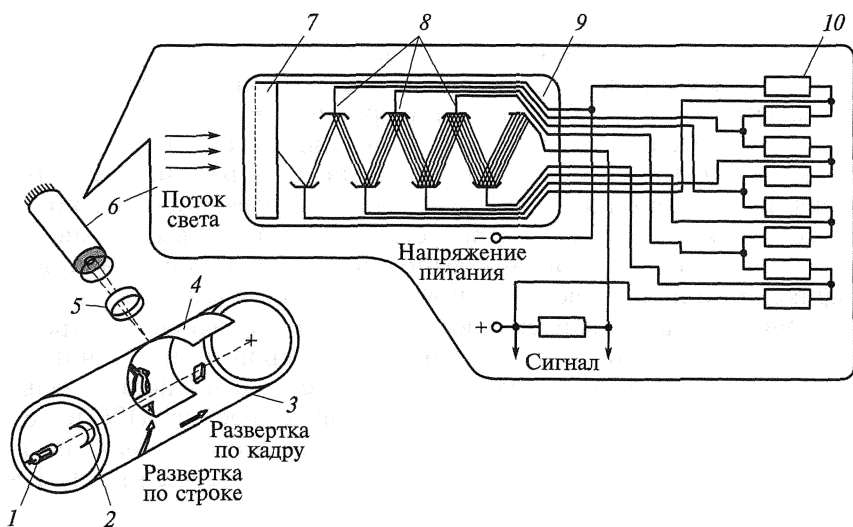


Рис. 14.7. Способ формирования изображения в барабанных сканерах: 1 — источник света; 2, 5 — объективы; 3 — барабан; 4 — оригинал; 6 — фотоэлектронный умножитель; 7 — катод; 8 — диноды; 9 — анод; 10 — делитель напряжения

ных широкому кругу пользователей ручных, планшетных, роликовых и проекционных сканерах.

В барабанных сканерах (рис. 14.7) в качестве светочувствительных приборов применяются фотоэлектронные умножители 6, а в качестве источника света 1 — ксеноновые или вольфрамогалогенные лампы, излучение которых с помощью конденсорных линз и волоконной оптики фокусируется на очень небольшой области оригинала 4. Фотоэлектронные умножители, представляющие собой электронные лампы, осуществляют электронное усиление интенсивности отраженного от оригинала света. Попадающий на катод ФЭУ поток света выбивает из него электроны, которые, проходя через пластины динодов 8, вызывают вторичную электронную эмиссию. Коэффициент усиления зависит от свойств материала и числа динодов. Напряжение, пропорциональное освещенности катода 7 ФЭУ, снимается с анода 9 и затем преобразуется в цифровой код.

В слайдовых сканерах, цифровых фото- и видеокамерах ПЗС-датчики обычно имеют форму прямоугольной матрицы, что позволяет формировать образ оригинала целиком, а не построчно. В этом случае говорят о *матричном* способе формирования изображения. Изображение с оригинала 1 (рис. 14.8) проецируется на матрицу 2, состоящую из более чем миллиона датчиков 3, каждый из которых формирует сигнал, соответствующий попавшей

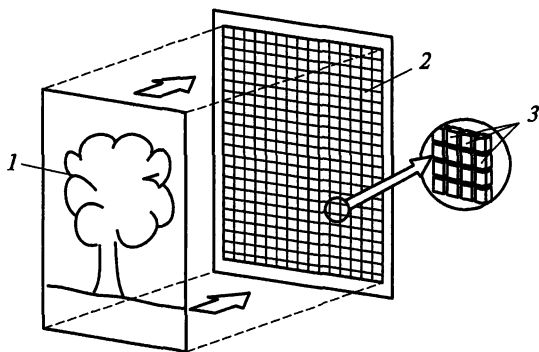


Рис. 14.8. Матричный способ формирования изображения:

1 — оригинал; 2 — матрица; 3 — датчики

на него порции светового потока. Указанный способ находит применение в так называемых нетрадиционных сканерах. К ним относятся цифровые камеры и устройства захвата видеоизображений, которые представляют собой плату расширения, устанавливаемую в слот ПК, имеющий входы для подключения видеокамеры, телевизора, видеомэгафона и др.

14.2. Печатающие устройства

Самым распространенным печатающим устройством является *принтер*. Существует большое разнообразие принтеров, отличающихся конструкцией, характеристиками, принципом действия. Принтеры печатают и текстовые, и графические материалы, т.е. рисунки и схемы.

Для вывода на бумажный носитель крупногабаритной графической информации (чертежи, географические карты, большие электрические схемы и т.п.) используются графопостроители, или *плоттеры*. Скорость печати у них несколько ниже, чем у хороших принтеров, но их применение зачастую оказывается экономичнее, чем применение для аналогичных целей хорошего принтера. Для изготовления рекламных материалов используются устройства, позволяющие отпечатать изображение размером в десятки метров.

По принципу действия различают матричные, струйные, лазерные, светодиодные и термические принтеры.

Матричные принтеры. До недавнего времени это был наиболее распространенный тип принтеров для IBM PC. Принцип печати матричных (или точно-матричных) принтеров следующий. Печатающая головка принтера содержит ряд тонких металлических стержней (иголок). Головка движется вдоль печатаемой строки, а стержни в нужный момент ударяют по бумаге через красящую

ленту, что обеспечивает формирование на бумаге символов и изображений. В дешевых моделях принтеров используется печатающая головка с девятью стержнями. Любая линия, отпечатанная на матричном принтере, будет не сплошной, а состоящей из отдельных точек. Качество печати у таких принтеров посредственное, но его можно несколько улучшить с помощью печати в несколько проходов (от двух до четырех). Более качественная и быстрая печать обеспечивается принтерами с 24 печатающими иглками (24-точечными принтерами). Бывают принтеры с 48 и даже 60 иглками. Скорость печати матричных принтеров — от 60 до 10 с на машинописную страницу, печать рисунков выполняется медленнее — до 5 мин на страницу. У специальных высокопроизводительных матричных принтеров скорость печати достигает нескольких тысяч строк в минуту. Наличие движущихся частей и необходимость удара по бумаге делает матричные принтеры шумными и ненадежными. Поэтому на смену им пришли значительно менее шумные и более надежные струйные и лазерные принтеры. Матричные принтеры продолжают использоваться в специальных устройствах для ввода штриховых кодов и специальных меток, которые в массовом порядке выполняются на различных документах в банковской и торговой сферах, в телефонных компаниях, при сортировке писем.

Струйные принтеры. В этих принтерах изображение формируется микрокаплями специальных чернил, выдуваемых на лист бумаги через сопло. Такой способ обеспечивает более высокое качество печати по сравнению с матричными принтерами. Для цветной печати струйные принтеры часто являются самым оптимальным решением, так как хороший цветной струйный принтер в несколько раз дешевле цветного лазерного принтера, дающего не намного лучшее изображение.

Принцип работы струйных принтеров схож с принципом работы матричных принтеров, только вместо иголок применяются тонкие, как волос, сопла, которые находятся в головке принтера. В этой головке установлен резервуар с жидкими чернилами, микрочастицы которых через сопла переносятся на материал носителя. Число сопел (минимум 16) зависит от модели принтера и фирмы-изготовителя. Например, головка принтера HP DeskJet 1600 имеет 300 сопел для черных чернил и 416 для цветных.

Современные модели струйных принтеров в своей работе могут использовать следующие методы: пьезоэлектрический, газовых пузырей, капля по запросу.

Для реализации пьезоэлектрического метода (рис. 14.9, а) в каждое сопло устанавливают плоский пьезоэлемент 1, связанный с диафрагмой 2. Под воздействием электрического поля происходит деформация пьезоэлемента, который, нажимая на диафрагму, выдавливает из сопла каплю чернил, подводимых по трубкам (системе капилляров) из резервуара. Выдавленная нару-

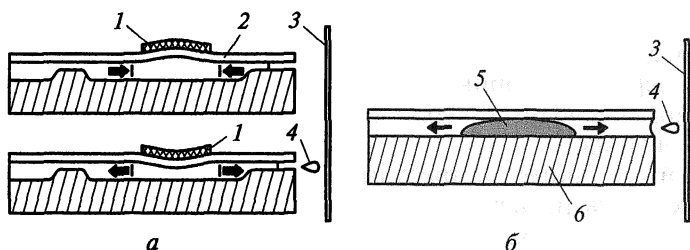


Рис. 14.9. Методы работы струйных принтеров:

а — пьезоэлектрический; *б* — газовых пузырей; 1 — пьезоэлемент; 2 — диафрагма; 3 — бумага; 4 — капля чернил; 5 — газовый пузырь; 6 — нагревательный элемент

жу капля оставляет на бумаге точку. Подобные устройства выпускают компании Epson, Brother и ряд других.

Метод газ о в ы х п у з ы р е й, или инжектируемых пузырьков (рис. 14.9, б), является термическим. При использовании этого метода каждое сопло оборудуют нагревательным элементом 6, который при пропускании через него тока за несколько микросекунд нагревается до температуры около 500 °С. Возникающий при резком нагревании газовый пузырь 5 выталкивает через выходное отверстие сопла каплю чернил, которая переносится на бумагу. При отключении тока нагревательный элемент остывает, пузырь уменьшается и через входное отверстие в сопло поступает новая порция чернил. Подобную технологию использует фирма Canon.

Метод, разработанный фирмой Hewlett Packard, называется методом «капля по запросу» (drop-on-demand). Как и в методе газовых пузырей, в каждом сопле используется нагревательный элемент. Капля чернил нагревается до испарения, а затем при выходе из сопла происходит конденсация чернильных паров.

Благодаря тому, что в принтерах, использующих метод газовых пузырей, меньше конструктивных элементов, такие принтеры надежнее в работе и срок их эксплуатации более продолжителен. Кроме того, использование этого метода позволяет добиться более высокой разрешающей способности принтеров. Данный метод обеспечивает высокое качество при прорисовке линий, но дает не очень хороший результат при печати областей сплошного заполнения: они получаются несколько расплывчатыми. Метод газовых пузырей целесообразно применять для печати графиков, гистограмм и т. п. Печать полутоновых графических изображений получается более качественной при использовании метода «капля по запросу». Он обеспечивает наиболее быстрое впрыскивание чернил, что позволяет существенно повысить качество и скорость печати. Цветное изображение в этом случае более контрастно.

Для хранения чернил используется резервуар, который может быть объединен с головкой принтера (в этом случае замена резервуара с чернилами приводит к замене головки) или размещаться отдельно, обеспечивая чернилами головку принтера через систему капилляров.

Струйные принтеры требуют тщательного ухода и обслуживания. Скорость их печати — от 15 до 100 с на страницу, время печати цветных страниц обычно составляет 3...5 мин, иногда может достигать 10 мин.

Лазерные принтеры. Большинство изготовителей лазерных принтеров использует механизм печати, который применяется в ксероксах. Например, в принтерах HP и QMS используется механизм печати ксероксов Canon.

Основным конструктивным элементом лазерного принтера является вращающийся фотонаборный барабан 8 (рис. 14.10), с помощью которого изображение переносится на бумагу. Барабан представляет собой металлический цилиндр, покрытый тонкой пленкой светопроводящего полупроводника. Обычно в качестве такого полупроводника используется оксид цинка. По поверхности барабана равномерно распределяется статический заряд. Для этого служит тонкая проволока или сетка 11, называемая коронирующим проводом. На этот провод подается высокое напряжение, вызывающее возникновение вокруг него светящейся ионизированной области, называемой короной. Лазер 5, управляемый микроконтроллером, генерирует тонкий световой луч, отражающийся от вращающегося зеркала 7. Этот луч, приходя на фотонаборный барабан 8, изменяет его электрический заряд в точке прикосновения. У некоторых типов принтеров потенциал поверхности барабана в таких точках уменьшается с 900 до 200 В. Таким образом, на барабане возникает скрытая копия изображения.

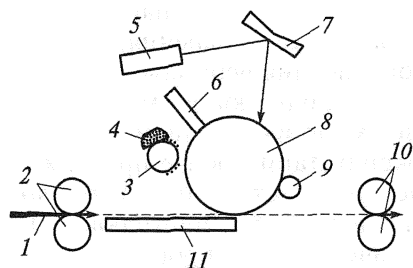


Рис. 14.10. Функциональная схема лазерного принтера:

1 — бумага; 2 — валики; 3 — барабан-девелопер; 4 — резервуар с тонером; 5 — лазер; 6 — провод разряда; 7 — зеркало; 8 — фотонаборный барабан; 9 — ролик очистки; 10 — фиксирующие ролики; 11 — коронирующий провод

На следующем этапе на фотонаборный барабан 8 наносится тонер — мельчайшая красящая пыль. Под действием статического заряда частицы тонера притягиваются к поверхности барабана в точках, подвергшихся облучению, и формируют изображение. Бумага 1 втягивается с подающего лотка и с помощью валиков 2 перемещается к барабану. Перед самым барабаном бу-

маге сообщается статический заряд. Затем бумага соприкасается с барабаном и благодаря своему заряду притягивает частицы тонера с барабана.

Для фиксации тонера бумага вновь заряжается и пропускается между двумя фиксирующими роликами 10, имеющими температуру около 180 °С. После передачи изображения на бумагу барабан полностью разряжается, очищается от прилипших частиц тонера и готов для нового процесса печати.

В цветном лазерном принтере изображение формируется на светочувствительной фотоприемной ленте последовательно для каждого цвета (циан — cyan, пурпурный — magenta, желтый — yellow, черный — black). Лист печатается за четыре прохода, что сказывается на скорости печати, имеются четыре резервуара для тонеров, выполняются несколько циклов проявления. Принтеры такого класса оборудованы имеющей большой объем памятью, процессором и, как правило, собственным винчестером.

Современные лазерные принтеры имеют разрешающую способность 600 точек на 1 дюйм и более. В некоторых принтерах используются специальные технологии повышения качества изображения (RET, TurboRes и др.), дающие результат, эквивалентный повышению разрешающей способности принтера в 1,5 раза.

Скорость печати лазерных принтеров — 5...15 с на страницу при выводе текстов. Страницы с рисунками могут выводиться значительно дольше, на вывод больших рисунков может потребоваться несколько минут. Выпускаются специальные высокопроизводительные (так называемые сетевые) принтеры, работающие со скоростью 15...40 страниц в минуту. Обычно такие принтеры подключаются к локальной сети и совместно используются ее пользователями.

Светодиодные принтеры. Альтернативой лазерному принтеру является так называемый светодиодный принтер, или LED-принтер (Light Emitting Diode). Вместо лазерных лучей, управляемых с помощью механизма зеркал, барабан освещает неподвижная диодная линейка, состоящая из 2 500 светодиодов, которая выделяет не каждую точку, а целую строку на сканируемой странице. Светодиодные принтеры (в отличие от лазерных) не выделяют озон, но качество отпечатков у них хуже, чем у лазерных.

Термические принтеры. Конструкция цветных лазерных принтеров еще не достигла совершенства. Для получения цветного изображения с качеством, близким к фотографическому, или изготовления допечатных цветных проб иллюстраций журналов и альбомов используются термические принтеры.

В настоящее время распространение получили три технологии цветной термопечати:

1) струйный перенос расплавленного красителя (термопластичная печать);

2) контактный перенос расплавленного красителя (термовосковая печать);

3) термоперенос красителя (сублимационная печать).

Общим для последних двух технологий является нагрев красителя и перенос его на бумагу (пленку) в жидкой или газообразной фазе.

14.3. Устройства отображения информации

Для отображения информации в вычислительной технике широко используются индикаторные устройства. Чаще всего применяются оптические индикаторы, поскольку с помощью зрения человек получает более трех четвертей всего воспринимаемого им объема информации. Кроме оптических используются звуковые (акустические) индикаторы. Например, для сигнализации об аварийных ситуациях наиболее приемлемы звуковые сигналы в сочетании с привлекающими внимание персонала световыми (т. е. оптическими) сигналами.

Рассмотрим оптические индикаторные устройства, в дальнейшем называемые для краткости индикаторами. Различают активные и пассивные оптические индикаторы. К активным относятся лампы накаливания, газоразрядные приборы, кинескопы и другие устройства, излучающие свет в видимой части спектра. К пассивным индикаторам относятся те устройства, которые сами не излучают свет, а лишь отражают свет внешних источников. К ним относятся шкалы измерительных приборов, цифровые индикаторы (например, счетчика электроэнергии в квартире или счетчика километров на приборной панели автомашины), жидкокристаллические индикаторные панели.

Одним из наиболее простых и распространенных активных индикаторов является *светодиод*. Принцип его действия основан на том, что при протекании прямого тока через полупроводниковый диод происходит излучение фотонов (т. е. световой энергии). Кремниевые и германиевые диоды излучают в невидимом глазом диапазоне длин волн, а вот диоды на основе арсенида-фосфида галлия (GaPAs) — в диапазоне длин волн 0,58...0,65 мкм. Такое излучение человек воспринимает как желтый (0,58 мкм), оранжевый (0,63 мкм) или красный (0,65 мкм) свет. Светодиод, изготовленный на основе фосфида галлия (GaP), излучает зеленый свет (0,56 мкм), а на основе арсенида галлия (GaAs) — инфракрасный (0,90 мкм), который хотя и является невидимым для человеческого глаза, но удобен для дистанционного управления объектами. В зависимости от числа и пропорций примесей можно изменять длину волны максимума излучения, т. е. цвет свечения светодиода. Материалы, используемые для изготовления светодиодов, дороже кремния и германия, поэтому светодиоды дороже обычных

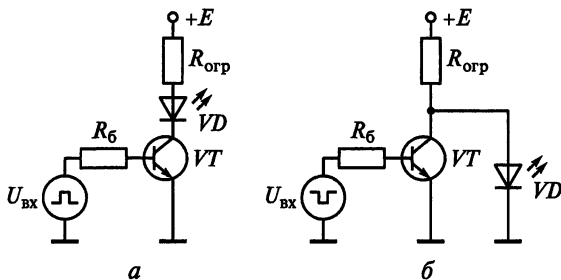


Рис. 14.11. Схемы включения светодиодов:

а — для высокого уровня входного сигнала $U_{вх}$; *б* — для низкого уровня

диодов. Коэффициент полезного действия светодиодов очень мал. У инфракрасных диодов он составляет примерно 5%, а у других — примерно в 100 раз меньше. Быстродействие светодиодов очень высокое: при подаче скачкообразного входного сигнала яркость диода изменяется за сотую долю миллисекунды.

Входным сигналом для светодиода является прямой ток. От его величины зависит яркость свечения. Хорошая видимость даже при дневном свете обеспечивается при прямом токе от 5 до 20 мА. При этом напряжение на светодиодах составляет 2...3 В. Светодиоды по своим параметрам хорошо согласуются с транзисторными и интегральными схемами. На рис. 14.11 показаны схемы включения светодиодов VD с помощью транзисторного ключа. Поскольку транзистор VT обладает усилительными свойствами, ток, поступающий от источника сигнала напряжением $U_{вх}$ через резистор базы $R_б$, в десятки раз меньше прямого тока светодиода.

Сопротивление резистора $R_{огр}$, ограничивающего прямой ток светодиода, подсчитывается по формуле $R_{огр} = (E - U_{VD})/I_{VD}$.

Светодиоды выпускаются в точечном, линейном и цифровом исполнении. Среди цифровых наибольшее распространение получили семисегментные цифровые светодиодные индикаторы (рис. 14.12). Стилизованное изображение цифры составляется из семи светодиодных сегментов (*a, b, c, d, e, f, g*), расположенных в виде цифры 8. При подаче сигналов на определенные сегменты высвечивается требуемая цифра. Например, для высвечивания цифры 5 необходимо подать сигналы на сегменты *a, f, g, c, d*. Из нескольких таких индикаторов может быть составлен многоразрядный индикатор. При этом для отделения целой части десятичного числа от дробной используется сегмент *h*.

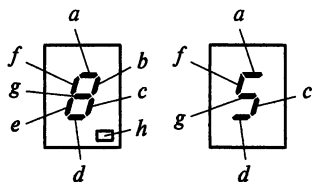


Рис. 14.12. Семисегментный цифровой светодиодный индикатор

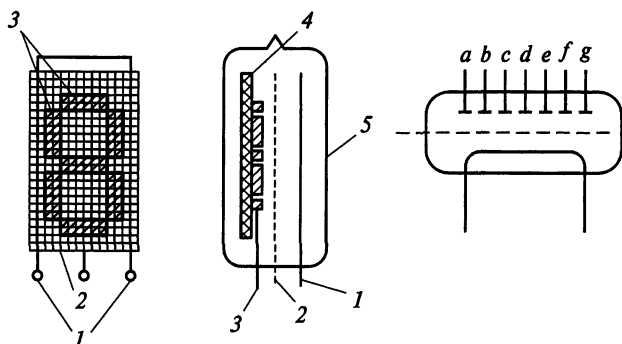


Рис. 14.13. Люминесцентный индикатор:

1 — катод; 2 — сетка; 3 — аноды; 4 — подложка; 5 — стеклянный корпус

Линейный светодиодный индикатор представляет собой интегральную схему в виде светящегося столбика, образованного последовательно включенными светодиодными сегментами, и блока управления. Внешне такой индикатор выглядит как линейная шкала. Он служит для отображения непрерывно меняющейся информации и является аналогом стрелочного измерительного прибора. Подобные устройства используются в многоканальных системах для индикации однотипной информации. Несколько расположенных рядом линейных шкал очень удобны для восприятия оператором.

Люминесцентный индикатор также относится к типу активных. Он представляет собой электронную вакуумную лампу с катодом 1 (рис. 14.13), управляющей сеткой 2 и несколькими анодами 3 на подложке 4. Аноды покрыты слоем люминофора, который светится, если на него попадает поток электронов, испускаемых катодом. Катод выполнен в виде двух тонких вольфрамовых нитей, натянутых параллельно анодам. Между катодом и анодами находится плоская сетка. На катод подается напряжение накала, он нагревается и испускает поток электронов. На сетку и аноды подаются положительные (по отношению к катоду) напряжения. Поток электронов из катода устремляется к положительно заряженной сетке, пролетает ее по инерции и попадает в ускоряющее поле тех анодов, на которые подано напряжение. Когда разогнавшиеся до большой скорости электроны достигают анодов, их кинетическая энергия переходит в световую энергию излучаемых люминофором квантов света (как и в обычной электронно-лучевой трубке). Серийно выпускаемые промышленностью люминесцентные индикаторы работают при напряжении накала до 5 В и сеточном напряжении 20...30 В.

К типу пассивных относится *жидкокристаллический индикатор*. Считывание с него информации возможно лишь при наличии

внешнего освещения — естественного или искусственного. Принцип действия таких индикаторов основан на изменении степени прозрачности органических жидкокристаллических веществ, находящихся в электрическом поле.

Конструктивно жидкокристаллический индикатор (рис. 14.14) выполнен в виде двух плоских стеклянных пластин 1, разделенных по периметру прокладкой 2. На

внутреннюю поверхность одной пластины наносят прозрачные проводящие электроды-сегменты 3, форма и взаимное расположение которых будут определять индицируемые знаки. На всю вторую пластину 1 наносят проводящий прозрачный электрод 4. Пространство между пластинами заполняют жидкокристаллическим веществом, толщина слоя которого составляет примерно 10 мкм. Собранный таким образом пакет из стеклянных пластин, электродов и жидкого кристалла герметизируют. Выводы от электродов проходят через герметик. Для управления индикатором между общим электродом и электродами-сегментами подается напряжение 5...15 В.

Ток индикатора очень мал, поскольку жидкокристаллическое вещество обладает большим удельным сопротивлением — несколько мегаом на 1 см. Поэтому потребление энергии таким индикатором существенно меньше, чем индикаторами других типов. В то же время для жидкокристаллического индикатора нужен внешний источник света и не всегда бывает достаточно естественной освещенности. В этом случае требуется дополнительная энергия для искусственного источника света. При хорошей внешней освещенности контрастность знаков по отношению к фону составляет 70...90 %.

Жидкокристаллические индикаторы относятся к высоконадежным элементам автоматики (наработка на отказ составляет несколько десятков тысяч часов), однако нельзя допускать их нагрев до температуры выше 60 °С, а также необходимо исключить постоянную составляющую в переменном напряжении питания. На жидких кристаллах созданы индикаторные панели и экраны. Такие устройства позволяют выводить большой объем информации — цифровые и буквенные тексты, графики, таблицы, схемы и рисунки.

Оптимальным с точки зрения сочетания качества изображения и стоимости является индикаторный экран на базе *электронно-лучевой трубки* (в телевизоре ее называют кинескопом). В мониторе персонального компьютера используется именно такая труб-



Рис. 14.14. Жидкокристаллический индикатор:

- 1 — стеклянные пластины; 2 — прокладка; 3 — электроды-сегменты; 4 — прозрачный электрод

ка (в ноутбуке — переносном компьютере с автономным питанием — в целях экономии энергии применяется жидкокристаллический экран). В электронно-лучевой трубке с помощью двух отклоняющих катушек можно изменять пространственное положение электронного луча на экране и получать изображение, состоящее более чем из миллиона точек. Однако устройства управления электронно-лучевой трубкой с трудом согласуются с наиболее перспективными цифровыми системами формирования изображения.

В настоящее время более удачным средством для индикации большого объема информации являются *плоские информационные экраны-панели*. Работа их основана на различных физических принципах, но все они выполняют две задачи: обеспечивают пространственное распределение электрических сигналов для включения любого элемента индикации на всей поверхности экрана-панели и осуществляют преобразование электрического сигнала в оптическое излучение. Для этого светоизлучающие элементы экрана располагаются в строго фиксированных точках. При этом возможны два способа адресации: параллельный (все элементы индикации независимы и могут включаться в любом порядке) и последовательный (в каждый, очень короткий момент времени включен лишь один элемент, и вся информация отображается путем поочередного включения всех необходимых элементов). При параллельной адресации каждый элемент (точка на экране) должен быть соединен с источником сигнала проводником. Технически это трудно осуществить. Например, для квадратного экрана с 10 000 элементов (100 точек в каждой из 100 строчек) потребуется 10 000 проводников и столько же ключей для управления экраном. При последовательной адресации число соединительных проводников и ключей (элементов управления) может быть резко уменьшено за счет применения матричного построения экрана. Такой экран выполняется слоистым с матричной (решетчатой) структурой,

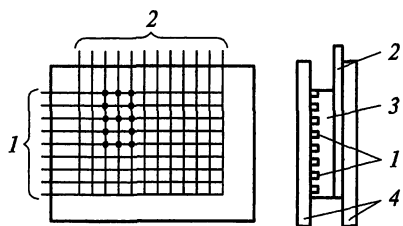


Рис. 14.15. Информационный матричный экран:

1 — горизонтальные электроды; 2 — вертикальные электроды; 3 — активный оптический слой; 4 — стеклянные пластины

как показано на рис. 14.15. На нижнюю стеклянную пластину 4 наносятся параллельные горизонтальные электроды 1, на верхнюю стеклянную пластину 4 — параллельные вертикальные электроды 2. Между электродами 1 и 2 помещается слой активного оптического материала 3, изменяющего свои оптические свойства при прохождении тока или под воздействием электрического поля. При одновременной подаче напряжения на один из горизонтальных электродов 1 и

один из вертикальных электродов 2 происходит включение элемента индикации, находящегося на их пересечении. При этом для экрана с 10 000 элементов при последовательной матричной адресации потребуется всего 200 соединительных проводников и ключевых элементов управления, т. е. в 50 раз меньше, чем при параллельной адресации. Однако при последовательной адресации необходимы весьма быстродействующие электрооптические преобразователи. Для нормального восприятия человеком-оператором картинки на экране она должна повторяться 50 раз в секунду. Следовательно, каждый элемент экрана должен включаться на время $1/(50 \cdot 10\,000) = 2 \cdot 10^{-6} \text{ с} = 2 \text{ мкс}$.

Большие экраны, созданные с помощью плазменной технологии, позволяют весьма значительно уменьшить размеры монитора. Плоский (толщиной менее 10 см) и легкий монитор с *плазменным экраном* можно разместить на стене или под потолком, что особенно удобно для оснащения компьютерных справочно-информационных систем. Основное достоинство плазменного экрана заключается в том, что каждая его светящаяся точка (пиксел) сразу окрашивается в красный, зеленый и голубой цвета, что в несколько раз сокращает размеры носителя цветовоспроизведения. Заряженные электроды, расположенные между стеклянными пластинами, вызывают появление мельчайших ячеек инертного газа, служащего для изменения состояния плазмы.

Принцип действия элементарной ячейки плазменного экрана поясняет рис. 14.16. Для того чтобы ячейка стала светиться, на соответствующие электроды 1, в точке пересечения которых эта ячейка находится, подается высокое переменное напряжение. Газ в ячейке переходит в состояние плазмы, т. е. его атомы разделяют-

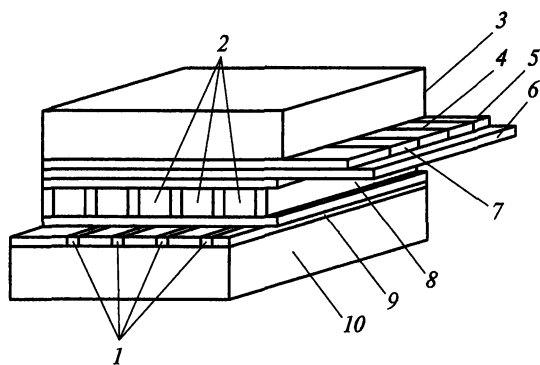


Рис. 14.16. Ячейка плазменного экрана:

1 — управляющие электроды; 2 — флуоресцирующее покрытие ячеек; 3 — наружное стекло; 4 — питающий электрод; 5, 9 — слои диэлектрика; 6 — защитный слой; 7 — сканирующий электрод; 8 — разделительная перегородка; 10 — внутреннее стекло

ся на ионы и электроны. В зависимости от полярности напряжения они попеременно собираются у электродов по разные стороны (электроны — там, где «плюс», ионы — там, где «минус») ячейки. Для «зажигания» на сканирующий электрод 7 подается импульс, одноименные потенциалы складываются и напряженность электростатического поля удваивается. Происходит разряд — часть ионов отдает энергию в виде излучения квантов света в ультрафиолетовом диапазоне. В свою очередь, флуоресцирующее покрытие 2 ячейки начинает излучать свет в видимом диапазоне, который и воспринимает наблюдатель. Каждый пиксел состоит из трех таких ячеек: красной, зеленой и голубой. Почти все вредное для глаз ультрафиолетовое излучение поглощается наружным стеклом 3. Яркость свечения каждой ячейки определяется величиной управляющего напряжения, а цвет пиксела является результатом смешения трех цветов разной яркости.

Плазменные экраны превосходят обычные по яркости более чем в два раза, а по контрастности — в 10 раз. По всей рабочей поверхности экрана сохраняется высокая четкость изображения. В отличие от мониторов с электронно-лучевыми трубками мониторы с плазменными экранами не подвержены помехам от электромагнитных полей. Это очень важное свойство плазменных экранов, позволяющее использовать их на промышленных предприятиях, где много всевозможных источников электромагнитных помех (электродвигатели, трансформаторы и т. п.).

Контрольные вопросы

1. Какие устройства относятся к периферийным?
2. Какие устройства используются для ввода информации в ЭВМ?
3. Что такое сканер?
4. Какие виды принтеров вы знаете?
5. Как работает лазерный принтер?
6. Какие устройства используются для отображения информации в вычислительной технике?
7. Расскажите о принципе действия плазменного экрана.

РАЗДЕЛ IV

ПРОГРАММНЫЕ СРЕДСТВА

Глава 15

ОПЕРАЦИОННЫЕ СИСТЕМЫ

15.1. Назначение, функции и состав операционных систем

Общение пользователя с ЭВМ осуществляется с помощью операционной системы (ОС). По мере развития вычислительной техники ОС развивались и совершенствовались, делая это общение более удобным для пользователя.

Операционная система представляет собой комплекс программ, обеспечивающих общее управление работой вычислительной машины и реализующих наиболее общие алгоритмы обработки информации. Основные функции ОС заключаются в упорядочении, сортировке, редактировании обрабатываемой и хранящейся в машине информации, трансляции прикладных программ, составленных на различных языках программирования, в воспринимаемые машиной (понятные ей) команды — машинные коды. Оператору (или пользователю) вычислительной машины ОС обеспечивает удобное обращение к различным разделам информации, к программам, решающим конкретные задачи, различным устройствам ввода—вывода, периферийным устройствам, компьютерным сетям.

Операционная система обеспечивает пользователю удобный способ общения (интерфейс) с устройствами компьютера.

При работе на компьютере используются программы трех категорий:

1) *прикладные программы*, непосредственно обеспечивающие выполнение необходимых пользователям работ (создание и редактирование текстов, создание и обработка графических изображений, обработка информационных массивов, создание электронных таблиц, математические вычисления, запись, воспроизведение и обработка музыки и т. д.);

2) *системные программы*, выполняющие различные вспомогательные функции (копирование и архивирование используемой информации, выдача справочной информации о компьютере, проверка работоспособности устройств компьютера, установка но-

вых устройств и замена действующих, размещение информации в запоминающих устройствах и т. д.);

3) *инструментальные программы, или системы*, обеспечивающие создание новых программ для компьютера.

Четких границ между этими тремя категориями нет. Системные программы непрерывно совершенствуются, в них включаются дополнительные возможности, позволяющие выполнять задачи прикладного характера, например, редактировать тексты, выполнять расчеты на калькуляторе, создавать программы для часто выполняемых конкретным пользователем задач (так называемые макрокоманды или макросы). Эти программы представляют собой удобную среду для оператора, не имеющего специальной подготовки для работы на вычислительной машине. В связи с широким распространением персональных компьютеров число таких операторов (их называют пользователями) вполне возможно уже достигло миллиарда. Операционные системы, усовершенствованные для огромного круга пользователей, получили название *операционные оболочки*.

Благодаря операционным оболочкам круг пользователей ЭВМ расширяется, поскольку не требуется наличие каких-то специфических знаний для общения с компьютером. Операционные оболочки обеспечивают пользователю более наглядные средства для выполнения часто используемых действий. Это, прежде всего, широкий набор средств для вывода изображений (включая тексты) на экран, внесения различных изменений в эти изображения, построения меню, окон на экране и т. д. Кроме того, операционные оболочки предоставляют дополнительные возможности для запускаемых программ, позволяют одновременно выполнять несколько программ, т. е. обеспечивают режим мультипрограммирования, предоставляют расширенные средства для обмена между программами, упрощают создание графических программ.

Проследить за направлением совершенствования ОС можно на следующем примере. В одной из первых ОС для компьютеров MS-DOS (первая версия была создана в 1981 г. фирмой Microsoft) для вызова нужной информации необходимо было набирать на клавиатуре определенное сочетание букв. В пришедшей ей на смену системе Norton Commander достаточно было выделить на экране требуемую строку из предложенного набора. Сейчас наиболее популярной является система Windows, при пользовании которой курсор с помощью мыши устанавливается на условное изображение требуемого раздела информации и вызов выполняется значительно быстрее и проще. По мере своего совершенствования ОС требуют все большего объема памяти. Если раньше она измерялась килобайтами, то теперь требуются десятки мегабайт.

Операционная система освобождает пользователя от выполнения часто повторяемых типовых операций, таких как запуск и

завершение программ, запись и чтение информации на магнитных дисках, ввод информации с клавиатуры и вывод ее на экран, отсчет времени и количества информации, определение причин некоторых сбоев в работе аппаратуры. Все программы и данные хранятся на магнитных дисках в виде файлов. Английское слово *file* имеет несколько значений: очередь, ряд, картотека, регистратор (для бумаг). В вычислительной технике файл означает блок однотипной информации, которому присвоено определенное имя. Файл — это информация, которая используется целиком. Его можно изменять, редактировать, при необходимости его можно объединять с другими файлами или даже разбивать на несколько файлов. Но работать с ним можно, только вызвав его целиком. Размер файла определяется количеством содержащейся в нем информации. Самые маленькие текстовые файлы имеют объем несколько байт, а самые большие — сотни килобайт и больше. Файл одной цветной фотографии может занимать около 10 Мбайт. Очень большой файл неудобен в использовании: он может не поместиться в запоминающем устройстве малой емкости (например, на гибком диске), много времени уходит на его вызов с жесткого диска.

Вся информация в ЭВМ состоит из некоторого, как правило, очень большого числа файлов. Для того чтобы облегчить поиск нужного файла, используется иерархическая (или древовидная) система. Операционная система должна знать, где находится тот или иной файл. Поэтому на каждом диске кроме самих файлов находятся еще несколько *каталогов* (или *директорий*, или *папок*), объединяющих группы файлов. Каталог также имеет имя, которое, как правило, отражает общее назначение файлов, которые данный каталог объединяет. В свою очередь несколько каталогов могут быть объединены в каталог более высокого уровня, которому присваивается определенное имя, и т. д. Каталог самого высокого уровня называется *корневым каталогом*. Вызывая с помощью ОС информацию о содержании жесткого диска, пользователь видит именно корневой каталог, который можно сравнить с оглавлением книги, где перечислены все разделы. Внутри каждого раздела имеются главы — это подкаталоги, т. е. каталоги более низкого уровня. Внутри главы есть параграфы, которые и являются аналогами файлов. Такую структуру называют древовидной, или деревом (рис. 15.1): от корня ЭВМ ствол ведет к крупным ветвям (дискам), от них отходят ветки (каталоги), а с ветками связаны листья (файлы).

Файловая система размещения информации была заложена уже в первых ОС. Эти системы обеспечивали управление взаимодействием процессора с устройствами ввода — вывода и всей информацией, размещаемой на дисках. Поэтому они назывались *дискowymi операционными системами* (ДОС).

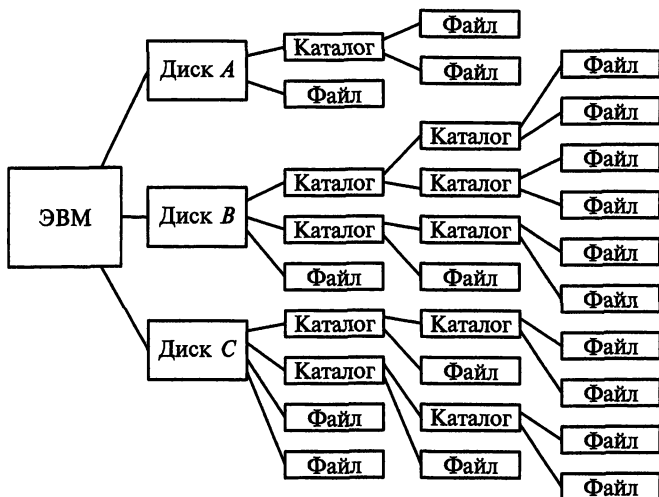


Рис. 15.1. Дерево файлов

Операционная система ДОС состоит из нескольких частей: базовой, резидентной и загружаемой.

Базовая система ввода—вывода (BIOS — Basic Input/Output System) находится в постоянной памяти (ПЗУ) компьютера. Эта часть ОС является неотъемлемой частью компьютера. Ее назначение состоит в выполнении наиболее простых и универсальных действий ОС, связанных с осуществлением ввода—вывода. Базовая система ввода—вывода содержит также тест функционирования компьютера, проверяющий работу памяти и других устройств компьютера при включении его электропитания. Кроме того, BIOS (по-русски так и говорят — БИОС) содержит программу вызова загрузчика ОС. Таким образом, БИОС — это программы ОС, хранящиеся в постоянной памяти компьютера.

Вводя с помощью клавиатуры команды ДОС, пользователь может выполнять некоторые изменения в настройке отдельных узлов компьютера, обрабатывать свои файлы, запускать требуемые программы. В промежутках между выполнением запускаемых пользователем команд ДОС находится в состоянии ожидания команд пользователя. Когда компьютер выключен, ОС хранится на жестком магнитном диске в виде так называемых системных файлов. После включения компьютера системные файлы загружаются в оперативную память, где и находятся в течение всего времени работы компьютера. Эта часть ОС называется резидентной.

Часть программ ДОС остается на жестком диске. Они загружаются в оперативную память лишь по мере необходимости и освобождают ее после отработки. Эта часть ОС называется загружаемой.

15.2. Операционная система MS-DOS

Широкое распространение в 1980-е годы получила разработанная фирмой Microsoft дисковая операционная система MS-DOS. С течением времени эта система совершенствовалась. Очередная, более совершенная версия получала более высокий номер. При включении компьютера прежде всего происходит выполнение некоторых программ для возможности дальнейшей работы. Эти программы указаны в так называемых системных файлах, наиболее важными из которых являются: IO.SYS; MSDOS.SYS; CONFIG.SYS; AUTOEXEC.BAT.

Название файла состоит из двух частей, разделяемых точкой. Эти части называются соответственно *имя* и *расширение*. Имя в MS-DOS может содержать от 1 до 8 символов, а расширение — от 1 до 3. Обычно для имени файла рекомендуется использовать латинские буквы, цифры и разделитель «-» или «_». Русские буквы (кириллица) не всеми версиями ДОС правильно отображаются на мониторе. Расширение в названии файла не является обязательным. Оно, как правило, поясняет содержание файла. Например, готовые к выполнению программы обычно имеют расширения .com; .exe; командные файлы — .bat; текстовые файлы — .txt; .rtf. В имени и расширении прописные и строчные латинские буквы воспринимаются одинаково, поскольку ДОС переводит все строчные буквы в соответствующие прописные. Некоторые сочетания символов (латинских букв) зарезервированы для обращения к различным устройствам компьютера: PRN (принтер), CON (при вводе — клавиатура, при выводе — монитор), LPT1 (устройство, подключенное к первому параллельному порту), COM3 (устройство, подключенное к третьему последовательному порту).

Имена каталогов обычно дают без расширения. Для вызова нужного файла в командной строке с помощью клавиатуры набирают название файла, перед которым указывается весь путь к каталогу, в котором этот файл находится. Например, пусть файл IVANOVU.TXT находится в подкаталоге PISMA, который в свою очередь находится в каталоге DOCUMENT корневого каталога диска C:\. Тогда для вызова указанного файла надо набрать команду

```
C:\DOCUMENT\PISMA\IVANOVU.TXT.
```

Операционная система взаимодействует с различными устройствами компьютера с помощью специальных программ, называемых драйверами. Такие устройства, как мышь, дисковые накопители, платы сетевого адаптера и принтеры, поставляются вместе со своими драйверами. Операционная система компьютера на распознает устройство до тех пор, пока не будет установлен необходимый драйвер (если операционная система не поддерживает спецификацию Plug and Play). Большинство драйверов предостав-

ляется производителями операционных систем. В некоторых случаях драйверы можно получить у оперативных служб типа The Microsoft Network (MSN) или CompuServe.

В сетевой среде каждый компьютер обладает платой сетевого адаптера и соответствующим ей драйвером, благодаря которым компьютер посылает данные по сети. Современные процедуры (с графическим интерфейсом пользователя) облегчают установку драйвера платы сетевого адаптера. Иногда возникает необходимость в обновлении или удалении драйвера.

В начале 1990-х годов широкую популярность приобрела система Norton Commander, повышающая эффективность работы дисковой ОС. Однако к тому времени для ОС MS-DOS уже было разработано большое число очень удобных и понравившихся пользователям программ. Поэтому при совершенствовании ОС в них стал предусматриваться так называемый режим эмуляции MS-DOS, благодаря которому можно использовать старые популярные программы.

15.3. Операционная система-оболочка Windows

Большой разницы между операционной системой и операционной оболочкой нет, но все же понятие оболочка более общее, чем система. Операционная оболочка, как уже отмечалось в подразд. 16.1, представляет собой надстройку над дисковой операционной системой, обеспечивающую пользователю дополнительные удобства и возможности. Операционная система Windows представляет много таких возможностей, поэтому правильнее отнести ее к операционным системам-оболочкам.

В течение долгих лет с момента своего появления персональные компьютеры (IBM и IBM-совместимые) обходились без специальных пользовательских оболочек, работая непосредственно под управлением операционной системы (MS-DOS или какой-то другой). Пользователи, садившиеся за такой компьютер, видели после включения на пустом экране только подсказку C:\>. Все операции управления компьютером производились путем ввода с помощью клавиатуры некоторых слов-директив или имени исполняемого файла.

В начале 1990-х годов появилась и быстро приобрела популярность операционная система Norton Commander, которая основывалась также на MS-DOS. Эта система сделала общение пользователя с компьютером наглядным. На экране монитора стало возможным увидеть содержимое всех дисков и директорий (каталогов) компьютера, включая и все подкаталоги. При этом одновременно воспроизводится содержимое двух директорий, поскольку на экране появляются две панели (рис. 15.2) и для работы в данный момент можно выбрать любую из них. На панелях содержатся

также сведения о времени создания и размерах (в байтах) каждого файла. Для вызова соответствующего файла достаточно подвести курсор к его имени и нажать клавишу «Ввод». Ниже панелей размещается и привычное приглашение MS-DOS, с помощью которого можно вводить команды прежним способом (т. е. набором имени файла на клавиатуре). А еще ниже помещается строка, помогающая выполнять целый ряд полезных операций с помощью функциональных клавиш. Система NC (под таким именем она запускается) способна создавать, переименовывать, перемещать и удалять каталоги, копировать, переименовывать, пересылать и удалять файлы, просматривать документы и текстовые файлы, редактировать их и делать многое другое. Кроме того, она помогает пользователю, предоставляя консультации по всем вопросам ее эксплуатации. В ней уже можно обеспечить совместную работу двух компьютеров, соединенных между собой коротким кабелем. Популярность системы Norton Commander в России была столь велика, что вскоре появилась и аналогичная ей русскоязычная система Volkov Commander. Развитие ОС на этом не остановилось.

Версия 3.0 системы-оболочки Windows и появившаяся следом версия 3.1 привнесли совершенно иные (по сравнению с MS-DOS) принципы в сферу общения пользователя с ЭВМ. Эти принципы можно считать относительно новыми, поскольку машины фирмы Apple строились на некоторых из них уже в течение ряда лет. Основная идея, заложенная в основу системы-оболочки Windows, — естественность представления информации. Другими словами, информация должна представляться в той форме, кото-

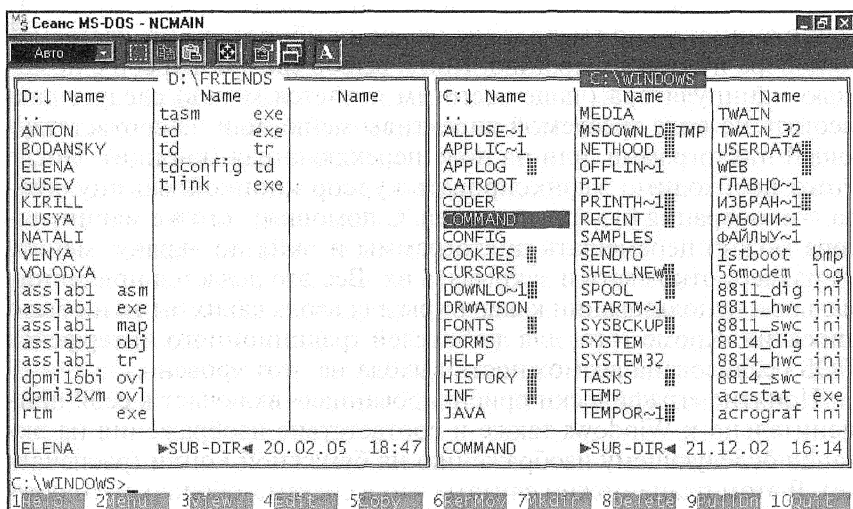


Рис. 15.2. Панели Norton Commander

рая обеспечивает наиболее эффективное усвоение ее человеком. Несмотря на простоту этого принципа, его реализация даже в рамках Windows 3.1 не лишена недостатков. В то же время эта система-оболочка представляет собой существенный шаг вперед по сравнению с предыдущими. Наиболее важными ее отличительными чертами являются следующие.

Система-оболочка Windows представляет собой замкнутую рабочую среду. Практически любые операции, доступные на уровне ОС, могут быть выполнены без выхода из Windows. Запуск прикладной программы, форматирование дискет, печать текстов — все это можно вызвать из Windows и вернуться к прежней программе по завершении операции. Многие основополагающие принципы и понятия среды Windows не отличаются от соответствующих принципов и понятий среды DOS.

Основными понятиями пользовательского интерфейса в среде Windows являются *окно* и *пиктограмма*. Пиктограмма (или ярлык, или иконка) — это миниатюрное графическое обозначение какого-либо раздела программного обеспечения, конкретной программы или даже отдельного файла. Все, что происходит в рамках системы-оболочки Windows, в определенном смысле представляет собой операцию либо с пиктограммой, либо с окном (или в окне). В среде Windows стандартизованы структура окон и расположение элементов управления ими, наборы операций и структура меню для сервисных программ, операции, выполняемые с помощью мыши для всех сервисных и прикладных программ.

Система-оболочка Windows представляет собой графически ориентированную оболочку. От пользователя не требуется ввод директив с помощью клавиатуры в виде текстовых строк. Ему достаточно внимательно смотреть на экран и выбирать из предлагаемого набора нужную операцию. Когда выбор сделан, курсор с помощью манипулятора (чаще всего им является мышь) следует разместить на поле требуемой директивы меню, или на соответствующей пиктограмме, или на поле переключателя (кнопки). После этого необходимо зафиксировать курсор кнопкой манипулятора — и операция будет выполнена. С помощью того же манипулятора можно перемещать пиктограммы и окна по экрану, менять их размер, открывать и закрывать их. Все это делается при минимальном использовании клавиатуры для ввода каких бы то ни было директив. Кроме того, для любителей традиционного интерфейса DOS реализована возможность выхода на этот уровень.

Понятие «графически ориентированная» включает в себя применительно к Windows также и соответствие изображения на экране последующему изображению на бумажной копии (распечатке). В этом плане можно считать, что в системе-оболочке Windows реализован принцип WYSIWYG (What You See Is What You Get — что вы видите, то и получаете), ранее являвшийся привилегией

относительно небольшого числа программ. С помощью True Type-шрифтов (при выборе шрифта видишь на экране его реальное изображение) этот принцип нашел в системе-оболочке Windows свое дальнейшее развитие.

Система-оболочка Windows обеспечивает независимый запуск и параллельное выполнение нескольких программ, в то время как большинство других операционных оболочек и систем рассчитано на выполнение в данный момент только одной программы. Каждая из параллельно (независимо) выполняемых программ имеет свое собственное окно. Переключение между выполняемыми программами производится фиксацией курсора с помощью манипулятора в окне требуемой программы.

Под управлением системы-оболочки Windows могут работать не только специальные программы, разработанные для эксплуатации в среде Windows (Windows-приложения), но и обычные программы, работающие в среде DOS, или так называемые DOS-приложения (DOS-прикладные программы). Система-оболочка Windows обеспечивает эффективный и комфортабельный обмен информацией между отдельными программами, выполняемыми под ее управлением.

Для начинающих в Windows предусмотрена обучающая программа, к которой можно обратиться уже на этапе инсталляции, т. е. при установке операционной системы-оболочки.

ОС Windows постоянно совершенствуется: последовательно появлялись системы Windows 95, 98, 2000, XP, а в 2007 г. — Виста.

15.4. Многопользовательская операционная система UNIX

До появления персональных компьютеров обработка информации и разнообразные вычисления производились в вычислительных центрах, оснащенных несколькими ЭВМ, среди которых были и очень мощные (следовательно, дорогие). Пользователями таких машин являлись научные сотрудники и инженеры-исследователи. Каждый из них был заинтересован в скорейшем выполнении своей задачи. Поэтому возникла потребность в многопользовательском режиме обработки данных на крупных ЭВМ. Для обеспечения такого режима и мультипрограммирования были созданы специальные многопользовательские и многозадачные ОС. Среди них первой функционально полноценной стала операционная система-оболочка OS/360, предназначенная для ЭВМ типа IBM/360. Использование данной ОС и последующих ее версий позволило существенно повысить производительность ЭВМ и оптимизировать загрузку технических средств, в частности периферийных устройств. Операционные системы коллективного пользования разграничивают функции операторов, администраторов, программистов и пользователей. Ранние версии таких ОС получа-

ли поток заданий в виде пакета перфокарт (тогда и появилось выражение «пакетная обработка»), поздние — на магнитной ленте и дисках. Выполнение заданий заканчивалось распечаткой на бумаге, которая передавалась пользователю. Анализируя отпечатанные данные, пользователь либо удовлетворялся решением своей задачи, либо вносил коррективы в программу ее решения (обычно с помощью программиста). Последующие версии многопользовательских ОС были ориентированы на диалоговую обработку данных. Пользователь получал в свое распоряжение виртуальную персональную ЭВМ и общался с ней в интерактивном режиме. Он получал возможность вносить необходимые коррективы непосредственно в процессе выполнения своей задачи, становился активным участником этого процесса. Практически одновременно с ним и другие пользователи решали свои задачи. Каждому из них была доступна полная мощность вычислительной установки.

В настоящее время самой распространенной из ОС коллективного пользования является система UNIX. Она установлена на многих высокопроизводительных компьютерах. Следует отметить, что MS-DOS функционально является подсистемой этой системы. Операционная система UNIX — одна из самых популярных в мире ОС благодаря тому, что ее сопровождает (т. е. создает для нее свои программы) и распространяет большое число компаний. Эта система была разработана фирмой Bell Laboratories (современное название Lucent Technologies) в 1969 г. как многозадачная система для мэйнфреймов. Она вобрала в себя целый ряд новых разработок в области ОС. В принципе, система UNIX создавалась как ОС для исследователей. При ее разработке была поставлена задача создать систему, которая могла бы удовлетворять непрерывно изменяющимся требованиям большого числа сотрудников, занимающихся разнообразными исследованиями.

В 1970 г. код системы с машинно-зависимого языка ассемблер (на котором тогда писались все операционные системы) был переписан на язык высокого уровня Си. Это позволило иметь всего одну версию системы UNIX, которую потом можно было компилировать Си-компиляторами на различных машинах. Система UNIX получила способность работать на различных типах машин практически без перепрограммирования. Кроме того, она позволяла иметь несколько видов интерфейсов взаимодействия между ядром и пользователем, или интерпретаторов.

В 1974 г. система UNIX была передана университетам «для образовательных целей», а несколько лет спустя нашла коммерческое применение. Постепенно она превратилась в одну из наиболее распространенных ОС. Сейчас существуют версии UNIX для многих ЭВМ, начиная от персонального компьютера до суперкомпьютеров.

Для проведения сложных экспериментальных исследований, связанных с большим количеством громоздких вычислений, об-

ширным объемом данных, требуются значительные системные ресурсы.

В этом случае система UNIX позволяет организовать кластер, т.е. многомашинный вычислительный комплекс, где все ресурсы компьютеров (дисковое пространство, память, ресурсы процессора) являются разделяемыми и доступными для любого пользователя в соответствии с его правами. В такой системе существует возможность постоянного наращивания мощности кластера путем подсоединения дополнительных компьютеров, а работа в ней остается для пользователя абсолютно «прозрачной», как если бы он работал на одном компьютере с огромными ресурсами.

Операционная система UNIX содержит следующие основные компоненты:

- ядро системы (core);
- оболочка ядра системы (kernel);
- средства разработки программ (development system);
- экранные руководства пользователя и программиста (manuals);
- эмулятор MS-DOS (VP/ix);
- средства передачи данных по каналам связи (UUCP);
- механизм сетевых протоколов (STREAMS);
- сетевой протокол (TCP/IP);
- сетевая файловая система (NFS);
- средства сетевых графических интерфейсов пользователя (X Window);

- командная оболочка на базе этих интерфейсов (Looking Glass).

Ядро управляет основными ресурсами (процессор, оперативная память) и периферийными устройствами обмена и хранения данных (магнитные диски, магнитные ленты, принтеры, терминалы, линии связи и т.д.). Одной из функций ядра системы UNIX является программная поддержка файловой системы. Командный интерпретатор Shell обслуживает терминал пользователя.

Пользователь ОС UNIX не один работает с этой системой, для него отведен рабочий период, т.е. *начало* и *конец* сеанса работ. Каждый пользователь системы имеет *имя пользователя* (для установления взаимодействия пользователей и начисления расходов) и *пароль пользователя* (для контроля входа в систему и защиты своих данных).

Пользователи могут быть объединены в группы (например, во время работы над совместными проектами) для разделения общих ресурсов, тогда дается еще имя группе пользователей. Один пользователь считается старшим (его называют *superuser*), он выполняет функции администратора системы (*root*). В частности, он «заводит» (регистрирует) всех прочих пользователей, а сам относится к числу так называемых спецпользователей. В функции администратора системы входит повседневное управление системой во всех аспектах ее существования, в том числе подключение но-

вых пользователей, управление файловой системой, изменение конфигурации и т. п. На персональных компьютерах эти функции может выполнять системный программист.

Основной проблемой многопользовательского и мультипрограммного режима в любой ОС является организация планирования параллельного выполнения нескольких процессов. Операционная система должна обладать четкими критериями для определения того, какому готовому к выполнению процессу и когда предоставить ресурс процессора. Система UNIX является системой разделения времен, т. е. она стремится к «справедливому» разделению ресурсов одного или нескольких процессоров между задачами разных пользователей. При этом необходимо, чтобы сами пользователи не ощущали того, что решение их задачи замедляется. Это означает, что время реакции на каждое действие интерактивного пользователя должно находиться в допустимых пределах. В последнее время возрастает тенденция к использованию ОС UNIX в приложениях реального времени, что повлияло на алгоритмы планирования.

В семейство UNIX-подобных ОС входит и система LINUX — многозадачная и многопользовательская ОС для бизнеса, образования и индивидуального программирования. Эта система применяется для обработки данных в сфере финансов и бизнеса, медицине, телекоммуникациях и т. д. В ней используется графический пользовательский интерфейс X Window, для которого разработан целый ряд программ управления окнами, предоставляющих многочисленные удобства пользователю в интерактивном режиме общения с компьютером.

Контрольные вопросы

1. Для чего нужна операционная система?
2. Как работает операционная система MS-DOS?
3. Как вводится команда в системе Norton Commander?
4. Какие основные достоинства у операционной системы-оболочки Windows?
5. В чем особенности операционной системы UNIX?

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**16.1. Назначение и виды программного обеспечения**

Программное обеспечение (ПО) компьютера можно подразделить на общесистемное и прикладное. Основой общесистемного ПО является *операционная система*, которая обеспечивает функционирование и взаимосвязь всех компонентов компьютера и предоставляет пользователю доступ к его аппаратным возможностям.

При традиционной классификации программных средств различают программы пользователей, часто называемые прикладными, или проблемными, и системные программы, поддерживающие работу вычислительных систем, комплексов и сетей в автоматическом режиме. Программные средства пользователей включают в себя исполняемые программные модули, или пакеты программ, и библиотеки объектных модулей — комплексы долговременно сохраняемых программ для решения разнообразных задач из узкой предметной области пользователя.

Управляющие системные программы, организующие корректное выполнение процессов и функционирование всех устройств системы при решении задач, как правило, постоянно сохраняются в основной памяти. Они составляют так называемое ядро ОС, постоянно находятся в основной памяти и поэтому называются *резидентными* (resident) программами. Те управляющие программы, для которых не хватает места в основной памяти и которые загружаются в память непосредственно перед выполнением, принято называть *транзитными* (transitive) программами.

Обрабатывающие системные программы выполняются как специальные проблемные задачи или приложения ОС (applications), решаемые пользователем или программистом при создании новых и модификации ранее разработанных программ.

Прикладное программное обеспечение можно в свою очередь разделить на две группы — средства разработки и приложения.

Средства разработки — это инструменты программиста. Традиционными средствами разработки являются системы (среды) программирования (СП), использующие алгоритмические языки программирования (ЯП). Основой систем программирования являются трансляторы, т. е. программы, обеспечивающие перевод исходного текста программы (на ЯП) на машинный язык (объектный код). Различают два вида трансляторов: интерпретаторы и компиляторы (см. подразд. 17.5).

Приложения — это программные продукты, предназначенные для решения задач в какой-либо конкретной предметной области. Многообразии приложений соответствует спектру задач, которые могут быть решены алгоритмически.

Рассмотрим несколько подробнее обе группы прикладного программного обеспечения.

Системы программирования предназначены для профессиональных программистов. Они позволяют разрабатывать программы на различных языках программирования. В последнее время широкое распространение получили системы визуального программирования (Visual Basic, Borland Delphi и др.), которые позволяют даже начинающему пользователю компьютера создавать не сложные программы.

Приложения позволяют пользователю обрабатывать текстовую, графическую, числовую, аудио- и видеoinформацию, а также работать в компьютерных сетях, не владея программированием. Приложение функционирует под управлением определенной ОС. Так, текстовый редактор Word является приложением ОС Windows, а редактор Edit — приложением ОС MS-DOS.

Для всех пользователей компьютера нужны *приложения общего назначения*, к числу которых относятся текстовые и графические редакторы, электронные таблицы, системы управления базами данных, а также приложения для создания мультимедиа-презентаций. В настоящее время наибольшее распространение получили пакеты приложений общего назначения Microsoft Office и StarOffice.

Быстрое развитие глобальных и локальных компьютерных сетей потребовало создания различных *коммуникационных программ*, которые включаются непосредственно в состав ОС, в частности ОС Windows.

В связи с широким распространением компьютерных вирусов потребовались специальные *антивирусные программы*.

Для использования в различных сферах профессиональной деятельности созданы *приложения специального назначения*. К ним относятся системы компьютерной графики (в том числе трехмерной), системы автоматизированного проектирования (САПР) и моделирования, программы бухгалтерского учета, компьютерные словари, системы автоматического перевода и др.

Все шире используются *обучающие программы* для самообразования или учебного процесса. Прежде всего это программы обучения иностранным языкам, работе на компьютере, тесты по различным предметам и т. д.

Большую пользу приносят различные *мультимедиа-приложения* (энциклопедии, географические атласы, справочники и т. д.) на лазерных дисках, которые содержат огромный объем информации и средства быстрого ее поиска.

Многие пользователи начинают знакомство с компьютером с компьютерных игр, которые бывают различных типов: логические, стратегические, спортивные и т.д.

По мере развития ОС они включают в себя все больше приложений, становясь в результате операционными оболочками и средами.

Драйвер (англ. driver) — это программное обеспечение, позволяющее компьютеру работать с различными устройствами. Даже если некоторое устройство и подключено к компьютеру, операционная система не сможет взаимодействовать с ним до тех пор, пока не будет установлен и правильно сконфигурирован драйвер этого устройства. Драйвер — это программа, которая «говорит» компьютеру, как надо управлять или работать с устройством, чтобы оно правильно выполняло свои функции.

Драйверы существуют почти для каждого типа устройств компьютера и периферии, например:

- для устройства ввода (мыши);
- жестких и гибких дисков;
- устройств мультимедиа (микрофонов, видеокамер, записывающих устройств);
- плат сетевого адаптера;
- принтеров, плоттеров, накопителей на магнитной ленте и т.д.

Примером использования драйверов может служить драйвер принтера. Принтеры производятся большим количеством фирм и обладают различными функциями и особенностями. Производители компьютеров просто не в состоянии оснастить свои компьютеры программным обеспечением для работы с каждым типом принтера. Для того, чтобы компьютер мог посылать документы на принтер, сначала надо загрузить драйвер этого принтера, который обеспечит взаимодействие компьютера с этим устройством.

16.2. Текстовые редакторы

Широкое применение персональных компьютеров потребовало разработки программного обеспечения для подготовки различных текстовых документов. Раньше такие документы выполнялись на пишущих машинках. В настоящее время стало значительно проще и удобнее, а следовательно, и выгоднее подготавливать текстовые документы на ПК. Пользователь может также составлять и редактировать на ПК различные программы для последующего их исполнения. Для этого имеется широкий выбор редакторов текстов программ.

Редакторы текстов программ рассчитаны на редактирование программ на том или ином языке программирования. Часто они встроены в систему программирования, с помощью которой можно запускать программы на компиляцию и выполнение.

Редакторы, рассчитанные на тексты программ, как правило, выполняют следующие функции:

- диалоговый просмотр текста;
- редактирование строк программы;
- копирование и перенос блоков текста;
- перенос одной программы или ее части в указанное место другой программы;
- контекстный поиск и замену подстрок текста;
- автоматический поиск строки, содержащей ошибку;
- распечатка программы или ее части.

Кроме того, редакторы текстов программ часто позволяют автоматически проверять синтаксическую правильность программ. Иногда эти редакторы бывают объединены с отладчиками программ на уровне исходного текста.

Редакторы текстов программ, как правило, можно использовать для создания и корректировки небольших документов. Однако для серьезной работы с документами лучше использовать редакторы, ориентированные на работу не с программами, а непосредственно с документами.

Удобство и эффективность применения ПК для подготовки текстов привели к созданию множества программ для обработки документов. Такие программы называются *редакторами текстов* (Word Processors), или *текстовыми редакторами*, или *редакторами документов*. Возможности этих программ различны; одни предназначены для подготовки небольших документов простой структуры, другие — для набора, оформления и полной подготовки к типографскому изданию книг и журналов (издательские системы).

Редакторы документов (текстовые редакторы) — это программы для обработки документов. В отличие от редакторов текстов программ они ориентированы на работу с текстами, имеющими структуру документа, т.е. состоящими из разделов, страниц, абзацев, предложений, слов и т.д. Поэтому редакторы документов могут обеспечивать функции, ориентированные на структуру документа:

- использование различных шрифтов для буквенных и иных символов в тексте документа;
- работу с пропорциональными шрифтами, имеющими разные размеры при печати на бумаге;
- задание произвольных межстрочных промежутков в тексте документа;
- автоматический перенос слов на новую строку;
- автоматическую нумерацию страниц и изменение ее при редактировании;
- обработку и нумерацию сносок;
- печать верхних и нижних заголовков страниц (колонтитулов);
- выравнивание краев абзаца;
- набор текста в несколько столбцов;

- создание таблиц и построение диаграмм;
- проверку правописания (орфографии и знаков препинания) на разных языках и подбор синонимов;
- построение оглавления, индексов и т. д.

Всего существует несколько сотен текстовых редакторов, от самых простых до весьма мощных и сложных. К наиболее распространенным в мире редакторам относятся Microsoft Word, WordPerfect, WordStar. Они обладают очень широкими возможностями, но вполне доступны для изучения и эффективного повседневного использования в работе.

Среди простых текстовых редакторов в России наибольшее распространение получил Лексикон. Он имеет интерфейс на русском языке и позволяет подготавливать несложные документы на русском и английском языках.

16.3. Электронные таблицы

Таблицы являются важным вспомогательным средством для представления данных и выполнения различных расчетов в математике, технике, быту. Их применяли еще в глубокой древности. Компьютерные программы, позволяющие составлять таблицы на ЭВМ, получили название *электронные таблицы*.

С помощью таких программ можно выполнять технические и бухгалтерские расчеты с большим количеством данных, строить графическое изображение результатов расчета, обновлять и изменять исходные данные. Современные электронные таблицы обладают очень широкими функциональными возможностями (некоторые из них можно использовать и как текстовые редакторы), но главными из них являются именно расчеты и построение диаграмм. Рассмотрим подробнее выполнение с помощью расчетов электронных таблиц.

После запуска программы на экране монитора появляется таблица, разделенная на ячейки (табл. 16.1).

Таблица 16.1

Структура электронной таблицы

	A	B	C	...	AA	AB	..	IU	IV
1									
2									
...									
16383									
16384									

Столбцы таблицы обозначают буквами. Так, в программе Excel используются буквы английского алфавита: А, В, С и т.д. Всего их 26. Для большего числа столбцов применяют двухбуквенные сочетания АА, АВ, АС и так до IV (не путайте с римскими цифрами, это английские буквы «ай», «ви»). Таким образом, общее число столбцов может достигать 256.

Каждая ячейка имеет собственное обозначение, состоящее из одной или двух латинских букв, обозначающих столбец, и цифр, обозначающих строку.

Для заполнения нужной ячейки необходимо подвести к ней курсор и щелкнуть левой клавишей мыши — ячейка помечается серой рамкой. Слева на строке формул, расположенной под панелями инструментов, будет показан адрес этой текущей ячейки, например F2. В указанную ячейку можно заносить цифровые данные или текст, поясняющий содержание столбца, строки или всей таблицы.

Большим удобством является возможность ввода в ячейку формулы, по которой вычисляется содержимое данной ячейки. Например, при подсчете расходов за I квартал в ячейку B2 заносят расходы за январь, в ячейку C2 — за февраль, в ячейку D2 — за март. В ячейку F2, отведенную для суммарных расходов за все три месяца, записывают формулу = B2 + C2 + D2. В строке формул появится та же запись. Теперь при каждом изменении содержания ячеек с расходами за январь, февраль и март будет автоматически изменяться и их сумма в ячейке F2. Аналогично вводятся формулы для подсчетов в столбцах.

Для экономии времени при вводе формул программа предлагает функцию автоматической суммы. На панели инструментов имеется кнопка с изображением знака суммы Σ .

Кроме знака “+” в формулах можно использовать знаки “-” (для вычитания), “*” (для умножения), “/” (для деления), “^” (для возведения в степень), а также скобки, проценты, операторы сравнения (больше, меньше, равно). Осваивать электронные таблицы следует по специальным руководствам в виде обычной книги или по электронному учебнику, который является частью программы Excel. Для запуска справки обычно требуется нажать клавишу F1 на клавиатуре или указать курсором на вопросительный знак на панели инструментов.

Осуществлять сортировку данных, т.е. производить их упорядочение, позволяют электронные таблицы, данные в которых (числа, текст, даты) можно сортировать по возрастанию или убыванию. При сортировке по возрастанию данные выстраиваются в следующем порядке: числа — от наименьшего отрицательного до наибольшего положительного; текст — сначала числа, затем знаки, далее слова по алфавиту (сначала латинскому, потом русскому); пустые ячейки всегда помещаются в конце списка.

Для сортировки строк таблицы необходимо выбрать столбец, по данным которого будет изменяться порядок следования строк.

Можно сортировать данные последовательно, по нескольким столбцам (так называемая *вложенная сортировка*). При этом строки, имеющие одинаковые значения в ячейках первого столбца, будут упорядочены по значениям в ячейках второго столбца, а строки, имеющие одинаковые значения в ячейках второго столбца, — по значениям в ячейках третьего столбца и т. д.

Например, табл. 16.2 является результатом вложенной сортировки таблицы, содержащей данные о компьютерах. Сортировка столбца А выполнена по возрастанию, столбца В — по убыванию и столбца С — по возрастанию.

В соответствии с заданными условиями, которые называются *фильтром*, в электронных таблицах можно осуществлять поиск данных (строк). В результате такого поиска будут найдены строки, удовлетворяющие заданному фильтру. Условия задаются с помощью операций сравнения. Для числовых данных это операции *равно* (знак =), *меньше* (<), *больше* (>), *меньше или равно* (\leq) и *больше или равно* (знак \geq).

Для текстовых данных возможны операции сравнения *равно* (ищутся одинаковые слова или сочетания знаков), *начинается с* (сравниваются первые символы), *заканчивается на* (сравниваются последние символы), *содержит* (сравниваются символы в любой части текста). Для задания условия необходимо выбрать операцию сравнения и задать последовательность символов.

Поиск данных можно осуществлять, вводя условия поиска для нескольких столбцов. Фильтр в этом случае будет содержать несколько условий, которые должны выполняться одновременно. Например, если мы хотим в таблице данных компьютеров (см. табл. 16.2) найти данные о настольном компьютере с процессо-

Таблица 16.2

Данные компьютеров после сортировки

	А	В	С
1	Тип компьютера	Процессор	Объем памяти, Гбайт
2	Настольный	Pentium 4	20
3	Настольный	Pentium 4	120
4	Настольный	Athlon	20
5	Настольный	Athlon	80
6	Ноутбук	Pentium 4	20
7	Ноутбук	Pentium 4	30

ром Pentium 4 и памятью 120 Гбайт, то необходимо задать фильтр, состоящий из трех условий: для столбца «Тип компьютера» равно *Настольный*; для столбца «Процессор» равно *Pentium 4*; для столбца «Объем памяти» больше или равно *120*.

Электронные таблицы позволяют представить данные, имеющиеся в документе, в виде диаграмм, которые наглядно отображают зависимости между этими данными, облегчают восприятие и помогают при их анализе и сравнении. При этом диаграммы могут быть различных типов и, следовательно, представлять данные в различной форме.

Для каждого набора данных важно правильно подобрать тип создаваемой диаграммы. Для наглядного сравнения различных величин используется *линейчатая диаграмма* (рис. 16.1, а), с помощью которой можно, например, представить данные о численности населения различных стран. Для отображения величин частей от целого применяется *круговая диаграмма* (рис. 16.1, б),

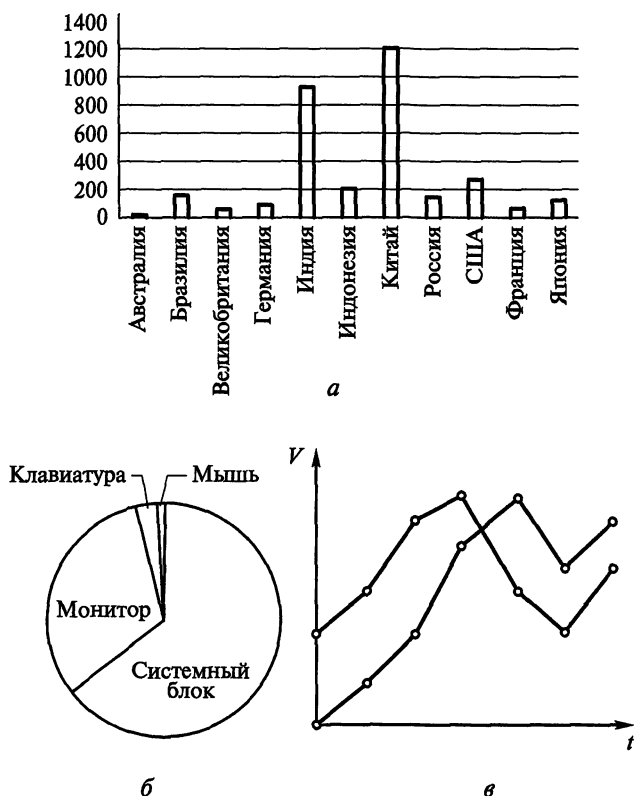


Рис. 16.1. Типы диаграмм:

а — линейчатая; б — круговая; в — график

которая позволяет, к примеру, наглядно показать доли стоимости отдельных устройств компьютера в его общей цене. Для отображения изменения величин в зависимости от времени или каких-то иных параметров используется *график* (рис. 16.1, в). Графики позволяют анализировать закономерности изменения величин.

Диаграммы могут располагаться как на листе с данными (внедренные диаграммы), так и на отдельных листах. Поскольку диаграммы связаны с исходными данными в документе, то они автоматически обновляются при замене этих данных. Для построения диаграмм используется опция (т.е. услуга, предоставляемая программой Excel) «Мастер диаграмм», которая позволяет создавать диаграмму с помощью диалоговых панелей.

16.4. Базы данных

Современные системы обработки информации (СОИ) выполняют преобразование больших массивов данных. Синонимом понятия «данные» служат термины «информация» и «сведения». Как показывает опыт разработки и эксплуатации СОИ, способы хранения и обработки данных оказывают решающее влияние на показатели функционирования системы в целом, на ее практическую эффективность. Систематическая организация данных и способов их обработки осуществляется в банках данных.

Банк данных (БНД) — это совокупность базы данных и систем управления базами данных.

База данных (БД) — это структурированная (т.е. определенным образом организованная) совокупность данных. Наименьшая единица описания данных называется *элементом описания*. Совокупность элементов описания, объединенных отношением принадлежности к одному описываемому объекту, называется *записью*. Если элементы описания соответствуют отдельным свойствам объекта, то запись описывает объект в целом. Например, код типа микросхемы, логическая функция, мощность потребления, коэффициент разветвления в совокупности составляют запись и описывают свойства конкретного объекта — микросхемы.

Система управления базами данных (СУБД) состоит из языковых и программных средств, предназначенных для создания и использования БД прикладными программами, а также непосредственно пользователями-непрограммистами.

Применение БНД позволяет решить следующие проблемы организации и введения больших массивов информации:

- сокращение избыточности;
- обеспечение целостности;
- разграничение доступа;
- обеспечение независимости представления данных.

Избыточность данных вызывается наличием различных форм представления одних и тех же данных, размножением части данных для дальнейшего использования прикладными программами, повторными записями одинаковых данных на различных физических носителях.

Целостностью называется свойство БД в любой момент времени содержать лишь достоверные данные. Наличие избыточности, противоречивых и неверно составленных данных нарушает целостность БД.

Для сокращения избыточности производится объединение одинаковых по смыслу, но имеющих различный тип данных в единую БД с приведением к общему, стандартизованному виду. Процесс объединения данных, используемых различными пользователями, в одну общую БД называется *интеграцией базы данных*.

Однако каждый конкретный пользователь должен получить доступ лишь к некоторому подмножеству данных из БД, необходимому ему для выполнения своих прикладных программ. Одновременно с разграничением доступа обеспечивается режим секретности и повышается степень защищенности данных от несанкционированного доступа.

Одним из важнейших преимуществ применения БД является возможность обеспечения независимости представления данных в прикладных программах от типов запоминающих устройств и способов их физической организации. В основном это достигается построением двух уровней представления данных — логического и физического.

На логическом уровне данные представляются в виде, удобном для использования в прикладных программах или непосредственно проектировщиками. Физический уровень представления данных отражает способ хранения и структуру данных с учетом их расположения на носителях информации в запоминающих устройствах ЭВМ.

Важнейшим понятием БД является *модель данных* — формализованное описание, отражающее состав и типы данных, а также взаимосвязи между ними. Модели данных классифицируются по ряду признаков.

В зависимости от объемов описываемой информации на логическом уровне различают внешнюю и внутреннюю модели данных. Внешняя модель данных (логическая подсхема) описывает структуру информации, относящейся к некоторой конкретной процедуре или к группе родственных проектных процедур. Внутренняя логическая модель данных (логическая схема) объединяет все подсхемы БД.

По способам отражения связей между данными на логическом уровне различают иерархическую, сетевую и реляционную модели. Модель называют *сетевой*, если данные и их связи имеют струк-

туру графа. Если структура отражаемых связей представляется в виде дерева, то модель называется *иерархической*. Представление данных в форме таблицы соответствует *реляционной* модели данных. Этот тип БД наиболее распространен. Рассмотренные в подразд. 16.3 электронные таблицы при выполнении функций сортировки и поиска действуют аналогично БД реляционного типа.

Базы данных находят широкое применение в электронных каталогах, картотеках, информационных и статистических системах, электронных магазинах. В сети Интернет давно и успешно используются многочисленные поисковые системы, в основе которых лежат СУБД.

В последнее время все более широкое распространение получают так называемые графические *геоинформационные системы* (ГИС). Заказчиками ГИС различного назначения являются крупные международные организации, правительства ряда стран, промышленные, транспортные, торговые, туристические фирмы и т.д. Популярность ГИС объясняется большими удобствами их использования, а также значительным экономическим эффектом от их внедрения.

Как отмечает один из ведущих специалистов по таким системам Юджин Бодански, ГИС предназначена для работы с пространственно распределенной информацией, которая используется для описания объектов, характеризуемых не только своими атрибутами (название, вес, цвет, стоимость, год выпуска и др.), но и координатами, геометрической формой и топологией. К указанным объектам относятся, например, географические и картографические объекты, отдельные узлы и линии машиностроительных и архитектурных чертежей, а также электрических схем. Город на мелкомасштабной карте имеет координаты в заданной системе координат (например, в прямоугольной системе координат какой-то карты или в географических координатах широты и долготы), река и дорога могут быть описаны с помощью координат вершин ломанных линий, место какого-либо события (например, дорожно-транспортного происшествия) может быть задано координатами или расстоянием вдоль дороги от определенного пункта.

Географическая информационная система содержит инструменты, позволяющие создавать пространственно распределенные базы данных, поддерживать их, заполнять информацией, трансформировать (например, переводить из одной системы координат в другую), хранить растровые изображения (например, аэро- и космические снимки, отсканированные чертежи, карты и др.) и манипулировать или формулировать запросы и отображать ответы на них в виде таблиц, фрагментов карт и схем, графиков и т.п.

Обрабатываемые ГИС запросы могут быть как весьма простыми, например: «Определить расстояние от пункта А до пункта В»,

«Найти площадь района С», «Вычертить все реки, показанные на данной карте, заданными условными обозначениями», так и достаточно сложными, например, «Определить расстояние от пункта А до пункта В по проселочным дорогам», «Определить то же расстояние по асфальтированным дорогам», «Подсчитать, сколько сосен старше 50 лет растет на данной территории на расстоянии более 100 м от берегов рек», «Показать контуры участков, удовлетворяющих следующим условиям:

а) находятся не далее 1 км от шоссе, по которому за день проезжает не менее 10 000 автомобилей;

б) на расстоянии менее 5 км от участка нет ни одного супермаркета (для выбора места строительства нового супермаркета)».

Современные географические базы данных часто содержат очень большое количество информации. Так, объем баз данных космических снимков либо данных о рельефе или загрязнении воздуха исчисляется сотнями терабайт.

16.5. Системы автоматизированного проектирования

Система автоматизированного проектирования (САПР) представляет собой комплекс технических и математических средств, предназначенных для автоматизации процессов проектирования с участием человека. Набор этих средств, последовательность их использования, участие человека на различных этапах проектирования определяет специальная программа, которая вводится в ЭВМ.

Проектирование любых изделий представляет собой творческий процесс, в ходе которого человек-проектировщик рассматривает несколько вариантов решения поставленной задачи, сформулированной в техническом задании. При этом варианты решения могут быть типовыми, основанными на уже ранее реализованных проектах, созданных при решении сходных, аналогичных задач. Новая задача может отличаться от прежних только иными числовыми характеристиками. Например, требуется спроектировать двигатель большей (по сравнению с уже выпускаемым) мощности или электротехническое изделие, отличающееся от прежних применением новых, более совершенных элементов, или механическое устройство, в конструкции которого использованы более качественные материалы, и т. п. В этих случаях программируется уже известная последовательность расчета, который затем и выполняется на основе иных исходных данных с помощью ЭВМ. В более сложных случаях, когда отсутствует типовое решение, человек-проектировщик на основе собственного опыта задает несколько вариантов возможного решения. Для каждого из этих вариантов существуют свой метод расчета и своя программа его выполнения. После выполнения этих расчетов на ЭВМ человек оценивает (срав-

нивает) их результаты и принимает решение о выборе наилучшего. В некоторых случаях выбор оптимального варианта, т. е. принятие решения, тоже может быть доверен ЭВМ, которая рассчитывает программным путем так называемый критерий оптимальности. Итогом автоматизированного проектирования должна стать техническая документация на проектируемое изделие, которая обычно содержит различные чертежи и схемы. Следовательно, САПР должна обладать хорошо развитыми средствами ввода, вывода и размножения графической информации и документации.

Обычно последовательность проектирования от технического задания к комплексу технической документации состоит из трех этапов: эскизного, технического и рабочего проектов. На каждом из этих этапов ЭВМ эффективно используется как для расчета, так и для моделирования. Любую последовательность действий можно формализовать с помощью алгоритма. Но проектирование, как уже отмечалось, является творческим процессом, который порой не поддается жесткой алгоритмизации. Поэтому в ходе проектирования возникает необходимость вмешательства человека, его участие в этом процессе. Для реализации этого участия в составе технических средств САПР необходимы устройства оперативного отображения графической информации и специальные пульты управления системой.

Программное обеспечение САПР состоит из двух основных частей: внешнего и внутреннего ПО.

Внешнее ПО — это программные средства общения человека-проектировщика с системой. В его состав входят языки представления исходной информации, средства пополнения информационной системы и языки управления работой САПР, позволяющие вести диалог между человеком и системой. Такое общение человека с машиной носит название *интерактивный режим*. Языки управления работой САПР называются *сервисными*.

Современные САПР имеют развитую систему сервисных команд, позволяющих по чертежам в трех проекциях делать трехмерное изображение детали или всего изделия, изменять его пропорции, показывать его с разных точек зрения.

Некоторые САПР являются векторными графическими редакторами, предназначенными для создания чертежей. При выполнении чертежей вручную с помощью карандаша, линейки и циркуля производится построение элементов чертежа (отрезков, окружностей и т. д.) с точностью, которую предоставляют чертежные инструменты. Использование САПР позволяет выполнять чертежи деталей и узлов с очень высокой точностью. На основе компьютерных чертежей затем создаются управляющие программы для станков с числовым программным управлением (ЧПУ), на которых по компьютерным чертежам многократно изготавливаются высокоточные детали из металла, дерева и т. д.

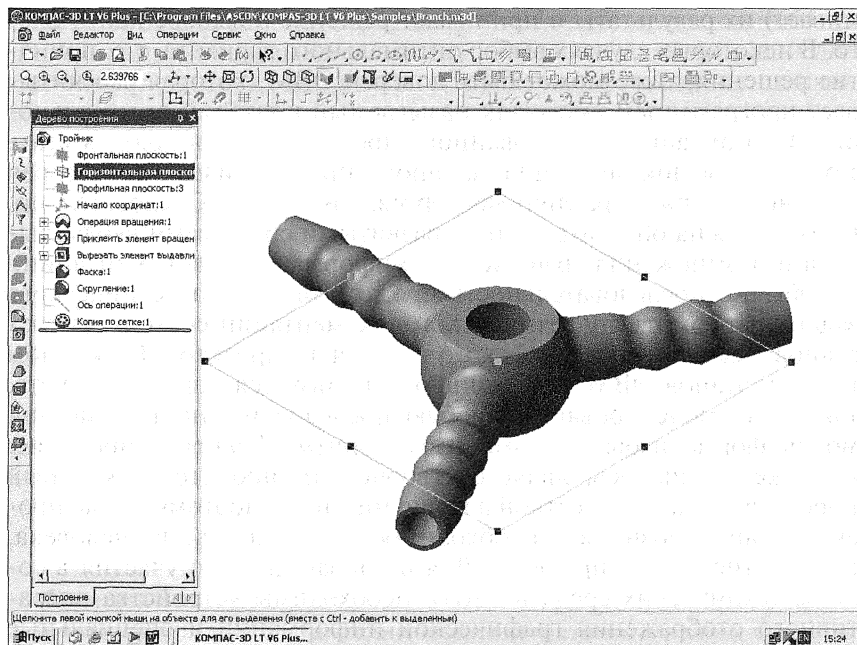


Рис. 16.2. Окно САПР «Компас-3D»

В качестве примера САПР рассмотрим систему «Компас-3D», которая позволяет создавать чертежи в соответствии с российскими стандартами. Версия «Компас-3D LT» предназначена специально для обучения компьютерному черчению в школах, техникумах и вузах. В центре рабочего окна «Компас-3D» (рис. 16.2) размещается система координат. Положение курсора отсчитывается от начала системы координат, а текущие значения его координат X и Y отображаются в правой части строки текущего состояния, расположенной в нижней части окна приложения, где указан и масштаб.

При разработке и редактировании чертежа используется инструментальная панель (рис. 16.3), которая по умолчанию размещается слева от чертежа. Она включает в себя пять вариантов рабочей панели («Геометрические построения», «Редактирование», «Выделение», «Измерения», «Размеры и технологические обозначения»), каждый из которых содержит набор кнопок определенного функционального назначения. Для выбора необходимого вариан-

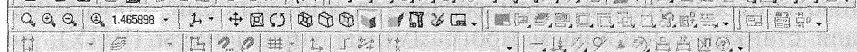


Рис. 16.3. Инструментальная панель

та рабочей панели служит панель переключения, которая размещена в верхней части инструментальной.

Панель «Геометрические построения» содержит кнопки, позволяющие рисовать на чертеже определенные объекты: точку, отрезок, окружность, овал, дугу, прямоугольник и др.

Панель «Редактирование» содержит кнопки, которые позволяют вносить изменения в чертеж, производя над объектами различные операции: перемещение, поворот, копирование, удаление, масштабирование и пр.

Панель «Выделение» позволяет осуществлять различные варианты выделения отдельных объектов и групп объектов.

Панель «Измерения» позволяет измерять расстояния (в миллиметрах), углы (в градусах), периметры и площади различных объектов.

Панель «Размеры и технологические обозначения» позволяет грамотно оформить чертеж: обозначить на чертеже размеры деталей, обработку, сделать надписи и т.д.

Для выбора создаваемого чертежного объекта (точка, отрезок, окружность, прямоугольник и пр.) используется панель «Геометрические построения». Каждый объект обладает определенным набором параметров, которые характеризуют его размеры и положение на чертеже. Они отображаются в строке параметров объекта. Например, после нажатия на панели «Геометрические построения» кнопки *Ввод отрезка* появится строка с полями для задания значений параметров отрезка: координат его начальной ($p1$) и конечной ($p2$) точек, длины (ln), угла наклона (an) и стиля линии. Строка параметров включает в себя также кнопку *Состояние полей*. По внешнему виду кнопки можно судить, в каком из трех состояний находится поле — фиксированном (обозначено крестиком), ожидания ввода (обозначено галочкой) или просто доступном для ввода (без каких-либо обозначений).

Создание и редактирование объектов с помощью строки параметров сводится к активизации нужных полей и вводу в них значений параметров. После ввода минимального набора значений параметров, достаточного для построения объекта (для отрезка — это координаты $p1$ и $p2$), система автоматически создает объект.

Можно осуществлять ввод параметров, автоматический, ручной и с использованием геометрического калькулятора.

В английском языке слова Computer-Aided Design (CAD) означают компьютерная помощь проектирования, т.е. имеют тот же смысл, что и наш термин САПР. Рассмотрим некоторые из таких CAD-систем, которые широко используются в качестве программного обеспечения. Эти системы находят применение в широком диапазоне инженерно-технической деятельности, начиная с решения сравнительно простых задач проектирования и изготовления конструкторско-технологической документации и кончая задачами объемного геометрического моделирования, веде-

ния проекта, управления распределенным процессом проектирования и т. п.

Данные системы значительно упрощают производственный процесс, тем самым снижая умственную нагрузку человека. С этой точки зрения их можно назвать интеллектуальными системами.

Система P-CAD выполняет полный цикл проектирования печатных плат, который включает в себя графический ввод в проектную документацию электрических схем, моделирование смешанных аналого-цифровых устройств, перенос схемы на печатную плату, интерактивное размещение компонентов, ручную, интерактивную и (или) автоматическую трассировку проводников, контроль ошибок в схеме и печатной плате и выпуск документации.

Система T-FLEX CAD LT предназначена для быстрого создания чертежной документации в полном соответствии с ЕСКД и международными стандартами, а также для управления параметрами уже созданного в системе T-FLEX CAD 2D чертежа. Используя функции эскизирования T-FLEX CAD LT, можно создавать отрезки, различные варианты дуг, полные окружности, многоугольники, автоматически наносить осевые линии, фаски, скругления и другие элементы.

Система T-FLEX CAD позволяет выполнять специальные конструкторские элементы оформления: наносить основную надпись, приводить на чертеже технические требования, автоматически подбирать форматку, наносить осевые линии и т. п. При заполнении углового штампа и таблицы спецификации пользователи работают в соответствующих полях прямо на чертеже.

Система T-FLEX CAD предоставляет возможность вводить текстовую информацию на поле чертежа в реальном режиме отображения. Текстовый редактор поддерживает форматирование абзацев текста (выравнивание, отступы, интервалы и т. д.), выделение фрагментов текста, использование разных шрифтов, размеров, стилей, цветов для фрагментов одного текста. В тексте можно использовать элементы оформления чертежа и даже его фрагменты. Для автоматизации ввода повторяющихся текстовых строк предусмотрен выбор из словаря.

Система T-FLEX CAD позволяет создавать любые таблицы, объединять их ячейки, задавать и изменять размеры полей, определять граничные линии, работать с многостраничными текстовыми документами и чертежами. Например, вся многостраничная спецификация на изделие может находиться в одном документе системы T-FLEX CAD вместе со сборочным чертежом.

Одной из передовых программ в области черчения и расчетов является также программа AutoCAD. С ее помощью выполняется компьютерная графика при конструировании пространственных объектов. Последние разработки в области САПР (CAD) обеспечивают создание трехмерного изображения и позволяют его рас-

смагивать под любыми углами, т. е. как бы с разных точек зрения.

Особое место среди множества систем математической поддержки исследований и проектирования занимает программный комплекс MathCAD — мощная и в то же время простая универсальная среда для решения задач в различных отраслях науки и техники, единственная система, в которой описание решения математических задач задается с помощью привычных математических формул и знаков. Программный комплекс MathCAD позволяет выполнять как численные, так и аналитические (символьные) вычисления, имеет чрезвычайно удобный математико-ориентированный интерфейс и прекрасные средства научной графики.

При проектировании электрических устройств и систем автоматизации большую помощь оказывают так называемые *электронные лаборатории* (Electronics Workbench). Они являются моделирующими системами, с помощью которых можно на экране компьютера собирать различные схемы из большого набора электрических элементов, а затем исследовать их поведение в различных режимах. На рис. 16.4 показаны собранная таким образом схема системы автоматического регулирования и график переходной характеристики, полученный на экране осциллографа.

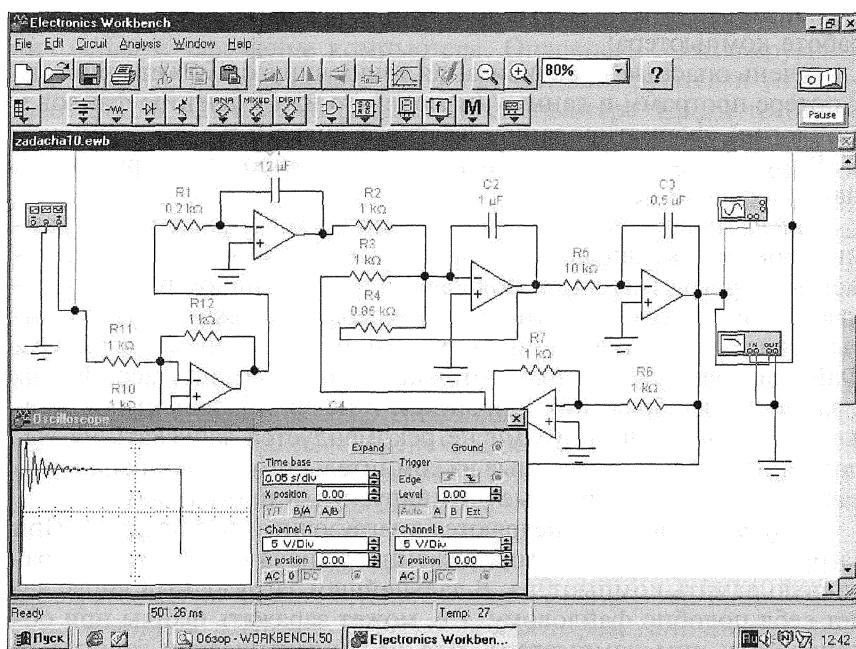


Рис. 16.4. Схема, смоделированная с помощью электронной лаборатории

16.6. Антивирусные программы

В настоящее время существует несколько десятков тысяч вирусов, заражающих компьютеры с различными ОС и распространяющихся по компьютерным сетям. Первая массовая эпидемия компьютерного вируса произошла в середине 1980-х годов, когда созданная с недобрыми намерениями программа стала «заражать» дискеты для первых массовых ПК. Способность к размножению (самокопированию) и незаметному для пользователя внедрению в файлы, загрузочные секторы дисков и документы — основное свойство компьютерного вируса. Именно по признаку способности к саморазмножению существующее в биологии понятие «вирус» стало применяться по отношению к компьютерным программам. После заражения компьютера вирус может активизироваться и заставить компьютер выполнять какие-либо действия. Начало активизации может быть связано с какими-либо событиями (наступлением определенной даты или дня недели, открытием документа и т.д.).

По степени вредного влияния вирусы подразделяются на следующие виды:

- неопасные — их влияние ограничивается уменьшением свободной памяти на диске, графическими, звуковыми и другими внешними эффектами;
- опасные — они могут привести к сбоям и зависаниям при работе компьютера;
- очень опасные — активизация таких вирусов может привести к потере программ и данных (изменению или удалению файлов и каталогов), форматированию жесткого диска и т.д.

Вирусы можно подразделить также на файловые, загрузочные, макровирусы и сетевые.

Файловые вирусы, внедряясь в исполняемые файлы (программы), обычно активизируются при их запуске. После запуска зараженной программы вирус появляется в оперативной памяти компьютера и становится активным (т.е. может заражать другие файлы) вплоть до выключения компьютера или перезагрузки ОС. Файловые вирусы не заражают файлы данных (например, файлы с записями изображения или звука). В целях профилактики заражения файловыми вирусами не рекомендуется запускать на выполнение файлы, полученные из сомнительного источника и предварительно не проверенные антивирусными программами.

Загрузочные вирусы внедряются в загрузочный сектор диска. При загрузке ОС вирусы с этого зараженного диска попадают в оперативную память компьютера. В дальнейшем загрузочный вирус ведет себя подобно файловому, т.е. может заражать файлы при обращении к ним компьютера. Для профилактической защиты от таких вирусов следует избегать загрузки компьютера с подозри-

тельных гибких дисков и устанавливать в BIOS компьютера защиту загрузочного сектора от изменений.

Макровирусы способны заражать файлы документов в приложениях Word и электронных таблиц в приложении Excel. Фактически они являются макрокомандами (макросами), которые встраиваются в документ. После загрузки зараженного документа в приложение макровирусы постоянно присутствуют в памяти компьютера и могут заражать другие документы. Только после закрытия приложения угроза заражения исчезает. Профилактическая защита от макровирусов — предотвращение их запуска. При открытии документа в приложениях Word и Excel на экране появляется сообщение о присутствии в них макросов (потенциальных вирусов) и предлагается запретить их загрузку. Запрет на загрузку макросов защищает компьютер от заражения макровирусами, однако отключает и полезные макросы, содержащиеся в документе.

Сетевые вирусы, распространяющиеся по компьютерной сети, могут быть любыми из вышеперечисленных вирусов. Заражение может произойти, например, при получении зараженных файлов с серверов файловых архивов. Существуют и специфические сетевые вирусы, которые используют для своего распространения электронную почту и сеть Интернет. Вирусы, которые распространяются по компьютерной сети во вложенных в почтовое сообщение файлах, носят название *интернет-черви* (worm). При обычном просмотре сообщения могут произойти автоматическая активизация червя и заражение компьютера. Опасность таких вирусов в том, что они по определенным датам активизируются и уничтожают файлы на дисках зараженного компьютера. Кроме того, интернет-черви выполняют роль троянского коня, внедренного в ОС пользователя. Они запоминают идентификатор и пароль пользователя для доступа в Интернет и передают их на определенный почтовый адрес, в результате чего злоумышленники получают возможность доступа в Интернет за чужие деньги.

Распространение сетевого вируса происходит лавинообразно. Вирус после заражения компьютера начинает рассылать себя по всем адресам электронной почты, которые имеются в адресной книге пользователя. Кроме того, может происходить заражение и по локальной сети, так как червь проникает в локальные и сетевые диски под случайным именем. Для профилактической защиты от Интернет-червей не рекомендуется открывать файлы, вложенные в почтовые сообщения, которые получены из сомнительных источников.

Особой разновидностью вирусов являются активные элементы (программы) на языках JavaScript или VBScript, так называемые *скрипт-вирусы*, которые могут выполнять разрушительные действия. Указанные элементы передаются по сети в процессе загрузки веб-страниц с серверов Интернета в браузер локального ком-

пьютера. Профилактическая защита от скрипт-вирусов состоит в том, что в браузере локального компьютера вводится запрет на получение активных элементов.

Для борьбы с компьютерными вирусами используют антивирусные программы (полифаги, ревизоры, блокировщики), которые основаны на различных принципах поиска и лечения зараженных файлов. Самыми популярными и эффективными являются *полифаги* (например, созданный в России лабораторией Е. Касперского Kaspersky Anti-Virus или американская программа Dr.Web). Принцип их работы основан на проверке файлов, загрузочных секторов дисков и оперативной памяти и поиске в них известных и новых (неизвестных полифагу) вирусов.

Для поиска известных вирусов используются так называемые *маски*. Маска вируса — это некоторая постоянная последовательность программного кода, специфичная именно для данного вируса. Если антивирусная программа обнаруживает такую последовательность в каком-либо файле, то этот файл подлежит лечению или удалению. Для поиска новых вирусов используются алгоритмы эвристического сканирования, т. е. анализ последовательности команд в проверяемом объекте. Если выявлена «подозрительная» последовательность команд, то полифаг выдает сообщение о возможном заражении объекта.

Полифаги могут обеспечивать проверку файлов в процессе их загрузки в оперативную память. Такие программы называются *антивирусными мониторами*.

Достоинством полифагов является их универсальность. К недостаткам следует отнести большие размеры используемых ими антивирусных БД, которые должны содержать информацию о максимально возможном числе вирусов. Это приводит к относительно небольшой скорости поиска вирусов.

Принцип работы *ревизоров* (например, ADInf) основан на подсчете контрольных сумм для присутствующих на диске файлов. Эти контрольные суммы затем сохраняются в БД антивируса, как и некоторая другая информация (длина файлов, даты их последней модификации и т. д.). При очередном запуске файла ревизоры сверяют суммы, содержащиеся в БД, с реально подсчитанными значениями. Если информация о файле, записанная в БД, не совпадает с этими реальными значениями, то ревизор сигнализирует о том, что файл изменен или заражен вирусом. Недостаток ревизоров в том, что они не способны обнаружить вирус в новых файлах, поскольку в их БД отсутствует информация об этих файлах.

Антивирусные *блокировщики* — это программы, выявляющие подозрительные ситуации и сообщающие об этом пользователю. К таким ситуациям относится, например, запись в загрузочный сектор диска, которая происходит при установке на компьютер

новой ОС или при заражении загрузочным вирусом. Наибольшее распространение получили антивирусные блокировщики в BIOS компьютера. С помощью программы BIOS Setup выполняют настройку BIOS таким образом, чтобы была запрещена любая запись в загрузочный сектор диска, т.е. чтобы компьютер был защищен от заражения загрузочными вирусами. Достоинство блокировщиков заключается в их способности обнаруживать и останавливать вирус на самой ранней стадии его размножения.

Контрольные вопросы

1. Какие существуют виды программного обеспечения?
2. Что можно делать с помощью текстового редактора?
3. Зачем нужны электронные таблицы?
4. Как можно наглядно представить данные из электронной таблицы?
5. Что такое база данных?
6. Для чего применяются системы автоматизированного проектирования?
7. Что собой представляют антивирусные программы и почему они нужны?
8. Что необходимо сделать перед загрузкой файла с гибкого диска, полученного от коллеги?
9. Зачем нужны драйверы?

ОСНОВЫ ПРОГРАММИРОВАНИЯ**17.1. Принципы программирования**

Задачей программирования в широком смысле слова является составление самых разнообразных программ для ЭВМ, с помощью которых выполняются вычисления, обрабатывается информация, вырабатываются сигналы, управляющие ходом технологических и производственных процессов, и осуществляются иные действия, которые облегчают умственную работу человека. Неслучайно в первые годы после появления ЭВМ журналисты называли их «думающими машинами», «электронным мозгом».

Действительно, работа ЭВМ во многом напоминает работу человеческого мозга. С помощью ЭВМ решаются сложнейшие интеллектуальные задачи. Благодаря огромному объему памяти и высокому быстродействию (по этим показателям вычислительная техника уже превзошла возможности не только обычных, но и самых одаренных людей) ЭВМ выполняет некоторые из таких задач более успешно, чем человек, но происходит это в соответствии с программой, которую составил человек. И хотя ЭВМ уже выигрывает в шахматы у чемпиона мира, нельзя сказать, что машина «умнее» человека. Есть целые области интеллектуальной деятельности человека, где основными факторами успеха являются талант, творческие способности, вдохновение. Пользуясь методом перебора различных слов (или нот), ЭВМ в конце концов может написать стихотворение «Я помню чудное мгновение» (или, например, мелодию «Yesterday»), но выбрать единственный из тьмы вариантов она не сможет. Причин тому много, но, наверное, главной является то, что машина не может любить, у нее нет души... По мнению Билла Гейтса, электронные орудия труда умножают те способности человека, которые делают его уникальным: способность мыслить, высказывать свои мысли и работать в коллективе над воплощением их в жизнь.

Различают три основных вида программирования для вычислительной техники: ручное, автоматическое, системное.

Ручное программирование для ЭВМ заключается в составлении человеком программ на машинном языке. *Машинный язык* — это язык команд конкретной машины, на которой будет решаться данная задача. Каждая команда задает машине информацию об одной операции (например, сложении, умножении и т. д.), адреса исходных чисел и результата операции. Адресами являются но-

мера ячеек памяти, в которых хранятся эти числа. Последовательность команд называется программой. Команды выполняются в том порядке, в каком они написаны в программе, за исключением случаев, когда имеются так называемые команды перехода, указывающие номер команды в программе, к которой нужно перейти после проверки какого-либо условия.

Основными приемами программирования являются построение циклов, подпрограмм и модификация программ. Модификация команд — это изменение адресов в команде, обеспечивающее применение данной команды для операции над величинами, находящимися в других ячейках памяти. Для решения часто встречающихся типовых задач или их отдельных частей составляют *стандартные подпрограммы*. Из них формируют *библиотеку стандартных подпрограмм*, которую используют при составлении программ для решения новых задач. Ответственным этапом программирования является так называемая отладка программ, заключающаяся в пробном решении на машине задач, результаты решения которых уже известны.

Перед написанием программы на языке машины составляют алгоритм задачи, определяющий общий ход вычислительного процесса, а также распределение памяти для данных (исходных, промежуточных и окончательных) в запоминающих устройствах машины.

Слово «алгоритм» произошло от имени выдающегося среднеазиатского ученого IX в. аль-Хорезми, который сформулировал правила выполнения арифметических действий. Первоначально ЭВМ предназначались для вычислений, т. е. для выполнения математических операций при решении задач. Понятие алгоритм относилось именно к последовательности выполнения математических операций. В настоящее время вычислительная техника применяется для обработки (преобразования) самой разнообразной информации, заданной не только в виде числовых величин, но и в виде графических образов, звуковых сигналов, а в более широком смысле — в виде любых сигналов, которые могут восприниматься как человеком, так и любым чувствительным прибором. Поэтому под алгоритмом теперь понимается последовательность действий, обеспечивающая преобразование объекта из исходного состояния в требуемое конечное. Преобразование заключается в выполнении нескольких простых операций, каждая из которых является командой. Алгоритм позволяет формализовать процесс достижения цели. Исполнитель (ЭВМ или человек) может выполнять алгоритм, не вникая в содержание поставленной задачи и даже ничего не зная о поставленной цели, поскольку при строгом следовании алгоритму задача все равно будет решена, а цель достигнута.

Для наглядности алгоритма используют его графическое изображение, которое называется схемой алгоритма (см. подразд. 10.4).

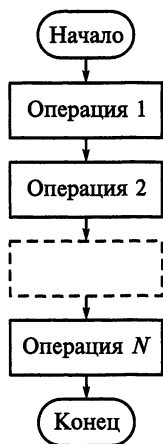


Рис. 17.1. Схема линейного алгоритма

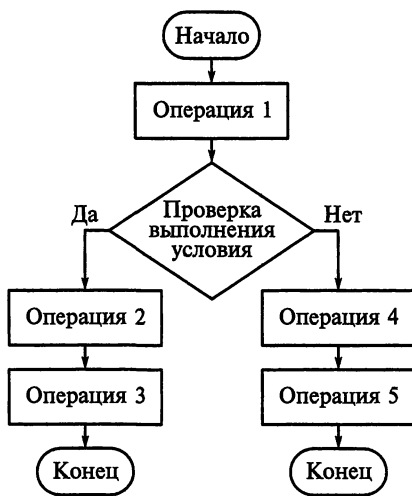


Рис. 17.2. Схема алгоритма типа ветвление

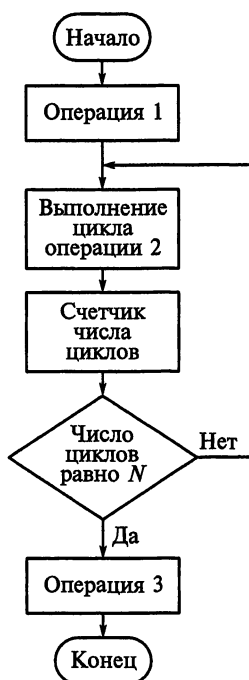


Рис. 17.3. Схема циклического алгоритма

На рис. 17.1 показана схема *линейного* алгоритма, по которому команды выполняются одна за другой. Так решаются простые задачи.

Во многих случаях линейная последовательность выполнения команд невозможна. В зависимости от полученного промежуточного результата выбирается та или иная последовательность действий. Схема такого алгоритма, называемого *ветвлением*, показана на рис. 17.2. Проверка промежуточного результата на соответствие заданному условию осуществляется в элементе, который изображен в виде ромба. Само условие записывается внутри ромба. При выполнении условия путь к следующей команде указывает стрелка с надписью «Да»; если условие не выполняется, то путь к следующей команде указывает стрелка с надписью «Нет». Если требуется выбирать более чем из двух вариантов, на схеме используют большее число ромбов, но из каждого ромба могут выходить только две стрелки (да или нет, третье не дано).

При решении некоторых задач порой требуется выполнение какой-либо команды (или

группы команд) многократно. При этом используется *циклический* алгоритм, схема которого показана на рис. 17.3. Переход к новой операции происходит после достижения заданного числа циклов N . Возможен выход из цикла и после того, как результат будет соответствовать поставленному условию.

17.2. Программирование на машинном языке

Уровень машинных команд наиболее близок процессору ЭВМ. Чтобы их понять, ему не требуются какие-либо посредники. Система машинных команд должна быть достаточной для того, чтобы реализовать требуемые действия, выдавая указания блокам машины. Каждая машинная команда имеет две части: операционную, определяющую, что надо делать, и операндную, определяющую объекты обработки, т. е. то, над чем надо выполнять действия. Возможны следующие размещения операндов: регистр — регистр; регистр — память; память — регистр; непосредственный операнд — регистр; непосредственный операнд — память. Существуют команды работы с цепочками, которые перемещают данные из памяти в память, команды работы со стеком, которые могут переносить данные из памяти в стек, также находящийся в памяти, команды типа умножения, которые кроме операнда, указанного в команде, используют еще второй, неявный операнд.

При программировании на машинном языке все элементы, входящие в команду, записываются в виде двоичных кодов. Так обозначаются и выполняемые операции, и адреса ячеек, в которых содержатся команды, и исходные данные, и полученные результаты.

В качестве примера рассмотрим программирование расчета по формуле $x = (a + b)(c + d)$. При непосредственном машинном программировании необходимо прежде всего составить таблицу размещения чисел в ячейках памяти машины. Пусть число a находится в ячейке с адресом 0100, b — в ячейке с адресом 0101, c и d — в ячейках с адресами соответственно 0102 и 0103. Для размещения числа x отведем ячейку с адресом 0104. Пусть команда сложения

Таблица 17.1

Фрагмент машинной программы

Адрес команды	Код операции	Первый операнд	Второй операнд	Адрес результата
0010	01	0100	0101	0100
0011	01	0102	0103	0102
0012	02	0100	0102	0104

имеет код 01, а команда умножения — код 02. Тогда, используя трехадресные команды, можно написать участок машинной программы, представленный в виде табл. 17.1.

Команды программы, как и числа, находятся в ячейках памяти машины. Первая команда, размещенная в ячейке с адресом 0010, показывает, что нужно взять одно число из ячейки с адресом 0100, другое число — из ячейки с адресом 0101, сложить их (код операции 01) и послать результат в ячейку с адресом 0100. Как видим, адрес результата совпадает в данном случае с адресом первого операнда. Это означает, что после выполнения команды в ячейке с адресом 0100 будет находиться уже не число a , а сумма чисел $a + b$. Посылка какого-либо значения в определенную ячейку приводит к замещению прежнего содержимого ячейки новым значением. Вторая команда (в ячейке с адресом 0011) имеет аналогичный смысл. Третья команда (в ячейке с адресом 0012) показывает, что нужно выполнить умножение (код операции 02) двух промежуточных значений, находящихся в ячейках с адресами 0100 и 0102, и посылку результата в ячейку с адресом 0104.

В данном примере номера ячеек записаны в десятичной системе (использованы цифры 2, 3, 4); в машинном коде они записываются в двоичной системе (с использованием только цифр 0 и 1). Например, номер ячейки 0104 выглядит так: 0000 0001 0000 0100. Запомнить все эти двоичные комбинации человеку-программисту очень сложно. Поэтому используются языки программирования высокого уровня, запись на которых во много раз привычнее и удобнее. Например, на языке Бейсик эта же программа будет иметь вид

$$x = (a + b)(c + d).$$

Перевод данной команды на машинный язык выполняют специальные программы, называемые компиляторами и интерпретаторами.

17.3. Программирование на языке ассемблер

Язык ассемблер занимает промежуточное положение между машинными кодами и языками высокого уровня. Можно сказать, что он является машинно-ориентированным языком программирования (см. подразд. 17.4). Программировать на этом языке проще, чем в машинных кодах, поскольку не надо помнить множество двоичных комбинаций. В то же время программа на этом языке более рационально использует возможности конкретной машины (точнее — процессора), чем программа на языке высокого уровня (который в свою очередь еще более прост для программирования, чем ассемблер). Основные принципы программирования на машинно-ориентированных языках рассмотрим на приме-

ре языка ассемблер для МП семейства *ix86* фирмы Intel, которые широко используются в информационных системах, выполняющих преобразование информации, и в вычислительных комплексах, производящих сложные расчеты.

Ассемблер для МП семейства *ix86* отнюдь не следует рассматривать как единственный или самый главный ассемблер для микропроцессорных систем. В настоящее время получили довольно широкое распространение процессоры десятка других фирм (например, DEC, HP и др.), да и сама фирма Интел выпускает более совершенные МП Пентиум. Разные МП имеют разные системы команд и языки ассемблер, среди них и значительно отличающиеся от языка для *ix86*. Например, для первых МП семейства *ix86* список команд состоял из 78 базовых команд (как и для отечественного МП серии КР580), а для МП *i586* (Пентиум) — уже из 138. Однако для программирования на языках ассемблер используется общая методика. Конкретные же технические приемы записи программ связаны с особенностями архитектуры и системы команд целевого МП.

Основу традиционной и наиболее распространенной технологии составления программ обеспечивают четыре наиболее важных вида программ: редактор, транслятор, компоновщик, отладчик.

Редакторы — это программы для ввода или модификации текста, который нужно откорректировать или поместить во внешнюю память.

Трансляторы преобразуют программы, составленные на входном языке программирования, в режиме интерпретации или компиляции. Транслятор программ с языка ассемблер называется ассемблером, а с языка высокого уровня — компилятором или интерпретатором в зависимости от способа реализации. Ассемблеры и компиляторы формируют так называемые *объектные модули*, что позволяет объединять программные фрагменты, составленные на разных языках.

Компоновщики, или редакторы связей, — это программы, объединяющие выполняемую программу с библиотечными и другими ранее оттранслированными модулями.

Отладчики размещают выполняемую программу в памяти, позволяют выполнять программы в режимах трассировки с отображением результатов, отладку шаг за шагом и отладку в режиме использования адресов отладочных прерываний.

В процессе отладки наиболее сложной и наименее автоматизированной задачей является установление информационной связи между отладчиками и текстовыми редакторами для локализации ошибок.

Такие составные части оборудования, как некоторые внутренние рабочие регистры, кэш-память, буферы, схемы синхронизации, внутренняя и системная шины, а также устройства управле-

ния ими, как и их конструктивное построение, остаются недоступными или незаметными для программиста на языке ассемблер.

Рассмотрим кратко последовательность создания программы на этом языке. Сначала производится ввод исходного текста программы. Для этого можно использовать текстовый редактор «Блокнот», встроенный в операционную систему Windows, или любой другой, не вставляющий в текст программы посторонних символов (символов редактирования). Расширение созданного файла должно быть изменено с .TXT на .ASM. Это возможно с помощью функции «Переименовать файл».

Затем создается объектный модуль, т.е. выполняется трансляция программы. Объектный модуль представляет собой исходную программу уже в машинных кодах. Для получения объектного модуля исходный файл необходимо подвергнуть трансляции с помощью программы `tasm.exe` из пакета TASM. Формат командной строки для запуска `tasm.exe` следующий:

TASM.EXE [/опции] имя исходного файла. В квадратные скобки помещены параметры, которые могут отсутствовать.

Пример: C:\TASM\Work\TASM.EXE /zi C:\TASM\Work\имя_исходного_файла.asm

Наконец, создается загрузочный модуль, т.е. происходит компоновка программы. Формат командной строки для запуска компоновщика следующий:

TLINK.EXE [/опции] имя объектного файла

Пример: C:\TASM\Work\TLINK.EXE /v

C:\TASM\Work\имя_исходного_файла.obj

В результате получается исполняемый модуль с расширением .EXE. Созданная программа может быть выполнена на компьютере, но в ней возможны ошибки, для устранения которых используется отладчик.

Отладчик Turbo Debugger (TD) представляет собой оконную среду отладчика программ на уровне исходного текста на языке ассемблер.

Он позволяет решить две главные задачи — определить место логической ошибки и найти причину этой ошибки. Отладчик TD может также применяться для исследования созданных программ. После запуска TD в меню FILE надо выбрать пункт OPEN и открыть исследуемый файл. В меню VIEW нужно выбрать пункт CPU. Исследовать программу рекомендуется в пошаговом режиме, используя клавишу F7. Окно CPU отражает состояние МП и состоит из пяти подокон:

- 1) окна с исходной программой в дизассемблированном виде;
- 2) окна регистров МП, отражающего текущее содержимое регистров;
- 3) окна флагов, которое отражает текущее состояние флагов МП в соответствии с их мнемоническими названиями;

4) окна стека Stack, отражающего содержимое памяти, выделенной для стека. Адрес области стека определяется содержимым регистров SS и SP;

5) окна памяти Dump, отражающего содержимое области памяти по адресу, который формируется из компонентов, указанных в левой части окна.

Каждое из пяти подокон независимо от других и в зависимости от настроек TD может как присутствовать, так и отсутствовать в окне CPU.

Рассмотрим теперь структуру программы на языке ассемблер. Создаваемые программы обычно содержат в себе некоторые стандартные блоки, которые одинаковы для всех программ, и разные для различных задач блоки, которые отражают конкретную цель решаемой задачи.

Пример стандартного оформления программы на языке ассемблер:

```
masm           ; режим работы
model small    ; модель памяти
.data          ; сегмент данных (при наличии исходных данных
               ; они описываются в этом сегменте)
stack 256      ; сегмент стека размером в 256 байт
.code         ; сегмент кода
start:        ; точка входа в программу
mov ax,@data  ; заносим адрес сегмента данных в регистр AX
mov ds,ax     ; перенос из регистра AX в регистр DS
               ; далее текст программы в соответствии с конкрет-
               ; ной задачей
mov ax,4c00h  ; пересылка 4c00h в регистр AX
int 21h       ; вызов прерывания с номером 21h
end start     ; конец программы с точкой входа start.
```

В языке ассемблер можно вводить данные как в десятичной, так и в шестнадцатеричной системе. В последнем случае после шестнадцатеричного числа ставится буква h.

17.4. Программирование на языках высокого уровня

По мере внедрения ЭВМ в практику и привлечения к программированию большого числа людей стало очевидно, что нельзя использовать при программировании только машинные коды. Появилась необходимость в языках программирования, которые были бы удобны для пользователей, обладающих различной квалификацией. С начала 1960-х годов языки программирования интенсивно развиваются. За это время были созданы сотни языков высокого уровня. Некоторые из них стали специализированными для конкретных областей применения.

Несмотря на большое разнообразие языков программирования только около десятка из них можно считать общепризнанными. Преимущественное использование какого-либо языка программирования определяется: ясностью, простотой и согласованностью понятий; стройностью и наглядностью структуры программы, написанной на данном языке; естественностью приложения; легкостью расширения; обеспеченностью пакетами прикладных программ; эффективностью реализации программ, написанных на данном языке, и др.

Первым из алгоритмических языков высокого уровня, получивших международное признание, был Алгол, рекомендованный для широкого применения в 1960 г. международной конференцией по вычислительной технике в Париже. При разработке его были использованы новые идеи, обобщающие предложения многочисленных специалистов из разных стран. Наиболее плодотворными из них были следующие:

- использование блочной структуры и области действия обозначений, позволяющее разделить работу по составлению больших программ на более мелкие части;
- возможность динамического распределения памяти;
- развитый аппарат вызова процедур.

Алгол был запроектирован не только как эффективный язык программирования, но и как средство записи алгоритмов. Он получил широкое распространение, на нем было написано большое число программ, из которых составлены целые библиотеки.

Язык Фортран является самым старым из применяемых в настоящее время языков программирования. Он приспособлен для сложных вычислений, располагает удобными средствами для ввода и вывода информации и удовлетворяет перечисленным выше требованиям.

В связи с появлением большого числа языков программирования возникла необходимость в их классификации. Прежде всего их разделили на машинно-ориентированные и процедурно-ориентированные языки.

В *машинно-ориентированных* языках используются такие типы данных, которые соответствуют вычислительным машинам определенного класса. Создание машинно-ориентированного языка преследует цель позволить пользователям составлять эффективные программы, учитывающие структуру вычислительных машин вообще или особенности какой-либо конкретной вычислительной машины.

Процедурно-ориентированные языки предназначены для решения определенных классов задач. Разделение на классы носит условный характер. Под классом задач понимают задачи, в которых рассматриваются аналогичные объекты и применяются сходные приемы решения. Всякий алгоритм решения задачи можно запи-

сать в виде программы для вычислительной машины, закодировав соответствующим образом рассматриваемые объекты.

С помощью процедурно-ориентированного языка специалисты в данной области могут описывать алгоритм решения задачи в привычных терминах, не вникая в особенности вычислительной машины и не прибегая к помощи программистов. Процедурно-ориентированные языки сформировались для задач: вычислительных, автоматической обработки данных; информационно-логических; обработки текстов; моделирования; управления.

В вычислительных задачах основными объектами являются числа и массивы чисел. Алгоритм решения может задаваться весьма сложными формулами с использованием рекурсивных определений (каждая последующая операция выбирается на основании предыдущего результата), индексных выражений, подстановок функций, сложных условий и т.д. Решение задач обычно связано с выполнением огромного числа арифметических операций. В области вычислительных задач методы решений были первыми формализованы и приспособлены для использования вычислительных машин, да и сами эти машины предназначались вначале преимущественно для решения задач этого класса. Поэтому первые процедурно-ориентированные языки появились именно в данной области.

В задачах автоматической обработки данных основным объектом являются массивы данных (например, подшивки документов), состоящие из логических записей (отдельных документов). Для таких массивов характерна иерархическая структура записей, на нижнем уровне которой расположены цифровые, алфавитные и алфавитно-цифровые элементы. Особое значение приобретает возможность ввода и вывода по сложным форматам с заданием особых операций (устранение нулей, защита чеков и т.д.), запоминание, хранение и выборка данных при работе с внешней памятью (ленты, диски). Наиболее распространенным процедурно-ориентированным языком для описания решения задач этого класса является Кобол.

В информационно-логических задачах основными объектами являются сложные структуры, элементы которых связываются при помощи ссылок (списки, деревья). Такие ссылки обеспечивают оптимальное обращение к элементам, имеющим заданные значения признаков.

Наиболее распространенным процедурно-ориентированным языком для этих задач является Лисп. Он дал толчок к разработке специализированных языков искусственного интеллекта и языков представлений знаний.

В задачах обработки текстов основными объектами являются строки символов.

При решении этих задач выполняются операции определения вхождения данной цепочки символов, замены, выбрасывания, вставки цепочки и т. д. Примером процедурно-ориентированного языка может служить Снобол.

В задачах моделирования основными объектами служат процессы, параллельно протекающие во времени и взаимодействующие друг с другом.

В задачах управления основными объектами являются сигналы прерывания от внешней среды и временного датчика, обратные сигналы и приоритеты внешних сигналов.

Особую группу составляют языки, предназначенные для описания специальных проблем и называемые *проблемно-ориентированными*.

Программа работы на таком языке помимо описания условия задачи содержит указание решить задачу того или иного класса. Примером такого языка является Stress, предназначенный для описания задач конструирования. Программа на этом языке содержит ряд общих характеристик системы (размерности, число вершин и др.) и данные, а также указание решить задачу и представить определенные данные в виде некоторой таблицы.

Для использования таких языков разрабатывают или универсальный для данного класса задач алгоритм, интерпретирующий исходные данные, или алгоритм анализа исходных данных и определения частной задачи, для которой генерируется соответствующая разрешающая процедура. Проблема создания единого универсального языка программирования фактически находится в стадии развития.

Предпринимаются попытки создать универсальный язык, удобный и эффективный для решения всех проблем. Примером может служить язык программирования С (Си) и его последующие варианты С+, С++, С#. Язык Си создавался как инструментальное средство для реализации операционной системы UNIX (см. подразд. 15.4) на ЭВМ фирмы DEC, но рамки его применения значительно расширились. Сейчас это один из универсальных языков программирования. В язык Си включены средства программирования почти на уровне языка ассемблер. Он сравнительно прост для изучения пользователем ЭВМ и одновременно дает возможность создавать достаточно экономные программы. В языке Си широко развит аппарат описаний, позволяющий посредством соответствующего контекста определять новые операторы, вводить новые символы или предписывать введенным символам новые операционные значения.

Большое влияние на развитие языков программирования оказала операционная система-оболочка Windows (см. подразд. 16.3), которая предоставила пользователю удобный графический интерфейс и возможность интерактивного общения.

Стремление сочетать универсализм и удобства графического интерфейса привело к созданию *объектно-ориентированных* средств программирования, наиболее популярными из которых в настоящее время являются Visual Basic и Visual Basic for Applications (VBA). По существу, это уже не просто языки, а системы программирования. Кратко рассмотрим основы программирования с использованием этих средств.

Приложения на языках объектно-ориентированного программирования Visual Basic и VBA строятся из объектов, подобно тому, как из блоков и различных деталей строятся дома. Эти системы программирования содержат программные библиотеки готовых объектов.

Системы объектно-ориентированного программирования позволяют сделать наглядным и понятным процесс создания графического интерфейса разрабатываемого приложения, поскольку при построении объектов с заданными свойствами используются диалоговые окна системы программирования. Взаимодействие программных объектов между собой и их изменения описываются с помощью программного кода. Создание программного кода в объектно-ориентированном программировании базируется на использовании алгоритмических структур различных типов (линейной, ветвления, цикла), которые реализуются программными объектами.

Основной единицей в объектно-ориентированном программировании является программный *объект*, который объединяет в себе как описывающие его данные (*свойства*), так и средства обработки этих данных (*методы*). Если использовать термины из грамматики, то объекты — это «существительные», свойства объекта — «прилагательные», а методы объекта — «глаголы». В связи с этим часто используют словосочетание «грамматика языка программирования». Программные объекты обладают свойствами, могут использовать методы и реагируют на события.

Классы объектов являются «шаблонами», определяющими наборы свойств, методов и событий. По этим шаблонам создаются объекты. В языке Visual Basic основными являются классы объектов, реализующие графический интерфейс приложения. В языке VBA используются еще более ста различных классов объектов, которые существуют в среде Windows. Можно подключать и дополнительные библиотеки программных объектов, а также создавать новые классы объектов самим программистом.

Каждый из классов обладает специфическим набором свойств, методов и событий. Например, в приложении Word существует класс объектов Document (Документ), который обладает определенным набором свойств: Name (Имя), Full Name (Полное имя) и т. д., методов Open (Открыть документ), Print Out (Напечатать документ), Save (Сохранить документ) и т. д., событий Document

New (Открытие документа), Document Close (Закрытие документа) и т.д.

Объект, созданный по «шаблону» класса объектов, является экземпляром класса и наследует весь набор свойств, методов и событий данного класса. Каждый экземпляр класса имеет уникальное для данного класса имя, которое указывается в скобках после названия класса. Различные экземпляры класса обладают одинаковым набором свойств, однако значения свойств у них могут отличаться. Так, в приложении Word могут быть открыты несколько документов, являющихся экземплярами класса Document, которые имеют различные имена и хранятся в различных каталогах.

Семейство объектов также представляет собой объект, содержащий несколько объектов, экземпляров одного класса. Первоначальные значения свойств объекта можно установить с использованием диалогового окна системы программирования. Эти значения свойств объектов можно изменять в программном коде. Для присвоения свойству объекта нового значения в левой части строки программного кода необходимо указать имя объекта и затем название свойства.

В правой части строки после знака равенства необходимо записать конкретное значение свойства:

Объект. Свойство = Значение Свойства.

Для того чтобы объект выполнил какую-либо операцию, необходимо применить метод, которым он обладает. Многие методы имеют аргументы, которые позволяют задать параметры выполняемых действий.

Для присваивания аргументам конкретных значений используются двоеточие и знак равенства; друг от друга аргументы отделяются запятой. Обратиться к методу объекта можно так же, как и к свойству объекта.

Событие представляет собой действие, распознаваемое объектом. Событие может создаваться пользователем (например, щелчок кнопкой мыши или нажатие клавиши) или быть результатом воздействия других программных объектов. Реакцией на события является определенная процедура, которая может изменять значения свойств объекта, вызывать его методы и т.д. Например, объект Document (Документ) реагирует на события Open (Открытие), New (Создание) и Close (Закрытие), а объект Selection (Выделенный фрагмент документа) реагирует на события Cut (Вырезать), Copy (Копировать), Paste (Вставить), Delete (Удалить) и т.д.

При решении задачи обработки информации пользователь взаимодействует с аппаратными и программными средствами ЭВМ (рис. 17.4).

В распоряжении пользователя есть сама ЭВМ, т.е. аппаратные средства, а также набор операционных систем и пакет приклад-

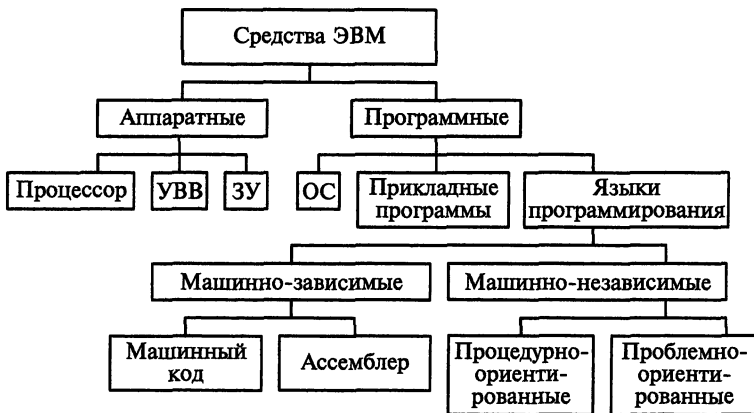


Рис. 17.4. Аппаратные и программные средства ЭВМ

ных программ. На универсальном языке пользователь ставит задачу, по которой определяется проблемно-ориентированный язык, дающий указание о том, какие процедуры должны быть выполнены при решении задачи.

Для них выбираются процедурно-ориентированные языки, на которых составляются программы. Программы затем интерпретируются на машинный язык, который дает понятные машине указания. Решение задачи происходит в пошаговом режиме, при этом пользователь выполняет отладку программы. В итоге программа доводится до рабочего состояния. Шутники говорят, что к этому времени она уже не нужна.

17.5. Автоматизация программирования

Автоматизация программирования — раздел программирования, разрабатывающий методы автоматического составления программ и решения задач на ЭВМ по данным, представленным в некотором формализованном виде — на формальном языке. Автоматизация программирования базируется на применении средств вычислительной техники, предназначенных для облегчения общения пользователя с ЭВМ.

Основу построения систем автоматизации программирования составляют алгоритмические языки, ориентированные на практические приложения, и базирующиеся на них языки программирования.

Целью создания систем автоматизации программирования является повышение эффективности использования ЭВМ в различных сферах путем разработки соответствующего математического обеспечения ЭВМ.

Работа пользователя ЭВМ при решении задач расчленяется на ряд этапов:

- изучение задачи (процесса переработки информации);
- выработку алгоритма решения задачи (алгоритма, моделирующего данный процесс);
- составление программы, реализующей этот алгоритм на избранной машине;
- обработку алгоритма и его отладку;
- подготовку данных;
- решение задачи;
- оформление результатов.

С расширением сфер применения ЭВМ наибольшую остроту приобрела проблема автоматизации этапов составления программ и их отладки как наиболее трудоемких. Описание сколько-нибудь сложного алгоритма в мелких машинных операциях связано с огромными техническими трудностями, поскольку при ручном программировании необходимо четкое представление о размещении в памяти ЭВМ (имеющей обычно многоступенчатую структуру) всех потоков информации на всех этапах работы программы, а также о связях и соотношениях между отдельными командами программы.

Дальнейшее развитие идей автоматизации программирования привело к созданию ОС, в которых помимо процесса программирования (или перевода алгоритмов на язык интерпретации) автоматизированы процессы решения задач на ЭВМ на всех этапах — от анализа задачи до синтеза программы и получения результатов в виде, пригодном для хранения, документирования или размножения. Вместе с тем благодаря использованию специальных системных программ ОС полностью автоматизирует работу обслуживающего персонала ЭВМ. Современные ОС представляют собой организованную совокупность алгоритмов, стандартных и обслуживающих программ, информационно-справочных систем и архивов данных, а также систем трансляции и интерпретации, обеспечивающих пакетную обработку программ и многопрограммную работу в режиме разделения времени и в реальном масштабе времени.

Труд специалиста по программированию при наличии таких систем приобретает более творческий характер, поскольку связан с разработкой новых и более совершенных методов решения задач на ЭВМ, а также мощных систем математического обеспечения ЭВМ и их комплексов. Развитие САПР оказывает существенное влияние на проектирование алгоритмических структур вычислительных систем, указывая пути их дальнейшего совершенствования прежде всего за счет повышения уровня непосредственной интерпретации этих систем.

Основной тенденцией развития автоматизации проектирования является стремление к созданию средств, обеспечивающих

реализацию задач (их программирование, отладку, решение и накопление решений для последующего использования, а также хранение программ и данных) при минимальных затратах труда программиста.

В свете этих работ особую остроту приобретает проблема унификации и стандартизации средств математического и технического обеспечения ЭВМ.

Разработка программ, сопровождающаяся изготовлением программной документации, выполняется в таком порядке.

1. Постановка задачи. Определение цели, критериев и направления обработки исходных данных, ограничений на ошибки преобразований и характеристики объектов и связей в моделях сферы исследований, типов выходных данных и результатов, а также аналитических отношений между ними. Этап заканчивается подготовкой технического задания на разработку программы.

2. Подбор языковых средств для программирования. Разработка форматов ввода и отображения исходных данных и результатов.

3. Разработка спецификаций программ, т. е. определение необходимого набора программ и взаимосвязи между входными данными и результатами. Этот этап сопровождается выдачей проектов эксплуатационных документов на программы.

4. Выбор метода решения задачи. Анализ возможности использования ранее разработанного программного обеспечения. Подбор метода обработки аналитических соотношений.

5. Разработка алгоритма решения задачи. Разбивка задач на подзадачи и фрагменты. Определение последовательности решения подзадач. Разработка структуры программы.

6. Кодирование программы средствами избранного языка программирования. Этот этап совместно с этапом 5 можно рассматривать как процесс эквивалентных преобразований математических и логических формальных записей в принятые для данного языка последовательности директив.

7. Проверка корректности программы путем аналитического доказательства и обоснования преобразований, а также определение ограничений на входные данные и результаты.

8. Тестирование программы. Разработка тестов и контрольных примеров. Сопоставление реальных и ожидаемых результатов.

9. Отладка программы в случае обнаружения ошибки. Локализация ошибок, обнаруженных в программе. Коррекция ошибок.

10. Оптимизация программы. Выбор оптимальных алгоритмов в соответствии со структурой и составом оборудования вычислительного комплекса.

11. Разработка текстовых описаний программы и заключительных вариантов эксплуатационных документов.

12. Исследовательская эксплуатация программ. Уточнение требований к представлению исходных данных и результатов про-

граммы. Анализ особенностей выполнения программы при наличии сбоев в ВК.

13. Промышленная эксплуатация программ и их коммерческое распространение. Сопровождение программ (т.е. контроль за их применением с целью выявления недостатков) и обработка требований к новым версиям программ.

14. Обобщение и усовершенствование программ. Определение сфер применения программ, в том числе на перспективу.

Рассмотренные этапы разработки программ осуществляются в соответствии со стандартами ЕСПД (единой системы программной документации). Этапы 1 и 2 выполняются на стадии разработки технического задания, 3 и 4 — при эскизном проектировании, 5 — при техническом проектировании, 6—11 — на стадии рабочего проектирования, 12—14 — в процессе внедрения программной продукции.

На каждом этапе используется системное программное обеспечение технологической поддержки, что повышает уровень автоматизации программирования.

Несколько подробнее рассмотрим процесс отладки программы (этап 9). Для того чтобы процессор мог выполнять программу, эта программа и данные, с которыми она работает, должны быть загружены в оперативную память. Программа написана на языке программирования (некоторый текст), а процессор «понимает» команды только на машинном языке. Поэтому необходимо, чтобы в оперативной памяти находилась программа-переводчик (транслятор), автоматически переводящая программу с языка программирования на машинный язык. Компьютер может выполнить программы, написанные только на том языке программирования, транслятор которого находится в оперативной памяти компьютера.

Трансляторы языков программирования, как уже отмечалось, бывают двух типов — интерпретаторы и компиляторы. Интерпретатор — это программа, которая последовательно переводит инструкции программы на машинный язык и тут же их выполняет. Эта процедура повторяется при каждом запуске программы на выполнение.

Достоинством интерпретаторов является удобство отладки программы (т.е. поиска и устранения в ней ошибок), так как возможно пошаговое ее выполнение, а недостатком — сравнительно малая скорость выполнения.

Компилятор действует иначе. После перевода всего текста программы на машинный язык он сохраняет его в исполняемом файле (обычно с расширением .exe).

Этот файл, уже записанный на машинном языке, можно запускать на исполнение многократно. Достоинством компиляторов является большая скорость выполнения программы, а недостат-

ком — трудоемкость отладки, поскольку пошаговое выполнение программы невозможно.

Современные системы программирования способны работать в режиме как интерпретатора, так и компилятора.

При разработке и отладке программы используется режим интерпретатора, а для получения готовой исполняемой программы — режим компилятора.

Контрольные вопросы

1. Что такое программирование?
2. Какие существуют виды языков программирования?
3. Что такое машинный язык?
4. Приведите два-три примера команд на языке ассемблер.
5. Каковы основные достоинства языков высокого уровня?
6. В чем заключается автоматизация программирования?
7. Какая разница между компьютером и интерпретатором?

ОСНОВЫ ПОСТРОЕНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ**18.1. Предпосылки к созданию сетей**

Можно выделить следующие этапы появления и развития сетей передачи информации.

В 1832 г. Павел Львович Шиллинг (1786—1837) создал первый практически пригодный электромагнитный телеграф. В 1837 г. американец Сэмюэл Морзе (1791—1872) изобрел электромеханический телеграфный аппарат. Уже в XIX в. существовали телеграфные сети, объединяющие несколько прямо-передающих станций электрическими проводами.

В 1876 г. американец Александр Белл (1847—1922) получил патент на первый практически пригодный телефон. В настоящее время телефонная сеть позволяет передавать голосовые сообщения между миллиардами телефонных аппаратов, которые соединяются через многие тысячи телефонных станций воздушными и кабельными (в том числе волоконно-оптическими) линиями связи, радиорелейными линиями, спутниковыми каналами связи.

В 1897 г. итальянец Гульельмо Маркони (1874—1937) получил патент на изобретение радиоприемника, принципиально тождественного изобретенному Александром Степановичем Поповым (1859—1906) в 1895 г. Сеть радиосвязи обеспечивает обмен информацией на практически неограниченных расстояниях и не требует при этом соединительных электрических проводов.

Указанные сети могут быть использованы и для создания компьютерных сетей.

Электронные вычислительные машины создавались для решения вычислительных задач, что и отражено в их названии. Многие формы человеческой деятельности могут быть сведены к решению задач. Если задача не очень сложная, то она может быть решена одним человеком. При большой сложности задачи к ее решению может быть привлечен коллектив людей. Задача решается в два этапа. На первом этапе ищется метод решения задачи и в соответствии с ним составляется последовательность правил ее решения, на втором по этой последовательности и по исходным данным производятся вычисления. Задачи, решение которых достигается путем вычислений по заранее найденным методам решений, называются вычислительными. В принципе любая задача может быть сведена к вычислительной.

Метод итераций состоит в получении последовательности решений системы

$$\left(x_1^{(0)}, \dots, x_n^{(0)}\right), \left(x_1^{(1)}, \dots, x_n^{(1)}\right), \left(x_1^{(k)}, \dots, x_n^{(k)}\right) \dots \quad (18.4)$$

Каждое последующее решение будет более точным. Если задаться требуемой точностью, то вычисления можно прекратить, когда она будет достигнута. Для этого после каждой итерации определяется разность

$$\left|x_i^{(k)} - x_i^{(k-1)}\right| < \varepsilon,$$

где ε — положительное число, определяющее точность вычислений.

Не всякая система уравнений решается методом итераций. Перед его использованием надо проверить выполнение одного из следующих условий:

$$1) \sum_{i=1}^n |\alpha_{ij}| < 1; \quad j = 1, \dots, n;$$

$$2) \sum_{j=1}^n |\alpha_{ij}| < 1; \quad i = 1, \dots, n;$$

$$3) \sum_{i=1}^n \sum_{j=1}^n |\alpha_{ij}|^2 < 1.$$

Если коэффициенты системы удовлетворяют одному из них (любому), то последовательность (18.4), построенная на основании формулы (18.3), сходится к решению системы (18.1).

При вычислении каждого приближенного значения выполняются n^2 операций умножения, n^2 операций сложения, n операций вычитания и n операций сравнения, т. е. всего $2n^2 + 2n$ операций. Для полного решения задачи необходимо в общем случае выполнить примерно n приближений. При решении задачи необходимо хранить в памяти $n^2 + 3n$ данных.

Такая задача вполне может быть решена на современной ЭВМ, но для ЭВМ первых поколений время вычислений порой оказывалось недопустимо большим, а иногда и памяти не хватало. Поэтому был разработан метод одновременного использования нескольких ЭВМ для решения общей вычислительной задачи. Этот метод, созданный академиком Э.В. Евреиновым, получил название *модель коллектива вычислителей*. По существу он стал основой не только вычислительных, но и всех других компьютерных сетей, включая Интернет.

Схема решения системы линейных уравнений коллективом из n вычислителей показана на рис. 18.1. Система распределяется между

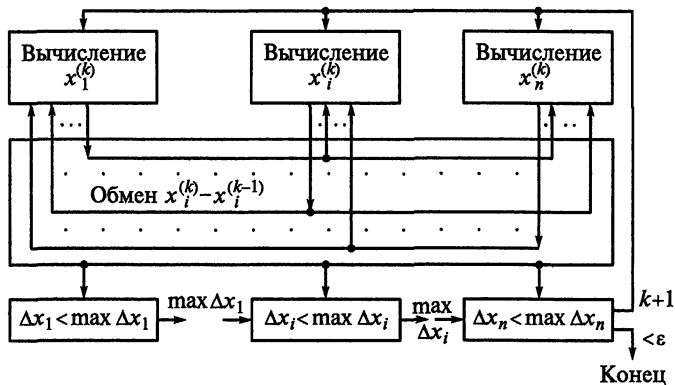


Рис. 18.1. Схема решения задачи коллективом вычислителей

n вычислителями таким образом, что на долю каждого из них приходится одно уравнение. Начальные приближения x_i^0 задаются всем вычислителям. В этом случае каждый может производить вычисления независимо от получения первого приближения ($x_1^{(1)}, \dots, x_n^{(1)}$). Это осуществляется за $2n$ шагов умножения и сложения при одновременной работе всех вычислителей.

Для продолжения вычислений необходимо выполнить обмен информацией между вычислителями. Поскольку каждый вычислитель имеет результат (приближение) только для своей переменной $x_i^{(k)}$, а для работы ему надо иметь все переменные ($x_1^{(k)}, \dots, x_n^{(k)}$), требуется выполнить цикл обмена переменными между вычислителями. Первый вычислитель посылает свою переменную $x_1^{(k)}$ остальным. Затем второй вычислитель пересылает другим вычислителям свой результат $x_2^{(k)}$ и т. д. За n обменов все вычислители будут иметь новые приближения, которые используются для вычислений на следующем шаге итераций. Обмен сообщениями, при котором один вычислитель передает сообщение, а все другие его принимают, называется *трансляционным*. Достоинство трансляционного обмена в том, что с ростом числа вычислителей растет и эффективность обмена. За один шаг фактически выполняются n обменов, а если бы вычислители обменивались попарно, то потребовались бы n шагов.

После выполнения трансляционных обменов вычислители продолжают независимые вычисления новых приближений. Это время можно использовать для проверки условий сходимости. Полученную разность между новым и предыдущим приближениями первый вычислитель отправляет второму, который сравнивает эту величину с полученной им и большую из них пересылает треть-

ему и т. д. Последний вычислитель сравнивает максимальную разность с заданным значением требуемой точности. Если максимальная разность не превышает этого значения, то следующий шаг приближений не нужен и вычисления заканчиваются.

Обмен между вычислителями, построенными в цепь, называется *конвейерным*. При организации такого обмена возможность передавать и принимать сообщение имеется у каждого вычислителя. В коллективе из n вычислителей за один шаг производится n обменов (как и при трансляционном обмене). Эти два способа обмена наиболее часто используются при решении сложных задач коллективом вычислителей.

В 1970-х годах возникла необходимость объединить разнотипные ЭВМ в вычислительные сети. На первом этапе были разработаны сетевые архитектуры, которые обеспечивали связь только между оборудованием одной и той же фирмы-изготовителя. Примерами таких сетевых архитектур являются SNA (системная сетевая архитектура) фирмы IBM и DNA (цифровая сетевая архитектура) фирмы DEC. Однако при указанном подходе замена существующего сетевого оборудования оборудованием других производителей вызвала серьезные денежные затраты. Появилась острая потребность в разработке некоторой концепции, позволяющей установить универсальные правила взаимодействия разнотипных ЭВМ между собой. В рамках этой концепции должны быть предусмотрены различные уровни взаимодействия ЭВМ — от самых простейших до самых сложных (высоких). При этом каждая из взаимодействующих ЭВМ должна рассматриваться как открытая система в том смысле, что реализуемое в ней программное обеспечение удовлетворяет некоторому набору универсальных соглашений, точное выполнение которых гарантирует возможность взаимодействия различных ЭВМ (открытых систем). Международная организация по стандартизации — МСС (ISO) приняла и рекомендовала так называемую эталонную модель взаимодействия открытых систем — ЭМ ВОС (OSI — Open System Interconnection), которая определяет стандарты соединения и взаимодействия элементов информационно-вычислительной сети.

При объединении компьютеров в сеть между ними происходит обмен информацией. Любой обмен информацией производится по так называемым каналам передачи информации, которые могут быть основаны на различных физических принципах. Так, при непосредственном общении людей информация передается с помощью звуковых волн, а при разговоре по телефону — с помощью электрических сигналов, которые распространяются по каналам связи. Компьютеры (как и телефоны) для обмена информацией используют каналы связи различной физической природы — кабельные, оптоволоконные, радиоканалы (в том числе через спутники на околоземной орбите).

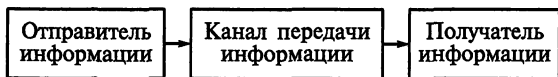


Рис. 18.2. Обобщенная схема передачи информации

Обобщенная схема передачи информации (рис. 18.2) включает в себя отправителя информации, канал передачи информации и получателя информации. При двустороннем обмене информацией отправитель и получатель могут меняться ролями.

Основной характеристикой каналов передачи информации является их пропускная способность, т. е. скорость передачи информации. Пропускная способность канала равна количеству информации, которое может передаваться по нему в единицу времени. Обычно пропускная способность измеряется в битах в секунду (бит/с) и кратных единицах (Кбит/с, Мбит/с, Гбит/с). Однако иногда в качестве единицы измерения используется байт в секунду (байт/с) и кратные ему единицы (Кбайт/с, Мбайт/с).

Соотношения между единицами пропускной способности канала передачи информации такие же, как между единицами измерения количества информации: $1 \text{ байт/с} = 2^3 \text{ бит/с} = 8 \text{ бит/с}$; $1 \text{ Кбит/с} = 2^{10} \text{ бит/с} = 1024 \text{ бит/с}$; $1 \text{ Мбит/с} = 2^{10} \text{ Кбит/с} = 1024 \text{ Кбит/с}$; $1 \text{ Гбит/с} = 2^{10} \text{ Мбит/с} = 1024 \text{ Мбит/с}$.

18.2. Локальные сети

При работе на персональном компьютере вне сети (т. е. в автономном режиме) пользователи могут обмениваться информацией (программами, файлами и т. п.), лишь копируя ее на дискеты и передавая их друг другу, что может занимать довольно продолжительное время.

В компьютерной сети обмен информацией происходит несравнимо быстрее. Создание компьютерных сетей было вызвано потребностью совместного использования информации у людей, работающих на удаленных друг от друга компьютерах. Сети предоставляют пользователям возможность не только быстрого обмена информацией, но и совместного использования принтеров и других периферийных устройств и даже одновременной работы с документами.

Локальная сеть объединяет компьютеры, установленные в одном помещении (например, в компьютерном классе, насчитывающем 10 компьютеров) или в одном здании (например, в здании колледжа объединены в локальную сеть несколько десятков компьютеров, установленных в различных предметных кабинетах).

В небольших локальных сетях все компьютеры обычно равноправны, т. е. пользователи самостоятельно решают, какие ресурсы

своего компьютера (диски, каталоги, файлы) сделать общедоступными по сети. Такие сети называются *одноранговыми*.

Если к локальной сети подключено более десяти компьютеров, одноранговая сеть может оказаться недостаточно производительной. Для увеличения производительности, а также для обеспечения большей надежности при хранении информации в сети некоторые компьютеры специально выделяются для хранения файлов и программных приложений. Такие компьютеры называются *серверами*, а локальная сеть — *сетью на основе сервера*. Каждый компьютер, подключенный к локальной сети, должен иметь специальную плату — сетевой адаптер. В первых моделях ПК эта плата вставлялась в слот системной платы, в современных — входит в ее состав.

Основной функцией сетевого адаптера является передача и прием информации из сети. В настоящее время наиболее часто используются сетевые адаптеры типа EtherNet. Соединение компьютеров (сетевых адаптеров) между собой производится с помощью кабелей различных типов (коаксиального, витой пары, оптоволоконного). Важнейшей характеристикой локальных сетей, которая определяется типом используемых сетевых адаптеров и кабелей, является скорость передачи информации по сети. Для локальной сети она обычно составляет от 10 до 100 Мбит/с.

Картина (вид) общей схемы соединения компьютеров в локальной сети называется *топологией сети*. Топологии сети могут быть различными.

При топологии «шина» (рис. 18.3) кабель проходит от одного компьютера к другому, последовательно соединяя компьютеры и периферийные устройства между собой. Электрические сигналы, передаваемые компьютерами, распространяются по всему кабелю — от одного его конца до другого. Достигая конца кабеля, они могут отражаться и мешать обмену информацией. Для того чтобы предотвратить помехи, на каждом конце кабеля устанавливают *терминаторы T*, поглощающие эти сигналы.

При топологии «кольцо» компьютеры подключаются к кабелю, замкнутому в кольцо. У такого кабеля нет свободного конца, а значит, не нужны и терминаторы.

Если к каждому компьютеру подходит отдельный кабель из одного центрального узла, то реализуется топология «звезда» (рис. 18.4). При такой схеме соединения центральным узлом может быть более мощный компьютер. Более удобным является использование в качестве цент-

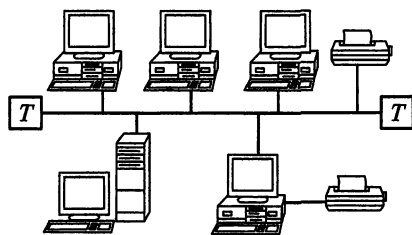


Рис. 18.3. Топология «шина»

рального узла специального устройства, которое называется *концентратор* (англ. hub).

Преимущество локальной сети топологии «звезда» перед локальной сетью топологии «шина» состоит в том, что при выходе из строя сетевого кабеля у одного компьютера локальная сеть в целом продолжает функционировать.

В операционной системе-оболочке Windows пользователь любого компьютера, подключенного к сети, может предоставить доступ к своим дискам, папкам или файлам, пользователям, работающим за другими компьютерами сети. Общими для всех пользователей сети могут быть и принтер, и сканер, и запоминающие устройства с большим объемом памяти. Для ограничения доступа к некоторым ресурсам можно вводить условное слово — пароль.

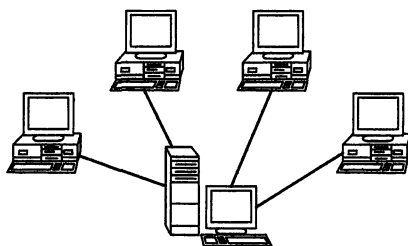


Рис. 18.4. Топология «звезда»

18.3. Сеть Интернет

Локальные сети обычно объединяют несколько десятков компьютеров, размещенных в одном здании, т. е. они не обеспечивают совместный доступ к информации пользователям, находящимся на большом расстоянии друг от друга, например в различных частях города. В этом случае используют *региональные сети*, которые объединяют компьютеры в пределах одного региона (города или страны).

Многие организации, заинтересованные в защите информации от несанкционированного доступа (например, военные, финансовые и др.), создают собственные так называемые *корпоративные сети*. Корпоративная сеть может объединять тысячи и десятки тысяч компьютеров, размещенных в различных городах и даже странах.

Стремление к формированию единого мирового информационного пространства привело к созданию глобальной компьютерной сети Интернет, которая предоставляет свои информационные ресурсы и услуги более чем миллиарду человек во всех странах мира. Эта сеть объединяет многие локальные, региональные и корпоративные сети (рис. 18.5).

В каждой входящей в сеть Интернет локальной или корпоративной сети обычно имеется по крайней мере один компьютер (*сервер Интернета*), который постоянно подключен к сети Интернет с помощью линии связи с высокой пропускной способностью. В качестве таких магистральных линий связи обычно исполь-

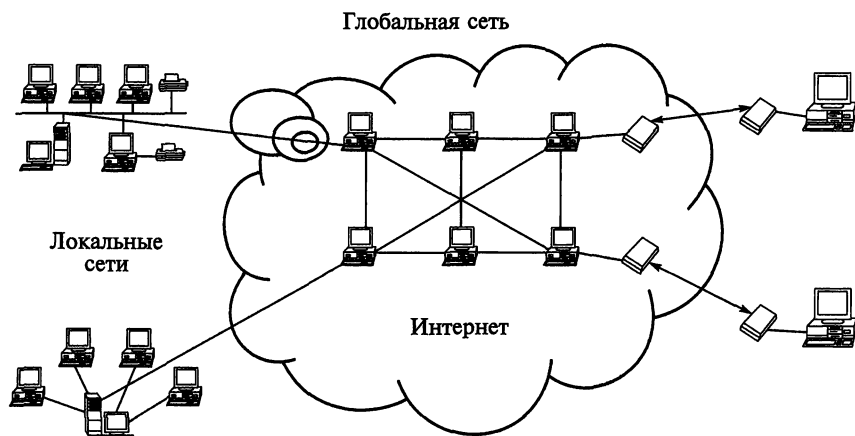


Рис. 18.5. Структура глобальной сети Интернет

зуются оптоволоконные линии с пропускной способностью несколько десятков гигабит в секунду.

Надежность функционирования глобальной сети обеспечивается большим числом линий связи между региональными сегментами сети. Например, российский региональный сегмент сети Интернет имеет несколько магистральных линий связи, соединяющих его с североамериканским, европейским и японским сегментами. Основу, «каркас» сети Интернет составляют более 150 млн серверов, постоянно подключенных к сети. К этим серверам пользователи могут подключаться с помощью локальных сетей или коммутируемых телефонных линий. Число пользователей Интернета уже превысило миллиард. В России миллионы людей пользуются услугами Интернета не только с домашнего или служебного ПК, но и из пунктов коллективного доступа. Во многих почтовых отделениях уже установлены компьютеры, подключенные к Интернету. За небольшую плату каждый человек может войти в глобальную сеть.

Для того чтобы в процессе обмена информацией компьютеры могли найти друг друга, в сети Интернет существует единая система адресации, основанная на использовании IP-адреса (Internet Protocol адрес). Каждый компьютер, подключенный к сети Интернет, имеет свой персональный 32-битный IP-адрес. Общее число возможных IP-адресов составляет $2^{32} = 4\,294\,967\,296$, т. е. более 4 млрд.

Система IP-адресации учитывает структуру сети Интернет, т. е. то, что Интернет является сетью сетей, а не объединением отдельных компьютеров. Каждый IP-адрес содержит адрес сети и адрес компьютера в данной сети.

В зависимости от числа компьютеров в сети IP-адреса разделяются на три класса: А, В, С. Первые биты адреса отводятся для

IP-адресация в сетях различных классов

Класс сети	Биты класса	Число бит адреса	
		сети	компьютера
А	0	7	24
В	1 0	14	16
С	1 1 0	21	8

идентификации класса, а остальные разделяются на адрес сети и адрес компьютера (табл. 18.1).

Например, адрес сети класса А (наиболее крупные сети) имеет только 7 бит для адреса сети и 24 бит для адреса компьютера. Таким образом, могут существовать лишь $2^7 = 128$ сетей класса А, зато в каждой из них могут содержаться $2^{24} = 16\,777\,216$ компьютеров.

В десятичной записи IP-адрес состоит из четырех разделенных точками чисел, каждое из которых находится в диапазоне от 0 до 255. Например, IP-адрес сервера компании МТУ-Интел записывается как 195.34.32.11.

По первому числу IP-адреса компьютера просто определить его принадлежность к сети того или иного класса. Если это число от 0 до 127, то сеть класса А, от 128 до 191 — класса В, от 192 до 223 — класса С. Следовательно, сервер компании МТУ-Интел относится к сети класса С, адрес которой 195, а адрес этого сервера в сети 34.32.11.

Компьютеры находят друг друга по числовому IP-адресу, однако человеку запомнить 12-значный числовой адрес нелегко. В связи с этим была введена *доменная система имен* (DNS — Domain Name System), в которой используются слова, записываемые латинскими буквами. Доменная система имен имеет иерархическую структуру, т. е. делится на домены первого уровня, второго уровня и т. д. Домены первого (верхнего) уровня бывают двух типов — географические (двухбуквенные — каждой стране соответствует двухбуквенный код) и административные (трехбуквенные).

России принадлежит географический домен ru. Давно существующие в нашей стране серверы могут относиться к домену su (СССР). Обозначение административного домена позволяет определить профиль организации — владельца домена (com — коммерческая, org — некоммерческая, edu — образовательная, net — компьютерная сеть).

Например, компания Microsoft зарегистрировала домен второго уровня microsoft в административном домене верхнего уровня com, а Министерство образования и науки РФ — домен третьего уровня ed в домене второго уровня gov, который в свою очередь

находится в географическом домене верхнего уровня ru. Имена компьютеров, которые являются серверами сети Интернет, включают в себя полное доменное имя и собственно имя компьютера. Таким образом, основной сервер компании Microsoft имеет имя www.microsoft.com, а сервер Минобразования — www.ed.gov.ru.

Сеть Интернет объединяет более миллиарда компьютеров, которые в свою очередь входят в громадное количество различных локальных, региональных и корпоративных сетей. Совместная работа в сети Интернет громадного числа отправителей и получателей информации возможна лишь при условии единых правил обмена информацией. Сводом таких правил является протокол передачи данных TCP/IP. Термин TCP/IP включает название двух протоколов — транспортного протокола TCP (Transmission Control Protocol) и протокола маршрутизации IP (Internet Protocol).

Протокол маршрутизации IP обеспечивает передачу информации между компьютерами сети. Пояснить работу данного протокола можно на примере работы обычной почты.

На конверте письма указываются адрес получателя (кому письмо) и адрес отправителя (от кого письмо). Аналогично передаваемая по сети информация «помещается в конверт», на котором указаны IP-адреса компьютеров получателя и отправителя. Информация в таком условном «конверте» называется *IP-пакетом*.

В процессе пересылки обыкновенных писем их сначала доставляют на ближайшее к отправителю почтовое отделение, а затем передают по цепочке почтамтов на ближайшее к получателю почтовое отделение. На промежуточных почтамтах письма сортируют, т.е. определяют, на какой следующий почтамт или почтовое отделение необходимо отправить данное письмо.

На пути от компьютера-отправителя к компьютеру-получателю IP-пакеты также проходят через многочисленные промежуточные серверы сети Интернет, в которых производится операция маршрутизации. В результате маршрутизации IP-пакеты направляются от одного сервера к другому, постепенно приближаясь к компьютеру-получателю. Скорость получения информации зависит не от удаленности веб-сервера (т.е. сервера, от которого начинают свой путь пакеты), а от числа промежуточных серверов, качества (их пропускной способности) линий связи и их загруженности в данный момент.

Скорость передачи и получения информации зависит также от ее размеров. Воспользуемся опять аналогией с работой почты. Пусть нам необходимо переслать по почте многостраничную рукопись, а почта бандероли и посылки не принимает. В этом случае, если рукопись не помещается в обычный почтовый конверт, ее надо разобрать на листы и переслать их в нескольких конвертах. При этом листы рукописи необходимо обязательно пронумеровать,

чтобы получатель мог их затем расположить в правильной последовательности.

В сети Интернет подобная ситуация возникает весьма часто, когда компьютеры обмениваются большими (и даже средними) по объему файлами.

При пересылке такого файла целиком он может надолго занять канал связи, сделав его недоступным для пересылки других сообщений. Для того чтобы этого не происходило, необходимо разбить большой файл на мелкие части, пронумеровать их и транспортировать в отдельных IP-пакетах до компьютера-получателя. Для IP-протокола, ответственного за маршрутизацию, эти пакеты совершенно не связаны между собой. Поэтому последний IP-пакет может прийти даже раньше первого, да и маршруты доставки этих пакетов могут оказаться разными. Благодаря протоколу TCP после получения всех пакетов на компьютере-получателе будет собран исходный файл в правильной последовательности.

Для предоставления различным пользователям возможности подключения к сети Интернет созданы многочисленные организации, которые называются *провайдерами*. Они используют разные способы подключения, которые отличаются предоставляемыми пользователю услугами и стоимостью подключения. Наилучшие возможности обеспечиваются при непосредственном подключении к сети Интернет с помощью высокоскоростного (оптоволоконного или спутникового) канала связи. Такое подключение достаточно дорого и обычно используется большими организациями для подключения локальных сетей. Провайдеры услуг сети Интернет имеют высокоскоростное соединение своих серверов с сетью, а своим клиентам-пользователям они предоставляют доступ к ней на коммерческой основе по коммутируемым телефонным каналам.

Осуществлять непосредственную передачу информации по коммутируемым телефонным линиям компьютеры не могут, так как обмениваются данными с помощью цифровых электронных импульсов, а по телефонной линии можно передавать только аналоговые (непрерывные) сигналы.

Для подключения компьютера к телефонной линии используется модем. На передающей стороне выполняется *модуляция* аналогового электрического сигнала определенной (несущей) частоты последовательностями электрических импульсов. Компьютер посылает модему такие последовательности, а модем преобразует цифровые сигналы компьютера в модулированный аналоговый сигнал (см. рис. 7.30).

Этот сигнал передается по телефонной линии. На принимающей стороне модем производит обратное преобразование — *демодуляцию*, т. е. преобразует входящий аналоговый сигнал в последовательность цифровых импульсов.

Для экономии времени передачи данных по линиям связи необходимо осуществлять *сжатие данных*. На передающей стороне получаемые из последовательного порта компьютера данные модем по определенным алгоритмам сжимает, а на принимающей — восстанавливает в исходном виде.

В процессе передачи данных модем выполняет также *коррекцию ошибок*. Это очень важно, так как при передаче программного файла ошибка только в одном бите может привести к неработоспособности всей программы. Если в процессе передачи пакета данных произошла ошибка, модем на принимающей стороне дает запрос на повторную передачу этого пакета.

18.4. Информационные технологии

Сеть Интернет развивается и расширяется очень быстрыми темпами, и найти нужную информацию среди миллиардов веб-страниц и файлов становится все сложнее. Для поиска информации используются специальные поисковые серверы, которые содержат достаточно полную и постоянно обновляемую информацию о веб-страницах, файлах и других документах, хранящихся на десятках миллионов серверов сети Интернет. Используются различные механизмы поиска, хранения и предоставления пользователю информации. Поисковые серверы сети Интернет можно разделить на две группы: поисковые системы общего назначения и специализированные поисковые системы.

Современные поисковые системы часто являются информационными *порталами*, которые предоставляют пользователям не только возможности поиска документов в сети Интернет, но и доступ к другим информационным ресурсам (новостям, прогнозу погоды, курсу валют, географическим картам и т.д.).

Поисковые системы общего назначения являются базами данных, которые содержат сгруппированную по темам информацию о всех ресурсах Всемирной паутины (www — World Wide Web). Такие поисковые системы находят веб-страницы по ключевым словам в БД или путем поиска в иерархической системе каталогов.

Интерфейс таких поисковых систем общего назначения содержит список разделов каталога и поле поиска. В поле поиска пользователь вводит ключевые слова для интересующего его документа, а для ускорения поиска выбирает определенный раздел каталога.

Пополнение БД осуществляется с помощью специальных программ-роботов, которые периодически обследуют веб-серверы сети Интернет. Программы-роботы просматривают все встречающиеся документы, выделяют в них ключевые слова и заносят в БД, содержащую URL-адреса документов (URL — Uniform Resource Locator — единообразный локатор ресурсов, т.е. идентификатор, или адрес ресурсов в сети Интернет).

Для обеспечения большего соответствия между содержанием БД поисковой системы и реальным состоянием сети Интернет большинство поисковых систем разрешают автору веб-сайта самому внести информацию в БД, заполнив регистрационную анкету. При этом разработчик сайта вносит URL-адрес сайта, его название, краткое описание содержания сайта, а также ключевые слова, по которым легче всего будет найти сайт.

Сайты в БД распределяются по числу их посещений в день, неделю или месяц. Посещаемость сайтов определяется с помощью специальных счетчиков, которые могут быть установлены на сайте. Счетчики фиксируют каждое посещение сайта и передают информацию о числе посещений на сервер поисковой системы.

Простой запрос содержит одно или несколько ключевых слов, которые, по мнению запрашивающего, являются главными для этого документа. Можно также использовать сложные запросы, использующие логические операции, шаблоны и т.д. Спустя некоторое время после отправки запроса поисковая система представляет список URL-адресов с кратким аннотациями тех документов, в которых были найдены указанные запрашивающим ключевые слова. Для просмотра этого документа в браузере достаточно активизировать указывающую на документ ссылку.

Если ключевые слова были выбраны неудачно, то список URL-адресов документов может оказаться слишком большим (содержать десятки и даже сотни тысяч ссылок). Для того чтобы уменьшить список, можно в поле поиска ввести дополнительные ключевые слова или воспользоваться каталогом поисковой системы.

Наиболее мощными поисковыми системами общего назначения в русскоязычной части сети Интернет являются серверы Рамблер (<http://www.rambler.ru>), Апорт (<http://www.aport.ru>) и Яндекс (<http://www.yandex.ru>), а по всей сети Интернет — сервер Google (<http://www.google.com>). Поисковая система Google принимает и обслуживает запросы и на русском языке.

Специализированные поисковые системы позволяют искать информацию в других информационных «слоях» сети Интернет — серверах файловых архивов, почтовых серверах и др.

Для поиска файлов в серверах файловых архивов используют специализированные поисковые системы двух типов: поисковые системы на основе использования баз данных и каталоги файлов. Для поиска файла в системе с использованием БД достаточно ввести имя файла в поле поиска и поисковая система выдаст URL-адреса мест хранения данного файла.

В БД российской файловой поисковой системы (<http://www.filesearch.ru>) содержатся сведения о нескольких миллионах файлов, размещенных на тысячах серверов файловых архивов российской части сети Интернет.

Специализированные поисковые системы позволяют искать адрес электронной почты по имени человека или, наоборот, имя человека по адресу электронной почты. Примером такой системы может служить поисковая система WhoWhere? (КтоГде?), адрес которой <http://www.whowhere.com>.

В сети Интернет существует достаточно большое число серверов, на которых хранятся мультимедиа (звуковые, графические и видео) файлы. Мультимедийные файлы имеют значительный информационный объем. Например, объем высококачественного звукового файла в цифровом формате продолжительностью звучания всего 1 с составляет 187,5 Кбайт, высококачественного графического файла — 1,37 Мбайт, а одна секунда видеофайла (из расчета 25 кадров в секунду) — 34,25 Мбайт.

Таким образом, для передачи по компьютерным сетям мультимедийных файлов в стандартных цифровых форматах требуются линии связи с высокой пропускной способностью, а для воспроизведения файлов нужно их предварительно загрузить в компьютер.

Для прослушивания и просмотра мультимедийных файлов непосредственно в процессе их получения из сети в режиме реального времени были разработаны специальные методы.

Для сжатия звуковых файлов в настоящее время широко используются стандарты MPEG player3 (сокращенно MP3) и MPEG4. Название свое они получили по имени организации-разработчика (Moving Picture Experts Group — экспертная группа кинематографии).

Стандарт MP3 при сжатии разбивает весь частотный спектр звукового сигнала на части, а затем удаляет звуки, не воспринимаемые человеком. Для передачи звука в формате стандарта MP3 по сети в режиме реального времени или прослушивания его достаточно пропускной способности линий связи 128 Кбит/с. Это обусловило высокую популярность данного формата и привело к накоплению на серверах сети Интернет огромного количества музыкальных и других записей в формате стандарта MP3. Для прослушивания таких записей используются различные программные проигрыватели. Однако прослушивание по сети звуковых файлов в формате стандарта MP3 в режиме реального времени с использованием модемного подключения (максимальная скорость — 56 Кбит/с) невозможно из-за недостаточной скорости передачи информации.

Широкое распространение в сети Интернет получили технологии передачи потокового звука и видео, которые можно использовать и при модемном подключении. Эти технологии предусматривают передачу звуковых и видеофайлов по частям в буфер локального компьютера, что обеспечивает возможность их потокового воспроизведения даже при использовании модемного

подключения. Для прослушивания и просмотра таких файлов необходимы специальные проигрыватели (RealPlayer или Windows Media Player).

Электронная коммерция (e-commerce) — это распространение товаров и услуг с помощью сети Интернет. В 2000 г. оборот электронной торговли превысил 100 млрд дол. Самой удобной для покупателя формой электронной торговли являются интернет-магазины. В российской сети Интернет существуют уже сотни магазинов, в которых можно купить самые разнообразные товары: компьютеры и программы, книги и CD, продукты питания и т.д.

В последнее время для расчетов через сеть Интернет стали использоваться так называемые *цифровые деньги*. Покупатель перечисляет определенную сумму обычных денег в банк, а взамен получает определенную сумму цифровых денег, которые существуют только в электронном виде (с использованием специальной программы) в компьютере покупателя. При расчетах через сеть Интернет цифровые деньги поступают к продавцу, который переводит их в банк, а взамен получает обычные деньги.

Информационные технологии применяются в образовании (дистанционное обучение), здравоохранении (телемедицина) и многих других важных для каждого человека областях.

Целевая государственная программа «Электронная Россия...» направлена на компьютеризацию всех областей человеческой деятельности в нашей стране. Но наряду с этой программой существует и успешно реализуется еще одна программа, направленная на применение вычислительной техники для решения одной конкретной задачи. Речь идет о создании отечественной навигационной системы, получившей название ГЛОНАСС. В этой системе совместно используются неоспоримые достоинства отечественной космической техники и пока еще относительно слабо развитая в нашей стране компьютерная техника.

Предназначена эта система для очень широкого применения в морском, воздушном, наземном транспорте, в землеустройстве, геодезии и картографии, строительстве и др. Первоочередное применение эта система находит для обороны, т.е. для обеспечения безопасности нашей страны. Поэтому понятно, что такая система должна быть полностью отечественного производства. Она не должна зависеть от работы подобных иностранных систем (например, американской системы GPS, европейской системы Галилео или разрабатываемой японской подсистемы MSAS).

Понятие «навигация» появилось и чаще всего употребляется в морском судоходстве. Задача навигации — обеспечить такое направление (курс) корабля, который обеспечит его приход из порта отправления в порт назначения. При этом необходимо точно знать координаты корабля (его место) в данный момент времени. Задача определения местоположения объекта называется *обсерва-*

цией. Следовательно, при навигации необходимо решать и задачу обсервации. При плавании по морям обсервация выполняется следующим образом. Определяют направления (углы) на объекты, координаты которых точно известны. Пока это делается с помощью радионавигационного оборудования. Поворачивая антенну направленного действия, можно найти такое ее положение, при котором обеспечивается наиболее громкая слышимость специального берегового передатчика. На практике, правда, оказывается более удобным выбирать такое положение антенны, при котором будет самый слабый сигнал. Направление антенны и дает угол, под которым и находится передатчик. Если взять два направления на два передатчика, то мы уже имеем два значения угла. На карте проводят под этими углами две прямых. Точка пересечения этих прямых и есть место корабля. Поскольку эти углы определены с некоторой погрешностью, то и точка, обозначающая место корабля, не является точным местом. Большей достоверностью обладает способ определения места по трем углам, т. е. с использованием трех разных береговых навигационных радиостанций, по возможности наиболее удаленных друг от друга. В этом случае проводятся три прямых. Как правило, они не имеют общей точки пересечения. Зато они выделяют некоторый треугольник, внутри которого и находится истинное место корабля. Координаты центра этого треугольника принимаются за координаты места, а расстояние до вершин треугольника дает нам величину погрешности, с которой эти координаты определены.

Подобный способ определения места объекта использован и в спутниковых навигационных системах, объединенных с мощными компьютерными коммуникационными устройствами. Система ГЛОНАСС основана на определении координат с помощью большого количества спутников, находящихся на большом удалении от Земли. «Созвездие» ГЛОНАСС состоит из 24 геоцентрических спутников, равномерно распределенных в трех орбитальных плоскостях, разнесенных на 120° вокруг планеты Земля. Каждая плоскость содержит 8 спутников, распределенных через 45° . Все спутники имеют круговые орбиты и находятся на расстоянии 19 100 км от Земли. При определении координат объекта определяются расстояния от него до трех спутников, расположенных в трех разных плоскостях. Принимаются сигналы не менее чем от четырех радиовидимых навигационных космических аппаратов.

При определении параметров, характеризующих положение наземного объекта, используется эффект Доплера, который заключается в изменении частоты в зависимости от скорости. С помощью доплеровского сдвига частоты определялись параметры движения еще первого искусственного спутника Земли, запущенного в нашей стране 4 октября 1957 г.

В навигационных системах расстояния между наземным объектом и навигационными космическими аппаратами вычисляются с помощью радиолуча, направленного на спутник. Определив разницу во времени между сигналом, посланным на спутник, и отраженным сигналом, можно узнать и расстояние до спутника. Поскольку координаты спутника известны, то можно вычислить и координаты объекта. Точность этих координат в значительной степени зависит от точности измерения времени. Так вот, в навигационных системах используются атомные стандарты частоты, обеспечивающие высочайшую точность. Система ГЛОНАСС уже работает, хотя число спутников еще почти втрое меньше, чем запланировано. При полном их количестве будет обеспечено определение координат подвижных объектов с погрешностью менее 5 м.

Выпуск сравнительно недорогой потребительской аппаратуры позволяет использовать ее в быту, особенно в автомобильных поездках, туризме, путешествиях, в спорте.

Поразительное распространение получило использование спутниковой аппаратуры для автомобильной навигации. Например, на европейском рынке удельный вес стоимости таких устройств составил к 2007 г. 80 %, тогда как доли авиации и морского флота составили 5 и 4 % соответственно.

Очень нужны такие системы для успешной работы экстренных служб (службы скорой помощи, ликвидации аварий, милиция и т. п.). Распоряжением президента России система ГЛОНАСС получила статус системы двойного назначения: военного и гражданского.

18.5. Коммуникационные возможности компьютеров

Как уже неоднократно отмечалось, вместо понятия «вычислительная техника» все чаще употребляется понятие «информационно-коммуникационная техника». Тем самым подчеркивается, что эта техника используется для обработки информации и для коммуникации (связи). В докомпьютерную эру основными видами связи были почтовая, проводная (телефонная и телеграфная) и радиосвязь. Теперь повсеместно распространяется связь с помощью компьютеров.

В последнее время расширяется интерактивное общение через сеть Интернет в режиме реального времени. Возросшие скорость передачи данных и производительность компьютеров позволяют пользователям не только обмениваться текстовыми сообщениями в этом режиме, но и осуществлять аудио- и видеосвязь.

В сети Интернет существуют тысячи серверов Internet Relay Chat (IRC), с помощью которых возможно интерактивное общение. Самый простой способ общения — это обмен сообщениями, на-

бираемыми с помощью клавиатуры. Пользователь вводит таким образом сообщение, и оно высвечивается в окне, которое одновременно видят все участники встречи (посетители чата). Если компьютеры собеседников оборудованы звуковой картой, микрофоном и наушниками или акустическими колонками, то можно обмениваться и звуковыми сообщениями. Однако «живой» разговор одновременно возможен только между двумя собеседниками. Для того чтобы собеседники могли видеть друг друга, к компьютерам должны быть подключены видеокамеры. Аналоговые видеокамеры подключаются через специальные видеоплаты, а цифровые — через параллельный порт компьютера. Качество звука и изображения зависит от скорости модема и пропускной способности канала связи, которые должны быть не менее 28,8 Кбит/с.

Для организации интерактивного общения необходимо специальное программное обеспечение (например, программа NetMeeting, которая входит в состав программы Internet Explorer). В последние годы большую популярность приобрело интерактивное общение через серверы ICQ (читается ай-си-кью и звучит как «I seek you» — «Я ищу тебя»). В настоящее время в системе ICQ зарегистрировано более 150 млн пользователей, причем каждый пользователь имеет уникальный идентификационный номер. После подключения к сети Интернет пользователь может начать общение с любым зарегистрированным в системе ICQ и подключенным в это время к сети Интернет пользователем. Программа уведомляет о присутствии в данный момент в сети Интернет абонентов из предварительного списка (My Contact List) и позволяет установить с ними контакт. Чтобы стать абонентом системы ICQ, достаточно загрузить программу ICQ-клиент с файлового сервера (например, www.freeware.ru) и зарегистрироваться в процессе ее установки на компьютер. Система интерактивного общения ICQ предоставляет различные формы общения: электронную почту, обмен текстовыми сообщениями, интернет-телефонию, передачу файлов, поиск в сети людей и т. д.

Интернет-телефония дает возможность пользователю сети Интернет использовать телефонную связь компьютер — телефон, компьютер — компьютер или телефон — компьютер. Провайдеры Интернет-телефонии обеспечивают такую связь с помощью специальных телефонных серверов сети Интернет, которые подключены и к ней, и к телефонной сети. С мобильного телефона на компьютер, подключенный к сети Интернет, и с компьютера на мобильный телефон можно отправлять SMS (Short Message Service — короткие текстовые сообщения). Для беспроводного доступа с мобильных телефонов к информационным и сервисным ресурсам сети Интернет используется протокол WAP (Wireless Application Protocol). Для работы в сети Интернет по этому протоколу не нужны дополнительные устройства (компьютер и модем), достаточно

одного мобильного телефона с поддержкой WAP. Высокоскоростной доступ в сеть Интернет с мобильного телефона можно осуществить по технологии GPRS (General Packet Radio Service). В этом случае можно работать с WAP-сайтами непосредственно с мобильного телефона, а на подключенном к нему компьютере можно просматривать веб-страницы, загружать файлы, работать с электронной почтой и любыми другими ресурсами сети Интернет.

При использовании технологии GPRS максимально возможная скорость передачи данных составляет 171,2 Кбит/с. Это приблизительно в три раза больше скорости доступа по коммутируемым телефонным линиям и значительно превышает скорость передачи данных в мобильных телефонных сетях стандарта GSM (9,6 Кбит/с).

Большой популярностью пользуется голосовая связь между пользователями Интернета с помощью программы Skype, которую можно бесплатно загрузить по адресу www.skype.com.

Контрольные вопросы

1. Что такое сеть?
2. Приведите примеры сетей для снабжения продуктами, электроэнергией, теплом, водой.
3. Как устроены компьютерные сети?
4. В чем разница между сетью Интернет и локальной сетью?
5. Назовите основные топологические виды сетей.
6. Каким образом передается информация в сети от одного компьютера к другому?
7. Что такое модем?
8. Что такое информационные технологии?
9. Перечислите возможные применения информационных технологий.
10. Предложите другие применения компьютеров и информационных технологий.

РАБОТА НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ**19.1. Загрузка компьютера**

В минимальный набор оборудования, который необходим для того, чтобы работать на персональном компьютере, входят системный блок, монитор, клавиатура и манипулятор типа мышь. Главной составляющей является системный блок, в котором размещаются системная плата с процессором, жесткий диск (винчестер), дисководы для гибких и компакт-дисков. В магазинах, продающих вычислительную технику, именно системный блок и называют компьютером. Монитор, клавиатура, мышь относятся к комплектующим изделиям, которые подключаются к системному блоку. Без них полноценная работа на ПК практически невозможна. В системном блоке имеется блок питания, от которого отходит кабель с вилкой, вставляемой в розетку обычной бытовой электросети. Монитор подключается к системному блоку с помощью двух кабелей, по одному подаются сигналы, формирующие изображение на экране монитора, а по другому к монитору подводится напряжение питания. Иногда монитор имеет собственный кабель питания с обычной вилкой. Клавиатура и мышь также имеют собственные кабели для подключения к системному блоку, которые заканчиваются разъемами такой формы, что исключается их неправильное подсоединение. Сначала надо включить монитор, но обычно он бывает уже включен (даже при выключенном ПК).

Для того чтобы включить ПК, надо нажать на соответствующую кнопку на системном блоке. Эта кнопка выделяется своими размером и цветом. Возле нее имеется соответствующая надпись. Если надписи на лицевой панели системного блока сделаны на английском языке, то кнопка включения помечена словом «POWER». После нажатия на нее начинается процесс загрузки компьютера, т. е. подготовка его к работе.

В ходе процесса загрузки выполняются программы операционной системы, уже заранее записанные на жесткий диск фирмой, которая поставляет компьютеры. Сначала работают программы проверки оборудования, а информация о ходе проверки выводится на экран монитора. Эта информация быстро меняется, но можно успеть прочесть данные об объеме оперативной памяти, о том, какая базовая система ввода — вывода использована в компьютере и какие дисководы в нем установлены. Таким образом, проверяется конфигурация компьютера. Обычно перед этой проверкой

(тестированием) на экран выводится сообщение о том, на какую клавишу надо нажать, чтобы войти в программу конфигурирования компьютера. Как правило, для этого надо нажать на клавишу «Delete», после чего на экране появляется таблица, с помощью которой можно задать текущее время и дату, типы дисководов, некоторые другие параметры компьютера. При отсутствии опыта работы на ПК лучше ничего не менять. Однако в дальнейшем рекомендуется ознакомиться с возможностями этой программы. В частности, желательно иметь в запасе трехдюймовую загрузочную дискету. Если загрузка с жесткого диска по какой-либо причине не будет выполняться, то, вставив загрузочную дискету в дисковод, можно будет выполнить начальный процесс загрузки операционной системы.

Начальная загрузка может продолжаться несколько десятков секунд. Все это время информация на экране меняется. В большинстве компьютеров работа происходит в операционной оболочке Windows (той или иной версии). В конце загрузки на экране появляются так называемые иконки — значки, обозначающие самые разные основные программы, которые могут использоваться в данном компьютере. Примерно в середине экрана появляется курсор — стрелка, положение которой можно менять с помощью мыши. Если рядом с курсором еще продолжается вращение песочных часов, значит, процесс загрузки не закончился и надо подождать, когда песочные часы исчезнут с экрана.

Вызвать требующуюся программу можно двумя способами. Во-первых, если пиктограмма нужной программы уже имеется на экране (экран с ярлыками-пиктограммами, который виден по окончании процесса загрузки, называется рабочим столом), то надо подвести к этой пиктограмме курсор и нажать на левую клавишу мыши, говоря по-другому, щелкнуть левой кнопкой. Во-вторых, можно курсор подвести к показанной на рабочем столе кнопке «Пуск» и нажать на левую клавишу мыши. Появится меню. Если нужной программы в нем нет, следует перейти из данного меню в другое и т.д. Этот процесс продолжают до тех пор, пока не будет найдено меню с требующейся программой. Последовательность указанных переходов из одного меню в другое — это и есть путь (patch) к нужной программе.

19.2. Файлы, каталоги, папки

Вся информация, которая хранится на дисках компьютера, оформлена в виде файлов. Каждому файлу присваивается определенное имя, по которому его можно найти на соответствующем диске. Имя файла состоит из двух частей, разделенных точкой. Левая часть — это собственно имя, а правая часть определяет тип файла и называется *расширением*. Файлы создаются с помощью той или

иной программы. Если файл создан пользователем компьютера, то тот сам и дает ему имя. Расширение сообщает о том, с помощью какой программы создан данный файл. Например, файлам, созданным с помощью текстовых редакторов, часто присваиваются расширения .doc; .txt; .rtf; рисункам и другим графическим файлам — .jpg; .bmp; файлам, записанным на языках программирования — .bas (на языке Бейсик); .pas (на языке Паскаль). Файлы, представляющие собой готовые к исполнению программы, имеют обычно расширения .exe; .com. Расширение в имени файла позволяет легче ориентироваться в большом объеме информации, хранящейся в памяти компьютера.

Размеры имени файла (его левой части) зависят от используемой операционной системы. В MS-DOS собственно имя файла не должно превышать восьми букв латинского алфавита, а в операционной системе-оболочке Windows имя файла может содержать 255 символов, в том числе буквы русского алфавита. Создавая какой-либо файл, следует очень ответственно отнестись к выбору имени для него, чтобы облегчить себе в дальнейшем его поиск, поскольку случайно присвоенное имя пользователь спустя какое-то время, может просто забыть.

В компьютере может храниться огромное число файлов. Большинство из них находится на жестком диске, какая-то часть — на сменных дискетах (гибких дисках), достаточно много — на компакт-дисках, которые можно вставлять в устройства для их чтения (CD-ROM). Порядок хранения файлов обеспечивается принятой файловой системой. Практически все пользователи применяют многоуровневую иерархическую файловую систему, имеющую древовидную структуру. Каждый диск разбивается на две области: в одной хранятся файлы, в другой — каталог. В каталоге записаны имена файлов и указано место размещения самого каталога на диске. В многоуровневой файловой системе используются каталоги нескольких уровней. Начальный, самый главный каталог (его называют корневым) содержит перечень каталогов 1-го уровня, в каждом из которых могут находиться каталоги 2-го уровня и т. д. (см. рис. 15.1). На каждом уровне кроме самого перечня каталогов следующего уровня могут находиться и файлы. Каталог каждого уровня присваивается имя. К выбору его пользователь должен отнестись так же ответственно, как и к выбору имени файла.

Поскольку в компьютере можно использовать информацию, записанную на разных дисках, присваиваются имена и самим дискам, но это уже, как правило, делает не пользователь. Имя диска состоит из одной буквы латинского алфавита: А или В — для гибких дисков; С, D, Е и т. д. — для жестких и лазерных дисков. Иногда реальный физический диск для удобства разбивают на несколько условных, так называемых логических дисков.

Для выбора нужного файла надо знать, на каком диске и в каком каталоге он находится. Иными словами, надо знать *путь к файлу*, который представляет собой последовательную запись логического имени диска и каталогов (от высшего уровня — к низшему) с указанием в конце собственного имени файла. Все элементы записываются через разделитель «\». Иногда путь к файлу называют полным именем файла.

Слово «каталог» используется в операционной системе MS-DOS. В более удобной для пользователя системе Norton Commander вместо слова «каталог» применяются слова «директория» и «поддиректория» (для каталогов низшего уровня). В системе-оболочке Windows аналогом слова «каталог» является слово «папка» (folder).

В Windows на самом верхнем уровне иерархии папок находится папка «Рабочий стол». На следующем уровне обычно расположены папки «Мой компьютер», «Мои документы», «Корзина», «Сетевое окружение» и другие папки, некоторые из которых могут быть созданы самим пользователем.

19.3. Работа с дисками

Для того чтобы сохранять информацию на магнитном диске, он должен быть отформатирован, т. е. должны быть созданы физическая и логическая структура диска. Жесткий диск в компьютере обычно уже отформатирован и на него записаны необходимые для пользователя часто употребляемые программы. В повседневной работе приходится использовать и гибкие диски (дискеты). Рассмотрим форматирование дисков применительно к обычной, так называемой трехдюймовой дискете.

Форматирование физической структуры диска состоит в создании на диске концентрических дорожек, разделенных на секторы. Для этого магнитная головка дисковода записывает в определенных местах диска метки дорожек и секторов. После форматирования так называемой трехдюймовой дискеты она имеет следующие параметры: число дорожек на каждой из двух сторон — 80 (нумерация дорожек от нулевой до 79-й); число секторов на дорожке — 18; информационная емкость сектора — 512 байт.

Таким образом, на дискете минимальным адресуемым элементом является сектор. Если на дискету записан небольшой файл размером, например, 20 байт, то он занимает один сектор, и в этот сектор уже больше никакой информации записать нельзя, хотя в нем остается еще много свободного места. Записанный на дискету файл всегда занимает целое число секторов. При записи очередного файла используются свободные секторы, которые могут находиться на разных дорожках.

Общая информационная емкость трехдюймовой дискеты составляет 1 474 560 байт (см. подразд. 12.5). Но из 2 880 секторов

пользователю для записи доступны только 2 847. По окончании форматирования на экране появляется сообщение: 1 457 664 байт доступно на диске. Если часть секторов повреждена, то это число будет меньше.

Форматирование дискеты может быть полным или быстрым. Полное форматирование включает проверку качества магнитного покрытия дискеты и разметку дискеты на дорожки и секторы. После полного форматирования вся хранившаяся на дискете информация уничтожается. При быстром форматировании производится лишь очистка корневого каталога и таблицы размещения файлов. Сами файлы сохраняются (пока на дискету не записаны другие файлы), и в принципе их можно восстановить с помощью специальной программы. При форматировании можно присвоить дискете название, которое вводится в поле «Метка». Это поможет пользователю в дальнейшей работе, если он применяет много дискет. Для защиты информации на дискете от несанкционированного доступа можно задавать нестандартные параметры форматирования, например иное количество дорожек или секторов.

Логическая структура жестких дисков (винчестеров) несколько отличается от логической структуры гибких дисков. Минимальным адресуемым элементом жесткого диска является кластер, который может включать в себя несколько секторов. Применяемая обычно таблица размещения файлов FAT16 позволяет присвоить адреса 65 536 кластерам. Файлу всегда выделяется целое число кластеров. Если жесткий диск имеет большую информационную емкость, то и размер кластера оказывается очень большим. Например, для 4-гигабайтного диска размер кластера будет равен 64 Кбайт. Следовательно, далеко не все информационные возможности жесткого диска будут полезно использованы. Эта проблема частично решается с помощью таблицы FAT32, в которой объем кластера принят равным восьми секторам, или 4 Кбайт, для диска любого объема.

Преобразование FAT16 в FAT32 можно выполнить с помощью служебной программы «Преобразование», которая входит в состав Windows.

При работе на компьютере приходится удалять одни файлы и записывать другие. Новые файлы, как уже говорилось, записываются в свободные кластеры, которые могут оказаться не соседними, а весьма удаленными друг от друга. Это приводит к замедлению обмена данными, скорость считывания уменьшается. Размещение фрагментов файла в удаленных друг от друга частях диска называется *фрагментацией*. Рекомендуется время от времени проводить *дефрагментацию* диска, в процессе которой файлы переписываются в соседние кластеры, последовательно расположенные друг за другом.

19.4. Архиваторы

При эксплуатации компьютера возможна порча или потеря информации на магнитных дисках из-за физической порчи магнитного диска, неправильной корректировки или случайного уничтожения файлов, разрушения информации компьютерным вирусом и целого ряда других причин. Для того чтобы уменьшить потери в таких ситуациях, следует иметь архивные копии используемых файлов и систематически обновлять их в случае изменения файлов.

Для копирования файлов можно использовать команды COPY или XCOPY, а также BACKUP и RESTORE. Однако при этом архивные копии будут занимать столько же места, сколько и исходные файлы, и для копирования нужных файлов может потребоваться много дискет. Например, для копирования файлов с жесткого диска емкостью 20 Мбайт необходимо не менее 14 дискет емкостью по 1,44 Мбайт. В таком большом количестве дискет разобратся довольно сложно, следовательно, трудоемкость создания и обновления архива будет весьма значительной. Поэтому для создания архивов употребляются специализированные программы двух классов — программы-упаковщики (архиваторы) и программы резервного копирования.

Программы-упаковщики позволяют за счет применения специальных методов сжатия информации создавать копии файлов меньшего размера и объединять копии нескольких файлов в один архивный файл. В большинстве случаев значительно удобнее хранить на дискетах, а иногда и на кассетах для стримера (ленточного ЗУ), файлы, предварительно сжатые программами-упаковщиками.

Программы резервного копирования предназначаются для копирования информации с жесткого диска на кассеты стримера или дискеты. Из этих программ широко используются Norton Backup (для DOS и Windows), FastBack Plus и др.

При наличии стримера создание резервных копий осуществляется с помощью программ резервного копирования. При этом, в принципе, можно обойтись без применения программ-упаковщиков, так как современные программы резервного копирования сами умеют достаточно эффективно сжимать данные. При переносе архивов на дискеты или магнитооптические диски, как правило, удобнее использовать программы-упаковщики.

Различными разработчиками были созданы специальные программы для упаковки файлов. Часть из них распространяется бесплатно, часть — на коммерческой основе (за плату). Большинство программ может быть продемонстрировано бесплатно, но если вы хотите их использовать постоянно, то должны выслать авторам или распространителям указанное вознаграждение.

Как правило, программы для упаковки (архивации) файлов позволяют помещать копии файлов на диске в сжатом виде в архивный файл, извлекать файлы из архива, просматривать оглавление архива и т. д. Разные программы отличаются форматом архивных файлов, скоростью работы, степенью сжатия файлов при помещении в архив, удобством использования.

Наиболее распространенные программы-упаковщики имеют приблизительно одинаковые возможности, и ни одна из них не превосходит другие по всем параметрам: одни программы работают быстрее, другие обеспечивают лучшую степень сжатия файлов. Даже если сравнивать программы только по степени сжатия, то среди них нет лидера: разные файлы лучше сжимаются разными программами.

Среди наиболее распространенных программ-упаковщиков можно назвать ARJ, PKZIP, LHA, PKPAK, PAK.

19.5. Вычисления, таблицы, графики

На ПК можно выполнять вычисления и оформлять их результаты в виде таблиц и графиков.

Электронные калькуляторы являются специализированными программными приложениями, предназначенными для вычислений. По своим функциональным возможностям они соответствуют аппаратным микрокалькуляторам.

Простые микрокалькуляторы широко используются в повседневной жизни и позволяют осуществлять только арифметические операции над числами. Инженерные микрокалькуляторы дают также возможность вычислять значения различных функций (\sin , \cos и др.) и используются в процессе обучения и для инженерных расчетов. Программистские микрокалькуляторы позволяют проводить вычисления в различных системах счисления и другие операции.

Электронные калькуляторы могут обладать возможностями всех вышеперечисленных типов аппаратных микрокалькуляторов.

Электронный калькулятор является стандартным приложением (под названием «Калькулятор») операционной системы-оболочки Windows. С помощью электронных калькуляторов можно производить арифметические действия над целыми и дробными числами, переводить числа из одной системы счисления в другую (например, из десятичной в двоичную), вычислять значения математических функций (например, x^2 , $1/x$), статистических функций (например, среднее арифметическое заданных чисел), финансовых функций (например, сумму банковского вклада с заданным процентом) и др.

Электронные калькуляторы позволяют делать сложные вычисления с записью промежуточных результатов в ячейки памяти

калькулятора. По мере необходимости такие результаты извлекают из памяти для возможности использования в последующих вычислениях.

Электронные калькуляторы дают возможность обмениваться числовыми данными с другими приложениями с использованием *буфера обмена* операционной системы.

Большие массивы числовых данных (например, результаты экспериментов, статистические данные и т.д.) очень удобно обрабатывать с помощью электронных таблиц (см. подразд. 16.3). Широкое распространение получили электронные таблицы Excel. Ячейка, с которой производятся какие-то действия, выделяется рамкой и называется *активной*. Электронные таблицы, с которыми работает пользователь, называются *рабочими листами*. Можно вводить и изменять данные одновременно на нескольких рабочих листах, а также выполнять вычисления на основе данных из нескольких листов. Документы электронных таблиц могут включать несколько рабочих листов и называются *рабочими книгами*.

В работе с электронными таблицами выделяют три основных типа данных — число, текст и формулу. В зависимости от решаемой задачи возникает необходимость применять различные форматы представления данных. В каждом конкретном случае важно выбрать наиболее подходящий формат.

Для представления чисел по умолчанию электронные таблицы используют *числовой формат*, который отображает два десятичных знака после запятой (например, 735,29).

Экспоненциальный формат применяется в тех случаях, когда число, содержащее большое количество разрядов, не умещается в ячейке (например, число 2 050 000 в экспоненциальном формате имеет вид 2,05E + 06).

По умолчанию числа выравниваются в ячейке по правому краю. При таком размещении чисел друг под другом в столбце таблицы происходит выравнивание по разрядам (единицы под единицами, десятки под десятками и т.д.), что очень удобно.

Текстом в электронных таблицах является последовательность символов, например запись «64 Кбайт» является текстовой. По умолчанию текст выравнивается в ячейке по левому краю.

Формула должна начинаться со знака равенства и может включать в себя числа, имена ячеек, функции и знаки математических операций. В формулу не должен входить текст.

Например, формула = A1 + C5 обеспечивает сложение чисел, хранящихся в ячейках A1 и C5, а формула = A2 · 7 — умножение числа, хранящегося в ячейке A2, на 7.

При вводе формулы в ячейке отображается не сама формула, а результат вычислений по этой формуле. При изменении исходных значений, входящих в формулу, этот результат пересчитывается автоматически.

19.6. Оформление документов

Основными объектами документа являются страница, абзац и символ. Для каждого из этих объектов требуется задать значения параметров, которые определяют внешний вид документа.

Любой документ состоит из страниц, поэтому в начале работы над документом необходимо задать значения параметров страницы (формат, ориентацию, размер полей и др.).

При создании реферата или заявления обычно выбирают формат страницы А4 (210×297 мм), который соответствует размеру стандартного листа бумаги для принтера. Для объявлений и плакатов можно выбрать формат А3, площадь которого в два раза больше, чем у А4.

Существуют две возможные ориентации страницы: книжная и альбомная. Для обычных текстов чаще всего используется книжная ориентация, а для широких таблиц с большим числом столбцов — альбомная.

Параметры страницы задаются следующим образом.

После ввода команды *Файл — Параметры страницы* появляется диалоговая панель *Параметры страницы* с четырьмя вкладками: *Поля*, *Размер бумаги*, *Источник бумаги*, *Макет*. На вкладке *Размер бумаги* с помощью раскрывающегося списка *Размер бумаги* необходимо выбрать используемый формат (например, А4), а также ориентацию страницы с помощью переключателя *Ориентация* (книжная или альбомная).

С помощью вкладки *Поля* на странице задают требуемые размеры полей (верхнего, нижнего, правого и левого), которые определяют расстояние от краев страницы до границы текста. Для вывода на каждой странице документа одинакового текста (например, названия документа) удобно использовать верхний или нижний колонтитул и задать расстояние от края страницы до колонтитула.

Страницы документа можно нумеровать, причем номера размещать по-разному (вверху или внизу страницы, по центру, справа или слева). Для этого вводится команда *Вставка — Номера страниц* и задаются местоположение номера и его формат.

Абзац — это часть текста, представляющая собой законченный по смыслу фрагмент произведения, окончание которого служит естественной паузой для перехода к новой мысли. В компьютерных документах абзацем считается любой текст, заканчивающийся управляющим символом (маркером) конца абзаца. В конце абзаца надо нажать на клавишу Enter и на экране появится специальный символ, если включен режим отображения непечатаемых символов. Для включения этого режима следует ввести команду *Сервис — Параметры* и на вкладке *Вид* диалоговой панели *Параметры* установить флажок *символы абзацев*. Абзац может со-

стоять из любого набора символов, рисунков и объектов других приложений.

Выравниванием абзацев достигается требуемое расположение текста относительно границ полей страницы. Чаще всего используют следующие способы выравнивания абзацев:

- по левому краю — левый край ровный, а правый имеет неровные очертания;
- центру — оба края имеют неровные очертания, но каждая строка абзаца симметрична относительно середины;
- правому краю — правый край ровный, а левый имеет неровные очертания;
- ширине — оба края ровные, т. е. располагаются точно по границам страницы. В этом случае последняя строка абзаца ведет себя как при выравнивании по левому краю.

Чаще всего абзац начинается отступом первой строки, т. е. с красной строки. Отступ может быть различных типов. Обычно применяется положительный отступ, когда первая строка начинается правее всех остальных строк абзаца. Отрицательный отступ (выступ), когда первая строка выходит влево относительно остальных строк абзаца, применяется в словарях и определениях. Нулевой отступ применяется для абзацев, выровненных по центру, и иногда для обычного текста.

Расстояние между строками документа можно изменять, задавая различные междустрочные интервалы (одинарный, полуторный, двойной и т. д.). Для более наглядного выделения абзацев можно устанавливать увеличенные интервалы между ними. Междустрочный интервал выбирают с помощью появляющегося списка *Междустрочный*, а интервал перед (после) абзаца — с помощью счетчиков *Интервал перед* и *Интервал после*.

Вкладка *Положение на странице* позволяет установить требуемое распределение абзацев по страницам. Например, можно запретить разрывать абзац между страницами, оставлять на странице первую или последнюю (висячую) строку и т. д. Часто бывает полезно запретить в абзаце автоматический перенос слов.

Для размещения в документе различных перечней применяют списки нескольких типов — нумерованные (элементы списка обозначаются арабскими или римскими числами либо буквами), маркированные (элементы списка отмечаются с помощью специальных символов-маркеров) и др.

В создаваемый документ можно вставлять таблицы. Таблица является объектом, состоящим из строк и столбцов, на пересечении которых образуются ячейки. В ячейках таблиц могут быть размещены различные данные (текст, числа и изображения). С помощью таблиц можно форматировать документы, например располагать абзацы в несколько рядов, совмещать рисунок с текстовой подписью и т. д. При размещении в таблице чисел можно произ-

водить над ними вычисления по формулам (суммирование, умножение, поиск максимального и минимального чисел и др.).

Преобразовать имеющийся текст в таблицу можно с помощью команды *Таблица — Преобразовать в таблицу*, однако часто бывает удобнее сначала создать таблицу и лишь затем заполнить ее данными. Для вставки таблицы в документ используют команду *Таблица — Добавить таблицу* и задают на панели *Вставка таблицы* в соответствующих полях ввода число строк и столбцов вставляемой таблицы.

Символы являются теми основными объектами, из которых состоит документ. К ним относятся буквы, цифры, пробелы, знаки пунктуации, специальные символы, такие как @, %, \$, &.

Среди основных свойств символов можно выделить шрифт, размер, начертание и цвет. Шрифт — это полный набор символов определенного начертания, включая прописные и строчные буквы, знаки препинания, специальные символы, цифры и знаки арифметических действий. Каждый шрифт имеет свое название, например Times New Roman, Arial, Courier и др. Полезно ознакомиться с набором шрифтов, установленных на компьютере. Для этого необходимо ввести команду *Сервис — Настройка — Команды — Шрифты*.

Единицей измерения размера шрифта является пункт (1 пт = = 0,376 мм). Размеры шрифтов можно изменять в широких пределах (обычно от 1 до 1638 пунктов), причем в большинстве текстовых редакторов по умолчанию используется шрифт размером 10 пт. Ниже приведены примеры представления текста при различных размерах шрифта.

Шрифт размером 16 пт.

Шрифт размером 12 пт.

Шрифт размером 8 пт.

Кроме нормального начертания символов при необходимости применяют **полужирное**, *курсивное*, **полужирное курсивное**. Можно установить дополнительные параметры форматирования символов: подчеркивание символов различными типами линий; изменение вида символов (^{верхний индекс}, _{нижний индекс}, ~~зачеркнутый~~); изменение расстояния между символами (разреженный, уплотненный) и др. Если планируется многоцветная печать документа, то для различных групп символов можно задать различные цвета, выбранные из предлагаемой текстовым редактором палитры.

Для форматирования шрифта вводят команду *Формат — Шрифт*, которая открывает диалоговую панель *Шрифт*. На данной панели с помощью раскрывающихся списков выбирают шрифт, размер, начертание, цвет символов, варианты подчеркивания.

19.7. Работа в сети Интернет

Для работы в сети Интернет необходимо подключить к компьютеру модем и заключить с провайдером договор о предоставлении доступа к сети. Если модем внешний, надо подсоединить его к одному из COM-портов компьютера, а если внутренний — вставить в слот шины расширения компьютера. После включения компьютера система-оболочка Windows определит, что подсоединено новое устройство. На экране монитора появятся указания о дальнейших действиях. Если от Windows нет сообщения о том, что обнаружено новое устройство, то необходимо выполнить такую последовательность команд: *Пуск — Настройка — Панель управления — Модемы — Добавить — Установить новый модем*. Далее надо следовать указаниям Windows и производителя модема.

Теперь компьютер может быть подключен с помощью модема к телефонной сети. После того как провайдер сообщил пользователю идентификатор, пароль и другую необходимую информацию, выполняется настройка программного обеспечения.

Рассмотрим процесс настройки соединения с сетью Интернет с использованием специальной программы «Удаленный доступ к сети», которая входит в состав системы-оболочки Windows.

Создание соединения с сетью Интернет выполняется в такой последовательности.

1. Ввести команду *Стандартные — Связь — Удаленный доступ к сети — Новое соединение*. На появившейся диалоговой панели *Новое соединение* ввести название соединения (например, *Соединение1*) и выбрать из списка подключенный модем.

2. Настроить параметры подключения и режимы работы модема, для чего щелкнуть кнопкой *Настройка*. Появится диалоговая панель *Свойства* с тремя вкладками (*Общие, Подключение, Параметры*). На вкладке *Общие* в списке *Порт* выбрать порт, к которому подключен модем (обычно *Параллельный порт COM2*), и в списке *Наибольшая скорость* выбрать скорость передачи информации от порта компьютера к модему.

При использовании модемов стандарта V.34 и выше рекомендуется выбирать максимальную скорость передачи 115 200 бит/с. На вкладках *Подключение* и *Параметры* рекомендуется оставлять те значения, которые предлагаются по умолчанию. После настройки параметров подключения и работы модема возвратиться к диалоговой панели *Новое соединение* и щелкнуть кнопкой *Далее*.

3. На появившейся следующей диалоговой панели ввести в поле *Телефон* номер телефона провайдера, а также выбрать нужные пункты в списках *Код города* и *Код страны*.

Щелкнуть кнопкой *Далее*. На появившейся следующей диалоговой панели щелкнуть кнопкой *Готово* для подтверждения создания соединения *Соединение1*.

Для дальнейшей настройки соединения *Соединение 1* вызвать его контекстное меню и ввести команду *Свойства*. Появится диалоговая панель *Соединение 1* с четырьмя вкладками (*Общие, Тип сервера, Сценарии, Подключения*).

Вкладка *Общие* содержит установки, которые были введены выше при создании соединения *Соединение 1*, поэтому в них ничего изменять не надо. Содержимое вкладок *Сценарии* и *Подключения* также можно не изменять.

4. На вкладке *Тип сервера* из списка *Тип сервера удаленного доступа* выбрать *PPP*, например *Интернет, Windows NT Server* или *Windows 98*.

В группе флажков *Допустимые сетевые протоколы* установить флажок *TCP/IP*, а в группе флажков *Дополнительные параметры* флажки можно не устанавливать.

5. Для настройки протокола *TCP/IP* щелкнуть кнопкой *Настройка TCP/IP*, появится диалоговая панель *Настройка TCP/IP*. Обычно провайдер присваивает компьютеру переменный IP-адрес, который может меняться при очередном подключении к сети, поэтому верхний переключатель установить в положение *Адрес IP назначается сервером*.

Функционирование доменной системы имен в сети Интернет обеспечивается специальными серверами DNS, которые в целях надежности дублируются. Поэтому первичный и вторичный адреса DNS сообщаются пользователю провайдером и вводятся вручную по команде *Указать адреса сервера имен* (например, 195.210.128.3 и 212.248.0.1).

Теперь удаленное соединение *Соединение 1* для подключения к сети Интернет полностью настроено. Для запуска настроенного соединения ввести команду *Стандартные — Связь — Удаленный доступ к сети — Соединение 1*. Появится диалоговая панель *Установка связи*. В текстовое поле *Имя пользователя* ввести идентификатор, а в поле *Пароль* — пароль (обычно сочетание букв и цифр, которое присваивает пользователю провайдер). Затем щелкнуть кнопкой *Подключиться*.

Модем произведет набор указанного телефонного номера (динамик компьютера будет издавать звуки во время набора), и, если номер провайдера свободен, его модем «снимет трубку». В целях определения максимального высокоскоростного соединения модемы в течение нескольких секунд будут обмениваться сигналами (будет слышен звуковой шум различных частот).

Если модемам удастся установить связь и идентификатор и пароль введены правильно, то через несколько секунд произойдет подключение к сети Интернет. Последовательно появятся сообщения *Проверка имени пользователя и пароля*, а затем *Вход в сеть*. Можно открыть информационное окно, которое содержит информацию о подключении (скорость, длительность и др.).

Наиболее популярными браузерами, т.е. программами для работы в сети Интернет, являются Netscape Navigator и Microsoft Internet Explorer (встроена в ОС Windows). Можно установить обе эти программы. Они поддерживают также отправку и получение почты.

Контрольные вопросы

1. Что происходит при загрузке компьютера?
2. Что такое файл?
3. Что такое папка?
4. Какие диски используются в компьютере?
5. Как обратиться к нужному диску?
6. Зачем нужны архиваторы?
7. Как выполняются простые расчеты на компьютере?
8. Как оформить на компьютере контрольную работу?
9. Как войти в сеть Интернет?

ЗАКЛЮЧЕНИЕ

Роль вычислительной техники в наше время трудно переоценить. В конце второго тысячелетия для каждого цивилизованного человека было необходимо уметь читать, писать и считать. Без этих знаний невозможно не только выполнять свои профессиональные обязанности, но и просто существовать в современном мире, организовывать свой быт. В настоящее время грамотным человеком может считаться лишь тот, кто имеет хотя бы элементарные навыки владения персональным компьютером и разнообразными бытовыми приборами с микропроцессорным управлением.

Применение вычислительной техники многократно увеличивает производительность труда. Особенно важно знать и уметь использовать компьютерные технологии специалисту в области автоматизации и управления. В этой книге даны лишь основы знаний в области вычислительной техники. Эта отрасль бурно развивается. С каждым днем появляются все более совершенные устройства, новые программные средства. Поэтому современный специалист должен постоянно следить за обновлением аппаратного и программного обеспечения. Нельзя останавливаться в своем развитии даже на короткое время. Это относится к любой области деятельности человека, но особенно касается тех, кто свои профессиональные успехи связывает с вычислительной техникой.

Для того чтобы лучше знать вычислительную технику, надо прежде всего постоянно пользоваться ею. Кроме того, она сама является прекрасным средством обучения. В сети Интернет с помощью поисковых систем можно найти ответ на любой вопрос. Существуют разнообразные веб-страницы, где ежедневно появляются новости по элементам вычислительной техники, компьютерным сетям, информационно-коммуникационным технологиям. Пусть расширение знаний о компьютере с помощью компьютера станет вашей привычкой.

СПИСОК ЛИТЕРАТУРЫ

1. *Браммер Ю. А.* Импульсные и цифровые устройства / Ю. А. Браммер, И. Н. Пашук. — М. : Высш. шк., 2002. — 351 с.
2. *Голицына О. Л.* Базы данных / О. Л. Голицына, Н. В. Максимов, И. И. Попов. — М. : Форум, 2004. — 352 с.
3. *Гребенюк Е. И.* Технические средства информатизации / Е. И. Гребенюк, Н. А. Гребенюк. — 2-е изд. — М. : Изд. центр «Академия», 2005. — 272 с.
4. *Громов Г. Р.* Очерки информационной технологии / Г. Р. Громов. — М. : ИнфоАрт, 1993. — 336 с.
5. *Калабеков Б. А.* Цифровые устройства и микропроцессорные системы / Б. А. Калабеков. — М. : Горячая линия — Телеком, 2007. — 336 с.
6. *Калиш Г. Г.* Основы вычислительной техники / Г. Г. Калиш. — М. : Высш. шк., 2000. — 271 с.
7. *Келим Ю. М.* Типовые элементы систем автоматического управления / Ю. М. Келим. — М. : Форум, 2004. — 384 с.
8. *Кузин А. В.* Микропроцессорная техника / А. В. Кузин, М. А. Жаворонков. — 2-е изд. — М. : Изд. центр «Академия», 2006. — 304 с.
9. *Михеева Е. В.* Информационные технологии в профессиональной деятельности / Е. В. Михеева. — 6-е изд. — М. : Изд. центр «Академия», 2007. — 384 с.
10. *Мышляева И. М.* Цифровая схемотехника / И. М. Мышляева. — М. : Изд. центр «Академия», 2004. — 400 с.
11. *Партыка Т. Л.* Операционные системы, среды и оболочки / Т. Л. Партыка, И. И. Попов. — М. : Форум : Инфра-М, 2003. — 400 с.
12. *Попов И. И.* Компьютерные сети / И. И. Попов, Н. В. Максимов. — М. : Форум, 2004. — 336 с.
13. *Семакин И. Г.* Основы программирования / И. Г. Семакин, А. П. Шестаков. — 5-е изд. — М. : Изд. центр «Академия», 2006. — 432 с.
14. *Фуфаев Э. В.* Пакеты прикладных программ / Э. В. Фуфаев, Л. И. Фуфаева. — 2-е изд. — М. : Изд. центр «Академия», 2006. — 352 с.
15. *Хорев П. Б.* Технология объектно-ориентированного программирования / П. Б. Хорев. — М. : Изд. центр «Академия», 2004. — 400 с.

ОГЛАВЛЕНИЕ

Предисловие	3
Введение	5

РАЗДЕЛ I ОСНОВНЫЕ СВЕДЕНИЯ ОБ ЭЛЕКТРОННОЙ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ

Глава 1. Назначение и развитие вычислительной техники	9
1.1. Краткий исторический очерк развития вычислительной техники	9
1.2. Области применения вычислительной техники	24
1.3. Перспективы информатизации общества	29
Глава 2. Характеристики и классификация вычислительной техники	34
2.1. Основные характеристики ЭВМ	34
2.2. Классификация ЭВМ	38
2.3. Развитие производства ЭВМ	45
Глава 3. Принцип действия ЭВМ	52
3.1. Функциональная схема ЭВМ	52
3.2. Основные узлы ЭВМ	54
3.3. Принцип открытой архитектуры	60
3.4. Понятие о программном обеспечении	62
Глава 4. Способы представления информации в ЭВМ	67
4.1. Виды информации	67
4.2. Количественные характеристики информации	69
4.3. Достоинства дискретного сигнала	74

РАЗДЕЛ II ОСНОВЫ РАБОТЫ ЭВМ

Глава 5. Математические основы работы ЭВМ	76
5.1. Системы счисления	76
5.2. Перевод чисел из одной системы счисления в другую	78
5.3. Правила десятичной арифметики	82
5.4. Способы представления чисел в разрядной сетке ЭВМ	87
Глава 6. Логические основы работы ЭВМ	91
6.1. Элементарные логические функции	91

6.2. Формы представления логических функций	94
6.3. Законы алгебры логики	99
6.4. Основной базис алгебры логики	101
6.5. Минимизация логических функций	105

РАЗДЕЛ III

ЭЛЕМЕНТЫ И УСТРОЙСТВА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Глава 7. Типовые элементы вычислительной техники	109
7.1. Назначение типовых элементов	109
7.2. Основные логические элементы	111
7.3. Триггеры	114
7.4. Регистры	117
7.5. Счетчики	121
7.6. Сумматоры	124
7.7. Кодированные и декодирующие устройства	128
7.8. Компараторы	142
7.9. Типы коммутаторов. Мультиплексоры и демультиплексоры	146
7.10. Типовые элементы аналоговых вычислительных машин	149
Глава 8. Основы микропроцессорных систем	153
8.1. Назначение процессоров и микропроцессоров	153
8.2. Архитектура и структура микропроцессора	155
8.3. Характеристики и классификация процессоров и микропроцессоров	162
Глава 9. Арифметико-логические устройства процессора	165
9.1. Назначение и состав арифметико-логических устройств	165
9.2. Комбинационные схемы	169
9.3. Конечные автоматы	172
9.4. Работа арифметико-логического устройства	175
Глава 10. Управление процессом обработки информации	182
10.1. Назначение устройства управления	182
10.2. Аппаратное управление	183
10.3. Программное управление	187
10.4. Алгоритм управления	190
Глава 11. Работа микропроцессора	195
11.1. Система команд микропроцессора	195
11.2. Процедура выполнения команд	200
11.3. Система прерывания	205
11.4. Понятие о состоянии процессора	208
11.5. Микроконтроллеры	211
Глава 12. Запоминающие устройства	215
12.1. Виды и характеристики запоминающих устройств	215
12.2. Оперативные запоминающие устройства	217

12.3. Принцип магнитной записи	220
12.4. Магнитная лента	226
12.5. Гибкие диски	228
12.6. Жесткие диски	229
12.7. Магнитооптические и оптические диски	232
12.8. Полупроводниковые энергонезависимые запоминающие устройства	236
Глава 13. Организация интерфейсов в вычислительной технике	241
13.1. Назначение и характеристики интерфейса	241
13.2. Параллельный интерфейс	244
13.3. Последовательный интерфейс	247
13.4. Интерфейсы современных персональных компьютеров	250
Глава 14. Периферийные устройства вычислительной техники	254
14.1. Основные типы устройств ввода — вывода	254
14.2. Печатающие устройства	261
14.3. Устройства отображения информации	266
РАЗДЕЛ IV	
ПРОГРАММНЫЕ СРЕДСТВА	
Глава 15. Операционные системы	273
15.1. Назначение, функции и состав операционных систем	273
15.2. Операционная система MS-DOS	277
15.3. Операционная система-оболочка Windows	278
15.4. Многопользовательская операционная система UNIX	281
Глава 16. Программное обеспечение	285
16.1. Назначение и виды программного обеспечения	285
16.2. Текстовые редакторы	287
16.3. Электронные таблицы	289
16.4. Базы данных	293
16.5. Системы автоматизированного проектирования	296
16.6. Антивирусные программы	302
Глава 17. Основы программирования	306
17.1. Принципы программирования	306
17.2. Программирование на машинном языке	309
17.3. Программирование на языке ассемблер	310
17.4. Программирование на языках высокого уровня	313
17.5. Автоматизация программирования	319
Глава 18. Основы построения компьютерных сетей	324
18.1. Предпосылки к созданию сетей	324
18.2. Локальные сети	329
18.3. Сеть Интернет	331

18.4. Информационные технологии	336
18.5. Коммуникационные возможности компьютеров	341
Глава 19. Работа на персональном компьютере	344
19.1. Загрузка компьютера	344
19.2. Файлы, каталоги, папки	345
19.3. Работа с дисками	347
19.4. Архиваторы	349
19.5. Вычисления, таблицы, графики	350
19.6. Оформление документов	352
19.7. Работа в сети Интернет	355
Заключение	358
Список литературы	359

Учебное издание

Келим Юрий Михайлович
Вычислительная техника
Учебник

9-е издание, стереотипное

Редактор *Г. В. Первов*
Технический редактор *Е. Ф. Коржуева*
Компьютерная верстка: *А. А. Ратникова*
Корректоры *Н. Т. Захарова, Н. Л. Котелина*

Изд. № 109107854. Подписано в печать 01.07.2014. Формат 60×90/16.
Бумага офсетная № 1. Гарнитура «Таймс». Усл. печ. л. 23,0.
Тираж 1 000 экз. Заказ № 3934

ООО «Издательский центр «Академия». www.academia-moscow.ru
129085, Москва, пр-т Мира, 101В, стр. 1.
Тел./факс: (495) 648-0507, 616-00-29.

Санитарно-эпидемиологическое заключение № РОСС RU. АЕ51. Н 16592 от 29.04.2014.

Отпечатано способом ролевой струйной печати

в ОАО «Первая Образцовая типография»

Филиал «Чеховский Печатный Двор»

142300, Московская область, г. Чехов, ул. Полиграфистов, д. 1

Сайт: www.chpd.ru, E-mail: sales@chpd.ru, т/ф. 8(496)726-54-10