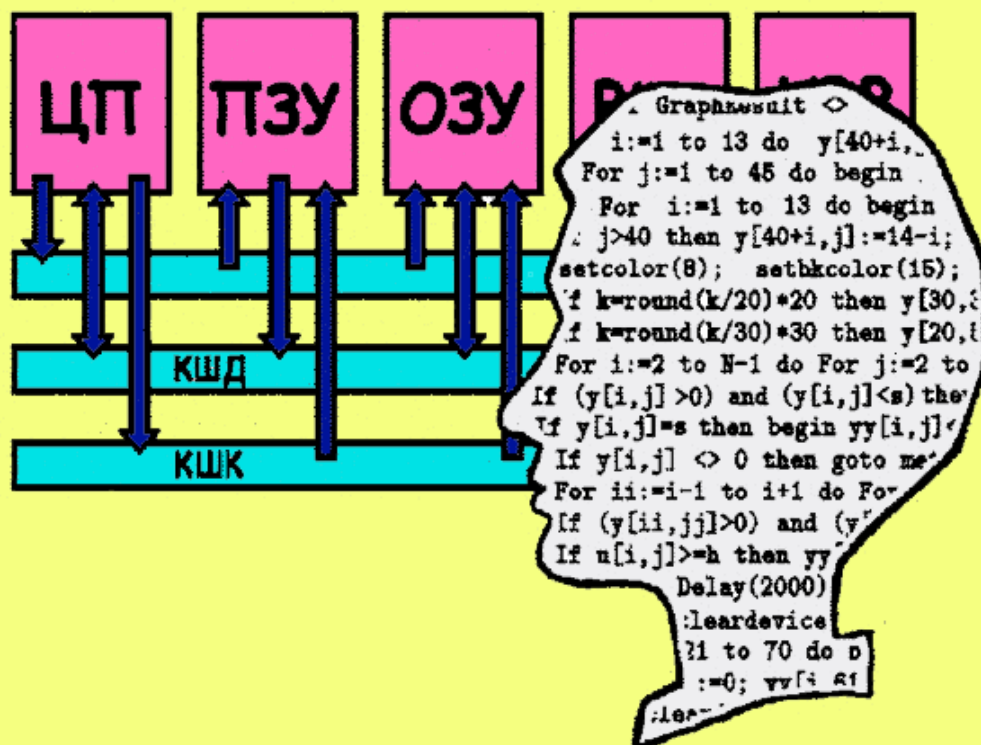


МАЙЕР Р.В.

КАК СТАТЬ КОМПЬЮТЕРНЫМ ГЕНИЕМ ИЛИ КНИГА О ИНФОРМАЦИОННЫХ СИСТЕМАХ И ТЕХНОЛОГИЯХ



Федеральное агентство по образованию РФ
ГОУ ВПО "Глазовский государственный педагогический институт
имени В. Г. Короленко"

Майер Р.В.

**КАК СТАТЬ
КОМПЬЮТЕРНЫМ ГЕНИЕМ
ИЛИ КНИГА
О ИНФОРМАЦИОННЫХ СИСТЕМАХ
И ТЕХНОЛОГИЯХ**

Глазов 2008

УДК 37.022: 681.3

ББК 32.81+32.97

М14

Печатается по решению кафедры информационных технологий в физическом образовании Глазовского государственного педагогического института имени В. Г. Короленко.

Автор: Майер Роберт Валерьевич.

Рецензенты:

В. А. Саранин, доктор физико-математических наук, профессор кафедры физики и дидактики физики ГОУ ВПО "Глазовский государственный педагогический институт".

А. Г. Русских, кандидат технических наук, доцент, заведующий кафедрой автоматизированных систем управления Глазовского инженерно-экономического института (филиала) ГОУ ВПО "Ижевский государственный технический университет".

Майер Р.В. Как стать компьютерным гением или книга о информационных системах и технологиях. — Глазов: ГГПИ, 2008. — 203 с. — ISBN 978-5-93008-111-4

В книге в доступной форме излагаются основные принципы работы современных информационных систем и сущность информационно-коммуникационных технологий. Рассматриваются основы аналоговой и цифровой электроники, принципы радио- и телевизионной связи, элементы теории информации и ее кодирования, теория алгоритмов. Описываются различные физические эксперименты с персональным компьютером, методы программирования на языке QBasic и некоторые другие вопросы, касающиеся сбора, хранения, обработки и передачи информации. Анализируются программы, написанные в средах Borland Pascal 7.0 и MS-DOS QBasic 1.0. Книга рассчитана на старших школьников и студентов, интересующихся информатикой и электроникой.

*Чтобы стать компьютерным гением,
Занимайся ЭВМ, а не пением.
Народная мудрость.*

ВСТУПЛЕНИЕ

21 век обещает стать веком информационных технологий. Наше общество приближается к такому состоянию, когда большая часть его граждан будут так или иначе связаны с компьютерной техникой. Компьютеры будут использоваться практически во всех сферах человеческой деятельности. При этом на смену постиндустриальному придет информационное общество, а Хомо Сапиенс превратится в Хомо Информатикус. Уже сейчас наиболее востребованы те специалисты, которые хорошо разбираются в информационно-коммуникационных технологиях, понимают основные принципы обработки информации, способны быстро обучаться и овладевать новыми знаниями в этой области. Компьютер, Интернет, искусственный интеллект, виртуальная реальность, нанотехнологии — вот что является перспективным и интересным для молодежи начала 21 века, которая уже через пару десятилетий определит направление научно-технического прогресса.

Книга "Как стать компьютерным гением" написана для тех молодых читателей, кто увлечен современной техникой и интересуется электроникой, информатикой, кибернетикой, современными системами связи. Она позволит найти ответ на некоторые существенные вопросы и поможет сделать важный шаг на пути к становлению специалиста в области компьютерной техники и коммуникационной технологии. Автор понимает претенциозность названия и осознает, что для того чтобы стать гением недостаточно прочитать эту книгу и даже выполнить все творческие задания, — для этого необходимо научиться самому ставить нетривиальные задачи и решать их. Однако он убежден, что эта и подобные ей книги помогают многим выбрать профессию, позволяют привлечь внимание читателя к важным проблемам информатики и подняться над среднестатистическим "пользователем ПК", а также создают предпосылки для изучения более сложной литературы по интересующим вопросам.

Хотя информатика — молодая наука, она уже имеет историю. Ее развитие всегда будет связано с именами таких выдающихся ученых, как Чарльз Бэббидж, Клод Шеннон, Джон Нейман, Норберт Винер и т.д. Среди физиков, математиков, кибернетиков, сделавших важные открытия и изобретения в области вычислительной техники и информационно-коммуникационных технологий есть и русские фамилии: Сергей Алексеевич Лебедев, Алексей Андреевич Ляпунов, Виктор Михайлович Глушков, Андрей Петрович Ершов и другие. Понимая тот значительный вклад, ко-

торый они внесли в развитие науки, следует помнить, что в самом начале своего жизненного пути они были обыкновенными людьми, которые в силу определенных причин заинтересовались математикой, физикой, техникой, сумели найти важную научную проблему и решить ее. Стремительное развитие науки и техники открывает новые горизонты, ставит новые задачи, дает более эффективные методы и требует новых специалистов, которые сумеют их решить.

Итак, что же все-таки нужно, чтобы стать "продвинутым" пользователем ПК, разобраться в современных информационно-коммуникационных технологиях? Во-первых, необходимо понять физические принципы сбора, обработки, хранения и передачи информации, овладеть основами электроники. Во-вторых, следует научиться программировать и с помощью компьютера управлять различными внешними устройствами. В-третьих, нужно освоить современные информационные технологии, работу с офисными приложениями (редакторами, электронными таблицами, базами данных), научиться использовать Интернет, электронную почту и т.д. И наконец, необходимо уметь ставить перед собой творческие задачи и решать их с помощью современных методов. Все это позволит Вам считать себя продвинутым в области информационных технологий и компьютерной техники и, если это важно, прослыть среди своих сверстников "компьютерным гением".

ЧТО ИЗУЧАЕТ ИНФОРМАТИКА

Мы живем в информационном океане. В мозг человека от различных органов чувств непрерывно поступают сигналы, нервные импульсы, позволяющие ему ориентироваться и адаптироваться в изменяющемся мире. Включаем ли мы телевизор, слушаем ли радио, разговариваем по телефону или общаемся посредством электронной почты, — мы практически непрерывно участвуем в информационном обмене с другими людьми и другими устройствами. Кто владеет информацией, тот правит миром, — так говорил один из мыслителей.

Наше общество скоро можно будет считать *информационным*, — уже сейчас большинство людей заняты производством, хранением, переработкой и реализацией информации. К критериям развитости информационного общества следует отнести наличие электронных средств связи, вычислительной техники, развитие рынка программного обеспечения, функционирование компьютерных сетей, а также уровень информационной культуры составляющих его граждан. Под *информационной культурой* понимают умение работать с информацией, используя для ее получения, обработки и передачи компьютерную технологию, современные технические средства и методы.

В истории человеческого общества произошло несколько *информационных революций*, каждая из которых представляла собой качественные изменения в методах обработки информации, приводящие к преобразо-

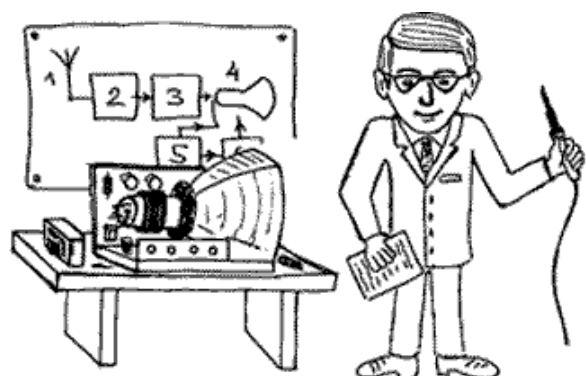
ванию общественных отношений. Перечислим основные вехи на пути к информационному обществу: 1) изобретение письменности, способствующее передаче знаний последующим поколениям; 2) изобретение книгопечатания, ускорившее развитие науки, техники и культуры; 3) изобретение телеграфа, телефона, радио, телевидения, что позволило оперативно передавать информацию; 4) изобретение микропроцессорной технологии и ЭВМ, вызвавшее появление компьютеров, компьютерных сетей, развитие информационно-коммуникационных технологий.

Информатика (information + automatique) — наука, изучающая общие свойства информации, информационные процессы, а также методы создания информационных систем и технологий. При этом под информацией понимают некоторые сведения об объекте познания, отражение одного объекта в другом, используемое для формирования управляющих сигналов. К *информационным процессам* относятся процессы создания новой информации, преобразования, уничтожения и передачи информации на расстояние, что осуществляется с помощью информационных систем, различных сетей и коммуникаций. Современные методы обработки информации, позволяющие создать новый информационный продукт, называются информационными технологиями. *Теоретическая информатика* — это фундаментальная наука, изучающая свойства информации, информационные процессы, методологию и теорию создания информационных систем и технологий. *Прикладная информатика* занимается разработкой информационно-коммуникационных систем, технических, алгоритмических и программных средств, используемых для обработки информации.

Информатика занимается решением следующих *задач*: 1) исследование информационных процессов любой природы; 2) разработка информационной техники и создание новейшей технологии переработки информации; 3) использование компьютерной техники и технологии во всех сферах общественной жизни. Перечислим *основные направления информатики как науки*: теория информации и ее кодирования, теория автоматов, теория алгоритмов, вычислительные системы, программирование, программное обеспечение, искусственный интеллект, кибернетика.

Информатика не сводится только к электронно-вычислительным машинам. Любой процесс обработки и передачи информации посредством проводной или беспроводной связи также является предметом изучения этой науки. Для того, чтобы разбираться в современных информационно-коммуникационных технологиях, необходимо понять физические принципы сбора, хранения, преобразования и передачи информации в различных электронных устройствах и системах.

Идея написания этой книги возникла в процессе работы на физическом факультете ГГПИ, руководимом профессором В. В. Майером. Автор выражает благодарность и другим преподавателям факультета: Е. С. Агафоновой, О. Е. Данилову, Р. В. Майеру (Акатову), А. В. Проказову, А. С. Рудину, В. А. Саранину, которые в той или иной степени способствовали пониманию сущности информационных процессов и написанию книги.



ГЛАВА 1 ОТ ТРАНЗИСТОРА ДО ТЕЛЕВИЗОРА

Наш мир — мир электронных приборов. Телевизор, радио, телефон, компьютер, видеомаягнитофон, DVD-плеер — вот далеко неполный список электронных устройств, подстерегающих нас на каждом шагу. Многие из них сочетают многофункциональность с малыми размерами, высокое качество работы и надежность с низкой стоимостью. Глядя на некоторые из них, начинаешь сомневаться в том, что они сделаны такими же людьми, как и мы с вами. Как же они работают? Чтобы ответить на этот вопрос, необходимо прежде всего познакомиться с элементной базой электронных устройств, применяющихся для обработки, хранения и передачи информации.

1.1. ЭЛЕМЕНТНАЯ БАЗА ЭВМ И ДРУГИХ ЭЛЕКТРОННЫХ УСТРОЙСТВ

Сначала были созданы электрические вычислительные машины релейного типа. С изобретением электронных ламп появились электронно-вычислительные машины первого поколения. Ко второму поколению ЭВМ относятся вычислительные машины на транзисторах. В ЭВМ третьего и четвертого поколения используются интегральные микросхемы.

1. Электромагнитное реле. Представим себе устройство, при подаче на управляющий вход которого постоянного напряжения происходит коммутация (замыкание и размыкание) соответствующих выводов. Таким устройством может быть *электромагнитное реле*, состоящее из обмотки, расположенной на магнитопроводе, якоря (подвижной части магнитопровода, замыкающей магнитный поток), пружины и системы контактов (рис. 1.1.1). Реле работает как электромагнитный ключ. При подаче на обмотку небольшого напряжения (например, 10 В) неподвижный магнитопровод притягивает якорь, тот растягивает пружину и замыкает или размыкает контакты. Эти контакты могут входить в другую цепь, которая будет замыкаться; при этом более мощная нагрузка (лампа) окажется подключенной к источнику напряжением 220 В. На базе реле можно создать электрическую цепь, способную находиться в двух устойчивых состояниях и хранить 1 бит информации.

2. Электронная лампа-триод. Электронная лампа-триод представляет собой вакууммированный стеклянный баллон с тремя электродами:

катодом, анодом и сеткой. Схема включения приведена на рис. 1.1.2. На анод подают положительный потенциал относительно катода (более 100 В). Катод разогревают с помощью нити накала, в результате термоэлектронной эмиссии вокруг него образуется электронное облако. Электроны, вылетевшие с поверхности разогретого катода, движутся по направлению к положительно заряженному аноду, проходя через сетку. Если на сетку подать отрицательный потенциал, то возникнет электрическое поле, препятствующее движению электронов к аноду, анодный ток уменьшится. Таким образом, изменяя сеточное напряжение, можно регулировать анодный ток, что приведет к соответствующим изменениям напряжения на нагрузке. Анодно-сеточная характеристика изображена на рис. 1.1.3. Если на сетку подать достаточно большой отрицательный потенциал $U = U_{отс}$, лампа-триод закроется, анодный ток будет равен нулю.

Лампа-триод использовалась в усилителях, генераторах, различных электронных ключах, триггерах, вычислительных машинах и т.д. Ее основные недостатки: большие размеры, необходимость разогрева катода нитью накала, высокое анодное напряжение.

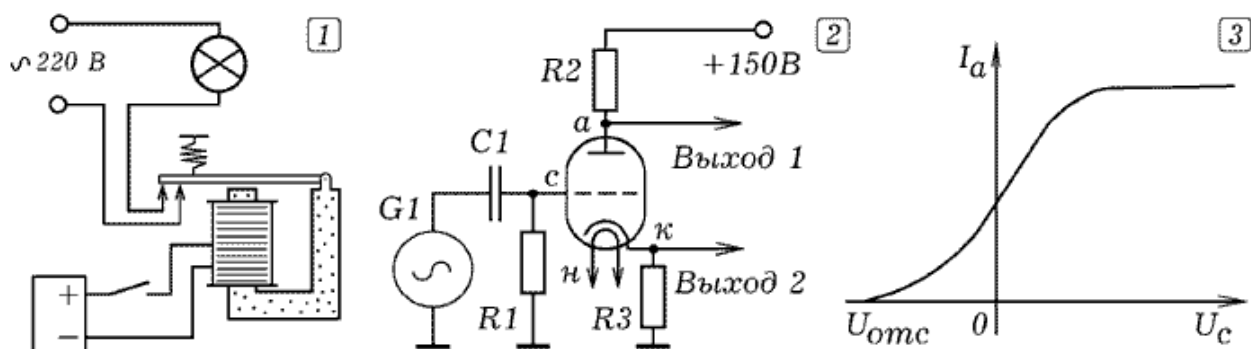


Рис. 1.1.

3. Полевой транзистор. Транзистор — полупроводниковый прибор с тремя выводами, у которого входное напряжение управляет током в выходной цепи. Полевой транзистор — униполярный полупроводниковый прибор, усилительные свойства которого обусловлены потоком основных носителей, протекающих через проводящий канал, управляемый электрическим полем. Он имеет три электрода: затвор, исток и сток. Из истока (аналог эмиттера) вытекают основные носители заряда, в сток (аналог коллектора) они втекают, затвор (аналог базы) — управляющий электрод.

Полевой транзистор с управляющим р-п-переходом и объемным каналом п-типа (рис. 1.2.1.) создается методом диффузии и напыления на кристалле Si п-типа (подложка). С противоположных сторон кристалла создаются р-п-переходы так, чтобы концентрация дырок в р-области существенно превышала концентрацию электронов в п-области. Для создания затвора, истока и стока соответствующие поверхности металлизированы. При подаче на затвор отрицательного потенциала дырки притягиваются к затвору, а электроны отталкиваются. В результате толщина запирающего слоя, обедненного основными носителями заряда, растет. Площадь поперечного сечения канала уменьшается, сопротивление исток-

сток возрастает, стоковый ток падает. Напряжение, при котором канал перекрывается полностью и ток стока падает до нуля, называется напряжением отсечки (рис.1.3.2). Если необходимо усилить переменный сигнал, то на затвор следует подать напряжение смещения, при котором транзистор будет полуоткрыт. Небольшие колебания потенциала приводят к соответствующим колебаниям стокового тока и напряжения на нагрузке.

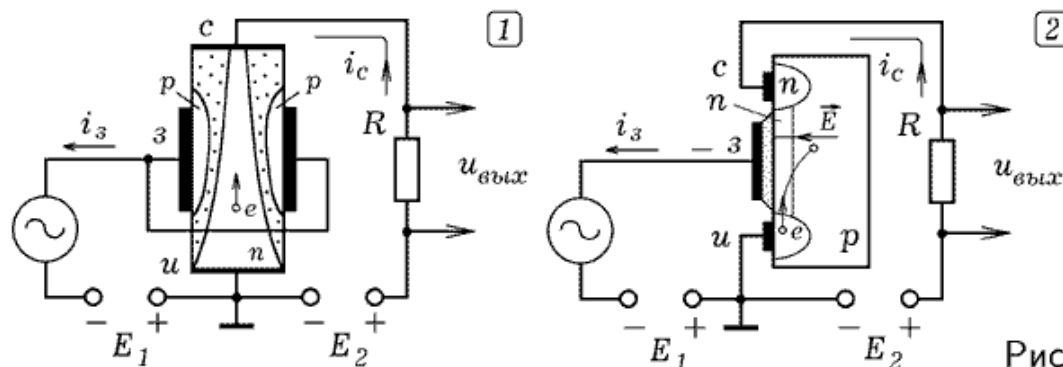


Рис. 1.2.

Транзистор с изолированным затвором типа МДП (металл–диэлектрик–полупроводник) или МОП (металл–оксид–полупроводник) имеет приповерхностный канал (рис.1.2.2). В слаболегированной пластине Si p-типа (подложка) созданы области n-типа: исток, сток и соединяющий их канал. На поверхность канала наносится слой оксида SiO_2 (или диэлектрика); его поверхности, а также поверхности областей n-типа металлизуются. При подаче на затвор положительного потенциала к поверхности оксида притягиваются электроны, проводимость канала увеличивается, ток стока растет. Если подать отрицательный потенциал, электроны, отталкиваясь от затвора, уходят в глубь кристалла кремния, где рекомбинируют с дырками. При этом сопротивление растет, ток стока падает.

МДП–транзисторы со встроенным каналом работают и при положительном и при отрицательном напряжении на затворе. Существуют также транзисторы с индуцированным каналом. Условные обозначения всех видов полевых транзисторов представлены на рис.1.3.1.

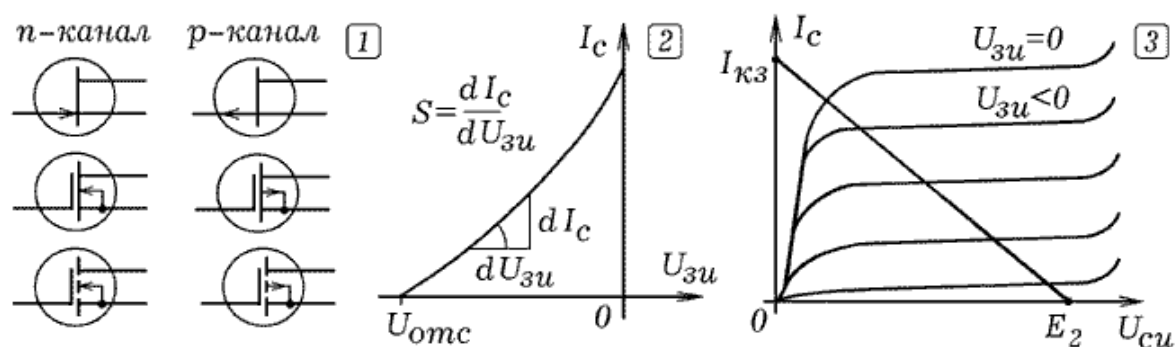


Рис. 1.3.

Стоковый ток является функцией напряжения на затворе и на стоке относительно истока: $I_c = f(U_{zu}, U_{cu})$. Семейство стоко-затворных (проходных) характеристик показывает зависимость тока стока от напряжения на затворе при постоянном напряжении на стоке $I_c = f(U_{zu})$ при $U_{cu} = const$. Каждая характеристика представляет собой возрастающую

кривую (рис. 1.3.2.) и характеризуется крутизной $S = dI_c/dU_{зи}$. Семейство выходных характеристик показывает зависимость $I_c = f(U_{си})$ при $U_{зи} = const$ (рис. 1.3.3). Горизонтальные участки графиков соответствуют режиму насыщения транзистора, в котором он полностью открыт.

Полевые транзисторы имеют высокое входное сопротивление: у транзисторов с управляющим р-п-переходом оно составляет 10^9 Ом, а у транзисторов с изолированным затвором типа МОП лежит в интервале $10^{12} - 10^{15}$ Ом. Они реагируют на статическое электричество, на их базе может быть создана оперативная память ЭВМ.

4. Биполярный транзистор. Работа биполярного транзистора обусловлена движением основных носителей заряда обоих знаков: отрицательных (электроны) и положительных (дырки). Транзистор имеет два р-п-перехода и три вывода: база — управляющий электрод, эмиттер — электрод, из которого вытекают основные носители заряда, и коллектор — электрод, в который втекают основные носители заряда. Чтобы изготовить транзистор прямой проводимости (р-п-р) берут пластину Si n-типа и методом диффузии создают на ее противоположных сторонах области типа р, соответствующие эмиттеру и коллектору (рис. 1.4.1). Для подключения выводов поверхности базы, эмиттера и коллектора металлизуют. Аналогично изготавливают транзистор обратной проводимости типа п-р-п. Структура транзисторов прямой и обратной проводимости изображена на рис. 1.4.3.

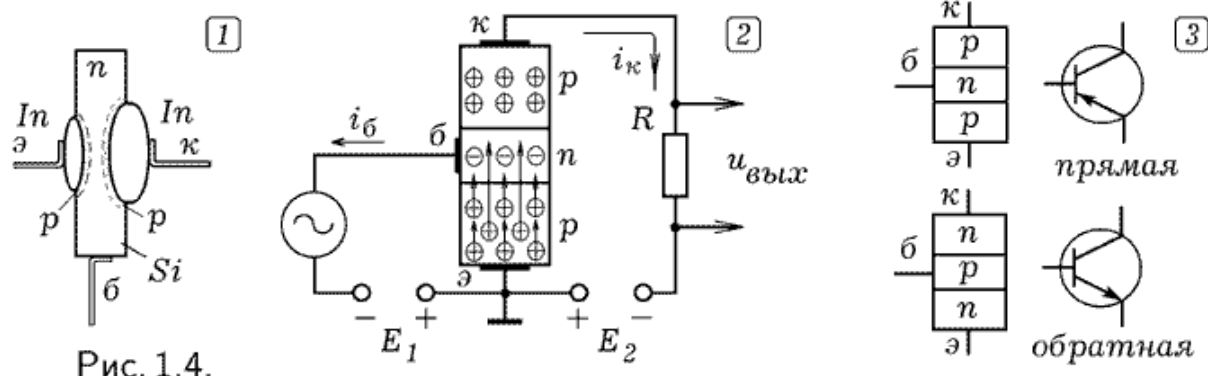


Рис. 1.4.

Схема включения транзистора типа р-п-р с общим эмиттером изображена на рис. 1.4.2. На коллектор подают отрицательный потенциал относительно эмиттера, р-п-переход база-эмиттер открыт, а р-п-переход база-коллектор закрыт, ток через нагрузку мал ($I_k = 0$). При подаче на базу отрицательного потенциала дырки из эмиттера переходят в базу, которая имеет небольшую толщину и невысокую концентрацию электронов. Небольшая часть дырок рекомбинирует с электронами базы, а остальные оказываются вблизи р-п-перехода база-коллектор. Сопротивление этого р-п-перехода падает, коллекторный ток растет. При периодическом изменении потенциала базы происходят соответствующие изменения тока коллектора и напряжения на нагрузке. Параметры цепи подбирают так, чтобы амплитуда колебаний выходного напряжения в десятки раз превышала амплитуду входного. Транзистор типа п-р-п работает аналогично, при его включении следует поменять полярность источников E_1 и E_2 .



ЛЕБЕДЕВ Сергей Алексеевич (1902–1974 гг.)

Занимался созданием энергетических систем в период электрификации страны (ГОЭЛРО). В институте электротехники АН УССР организовал лабораторию по разработке Малой электронно-счетной машины (МЭСМ), руководил созданием Больших электронных машин БЭСМ-1. Выполнял важные работы по созданию многомашиных и многопроцессорных комплексов.

Ток коллектора I_k зависит от напряжения на базе $U_{бэ}$ и на коллекторе $U_{кэ}$ относительно эмиттера и выражается функцией: $I_k = f(U_{бэ}, U_{кэ})$. Входная статическая характеристика $I_б = f(U_{бэ})$ при $U_{кэ} = const$ представляет собой вольт-амперную характеристику р-н-перехода база-эмиттер (рис. 1.5.1). Выходные статические характеристики $I_k = f(U_{кэ})$ при различных $U_{бэ} = const$ (рис. 1.5.2) — это семейство кривых с практически горизонтальными участками, соответствующими режиму насыщения, когда транзистор полностью открыт.

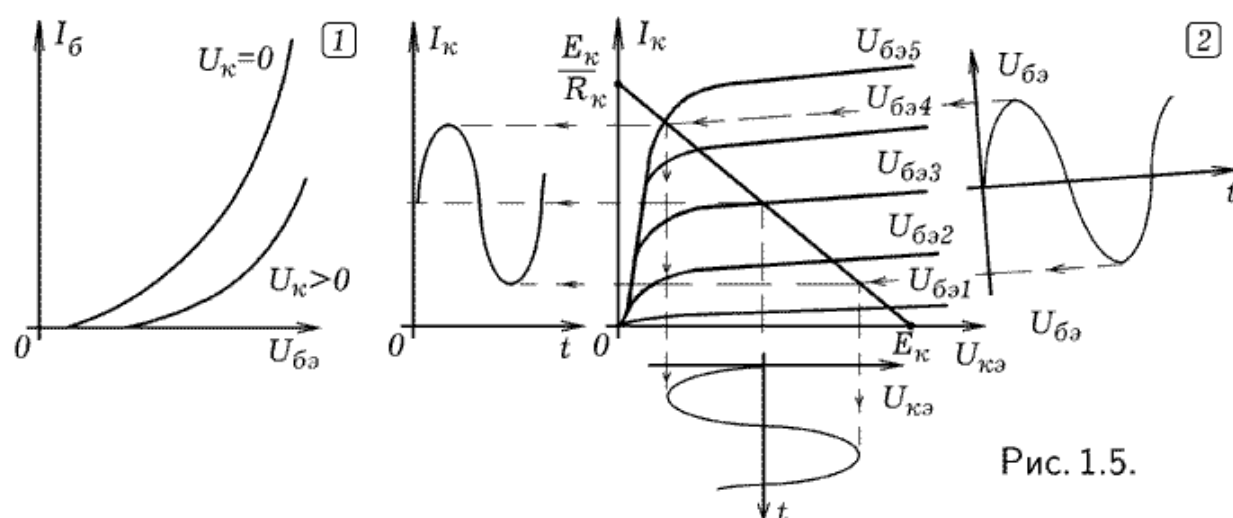


Рис. 1.5.

Чтобы представить работу транзистора в динамическом режиме, в семействе выходных статических характеристик строят нагрузочную прямую источника E_2 . Если напряжение $U_{бэ}$ изменяется, то рабочая точка А, характеризующая состояние транзистора, колеблется вдоль нагрузочной прямой, перескакивая с одной выходной характеристики на другую (рис. 1.5.2). В результате напряжение на коллекторе $U_{кэ}$ будет изменяться в противофазе с $U_{бэ}$.

5. Режимы работы транзистора. Рассмотрим проходную динамическую характеристику транзистора $I_k = f(U_{бэ})$ (рис. 1.6). Режим работы транзистора зависит от напряжения смещения $U_{см}$ — постоянной составляющей напряжения на базе, обуславливающей положение рабочей точки А на проходной характеристике. В режиме А ток в выходной цепи протекает в течение всего периода изменения $U_{бэ}$. При этом напряжение

смещения должно превышать амплитуду $U_{бэ}$ (рис. 1.6.1). Транзистор не вносит нелинейных искажений (форма выходного сигнала такая же как у входного), но он потребляет много энергии, так как полуоткрыт при отсутствии входного сигнала (в состоянии покоя).

В режиме В напряжение смещения на базе близко к нулю, поэтому ток в выходной цепи протекает только в течение половины периода входного сигнала $U_{бэ}$. Так как в состоянии покоя транзистор закрыт, он в потребляет меньше энергии. Форма выходного сигнала не совпадает с формой входного, имеют место нелинейные искажения. Промежуточным между режимами А и В является режим АВ, при котором ток в выходной цепи протекает в течение времени, большем половины периода $U_{бэ}$ (рис. 1.6.2). В режиме С ток в выходной цепи протекает в течение времени, меньшем половины периода входного сигнала $U_{бэ}$. В режиме D (импульсный режим) транзистор находится либо в состоянии *насыщения*, то есть полностью открыт, либо в состоянии *отсечки*, когда он полностью закрыт.

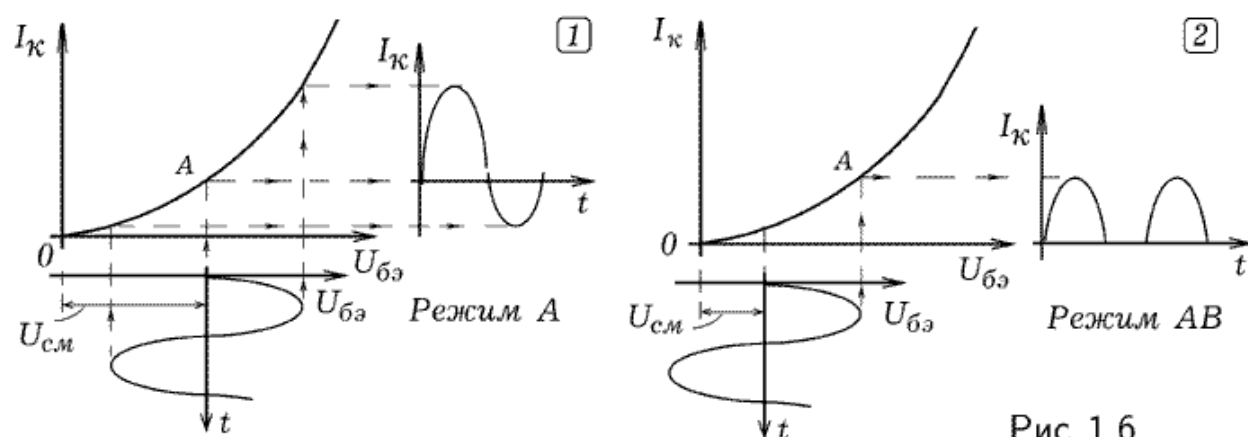


Рис. 1.6.

6. Светоизлучающие элементы. Важными элементами систем оптической связи являются светоизлучающие приборы, преобразующие импульсы напряжения в световые импульсы. К таким приборам относится лампа накаливания. Однако она имеет слишком большую инерционность и не позволяет осуществить быструю передачу информации по оптическому каналу связи. Поэтому сейчас в качестве светоизлучающих элементов используются светодиод и полупроводниковый лазер.

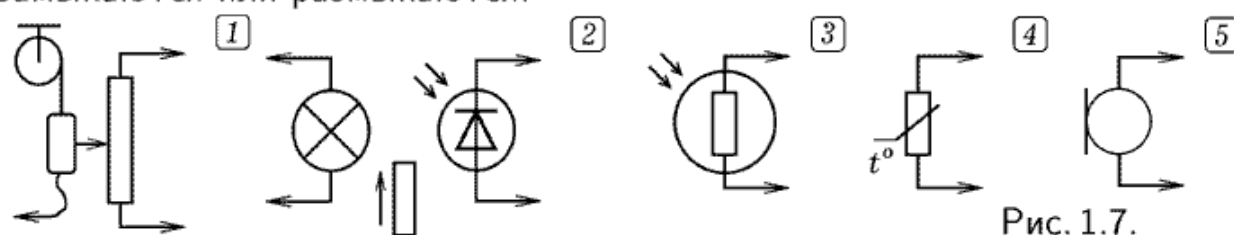
Светодиод — это полупроводниковый прибор с одним р-п-переходом, в котором при протекании тока генерируется излучение в инфракрасной, видимой или ультрафиолетовой областях спектра. В результате рекомбинации электронов проводимости и дырок вблизи р-п-перехода возникает монохроматическое излучение, спектр которого зависит от свойств полупроводника и количества примесей. **Полупроводниковый инжекционный лазер** состоит из р-п-перехода и оптического резонатора, состоящего из двух зеркал, в качестве которых используются отполированные грани кристалла полупроводника. При включении р-п-перехода в прямом направлении создается инверсная заселенность, приводящая к излучению света. Оптический резонатор позволяет получить свет с высокой монохроматичностью.

1.2. УСТРОЙСТВА, ВЫРАБАТЫВАЮЩИЕ И ПРЕОБРАЗУЮЩИЕ АНАЛОГОВЫЕ СИГНАЛЫ

Как работает информационная система? Датчик, реагируя на изменения измеряемой величины, вырабатывает аналоговый сигнал, несущий полезную информацию. Этот сигнал поступает в различные пассивные и активные преобразователи, фильтруется, смешивается с другими сигналами, преобразуется в цифровой сигнал, сохраняется в запоминающем устройстве и т.д. При этом к основным устройствам, осуществляющим генерацию, преобразование и передачу аналогового сигнала следует отнести всевозможные электрические фильтры, усилители, генераторы электрических колебаний различной формы, а также сумматоры, интеграторы, дифференциаторы и другие решающие элементы. Изучим работу перечисленных выше устройств и приборов.

1. Датчики. Электрическим датчиком называется устройство, преобразующее изменение исследуемой величины в напряжение (ток, сопротивление заряд). Датчики являются органами чувств электронных устройств, они переводят воспринимаемую информацию на язык электрических сигналов. *Пассивные (параметрические) датчики* под действием измеряемой величины изменяют свои параметры (резистивный, емкостный и индуктивный датчики координаты, терморезистор). Чтобы получить сигнал, несущий информацию, параметрический датчик включают в одну цепь с источником напряжения. *Активный (генераторный) датчик* сам вырабатывает напряжение, зависящее от измеряемой величины (термопара, фотоэлемент).

Простейшим является *датчик контактного типа*, представляющий собой пару контактов, замыкаемых, например, движущимся телом. *Магнитоконтактный датчик* или *геркон* (герметичный контакт) выполнен в виде стеклянного баллона с двумя или тремя электродами. При внесении датчика в магнитное поле происходит намагничивание электродов, и они замыкаются или размыкаются.



Резистивный датчик координаты преобразует изменение положения тела в изменение сопротивления (рис. 1.7.1). Он может быть выполнен в виде реостата и состоять из нихромовой проволоки, намотанной на цилиндр, по которой скользит подвижный контакт. Если на концы проволоки подать постоянное напряжение, то потенциал подвижного контакта будет изменяться пропорционально его координате.

Простейший *оптодатчик* состоит из источника и приемника света, в качестве которого часто используются фоторезистор и фотодиод (рис. 1.7.2). *Фоторезистор* — полупроводниковый резистор, при освеще-

нии которого изменяется его сопротивление (рис. 1.7.3). Действие фоторезистора объясняется внутренним фотоэффектом, то есть возникновением электронно-дырочных пар внутри полупроводника при его освещении.

Фотодиод — полупроводниковый прибор с одним р-п-переходом, при освещении которого увеличивается его проводимость. Это также обусловлено внутренним фотоэффектом: при поглощении света вблизи р-п-перехода образуются новые дырки и электроны. В *фотодиодном режиме* последовательно с фотодиодом включен источник постоянного напряжения так, что на катод подается положительный потенциал (обратное включение). Освещение фотодиода вызывает уменьшение его сопротивления, обратный ток растет. В *фотогальваническом режиме* источник отсутствует, при освещении фотодиода на его выводах возникает фотоэдс, то есть он ведет себя как фотоэлемент.

Для преобразования изменения температуры в изменение сопротивления используют терморезистор — полупроводниковый прибор, сопротивление которого при повышении температуры уменьшается (рис. 1.7.4). Акустическим датчиком, реагирующим на изменение звукового давления, является микрофон (рис. 1.7.5).

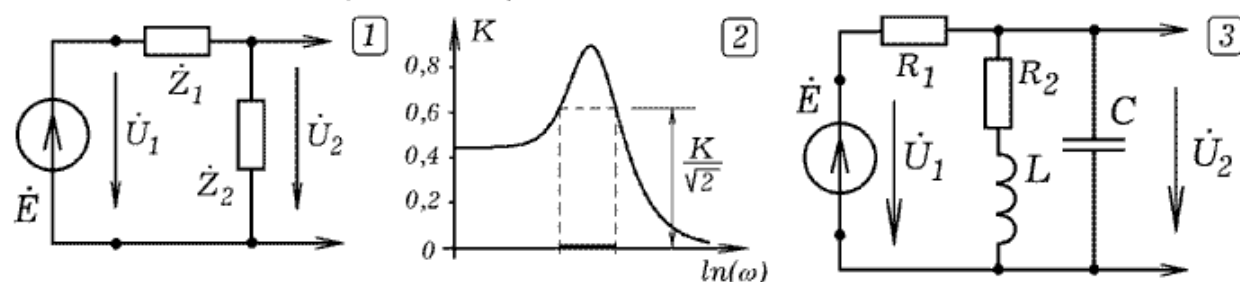


Рис. 1.8.

2. Фильтры. *Фильтром* называется пассивный четырехполюсник, составленный из резисторов, конденсаторов, катушек индуктивности и других элементов, предназначенный для разделения электрических колебаний различных частот. Различают *помехоподавляющие*, *сглаживающие* и *резонансные* фильтры. Последние делятся на фильтры верхних частот, нижних частот, полосовые и заградительные. Комплексный коэффициент передачи фильтра по напряжению равен отношению комплексов выходного \dot{U}_2 и входного \dot{U}_1 напряжений:

$$\dot{K}(\omega) = \dot{U}_2 / \dot{U}_1 = K(\omega) e^{j\varphi}, \quad K(\omega) = |\dot{K}(\omega)| = U_2 / U_1.$$

Г-образный фильтр состоит из двух последовательно соединенных ветвей с импедансами \dot{Z}_1 и \dot{Z}_2 , на которые подается входное напряжение \dot{U}_1 (рис. 1.8.1). Его коэффициент передачи равен:

$$\dot{K}(\omega) = \frac{\dot{U}_2}{\dot{U}_1} = \frac{i\dot{Z}_2}{i(\dot{Z}_1 + \dot{Z}_2)} = \frac{1}{1 + \dot{Z}_1 / \dot{Z}_2}, \quad K(\omega) = |\dot{K}(\omega)|.$$

При этом ток через нагрузку, подключенную параллельно \dot{Z}_2 считается пренебрежимо малым. *Полоса пропускания фильтра* — это диапазон

частот, на границах которого коэффициент передачи фильтра в $\sqrt{2}$ раз меньше максимального (рис. 1.8.2).

Рассмотрим Г-образный фильтр из резистора R_1 и параллельного колебательного контура, содержащего последовательно соединенные резистор R_2 , катушку индуктивности L , параллельно которым подключен конденсатор C (рис. 1.8.3). Импеданс параллельного колебательного контура равен

$$\dot{Z}_2 = \frac{(R_2 + j\omega L)(-j/\omega C)}{R_2 + j\omega L - j/\omega C}.$$

Вблизи резонансной частоты импеданс параллельного колебательного контура, а вместе с ним и модуль коэффициента передачи достигают максимального значения (рис. 1.8.2). Полоса пропускания показана толстой линией.

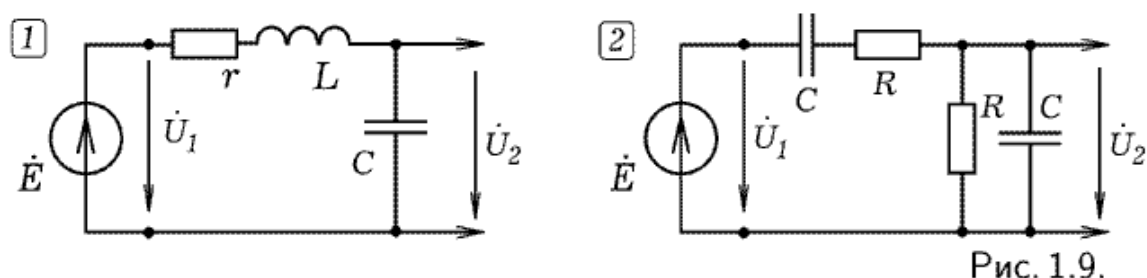


Рис. 1.9.

Рассмотрим другой фильтр, состоящий из активно-индуктивного и емкостного элементов (рис. 1.9.1). Импедансы ветвей фильтра и коэффициент передачи: $\dot{Z}_1 = R + j\omega L$, $\dot{Z}_2 = -j/(\omega C)$,

$$\dot{K} = \frac{1}{1 + (R + j\omega L)/(-j/\omega C)} = \frac{1}{1 + \omega^2 LC + j\omega RC}.$$

Графики зависимости коэффициента передачи от частоты $K = K(\omega)$ и сдвига фаз от частоты $\varphi = \varphi(\omega)$ показаны на рис. 1.10.1.

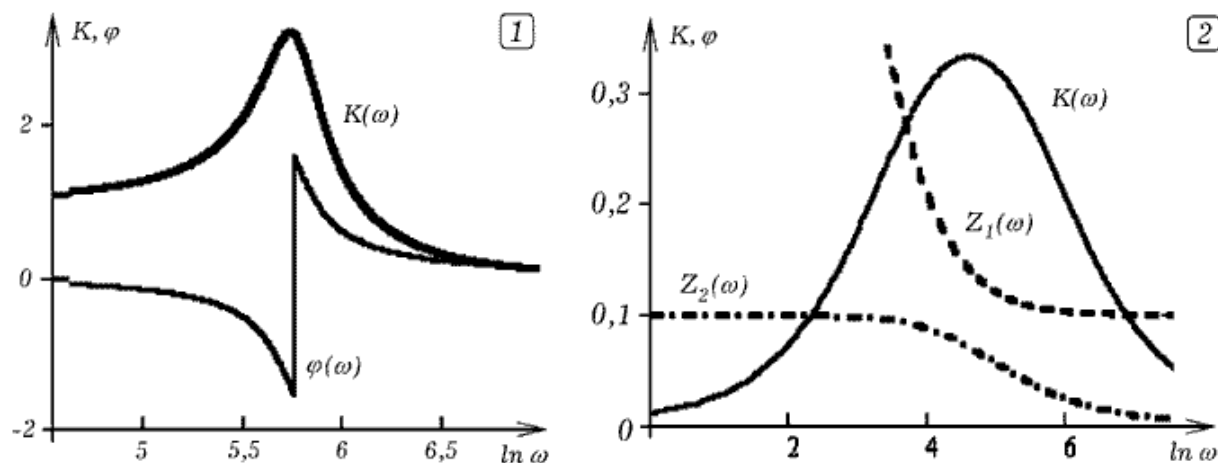


Рис. 1.10.

Изучим частотную зависимость коэффициента передачи моста Вина, представляющего собой Г-образный фильтр, обе ветви которого содержат последовательно и параллельно соединенные резистор и конденсатор

(рис. 1.9.2). Импедансы ветвей моста Вина и его коэффициент передачи равны:

$$\dot{Z}_1 = R - \frac{j}{\omega C}, \quad \dot{Z}_2 = \frac{R(-j/\omega C)}{R - j/\omega C}, \quad \dot{K} = \frac{1}{3 + j(\omega RC - 1/(\omega RC))}.$$

Графики зависимостей $K = K(\ln(\omega))$, $Z_1 = Z_1(\ln(\omega))$, $Z_2 = Z_2(\ln(\omega))$ построены на рис. 1.10.2.

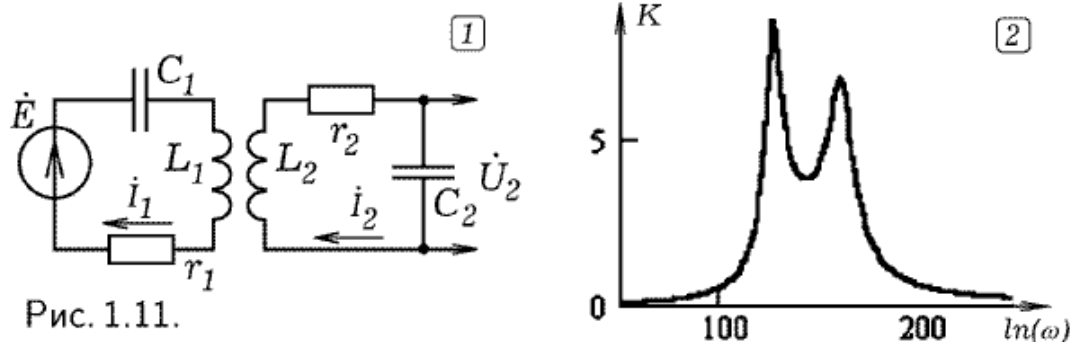


Рис. 1.11.

В идеале полосовой фильтр должен иметь П-образную характеристику $K = K(\omega)$, и линейную характеристику $\varphi = \varphi(\omega)$. Для получения П-образной зависимости $K = K(\omega)$ используют индуктивно связанные контуры, представляющие собой два колебательных контура, катушки индуктивности которых находятся на одном сердечнике (рис. 1.11.1). График зависимости $K = K(\omega)$ изображен на рис. 1.11.2.

3. Интегрирующие и дифференцирующие цепи. Последовательно соединим резистор, конденсатор и подключим их к генератору импульсов заданной формы. Напряжение, снимаемое с конденсатора (рис. 1.12.1), пропорционально интегралу от входного напряжения $u_{вх}$. Если выходное напряжение $u_{вых}$ снимать с резистора (рис. 1.12.2), то оно оказывается пропорциональным производной от входного напряжения $u_{вх}$ по времени.

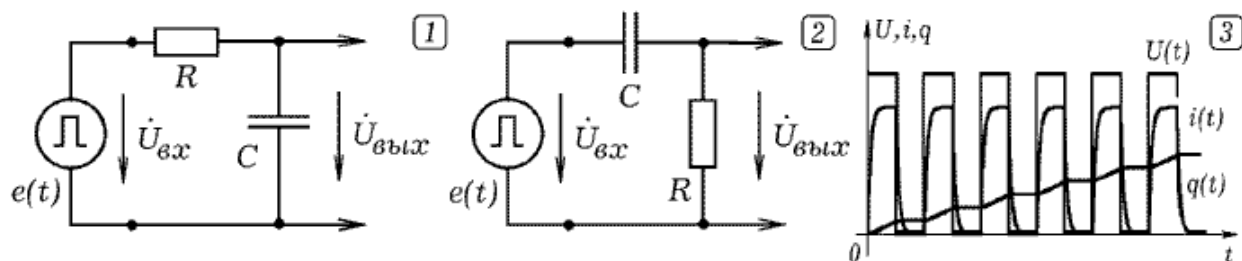


Рис. 1.12.

Рассмотрим *интегрирующую цепь*, на которую поступают импульсы напряжения с частотой ω (рис. 1.12.1). По второму закону Кирхгофа $u_{вх}(t) = u_R + u_C = iR + q/C$. Если частота достаточно велика ($\omega \gg 1/RC$), то емкостным сопротивлением можно пренебречь: $u_{вх}(t) \approx iR$ и $i = u_{вх}(t)/R$. Выходное напряжение равно:

$$u_{вых} = u_C = \frac{1}{C} \int \frac{u_{вх}}{R} dt = \frac{1}{RC} \int u_{вх} dt.$$

Коэффициент передачи интегрирующей цепи

$$\dot{K} = \frac{\dot{U}_{\text{вых}}}{\dot{U}_{\text{вх}}} = \frac{1/j\omega C}{R + 1/j\omega C} = \frac{1}{1 + j\omega RC}.$$

При поступлении на вход интегрирующей цепи прямоугольных импульсов, заряд конденсатора q и выходное напряжение $u_{\text{вых}} = q/C$ растут пропорционально числу импульсов (рис. 1.12.3).

Проанализируем дифференцирующую цепь (рис. 1.12.2). По второму закону Кирхгофа $u_{\text{вх}}(t) = u_R + u_C = iR + q/C$. Если частота достаточно мала ($\omega \ll 1/RC$), то сопротивлением R можно пренебречь: $u_{\text{вх}}(t) \approx q/C$ и $i = dq/dt = Cdu_{\text{вх}}/dt$. Выходное напряжение равно:

$$u_{\text{вых}} = u_R = iR = RC \frac{du_{\text{вх}}}{dt}.$$

Дифференцирующая цепь имеет коэффициент передачи

$$\dot{K} = \frac{\dot{U}_{\text{вых}}}{\dot{U}_{\text{вх}}} = \frac{R}{R + 1/j\omega C} = \frac{1}{1 + 1/j\omega CR}.$$

При подаче прямоугольных импульсов (рис. 1.13.1) на дифференцирующую и интегрирующую цепи напряжение на выходе изменяется как показано на рис. 1.13.2 и 1.13.3 соответственно.

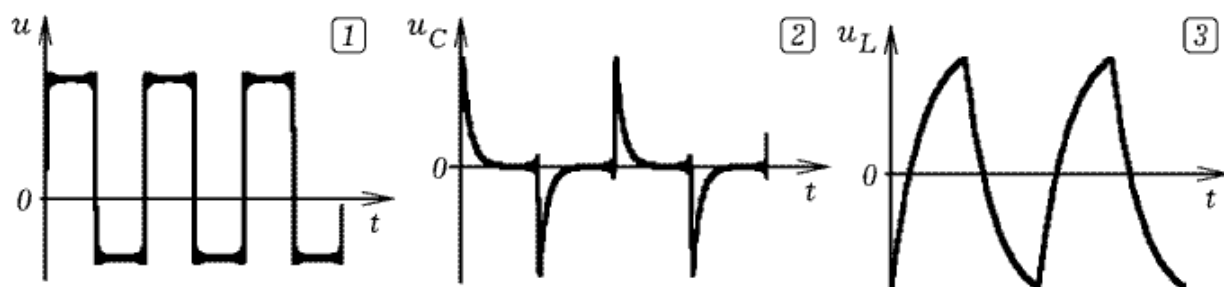


Рис. 1.13.

4. Усилители. Для увеличения амплитуды сигнала используются усилители. В качестве усиливающего элемента применяют электронные лампы, биполярные и полевые транзисторы, операционные усилители. Если коэффициент усиления не зависит от частоты, то амплитуда всех гармоник сигнала увеличивается в равное количество раз и сигнал сохраняет свою форму, то есть происходят линейные искажения. В случае, когда коэффициент усиления зависит от частоты, или усилитель достигает режима насыщения, усиливаемый сигнал изменяет свою форму, имеют место нелинейные искажения.

Резисторный усилитель (рис. 1.14.1) содержит транзистор VT1, включенный по схеме с общим эмиттером (ОЭ). Конденсаторы C1 и C3 разделительные, пропускают переменную составляющую сигнала. Делитель на R1 и R2 создает на базе транзистора необходимое напряжение смещения, задавая режим его работы. Чтобы исключить нелинейные искажения, транзистор должен быть полуоткрыт (режим А). При увеличении

потенциала на базе транзистор $n-p-n$ -типа открывается, сопротивление между коллектором и эмиттером падает, напряжение на выходе уменьшается. При уменьшении потенциала все происходит наоборот. Сдвиг фаз между входным и выходным сигналами равен π .

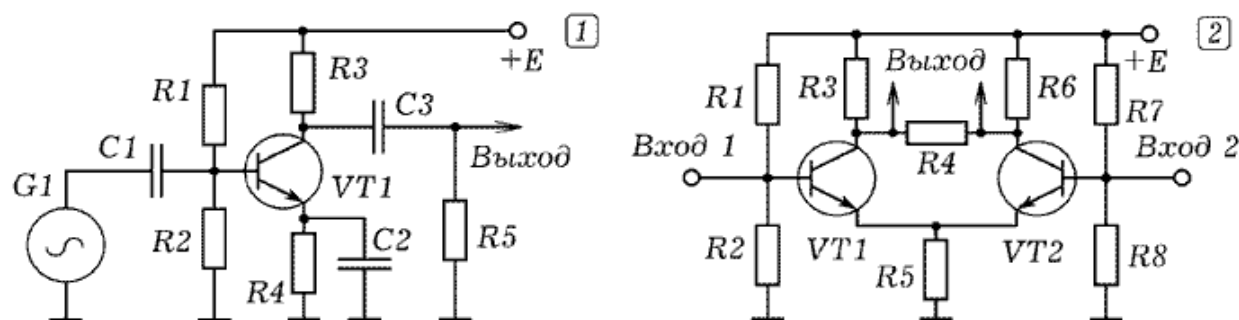


Рис. 1.14.

Резистор $R4$ необходим для температурной стабилизации работы усилителя: если температура растет, ток коллектора увеличивается. Это вызывает рост напряжения на $R4$, что приводит к уменьшению напряжения $u_{бэ}$ база-эмиттер. В результате коллекторный ток i_k стабилизируется. Чтобы на работу транзистора не влияла переменная составляющая тока i_k , резистор $R4$ шунтируют конденсатором $C2$ (рис. 1.14.1).

Дифференциальный усилитель усиливает разность входных напряжений. Он имеет два входа и два усилительных каскада (рис. 1.14.2). Транзистор $VT1$ усиливает сигнал, поступающий на вход 1, а транзистор $VT2$ — сигнал, поступающий на вход 2. Выходное напряжение, снимаемое с резистора нагрузки $R4$, равно разности потенциалов на коллекторах транзисторов: $u_{к1}, u_{к2}$: $u_{вых} = u_{к2} - u_{к1} = k(u_{вх2} - u_{вх1})$.

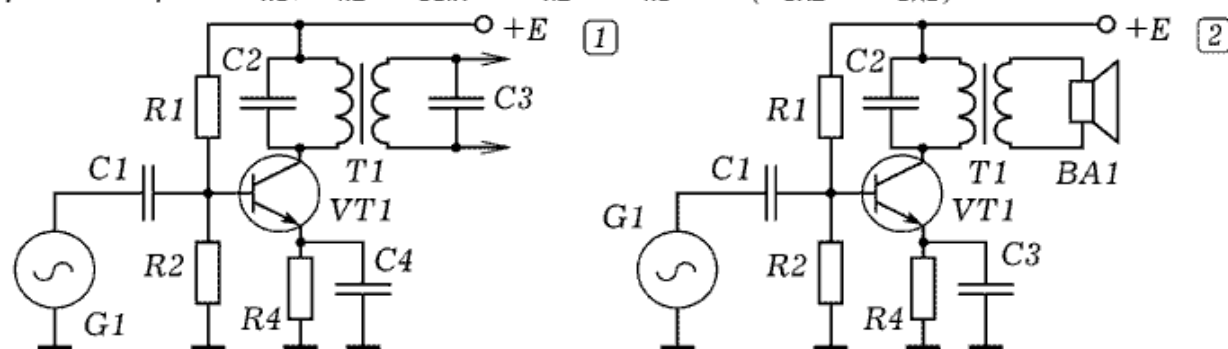


Рис. 1.15.

Резонансный усилитель отличается тем, что в качестве нагрузки содержит параллельный колебательный контур или два контура, связанных индуктивной связью (рис. 1.15.1). В этом случае коэффициент усиления сильно зависит от частоты: усиливаются сигналы, частота которых находится внутри полосы пропускания контура.

Однотактный усилитель мощности содержит трансформатор, первичная обмотка которого включена в цепь коллектора, а ко вторичной подсоединена нагрузка (рис. 1.15.2). В рассмотренных выше схемах транзистор работает в режиме А, то есть его рабочая точка — в середине линейного участка входной характеристики транзистора $i_b = i_b(u_{бэ})$. Их общий

недостаток состоит в том, что в состоянии покоя (при отсутствии входного сигнала) транзистор полуоткрыт и через цепь коллектора течет ток, батарея питания разряжается, КПД оказывается невысок.

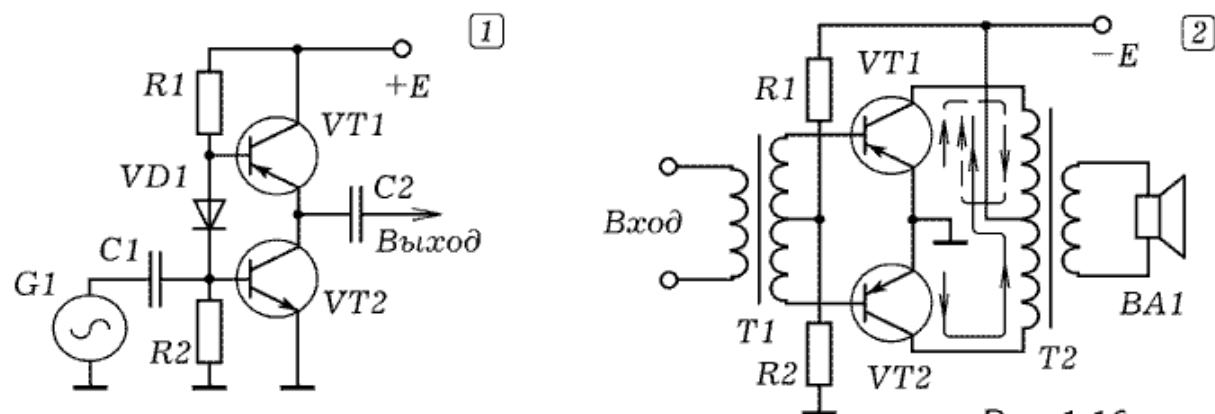


Рис. 1.16.

Двухтактный усилитель состоит из двух транзисторов, работающих в режиме В. В состоянии покоя транзисторы закрыты, а при поступлении на вход синусоидального сигнала поочередно открываются (рис. 1.16.1). Когда потенциал входа положительный, диод VD1 и транзистор VT1 закрыты, транзистор VT2 открывается и потенциал его коллектора уменьшается. При подаче на вход отрицательного потенциала транзистор VT2 закрывается, а диод VD1 открывается и потенциал базы VT1 уменьшается. Это приводит к открыванию VT1, потенциал его эмиттера растет. Конденсаторы C1 и C2 разделительные, они пропускают только переменную составляющую сигнала.

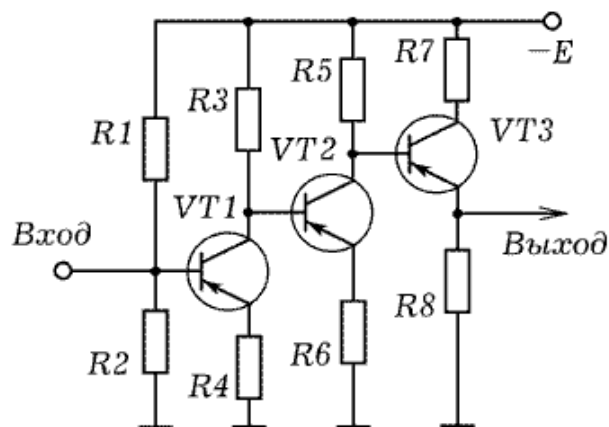


Рис. 1.17.

Схема двухтактного усилителя мощности изображена на рис. 1.16.2. Резисторы R1 и R2 образуют делитель напряжения, который создает необходимое напряжение смещения на базе транзисторов VT1 и VT2, работающих в режиме В. При подаче на вход переменного напряжения потенциалы баз транзисторов изменяются. В течение нечетных полупериодов, когда потенциал базы VT1 положительный, а потенциал базы VT2 отрицательный, VT1 закрыт, а VT2 открывается. При этом ток течет по пути, показанному стрелкой. В течение четных полупериодов открывается транзистор VT1, а VT2 закрыт, ток течет по пути, показанному пунктирной стрелкой.

Для усиления медленных изменений однополярного напряжения используется усилитель постоянного тока (рис. 1.17), отличающийся отсутствием разделительных конденсаторов. При подаче на вход положительного потенциала VT1 закрывается, потенциал базы VT2 становится более отрицательным и он открывается, что приводит к закрыванию VT3. В результате ток через резистор R8, а значит и напряжение на нем уменьшаются.

5. Генераторы колебаний. Автогенератор электрических колебаний преобразует энергию источника постоянного тока в энергию переменного тока (напряжения) требуемой формы. Различают генераторы синусоидальных колебаний и релаксационные генераторы импульсов (прямоугольных, треугольных и т.д.).

Генератор синусоидальных колебаний состоит из усилителя и цепи обратной связи, через которую часть выходного сигнала подается на вход (рис. 1.18.1). Если выходной сигнал подается на вход в фазе с входным, то такая обратная связь называется положительной (ПОС). В этом случае происходит самовозбуждение системы, амплитуда колебаний возрастает до тех пор, пока потери за период не уравновесят приток энергии от источника. Если обратная связь отрицательная (ООС), то есть выходной сигнал подается на вход усилителя в противофазе, то он ослабляет входной сигнал. В результате снижается коэффициент усиления усилителя, повышается устойчивость, становится сложнее перевести его в режим самовозбуждения.

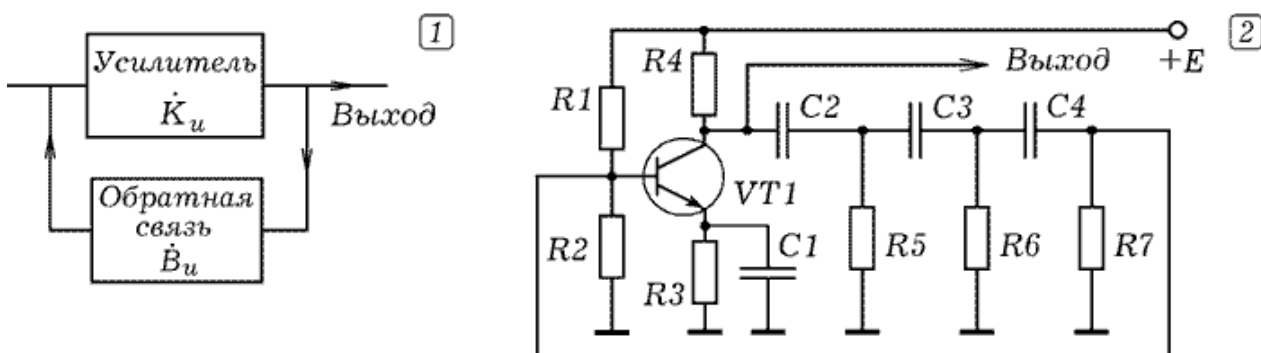


Рис. 1.18.

Коэффициенты передачи усилителя $\dot{K}_u = K_u e^{j\varphi_K}$ и цепи обратной связи $\dot{B}_u = B_u e^{j\varphi_B}$ зависят от частоты. Генератор вырабатывает колебания на частотах, для которых выполняются условия генерации: 1. Условие баланса фаз: суммарный сдвиг фаз, который приобретает сигнал при прохождении через усилитель и цепь обратной связи должен быть кратен 2π : $\varphi_K + \varphi_B = 2\pi k$. 2. Условие баланса амплитуд: произведение комплексных коэффициента усиления усилителя \dot{K}_u и коэффициента передачи цепи обратной связи \dot{B}_u по модулю равен 1: $|\dot{K}_u \cdot \dot{B}_u| = 1$.

RC-генератор с резистивно-емкостной цепью ПОС состоит из усилительного каскада на одном транзисторе, охваченный ПОС из трех Г-образных RC-фильтров (рис. 1.18.2). Транзистор включен по схеме с ОЭ и вносит сдвиг фаз π . Генератор вырабатывает синусоидальные колеба-

ния на частоте, на которой каждая RC -цепь создает сдвиг фаз $\pi/3$ (в сумме π) и выполняется баланс амплитуд.

LC -генератор представляет собой резонансный усилитель на одном транзисторе, охваченный положительной обратной связью (рис. 1.19.1). В цепи коллектора имеется колебательный контур с собственной частотой $\omega_0 = 1/\sqrt{L_1 C_1}$. На одном сердечнике с катушкой индуктивности L_1 расположена катушка связи $L_{св}$, напряжение нее подается на вход усилителя (базу транзистора VT1). При правильном включении катушки связи обратная связь получается положительной: когда ток через катушку L_1 направлен вниз по схеме, на базе VT1 появляется отрицательный потенциал. При этом транзистор открывается, сопротивление между коллектором и эмиттером падает. Часть обмотки АВ катушки L_1 оказывается подключенной к источнику так, что текущий через нее ток усиливает ток, возникающий при перезарядке конденсатора. В следующие полпериода, когда ток через катушку L_1 течет вверх по схеме, на базе VT1 появляется положительный потенциал, транзистор закрывается и т.д. Генератор вырабатывает синусоидальные колебания с частотой ω_0 .

Симметричный мультивибратор (рис. 1.19.2) представляет собой двухкаскадный усилитель, выход которого соединен с входом. Каждый транзистор поворачивает фазу на π , поэтому баланс фаз выполняется для всех частот. На выходе получается последовательность прямоугольных импульсов, частота которых тем выше, чем меньше емкость конденсаторов C_1, C_2 и сопротивление резисторов R_2, R_3 . Если на базу одного из транзисторов подать отрицательный потенциал, то он будет все время закрыт, генерация прекратится.

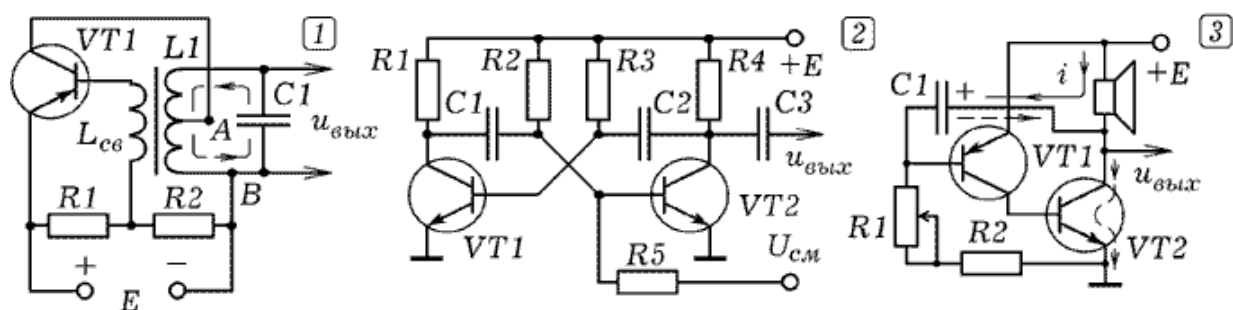


Рис. 1.19.

Схема несимметричного мультивибратора представлена на рис. 1.19.3. При включении конденсатор C_1 заряжается через динамик, резисторы R_1 и R_2 . При этом течет ток i , показанный стрелкой, потенциал базы VT1 уменьшается. Когда на базе VT1 возникает отрицательный относительно эмиттера потенциал, транзистор VT1 открывается, сопротивление между коллектором и эмиттером падает. База транзистора VT2 оказывается соединенной с положительным полюсом источника, транзистор VT2 открывается, и его сопротивление между коллектором и эмиттером падает. В результате через динамик, коллектор и эмиттер открытого VT2 течет ток, конденсатор C_1 разряжается через резисторы R_1, R_2 и транзистор VT2. Потенциал базы VT1 становится положительным, транзистор закрывается, вызывая закрытие транзистора VT2. После этого конден-

сатор $C1$ снова заряжается, все повторяется снова. Частота генерируемых импульсов обратно пропорциональна времени заряда конденсатора $\tau = (R_1 + R_2)C_1$. При увеличении сопротивления $R1$ частота щелчков динамика уменьшается.

Простейший генератор линейно-импульсного (пилообразного) напряжения может быть собран из тиристора (динистора или тринистора), резистора и конденсатора (рис. 1.20.1). Тринистор отличается от динистора наличием управляющего электрода. При включении тиристор закрыт, конденсатор $C1$ медленно заряжается от источника питания через резистор $R1$. Напряжение на конденсаторе растет до напряжения открывания тиристора. Когда тиристор открывается, его сопротивление резко падает и конденсатор быстро разряжается через него. При уменьшении анодного напряжения до напряжения закрывания тиристор закрывается, после чего все повторяется снова. Время заряда $\tau = R_1 \cdot C_1$, поэтому при увеличении $R1$ и $C1$ частота колебаний уменьшается. С ростом напряжения питания частота генерируемых импульсов увеличивается. Если использовать тринистор, то при подаче на управляющий электрод положительного относительно катода потенциала, напряжение открывания уменьшается, частота формируемых импульсов растет.

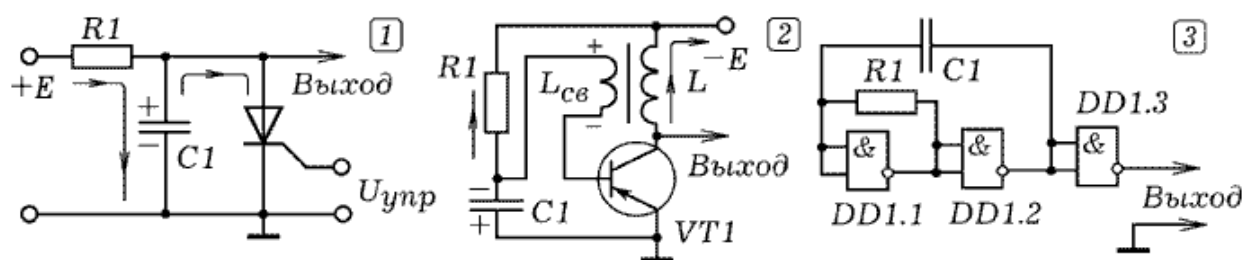


Рис. 1.20.

Блокинг-генератор — это релаксационный генератор с сильной трансформаторной обратной связью, вырабатывающий последовательность кратковременных импульсов, разделенных достаточно большим промежутком времени (рис. 1.20.2). При включении конденсатор $C1$ начинает заряжаться через резистор $R1$, на базе появляется отрицательный потенциал. Транзистор $VT1$ открывается, через катушку L начинает течь ток. В результате в катушке связи $L_{св}$ возникает напряжение и при правильном ее включении потенциал базы $VT1$ становится еще отрицательнее. Транзистор открывается еще сильнее до тех пор, система не войдет в насыщение. Через резистор $R1$ и открытый транзистор конденсатор $C1$ начинает разряжаться. Так как коллекторный ток перестает изменяться, то напряжение на катушке связи становится равным нулю и транзистор закрывается. Конденсатор $C1$ начинает заряжаться, потенциал базы транзистора уменьшается и т.д.

В цифровой технике также используется генератор прямоугольных импульсов, собранный на двух или трех логических элементах (рис. 1.20.3). Каждый элемент инвертирует импульс, изменяя фазу колебаний на π . Частота генерируемых импульсов тем больше, чем меньше сопротивление R и емкость C .



ЕРШОВ Андрей Петрович (1931–1988 гг.)

Академик АН СССР, занимался проблемами программирования и информатики. После окончания МГУ руководил работами на БЭСМ и ЭВМ "Стрела". Создал неформальную школу программистов, занимался разработкой языков программирования. Член Американской ассоциации по вычислительной технике, почетный член Британского общества по вычислительной технике.

6. Операционный усилитель. Операционный усилитель (ОУ) — усилитель постоянного тока с высоким коэффициентом передачи постоянного напряжения и большим выходным сопротивлением. На базе ОУ возможно создание аналоговой ЭВМ, работающей с плавно изменяющимся напряжением и осуществляющей решение соответствующего дифференциального или интегрального уравнения. В зависимости от вида уравнения создается схема АЭВМ, содержащая сумматоры, вычитатели, умножители, дифференциаторы, интеграторы, осуществляющие соответствующие операции над входным сигналом. Перечисленные выше блоки могут быть созданы с помощью резисторов, конденсаторов и катушек индуктивности, однако использование ОУ позволяет получить более точные результаты. Интегральные ОУ имеют коэффициент усиления по напряжению от 500 до 50000, входные токи $10 - 10^{-3}$ мкА.

ОУ имеет два входа (инвертирующий и неинвертирующий) и один выход, на котором получается усиленная разность входных сигналов: $U_{\text{вых}} = K_u(U_{\text{неинв}} - U_{\text{инв}})$. При подаче на инвертирующий вход переменного напряжения на выходе получают усиленные колебания той же частоты, сдвинутые по фазе на π . ОУ содержит дифференциальный усилитель, схема которого аналогична изображенной на рис. 1.14.2, и выходной каскад (рис. 1.21.1). При подаче положительного потенциала на базы транзисторов VT1 и VT2, работающих в режиме В, VT1 открывается, а VT2 закрывается. Если на входе отрицательный потенциал, то открывается VT2, VT1 закрывается.

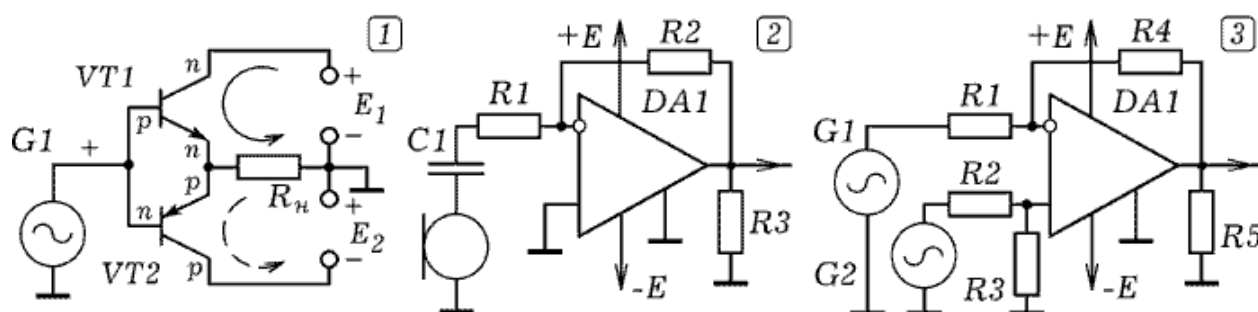


Рис. 1.21.

Чтобы из ОУ получить усилитель, необходимо на инвертирующий вход подать усиливаемый сигнал, а второй вход соединить с общим проводом (рис. 1.21.2). Резистор R2 соединяет выход ОУ с его инвертирующим вхо-

дом и образует цепь отрицательной обратной связи. Коэффициент усиления ОУ равен $K_u = R_2/R_1$. Схема включения ОУ как дифференциального усилителя приведена на (рис. 1.21.3); на его выходе получается усиленная разность входных сигналов.

Рассмотрим схему сумматора аналоговых сигналов с тремя входами (рис. 1.22.1). Выходное напряжение равно

$$U_{\text{вых}} = \left(\frac{R_5}{R_1} U_1 + \frac{R_5}{R_2} U_2 + \frac{R_5}{R_3} U_3 \right),$$

где U_1, U_2, U_3 — напряжения на входах сумматора. Если $R_1 = R_2 = R_3$, то на выходе $U_{\text{вых}} = (R_5/R_1)(U_1 + U_2 + U_3)$. Интегратор состоит из ОУ, в цепи обратной связи которого находится конденсатор (рис. 1.22.2). При подаче на его вход последовательности импульсов конденсатор накапливает заряд, потенциал инвертирующего входа растет, выходное напряжение изменяется пропорционально интегралу от входного напряжения по времени:

$$U_{\text{вых}} = -\frac{k_1}{R_1 C_1} \int U_{\text{вх}}(t) dt.$$

Если резистор R_1 и конденсатор C_1 поменять местами, получится *дифференциатор*. На его выходе получается напряжение, пропорциональное производной от входного напряжения по времени:

$$U_{\text{вых}} = -k_2 R_1 C_1 \frac{dU_{\text{вх}}}{dt}.$$

Для того чтобы получить RC -генератор синусоидальных колебаний необходимо подать сигнал с выхода ОУ на его неинвертирующий вход через *мост Вина* (рис. 1.22.3). Зависимость коэффициента передачи моста Вина от частоты похожа на резонансную кривую: она имеет максимум на частоте $\omega = 1/RC$, где $R = R_2 = R_4$, $C = C_1 = C_2$ (рис. 1.10.2). Цепь положительной обратной связи хорошо пропускает колебания этой частоты, сильно ослабляя колебания других частот. В результате на выходе ОУ генерируется синусоидальный сигнал с частотой $\omega = 1/RC$.

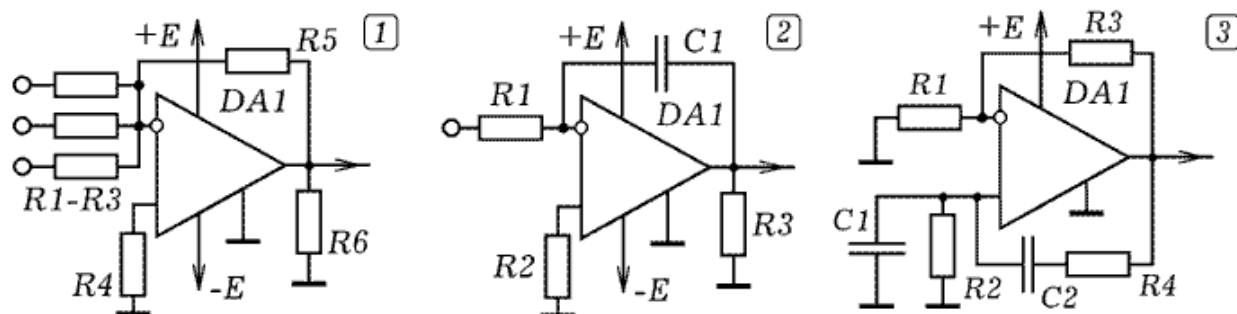


Рис. 1.22.

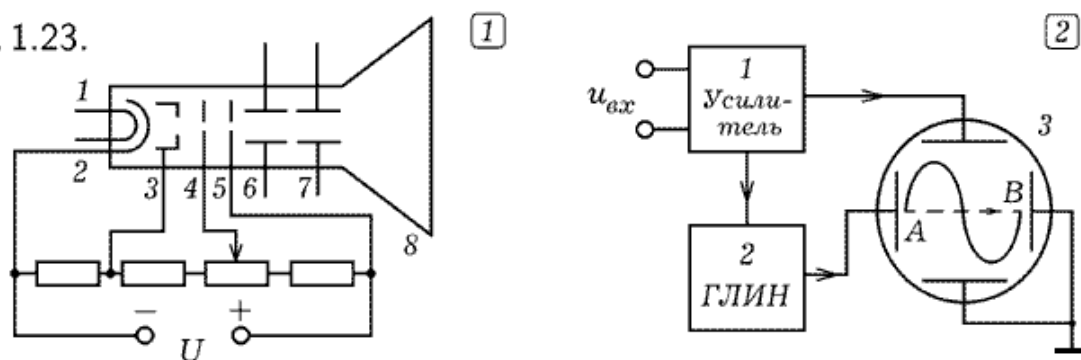
7. Способы регистрации аналоговых сигналов. Для обработки информации необходимо уметь ее сохранять. Для этого используются запоминающие устройства (ЗУ), способные записывать и сохранять аналоговые и цифровые сигналы. Кроме того, в электронных приборах широко

используются устройства отображения информации: различные индикаторы, мониторы, дисплеи. Познакомимся с принципами хранения и отображения информации.

Устройства записи аналоговой информации подразделяются на следующие классы: 1. *Приборы прямого преобразования*: входной сигнал непосредственно воздействует на регистрирующий орган (осциллограф, магнитофон). 2. *Приборы со следящим преобразованием*: на регистрирующий орган воздействует не входной сигнал, рассогласование между входным и вспомогательным (компенсирующим) сигналом (устройство, поворачивающее телескоп вслед за движением звезды по небесной сфере); 3. *Приборы развертывающего преобразования*: компенсирующий сигнал с высокой частотой периодически изменяется во всем диапазоне, и в момент совпадения компенсирующего сигнала со входным, формируется импульс, который поступает на регистрирующий орган (работа АЦП, рис. 2.23.1). Существуют три группы способов регистрации сигнала: 1) нанесение слоя вещества (самописец); 2) деформация или снятие слоя вещества (звукозапись на грампластинку); 3) изменение состояния вещества носителя (фотобумага, лазерный диск, магнитная пленка).

Осциллографический метод регистрации сигналов позволяет получить *осциллограмму* — график зависимости напряжения от времени. При этом используется *электронно-лучевая трубка* (ЭЛТ). Она представляет собой стеклянный вакуумированный баллон с экраном, который изнутри покрыт люминофором (рис. 1.23.1). В задней части ЭЛТ имеется электронный прожектор, состоящий из нити накала, катода и анода с отверстием в центре. Для фокусировки электронного пучка используется фокусирующий электрод.

Рис. 1.23.



Электронный осциллограф состоит из усилителя 1, генератора линейно-импульсного (пилообразного) напряжения 2 и электронно-лучевой трубки 3 (рис. 1.23.2). Генератор вырабатывает пилообразное напряжение, которое подается на горизонтально отклоняющие пластины. При этом электронный луч равномерно перемещается слева направо, затем быстро возвращается обратно. Исследуемый сигнал $u = u(t)$ усиливается и подается на вертикально отклоняющие пластины ЭЛТ. В результате на горизонтальное равномерное движение луча накладываются колебания по вертикали, происходящие в такт с колебаниями исследуемого напряжения. На экране ЭЛТ получается график $u = u(t)$.

1.3. ПЕРЕДАЧА ИНФОРМАЦИИ НА БОЛЬШИЕ РАССТОЯНИЯ

Способ передачи информации зависит от уровня технического прогресса. Раньше, чтобы передать сообщение посылали гонца или использовали цепь из людей, которые последовательно зажигали факелы или отбивали последовательность ударов в большие барабаны. Позже была изобретена проводная (телеграф, телефон) и беспроводная связь (радиотелеграф, радиотелефон, телевидение, оптическая связь и т.д.)

1. Различные виды связи. В общем случае система передачи информации состоит из источника сигналов, линии связи, ретранслятора, приемника сигналов (рис. 1.24). Источник сигналов включает в себя датчик 1, преобразователь 2, усилитель 3, передающее устройство 4 (антенна, светоизлучающий элемент). Ретранслятор состоит из приемного устройства 6 (антенна, фотодатчик), усилителя 7 и передающего устройства 8. Приемник сигналов имеет приемное устройство 10, усилитель 11, дешифратор 12, устройство вывода информации 13 (динамик, дисплей). Каналом связи 5 и 9 могут являться провода, передающие электрический сигнал, окружающее пространство, по которым распространяются радиоволны, прозрачная среда или оптоволоконный кабель, передающие световые сигналы.



Рассмотрим передачу сообщений (сигналов) по проводам. Информационная система при этом состоит из передающего устройства, линии связи (двух проводников), и приемного устройства. В простейшем случае передающим устройством является ключ, а принимающим — лампочка или звонок. С помощью азбуки Морзе или аналогичного способа кодирования букв и их сочетаний можно передать закодированное сообщение. При этом используются цифровые сигналы, а связь называется *телеграфной*.

Позже телеграфные аппараты были усовершенствованы до *телетайпов* содержащих клавиатуру, печатающий механизм и устройство, осуществляющее автоматическое кодирование и декодирование сообщения. При отправке телеграммы пользователь набирает текст на клавиатуре, телетайп осуществляет кодирование сообщения и выдает цифровой сигнал в линию связи. Телетайп, принимающий сообщение, автоматически декодирует поступающий сигнал и печатает текст на листе бумаги.

Телефонная проводная связь реализуется с помощью микрофона, усилителя, двухпроводной линии связи и динамика или телефона. Микрофон преобразует звуковые колебания в колебания напряжения. Получающийся при этом аналоговый сигнал усиливается усилителем и посылается в линию связи. Динамик осуществляет преобразование электрических колебаний в звуковые.

Оптическая связь заключается в передаче информации с помощью све-

та через прозрачную среду или по световоду. *Телеграфная оптическая связь* — передаваемое сообщение кодируется с помощью азбуки Морзе (точка–тире) или представляется в виде последовательности электрических импульсов. Передающее устройство имеет светоизлучающий элемент, посылающий световые импульсы на фотоэлемент удаленного приемника. На выходе фотоэлемента появляются импульсы напряжения, которые усиливаются, декодируются и поступают на регистрирующее устройство.

Телефонная оптическая связь может быть организована следующим образом. Сигнал с микрофона усиливается и подается на устройство, модулирующее световой пучок так, чтобы его интенсивность изменялась в такт со звуковыми колебаниями, несущими информацию. Этот световой пучок освещает фотоэлемент приемника, который преобразует изменения интенсивности в соответствующие изменения напряжения. Сигнал демодулируется, усиливается и подается на динамик.

Система радиопередачи включает в себя одну или несколько радиопередающих станций и радиоприемник. Если низкочастотный сигнал с микрофона усилить и подать на антенну, то осуществить передачу радиосообщения на достаточно большое расстояние не удастся потому, что: 1) электромагнитные волны звуковой частоты (ниже 20 кГц) плохо излучаются, так как интенсивность радиоволны пропорциональна четвертой степени частоты; 2) при наличии двух или более передатчиков сигналы будут глушить друг друга, отфильтровать их невозможно.

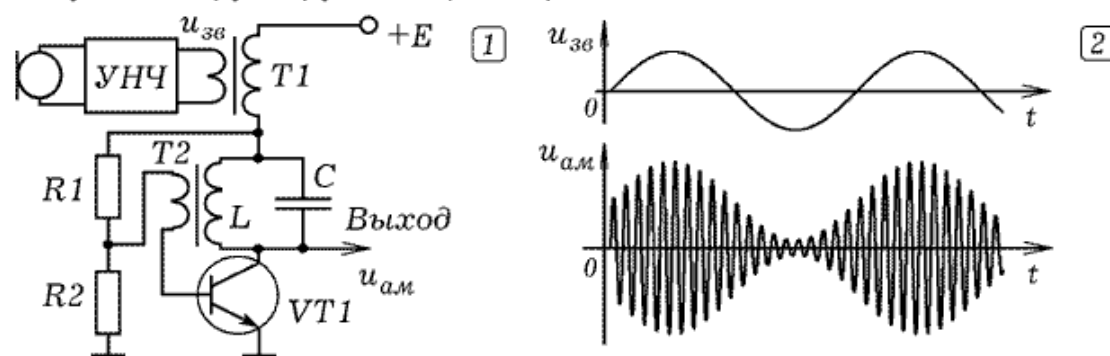


Рис. 1.25.

Поэтому используется *принцип частотного разделения*: каждая радиостанция работает на своей несущей частоте f_n . В передатчике низкочастотный сигнал с микрофона накладывается на колебания высокой частоты f_n . В результате модуляции происходят изменения амплитуды или частоты высокочастотных колебаний в такт с низкочастотными. Эти высокочастотные колебания усиливаются и излучаются антенной в пространство. Антенна радиоприемника воспринимает сигналы от всех радиостанций. С помощью резонансного фильтра выделяются колебания определенной частоты, на которой работает данный передатчик. С помощью детектора осуществляется детектирование — процесс, обратный модуляции, то есть выделение низкочастотного сигнала из модулированного сигнала высокой частоты.

2. Модуляция и детектирование. Модуляция — наложение сигнала

ла, несущего информацию, на колебания высокой частоты. В результате *амплитудной модуляции* происходит изменение амплитуды несущих колебаний высокой частоты в соответствии с низкочастотным сигналом, который переносит информацию. *Коллекторная модуляция* осуществляется с помощью схемы на рис. 1.25.1. *Модулятор* представляет собой генератор колебаний, работающий на высокой частоте $\omega_0 = 1/\sqrt{LC}$. Последовательно с источником включена вторичная обмотка трансформатора. На его первичную обмотку подается низкочастотный сигнал (например, гармонические колебания звуковой частоты). Так как амплитуда колебаний на выходе генератора с ростом напряжения питания увеличивается, то в результате получают высокочастотные колебания, амплитуда которых изменяется в такт с модулирующим сигналом (рис. 1.25.2).

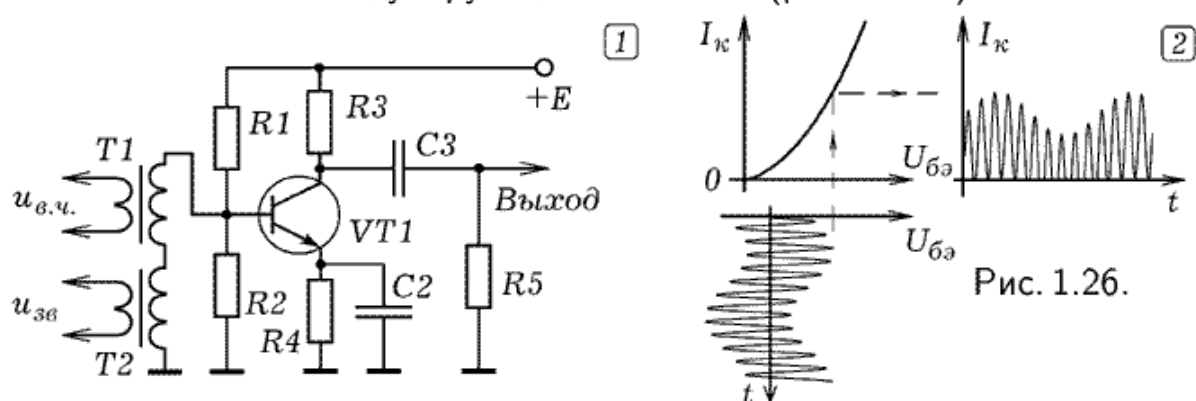


Рис. 1.26.

Схема, позволяющая осуществить *базовую модуляцию*, изображена на рис. 1.26.1. Высокочастотные несущие колебания и низкочастотный сигнал, передающий информацию, подаются на первичные обмотки трансформаторов T1 и T2. Их вторичные обмотки соединены последовательно, поэтому на базу транзистора поступает результат сложения двух колебаний, частоты которых сильно отличаются. Напряжение смещения, задаваемое резисторами R1 и R2, подобрано так, чтобы на выходе транзистора получалась последовательность импульсов, амплитуда которых изменяется в такт с низкочастотными колебаниями. Это хорошо видно из рис. 1.26.2, на котором показана динамическая проходная характеристика транзистора, входной и выходной сигналы.

Для осуществления *частотной модуляции* используют *варикап* — полупроводниковый прибор с одним р–п–переходом, емкость которого зависит от приложенного напряжения. Варикап включают параллельно конденсатору, входящему в колебательный контур генератора высокочастотных колебаний. При подаче на варикап медленно изменяющегося напряжения происходят соответствующие изменения частоты колебаний на выходе генератора. Частотно-модулированный сигнал имеет более высокую помехоустойчивость по сравнению с амплитудно-модулированным сигналом.

Процесс, обратный модуляции, в результате которого из модулированного сигнала выделяется сигнал, несущий информацию, называется *детектированием*. Простейший *амплитудный детектор* представляет собой полупроводниковый диод и параллельно соединенные резистор и конденсатор (рис. 1.27.1). Диод пропускает ток только от анода к катоду и

как бы обрезает отрицательную половину амплитудно-модулированного сигнала. Конденсатор выполняет роль фильтра, шунтируя резистор по переменной составляющей высокой частоты. Переменный ток идет через конденсатор, а через резистор течет медленно изменяющийся ток. На нем возникает напряжение, форма которого повторяет модулирующий низкочастотный сигнал.

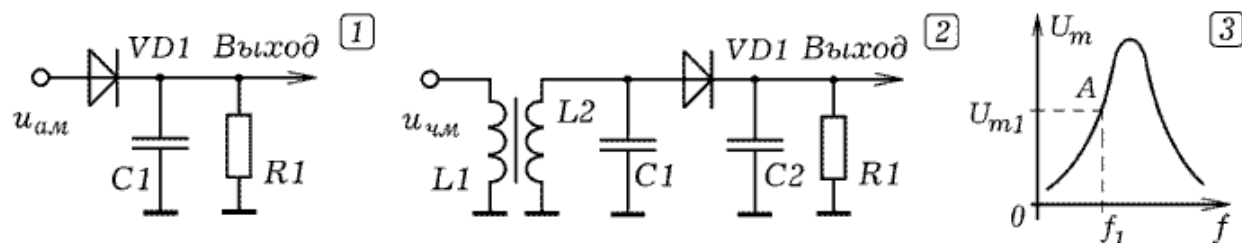


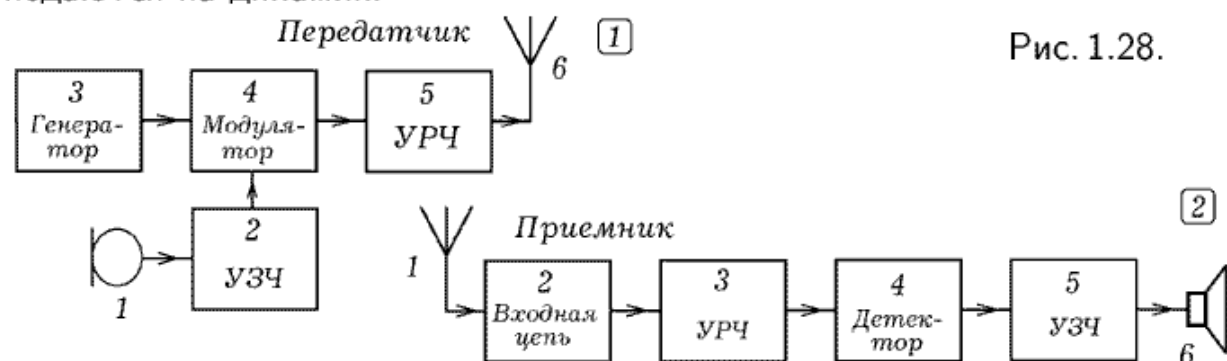
Рис. 1.27.

Частотный детектор (рис. 1.27.2) состоит из параллельного колебательного контура $L2C1$, преобразующего изменения частоты в изменения амплитуды колебаний, и амплитудного детектора. Собственная частота контура должна несколько отличаться от средней частоты частотно-модулированного сигнала так, чтобы соответствующая ей рабочая точка A находилась на прямолинейном участке резонансной кривой контура (рис. 1.27.3). Тогда увеличение частоты сигнала приведет к росту амплитуды, а уменьшение частоты — к уменьшению амплитуды колебаний напряжения, снимаемого с колебательного контура. В результате частотно-модулированный сигнал преобразуется в амплитудно-модулированный сигнал, который подается на амплитудный детектор, состоящий из диода $VD1$, конденсатора $C2$ и резистора $R1$.

3. Радиопередатчик и радиоприемник. Радиопередатчик состоит из микрофона 1, усилителя звуковой частоты 2, генератора колебаний несущей частоты 3, модулятора 4, усилителя радиочастоты 5 и излучающей антенны 6 (рис. 1.28.1). Низкочастотный сигнал с микрофона усиливается усилителем звуковой частоты и смешивается в модуляторе с высокочастотными колебаниями, вырабатываемыми генератором. Получающийся амплитудно-модулированный сигнал усиливается УРЧ и излучается антенной в пространство. Каждый радиопередатчик работает на своей уникальной частоте несущих колебаний. Весь частотный диапазон разбит на 4 поддиапазона: длинные волны ДВ (λ от 10^3 до 10^4 м); средние волны СВ (λ от 10^2 до 10^3 м); короткие волны КВ (λ от 10 до 10^2 м); ультракороткие волны УКВ (λ от 1 до 10 м).

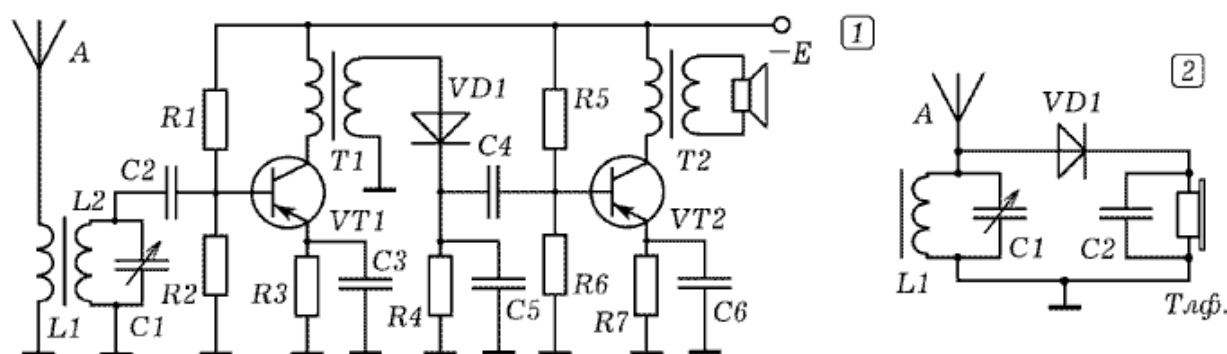
Радиоприемник прямого усиления состоит из приемной антенны 1, входной цепи 2, усилителя радиочастоты 3, детектора 4, усилителя звуковой частоты 5, динамика 6 (рис. 1.28.2). Радиоволны, достигнув антенны, вызывают вынужденные колебания электронов, то есть возникновение высокочастотного тока. Входная цепь содержит регулируемый резонансный фильтр, пропускающий колебания узкого диапазона частот, что позволяет настроить приемник на частоту, соответствующую тому или иному радиопередатчику. Сигнал усиливается УРЧ и поступает на

детектор. В результате детектирования происходит выделение из модулированного высокочастотного сигнала низкочастотных колебаний, которые несут полезную информацию. Эти колебания усиливаются УЗЧ и подаются на динамик.



Принципиальная схема транзисторного приемника прямого усиления приведена на рис. 1.29.1. Входная цепь содержит параллельный колебательный контур с конденсатором переменной емкости $C1$, позволяющим настраивать радиоприемник на ту или иную радиостанцию. На транзисторах $VT1$ и $VT2$ собраны однокаскадные усилители радиочастоты и звуковой частоты. Детектирование амплитудно-модулированного сигнала осуществляется с помощью диода $VD1$. На рис. 1.29.2 представлена схема детекторного приемника.

Недостатки приемника прямого усиления: 1) низкая чувствительность (при повышении коэффициента усиления в перенастраиваемом по частоте усилителе возникает самовозбуждение); 2) чувствительность сильно зависит от частоты; 3) частоты близких станций не различаются; на высоких частотах приемник теряет избирательные свойства и не ловит радиостанции в КВ и УКВ диапазонах; 4) АЧХ резонансного усилителя с одним колебательным контуром далека от П-образной формы.



4. Приемник-супергетеродин. Основными недостатками приемника прямого усиления являются плохая избирательная способность и невысокая чувствительность, сильно зависящая от частоты. Из-за этого он не может работать в диапазоне коротких волн. От перечисленных недостатков свободен более сложный *приемник-супергетеродин*.

Представим себе, что колебания несущей частоты f_n , воспринимаемые

приемной антенной, преобразуются в колебания промежуточной частоты $f_{пч}$, которые затем поступают в усилитель промежуточной частоты УПЧ, имеющий высокий коэффициент усиления и острую АЧХ. Такое преобразование осуществляется путем смешивания воспринимаемых колебаний несущей частоты f_n с колебаниями гетеродина (генератора регулируемой частоты) частотой f_r .

В результате наложения двух колебаний с частотами f_n , f_r получают биения с частотой $|f_n - f_r|$. Пусть две станции работают на частотах $f_{н1} = 30,2$ МГц и $f_{н2} = 30,5$ МГц, относительная расстройка составляет $\Delta f/f = |f_{н2} - f_{н1}|/f_{н1} = |30,5 - 30,2|/30,2 \approx 0,01$ то есть 1 %. Полоса пропускания колебательного контура 2–5 %, поэтому он не сможет их отфильтровать. В приемнике прямого усиления оба сигнала будут усилены; один из них будет помехой для другого.

Пусть в супергетеродине принимаемые антенной сигналы с частотами $f_{н1}$ и $f_{н2}$ смешиваются с колебаниями гетеродина с частотой $f_r = 30$ МГц. В результате получаются колебания $f'_1 = 0,2$ МГц и $f'_2 = 0,5$ МГц, их относительная расстройка $\Delta f/f = |f'_2 - f'_1|/f_{ср} = |0,5 - 0,2|/0,35 \approx 0,9$ то есть около 90 %. Такие колебания могут быть легко отфильтрованы.

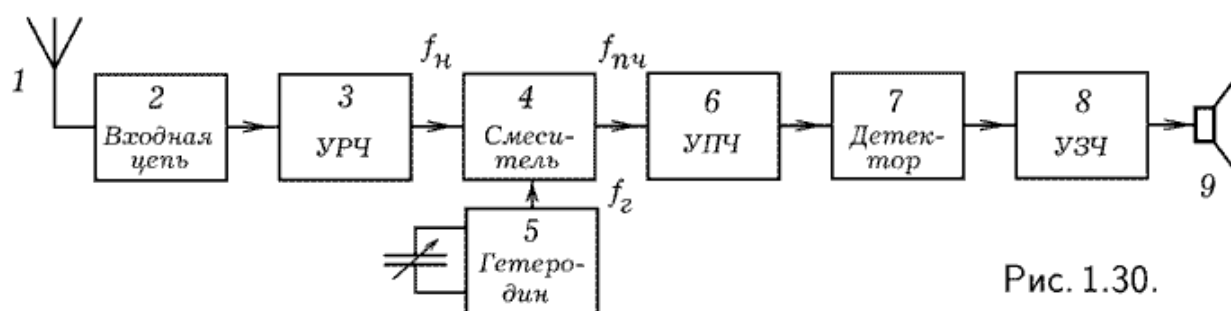


Рис. 1.30.

Блок-схема приемника-супергетеродина изображена на рис. 1.30. Сигналы, воспринимаемые антенной 1, усиливаются усилителем радиочастоты 3 и поступают в смеситель 4, в котором они складываются с колебаниями гетеродина 5 регулируемой частоты f_r . В результате сложения колебаний с частотой несущей f_n с колебаниями частотой f_r получаются биения частотой $|f_n - f_r|$. Они поступают на вход усилителя промежуточной частоты 6, который имеет высокий коэффициент усиления на промежуточной частоте $f_{пч}$ и острую АЧХ.

Таким образом, через УПЧ проходит сигнал с частотой $f_{пч}$, который несет информацию от радиостанции работающей на такой частоте f_n , что $|f_r - f_n| = f_{пч}$. Этому условию удовлетворяют две частоты $f'_n = f_r - f_{пч}$ и $f''_n = f_r + f_{пч}$. Сигнал частотой $f''_n = f_r + f_{пч}$ называется зеркальной помехой, он подавляется во входной цепи и в УРЧ. Настройка супергетеродина на другую радиостанцию осуществляется путем регулировки частоты гетеродина (изменяется емкость и индуктивность его колебательного контура). Сигнал, усиленный УПЧ, подается на детектор 7. После детектирования получаются низкочастотные колебания, которые поступают на усилитель звуковой частоты 8, с динамиком 9 на выходе.

Преимущества приемника-супергетеродина: 1) высокая избирательная способность; 2) так как УПЧ работает на фиксированной частоте, то в качестве фильтра с П-образной характеристикой можно использовать связанные контуры; 3) при фиксированной частоте сигнала легче повысить коэффициент усиления без самовозбуждения УРЧ и УПЧ; 4) чувствительность не зависит от несущей частоты.



Рис. 1.31.

Основные недостатки состоят в следующем: 1. Существуют две радиостанции с частотами $f'_H = f_r - f_{пч}$ и $f''_H = f_r + f_{пч} = f'_H + 2f_{пч}$, сигналы от которых проходят через УПЧ. Сигнал с частотой f''_H называется *зеркальной помехой*, он подавляется колебательными контурами во входной цепи и УРЧ. Чтобы f'_H и f''_H сильно отличались, промежуточная частота $f_{пч}$ должна быть достаточно большой. 2. Если частота сигнала от радиостанции равна промежуточной частоте, то УПЧ его тоже усилит и он создаст *помеху прямого прохождения*. Чтобы этого избежать во входной цепи приемника ставят фильтр-пробку, который не пропускает колебания промежуточной частоты. Кроме того, промежуточная частота выбирается в окне между диапазонами ДВ и СВ (на ней радиостанции не работают) и составляет 465 кГц.

5. Принципы телевидения. Телевидение — передача изображения движущихся объектов на большое расстояние. Представим себе источник сигнала — двумерную матрицу светочувствительных элементов, на которые проецируется изображение движущегося предмета (рис. 1.31). Каждый светочувствительный элемент соединен отдельным каналом связи с соответствующим светоизлучающим элементом приемника. При движении объекта происходит перемещение его изображения по светочувствительной матрице и на приемной матрице формируется его изображение.

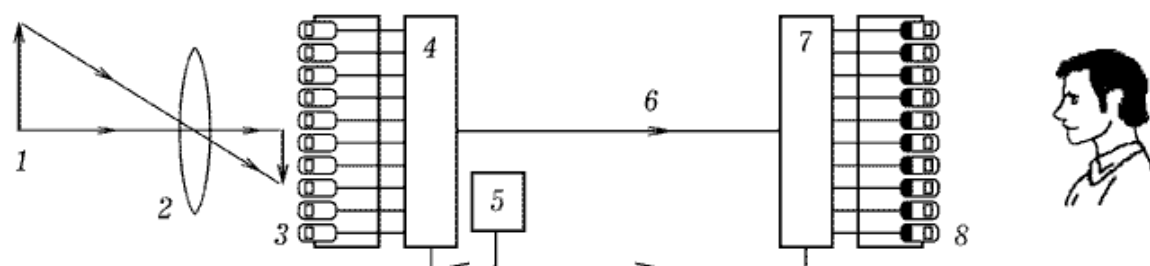


Рис. 1.32.

Для передачи черно-белого изображения не очень высокого качества, например, разрешением 640×480 , необходимо $640 \cdot 480 \approx 3 \cdot 10^5$ каналов связи. Такой параллельный способ передачи информации представляет определенные технические трудности и экономически не выгоден. Поэто-

му используют один канал связи, к которому последовательно "подключают" соответствующие светочувствительный и светоизлучающий элементы, расположенные на источнике и приемнике телевизионного сигнала. При таком последовательном способе передачи информации реализуется принцип временного разделения канала связи.

Итак, для передачи изображения движущихся объектов используется матрица светочувствительных элементов 3, мультиплексор 4, канал связи 6, демультиплексор 7, матрица светоизлучающих элементов 8, а также генератор тактовых импульсов 5, синхронизирующий работу мультиплексора и демультиплексора (рис. 1.32). Генератор вырабатывает тактовые импульсы, которые подсчитываются счетчиками и подаются на адресные входы мультиплексора и демультиплексора (глава 2). В результате происходит коммутация соответствующих светочувствительного и светоизлучающего элементов через канал связи. Частота переключений должна быть такой, чтобы время передачи одного кадра было бы не более $1/25$ с, — в этом случае человек не заметит смены кадров. Рассмотренная схема реализуется в цифровом телевидении, при передаче изображения по компьютерной сети с помощью Web-камеры, при записи изображения в цифровых фотоаппарате и видеокамере.

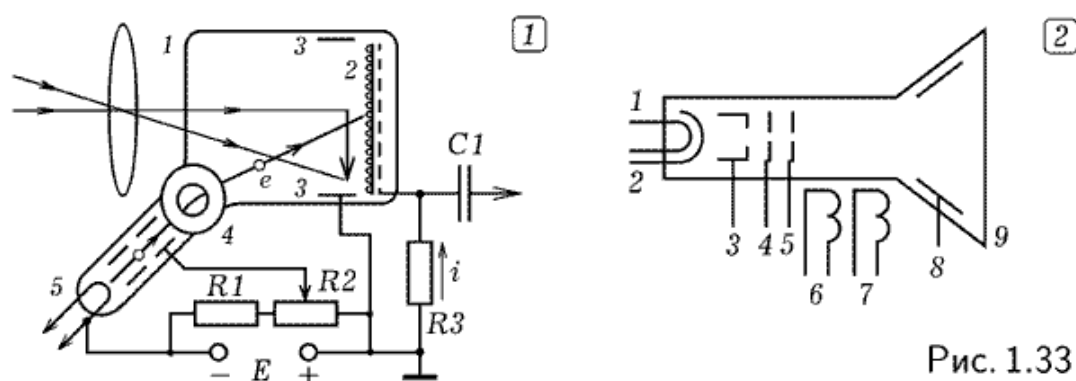


Рис. 1.33.

6. Передающая телевизионная трубка. Телевизионный сигнал.

Передача изображения может быть разложена на три процесса: 1) анализ изображения, то есть его преобразование в видеосигнал (в телевизионном передатчике); 2) передача сигнала по каналу связи; 3) синтез изображения, то есть его получение из электрического сигнала (в приемнике).

Для преобразования изображения движущегося предмета в последовательность электрических сигналов используются передающие телевизионные трубки (иконоскоп, ортископ, видикон и т.д.). Иконоскоп состоит из вакууммированного стеклянного баллона 1, сигнальной пластины 2, коллектора 3, магнитной отклоняющей системы 4, электронного прожектора 5 (рис. 1.33.1). Сигнальная пластина выполнена из слюды, на одну сторону которой нанесен слой металла, а на другую — мелкие зерна цезия (это металл с малой работой выхода электронов). С помощью объектива изображение предмета фокусируется на сигнальную пластину. Происходит внешний фотоэффект, свет выбивает из зерен электроны, при этом они заряжаются положительно. На сигнальной пластине запоминается

изображение предмета, образуется потенциальный (зарядовый) рельеф.

Электронный прожектор посылает электронный пучок, который под действием управляющего магнитного поля вычерчивает на сигнальной пластине *растр* — систему горизонтальных строк. Если зерно освещено, то оно заряжено положительно и захватывает электроны, при этом через резистор нагрузки течет ток. Для удаления медленных электронов, возникающих вблизи сигнальной пластины вследствие попадания в нее электронного пучка используется положительно заряженный электрод — коллектор.

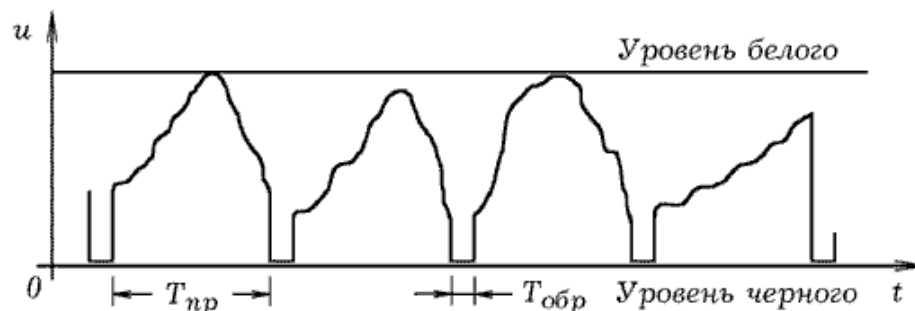


Рис. 1.34.

На рис. 1.34 изображена осциллограмма видеосигнала, формирующегося на резисторе нагрузки. Пока электронный луч движется в прямом направлении (время T_n), на выходе трубки возникает напряжение, пропорциональное распределению освещенности вдоль соответствующей строки. В течение времени обратного движения луча $T_{обр}$ напряжение обращается в ноль. В результате однократного сканирования всего кадра возникает последовательность импульсов, форма которых характеризует распределение освещенности внутри него. Полный управляющий телевизионный сигнал включает в себя видеосигнал, строковые и кадровые синхроимпульсы (рис. 1.35). Для получения из видеосигнала изображения используется кинескоп, в котором управление электронным пучком осуществляется с помощью магнитного поля (рис. 1.33.2). Он состоит из стеклянного вакуумированного баллона 9, нити накала 1, катода 2, модулирующего электрода 3, анода 4, фокусирующего электрода 5, обмоток вертикального и горизонтального отклонения 6 и 7. Внутренняя поверхность баллона вблизи экрана металлизирована и соединена с электродом 8 (аквадаг). Напряжение между катодом и аквадагом около 1 кВ.

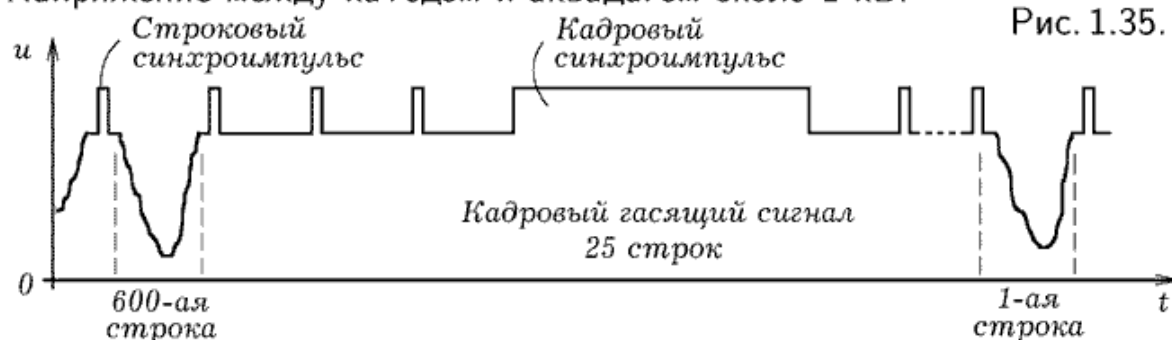


Рис. 1.35.

7. Параметры телевизионного сигнала. В системе телевизионного вещания реализуется принцип частотного разделения. По каналу связи (окружающему пространству) одновременно передается несколько

сигналов от различных телевизионных станций, работающих на разных несущих частотах. Переключая телевизионный канал, мы настраиваем телевизор на частоту, соответствующую требуемой телевизионной станции. Электронный луч вычерчивает растр, сканируя светочувствительный экран передающей телевизионной трубки, на который проецируется изображение предмета. Считанная информация об освещенности различных точек кадра передается последовательно (принцип временного разделения). Электронный луч приемной телевизионной трубки вычерчивает аналогичный растр, изменяя свою "яркость" и формируя изображение. Чтобы синхронизировать работу передающей и принимающей телевизионных трубок, передаваемый сигнал несет в себе синхроимпульсы строчной и кадровой разверток.

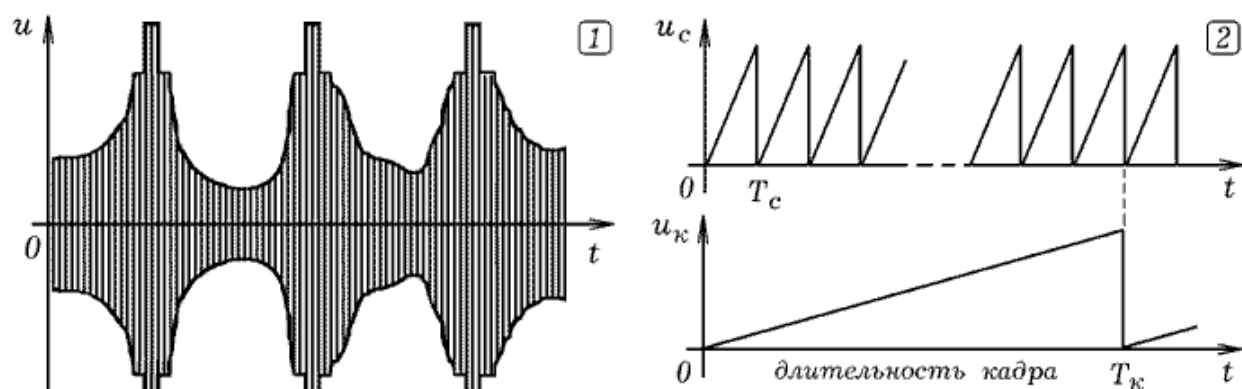


Рис. 1.36.

Соотношение сторон экрана телевизора, его разрешение, частота кадровой развертки, цветовые характеристики определяются физиологическими особенностями человека. В стандарте, принятом в СССР, отношение сторон экрана составляло $a/b = 4/3$, число строк (элементов изображения по вертикали) $N_y = 625$, количество элементов по горизонтали $N_x = (4/3) \cdot 625 \approx 830$. Общее количество элементов $N_{эл} = N_x \cdot N_y = (4/3) \cdot 625^2 \approx 521000$. Человеческий глаз в силу своей инерционности перестает замечать смену кадров при частоте $f_k = 25$ Гц. Частота генератора строковой развертки $f_{стр} = 25 \cdot 625 = 15625$ Гц.

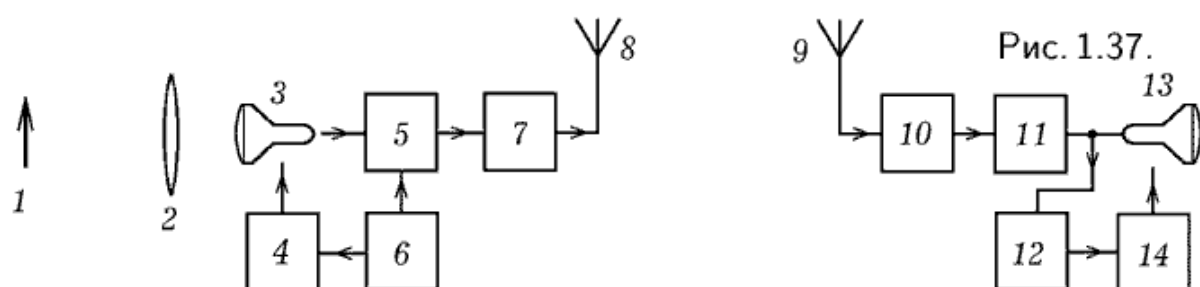


Рис. 1.37.

Рассчитаем наименьшую и наибольшую частоты видеосигнала для ЧБ телевизора. Наименьшей частоте соответствует ситуация, при которой верхняя половина экрана, допустим, белая, а нижняя — черная. Тогда $F_{min} = f_k = 25$ Гц. Максимальной частоте соответствует чередование белого элемента с черным, поэтому: $F_{max} = 0,5 f_k N_{эл} = 0,5 \cdot 25 \cdot 521000 =$

$6,5 \cdot 10^6$ Гц. По теореме Котельникова, чтобы закодировать видеосигнал с частотой 6,5 МГц, частота несущей должна быть не менее чем в 2 раза выше и составлять 13 МГц. В России сигнал звукового сопровождения модулируется по частоте. Частота несущей звукового сигнала превышает частоту несущей изображения на 6,5 МГц.

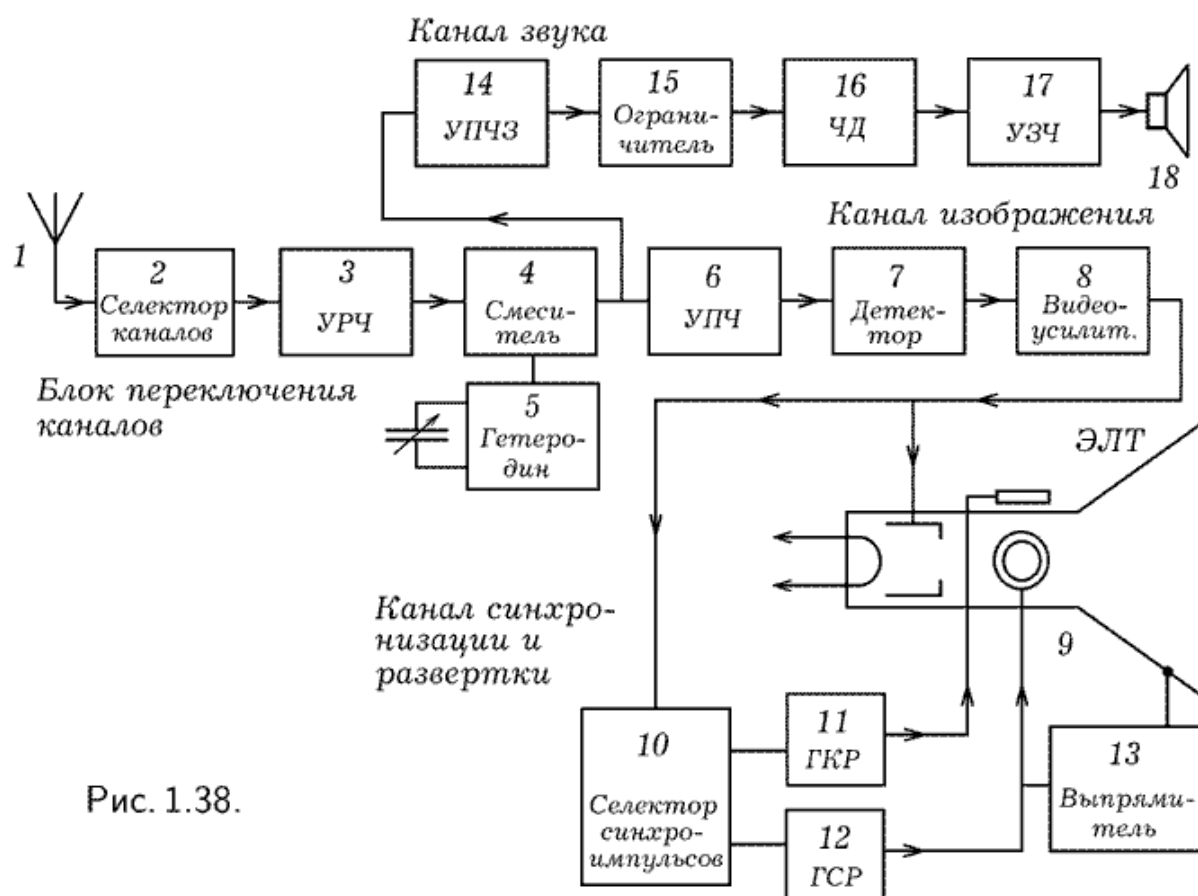


Рис. 1.38.

8. Структурная схема телевизионного передатчика и приемника.

Передатчик телевизионного изображения (рис. 1.37) состоит из передающей ЭЛТ 3, на которую с помощью объектива 2 проецируется изображение предмета 1, генератора развертки 4, видеоусилителя 5, синхрогенератора 6, радиопередатчика 7 и передающей антенны 8. Синхрогенератор 6 вырабатывает синхроимпульсы кадровой и строчной разверток, которые синхронизируют работу генератора развертки 4 и поступают в видеоусилитель 5, где они смешиваются с видеосигналом. Получается полный управляющий телевизионный сигнал, включающий в себя видеосигнал (сигнал яркости), в который введены строчные и кадровые импульсы для гашения обратного хода электронного луча в кинескопе в процессе телевизионной развертки (рис. 1.35). Синхроимпульсы находятся выше уровня гашения. В телевизионном передатчике 7 (рис. 1.37) осуществляется амплитудная модуляция колебаний несущей частоты полным управляющим сигналом. В результате получается телевизионный радиосигнал, подаваемый на передающую антенну (рис. 1.36.1).

Когда радиоволна достигает приемной антенны 9 телевизионного при-

емника (рис. 1.37), на ее выходе возникает телевизионный радиосигнал — высокочастотные колебания, амплитуда которых изменяется в такт с полным управляющим сигналом. В приемнике-супергетеродине 10 осуществляется преобразование принимаемых колебаний в колебания промежуточной частоты (около 35 МГц) с последующим детектированием. В результате выделяется полный управляющий сигнал, который подается в видеоусилитель 11. Видеосигнал подается на приемную телевизионную трубку 13 (кинескоп). Синхроимпульсы поступают в блок синхронизации 12, который управляет работой генератора строчной и кадровой развертки 14, вырабатывающего пилообразные напряжения с частотами f_c и f_k (рис. 1.36.2).

На рис. 1.38 приведена подробная блок-схема телевизионного приемника. Выделяют блок переключения каналов (селектор каналов 2, усилитель радиочастоты 3, гетеродин 5, смеситель 4), канал изображения (усилитель промежуточной частоты изображения 6, амплитудный детектор 7, видеоусилитель 8, кинескоп 9), блок синхронизации и развертки (селектор синхроимпульсов 10, генератор кадровой развертки 11, генератор строчной развертки 12), канал звука (усилитель промежуточной частоты звука 14, амплитудный ограничитель 15, частотный детектор 16, усилитель звуковой частоты 17, динамик 18). Выпрямитель 13 преобразует импульсы строчной развертки в постоянное высокое напряжение.

9. Цветное телевидение. Согласно трехкомпонентной цветовой модели RGB любой цвет может быть получен путем смешивания трех основных цветов: красного, зеленого и синего, взятых в соответствующих отношениях. Поэтому чтобы передать цветное изображение предмета необходимо получить его красную, зеленую и синюю составляющие, закодировать их с помощью электрических сигналов и послать по каналу связи. Представим себе иконоскоп, перед которым вращается диск, разделенный на три сектора: красный, зеленый, синий. Частота вращения диска должна быть синхронизирована с частотой кадровой развертки так, чтобы один кадр нес информацию о красной составляющей изображения, второй — о зеленой, третий — о синей и т.д.

В России, Франции, странах Восточной Европы используется система SECAM (SEquentiel Couleur A Memoire), в странах Западной Европы, Африке, Азии и Китае — PAL (Phase Alternation Line), в США, Канаде и Японии — NTSC (National Television System Color). В системе SECAM передающая телевизионная трубка формирует сигнал яркости $Y = 0,299R + 0,587G + 0,114B$ и еще два цвето-разностных сигнала $U = R - Y$ и $V = B - Y$ (красный минус яркостный и синий минус яркостный). Эти сигналы смешиваются с синхронизирующими и гасящими импульсами, в результате чего получается полный цветовой телевизионный сигнал, который поступает в модулятор. На его выходе получается телевизионный радиосигнал, который передается посредством радиоволн. В телевизионном приемнике происходит преобразование этих трех сигналов в четыре сигнала: яркостный, красный, зеленый и синий: $R = Y + U$, $B = Y + V$, $G = Y - 0,509U - 0,194V$.

Цветной кинескоп состоит из стеклянного баллона 1, трех электронных прожекторов 4, на модуляторы которых подаются сигналы, несущие информацию о красной, зеленой и синей составляющих изображения, а на общую цепь катодов — яркостный сигнал (рис. 1.39.1). Внутри кинескопа параллельно экрану установлена *тенивая маска* 2 — система струн или пластина с отверстиями (рис. 1.39.3). Управление электронным пучком осуществляется с помощью обмоток 3. Для создания осесимметричного ускоряющего поля и удаления электронов вблизи экрана используется аквадаг 5.

Экран разбит на пиксели, имеющие три сегмента. При попадании на них электронного луча сегменты светятся красным, зеленым и синим цветами (рис. 1.39.2). Три электронных луча проходят через одно отверстие, а затем расходятся, попадая в соответствующий сегмент (рис. 1.39.3). Изменяя величину модулирующих сигналов, подаваемых на электронные прожекторы, можно изменять соотношение уровня красного, зеленого и синего цветов и регулировать цвет пикселя. Электронные лучи вместе отклоняются магнитным полем обмоток вертикального и горизонтального отклонения. При этом они последовательно переходят от одного отверстия маски к другому, вычерчивая растр. Видеосигнал управляет яркостью этих лучей, формируя цветное изображение.

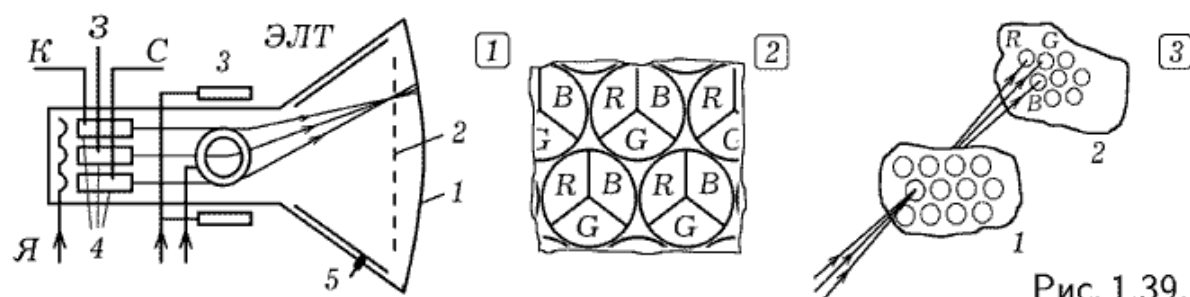
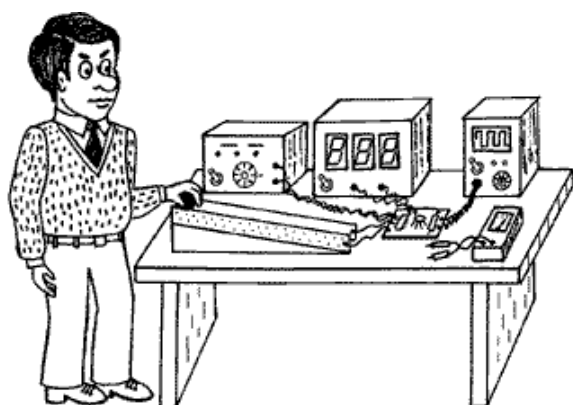


Рис. 1.39.

В настоящее время вместо передающих телевизионных трубок используются цифровые видеокамеры, содержащие двумерную фотоПЗС матрицу, которая оцифровывает изображение, формируя видеосигнал. Широкое распространение получили телевизоры с жидкокристаллическими мониторами. Видеосигнал с помощью видеомagnetofона может быть записан на магнитофонную ленту. Если звуковой сигнал содержит до 10^4 бит в секунду, то видеосигнал — до 10^7 бит в секунду. Записывающие и воспроизводящие головки видеомagnetofона укреплены на диске, который наклонен по отношению к ленте и быстро вращается. В результате на движущейся магнитной ленте образуются наклонные дорожки с записанной информацией.



ГЛАВА 2 ИЗУЧАЕМ ЦИФРОВУЮ ЭЛЕКТРОНИКУ

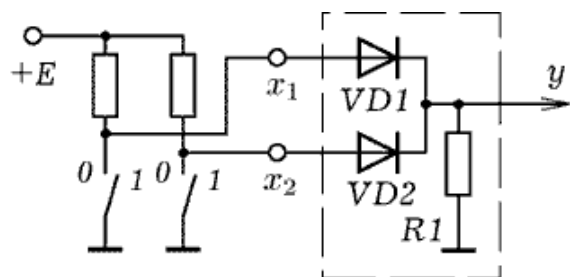
”Одним из важнейших технологических прорывов является перевод российского ТВ на цифру”, — так была сформулирована задача, стоящая перед конструкторами-телевизионщиками. Цифровой телевизор, цифровой фотоаппарат, цифровая видеокамера, цифровая ЭВМ, цифровой мультиметр — этот перечень можно продолжить. Что представляют собой цифровые приборы и в чем их преимущество? Чтобы понять это, придется начать с азов цифровой электроники — логических элементов, триггеров, счетчиков, дешифраторов, мультиплексоров ...

2.1. ОСНОВЫ ЦИФРОВОЙ ЭЛЕКТРОНИКИ

1. Аналоговый, цифровой и дискретный сигналы. Сигнал — это изменение некоторой физической величины (напряжения, громкости, освещенности и т.д.), которое переносит информацию. Виды электрических сигналов: 1) *аналоговый* — напряжение принимает всевозможные значения в интервале от U_{min} до U_{max} (сигнал на выходе микрофона); 2) *цифровой* — имеется два устойчивых уровня напряжения: напряжение высокого уровня ($U > 2,4 \text{ В}$) и напряжение низкого уровня ($U < 0,4 \text{ В}$) (сигнал на выходе логического элемента); 3) *дискретный* — напряжение принимает конечный набор дискретных значений $U = U_0 + i\Delta U$, где $i = 0, 1, \dots$ (сигнал на выходе цифро-аналогового преобразователя без фильтра).

Современные электронно-цифровые приборы созданы на основе логических элементов ИЛИ, И, НЕ. Эти элементы позволяют реализовать все возможные логические функции и построить цифровую ЭВМ. В настоящее время используются интегральные микросхемы, содержащие до нескольких сотен и тысяч логических элементов.

Минимальный теоретический предел для энергии переноса единицы информации должен быть сравним с энергией элементарного шумового выброса, которая равна $E_{min} = kT$. При комнатной температуре $T = 300 \text{ К}$ величина E_{min} составляет $1,38 \cdot 10^{-23} \cdot 300 = 4 \cdot 10^{-21} \text{ Дж}$. Эта величина на 9 порядков меньше энергии, потребляемой микросхемой для обработки 1 бита информации. В настоящее время широко используются большие интегральные микросхемы, содержащие более 10^5 транзисторов. Это микропроцессоры, контроллеры, запоминающие устройства и т.д. Например, микропроцессор на 32 разряда имеет около $3,5 \cdot 10^5$ транзисторов.

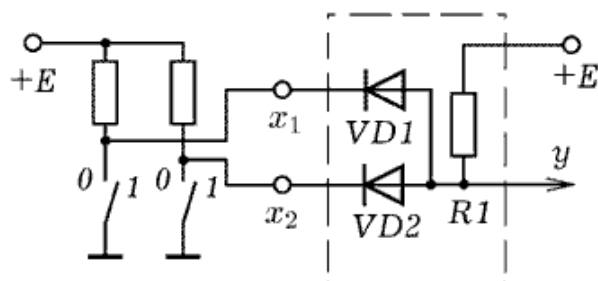


| x_1 | x_2 | y |
|-------|-------|-----|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Элемент ИЛИ
 $y = x_1 \text{ OR } x_2$



Рис. 2.1.



| x_1 | x_2 | y |
|-------|-------|-----|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Элемент И
 $y = x_1 \text{ AND } x_2$

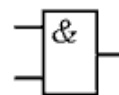
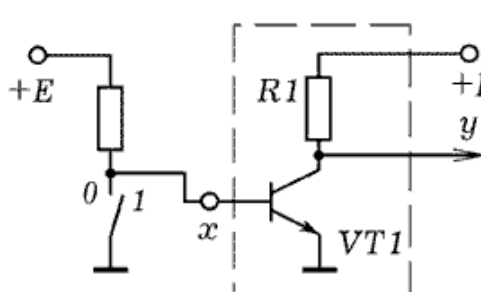


Рис. 2.2.



Элемент НЕ
 $y = \text{NOT } x$

| x | y |
|-----|-----|
| 0 | 1 |
| 1 | 0 |

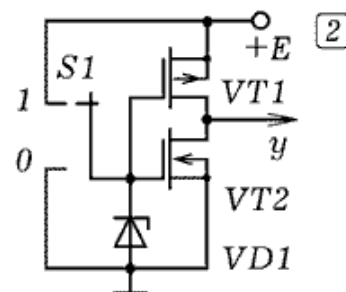
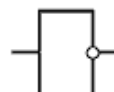
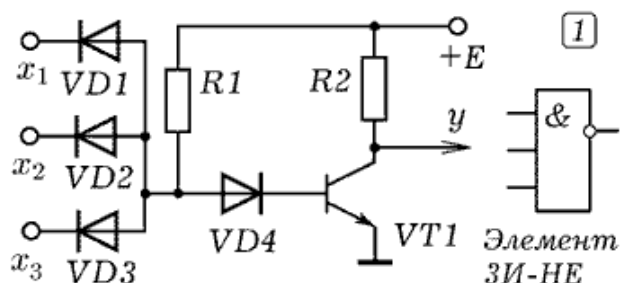
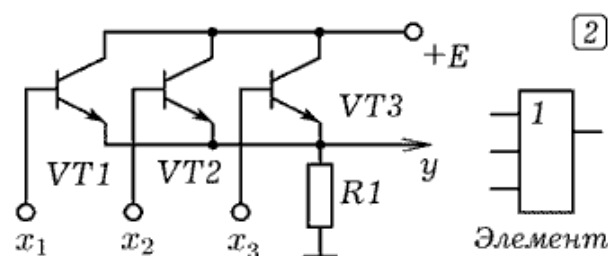


Рис. 2.3.

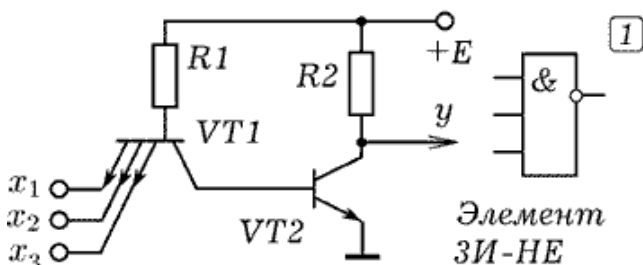


Элемент 3И-НЕ

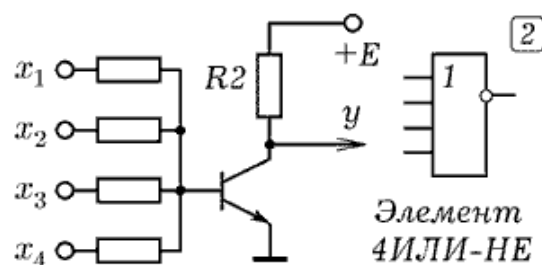


Элемент 3ИЛИ

Рис. 2.4.



Элемент 3И-НЕ



Элемент 4ИЛИ-НЕ

Рис. 2.5.

2. Логические цифровые элементы. Цифровые приборы собраны на логических элементах — устройствах, которые реализуют простейшие логические операции над входными сигналами. К ним относятся логическое сложение (операция ИЛИ, дизъюнкция), логическое умножение (операция И, конъюнкция), логическое отрицание (операция НЕ, инверсия). Получили распространение элементы резисторно-транзисторной логики (РТЛ), диодно-транзисторной логики (ДТЛ), транзисторно-транзисторной логики (ТТЛ), диодно-диодной логики (ДДЛ), резисторно-транзисторной логики (РТЛ). На рис. 2.1, 2.2 изображены схемы логических элементов ИЛИ и И (ДДЛ), их условные обозначения и представлены таблицы истинности. Если хотя бы на один вход элемента ИЛИ подать лог.1 (соединить с проводом $+E$), то на выходе — лог. 1. Если оба входа соединить с общим, то есть подать лог. 0, то на выходе — лог. 0 (рис. 2.1).

При подаче на оба входа элемента И (рис. 2.2) лог. 1 на выходе получается лог. 1. Если хотя бы на один из входов подать лог. 0, то на выходе — лог. 0. Инвертор (элемент НЕ) на транзисторе работает так (рис. 2.3.1): при подаче на вход лог. 1 транзистор $n-p-n$ открывается, сопротивление коллектор-эмиттер падает, выход y оказывается соединенным с общим, на нем лог. 0. Если на вход подать лог. 0, транзистор закроется, на выходе — лог. 1.

Возможно создание комбинированных схем, выполняющих несколько логических операций. На рис. 2.4.1 изображена схема элемента ЗИ-НЕ (ДТЛ). Если хотя бы на одном входе x_1, x_2, x_3 лог. 0 (соединен с общим), то потенциал базы VT1 невысок, транзистор закрыт, на выходе — лог. 1. Когда на всех входах лог. 1, на базе VT1 положительный потенциал, транзистор открыт, на выходе — лог. 0.

На рис. 2.4.2 представлена схема ЗИЛИ (ТТЛ). При подаче хотя бы на один вход лог. 1 соответствующий транзистор открывается, на выходе появляется лог. 1. Если на всех входах лог. 0 — все транзисторы закрыты, на выходе лог. 0. Элемент ЗИ-НЕ (ТТЛ) может быть создан на базе трехэмиттерного транзистора (рис. 2.5.1). При подаче на все 3 входа лог. 1, транзистор VT1 закрыт, транзистор VT2 открыт, на выходе лог. 0. Если хотя бы на один вход подать лог. 0, VT1 откроется (на его базе — положительный потенциал), и на базе VT2 окажется лог. 0. Транзистор VT2 закроется, на выходе — лог. 1.

На рис. 2.5.2 показан элемент 4ИЛИ-НЕ (РТЛ). Обычно логические элементы содержат выходной каскад (рис. 2.6.1), который позволяет получить напряжения, соответствующие лог. 0 и лог. 1. Часто в одной микросхеме размещают несколько логических элементов (рис. 2.6.4).

Большое распространение получили микросхемы типа КМОП и КМДП. Эти сокращения расшифровываются как комплиментарные полевые транзисторы со структурой металл-оксид-полупроводник или металл-диэлектрик-полупроводник. Комплиментарными называются взаимодополняющие транзисторы с одинаковыми параметрами, но противоположной проводимости.

Например, один биполярный транзистор р-п-р-типа, а другой п-р-п-типа, один полевой транзистор с р-каналом, другой — с п-каналом. На рис. 2.3.2, 2.6.2, 2.6.3 показаны схемы НЕ, И-НЕ и ИЛИ-НЕ на комплементарных полевых транзисторах.

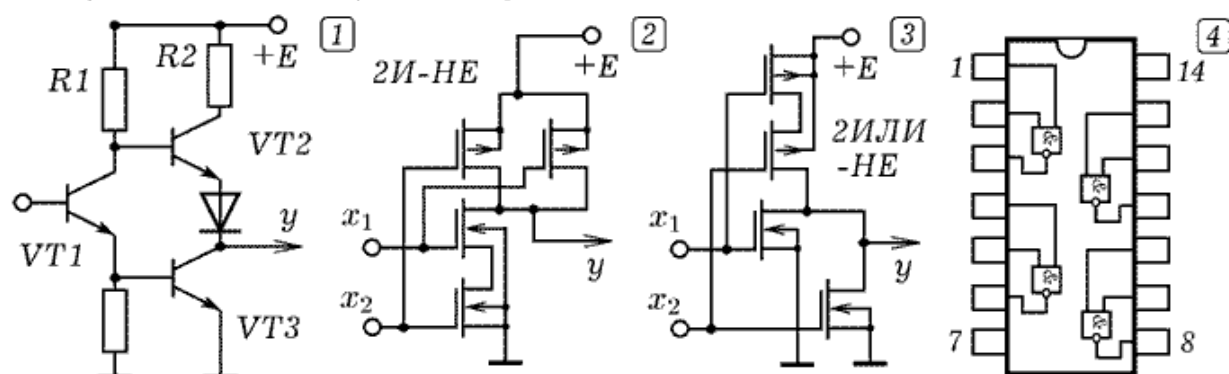


Рис. 2.6.

На базе логических элементов И, ИЛИ, НЕ можно создать все основные узлы ЭВМ, к которым относятся регистры памяти, сумматор, счетчик, шифратор и дешифратор.

3. Двухстабильные системы — триггеры. Обработка информации требует ее хранения, для чего используются различные типы запоминающих устройств. Элементарной ячейкой статической памяти (SRAM) являются триггеры — двухстабильные системы, способные хранить 1 бит информации. Простейший триггер состоит из двух инверторов, выход каждого из которых соединен с входом другого (рис. 2.8.2).

Асинхронный RS-триггер состоит из двух элементов 2ИЛИ-НЕ, соединенных симметрично (рис. 2.7.1). Он имеет два входа: S — set (установить), R — reset (переустановить). При $S=0, R=0$ реализуется режим хранения информации, если $S=1, R=0$ — запись лог. 1, если $S=0, R=1$ — запись лог. 0. Состояние $S=1, R=1$ не используется. Выход \bar{Q} является инвертирующим по отношению к выходу Q. Схема триггера на элементах 2И-НЕ приведена на рис. 2.7.2.

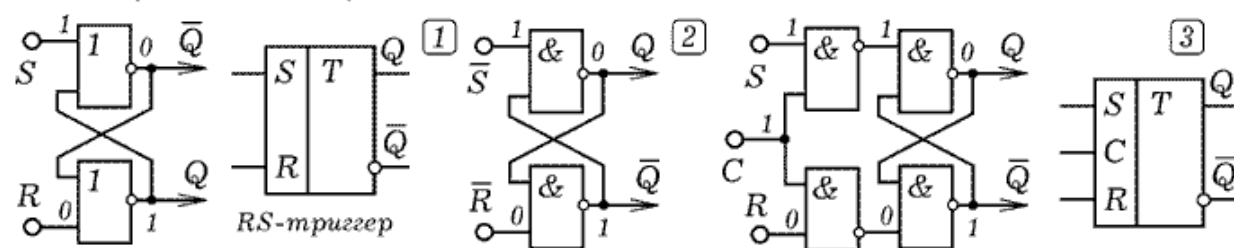


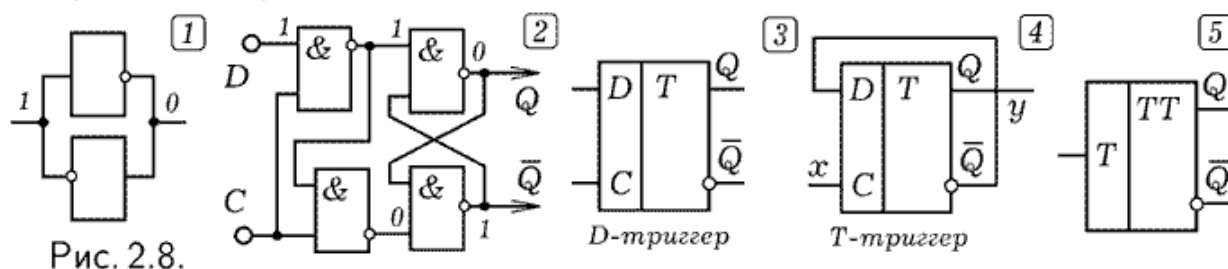
Рис. 2.7.

Синхронный триггер может быть собран на четырех элементах 2И-НЕ (рис. 2.7.3). Пока не пришел тактовый импульс (синхроимпульс) на вход C, состояние триггера не изменяется при любых состояниях R- и S-входов. Триггер опрокидывается в другое состояние с приходом тактового импульса ($C=1$).

D-триггер имеет один управляющий вход D и синхронизирующий вход C (рис. 2.8.2, 2.8.3). При $C=0$ состояние выхода не изменяется (режим

хранения). При поступлении тактового импульса ($C=1$) состояние выхода Q оказывается таким же, что и состояние входа D .

На основе D -триггера можно создать T -триггер, который "опрокидывается" в противоположное состояние по фронту входного импульса, то есть каждый раз, когда на входе C лог.0 сменяется на лог.1. Для этого выход \bar{Q} следует соединить с входом D (рис. 2.8.4). Тогда с фронтом входного импульса, подаваемого на вход C , триггер переключается из $Q=0$ в $Q=1$, а из $Q=1$ в $Q=0$ (рис. 2.10.1). Условное обозначение T -триггера изображено на рис. 2.8.5.



У JK -триггера имеется вход установки J , вход сброса K и два выхода Q и \bar{Q} . Когда на входе J лог.0 сменяется на лог.1, то независимо от состояния входа K , на выходе Q лог.1. При переходе входа K из состояния 0 в состояние 1 независимо от состояния входа J на выходе Q появляется лог.0. Одновременный переход входов J и K из состояния 0 в состояние 1 вызывает опрокидывание триггера в противоположное состояние. Смена лог.1 на лог.0 на любом из входов никак не сказывается на состоянии выходов триггера.

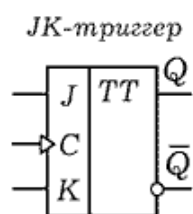


Рис. 2.9.

| C | J | K | Q_{t+1} | \bar{Q}_{t+1} | Режим |
|--------------|-----|-----|-------------|-----------------|------------|
| 0 или 1 | * | * | Q_t | \bar{Q}_t | Хранение |
| Спад 1 на 0 | * | * | Q_t | \bar{Q}_t | Хранение |
| Фронт 0 на 1 | 0 | 1 | 0 | 1 | Запись "0" |
| Фронт 0 на 1 | 1 | 0 | 1 | 0 | Запись "1" |
| Фронт 0 на 1 | 0 | 0 | Q_t | \bar{Q}_t | Хранение |
| Фронт 0 на 1 | 1 | 1 | \bar{Q}_t | Q_t | Инверсия |

Тактируемый JK -триггер кроме входов J и K имеет синхронизирующий вход C , его условное обозначение и таблица состояний приведены на рис. 2.9. Переход входа C из состояния $C=0$ в состояние $C=1$ (фронт синхроимпульса) вызывает: 1) $Q=0$, $\bar{Q}=1$, если $J=0$, $K=1$; 2) $Q=1$, $\bar{Q}=0$, если $J=1$, $K=0$; 3) сохранение предшествующего состояния, если $J=0$, $K=0$; 4) изменение выходного состояния на противоположное, если $J=1$, $K=1$. Триггер не изменяет своего состояния при отсутствии синхроимпульсов и не реагирует на смену $C=1$ на $C=0$ (хранение).

4. Параллельный и последовательный регистры памяти. Регистр статической памяти состоит из триггеров, каждый из которых хранит 1 бит информации. *Параллельный регистр* для хранения n –

разрядного двоичного числа содержит n D-триггеров, включенных "параллельно", как показано на рис. 2.10.2. Запись информации происходит после прихода тактового импульса на синхронизирующий вход С. Все триггеры устанавливаются в состояние, соответствующее состоянию входов A_1, A_2, \dots, A_n . Так, если на n входов поступило двоичное слово 101...1, то после прихода тактового импульса на выходах B_1, B_2, \dots, B_n регистра памяти появляется то же двоичное слово 101...1. Чтобы стереть информацию на вход переустановки (сброса) R следует подать лог. 1.

Сдвиговые или последовательные регистры состоят из D-триггеров, соединенных "последовательно", то есть друг за другом (рис. 2.11.1). С приходом каждого тактового импульса последовательный регистр принимает и выдает информацию. При этом входная последовательность нулей и единиц сдвигается вправо как единое целое на один разряд. Сдвиговые регистры используются в АЛУ для умножения и деления. Сдвиг двоичного числа влево на n разрядов означает умножение на 2^n , сдвиг вправо — умножение на 2^{-n} . Также они используются для задержки информации на n тактов машинного времени. На их основе создаются задерживающие цепочки FIFO (First Input First Output — первым вошел, первым вышел) и стековая память FILO (First Input Last Output — первым вошел, последним вышел).

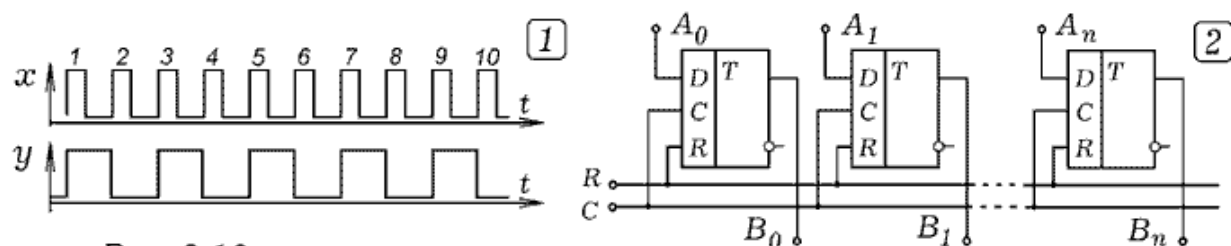


Рис. 2.10.

5. Счетчик, шифратор и дешифратор. Счетчик осуществляет счет числа импульсов и фиксацию результата в двоичном коде. Схема последовательного счетчика на T-триггерах изображена на рис. 2.11.2. При подаче сигнала на вход сброса R=1 все триггеры переходят в состояние $Q_i = 0$. Цепочка из n триггеров считает до 2^n , затем полностью обнуляется и все повторяется снова. Если на вход x поступают импульсы частотой f (рис. 2.12.1), то на выходе первого триггера получаются импульсы с частотой $0,5f$, на выходе второго — с частотой $0,25f$ и т.д. В результате на выходах A_0, A_1, A_2, A_3 получается двоичное число, равное количеству поступивших импульсов.

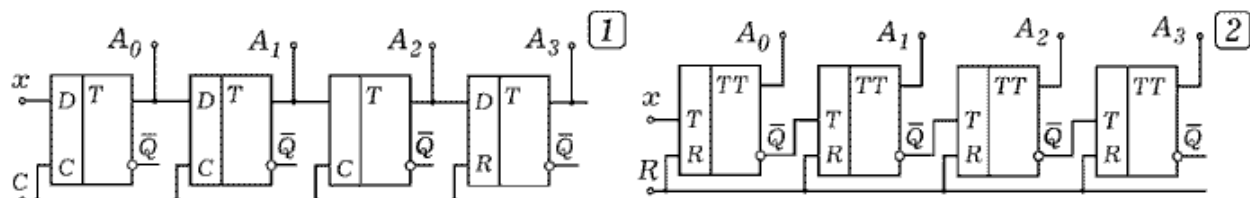


Рис. 2.11.

Шифратор — узел ЭВМ, переводящий число из системы счисления m в систему счисления n . Шифраторы используются для передачи ин-

формации между устройством управления, запоминающим устройством и устройствами ввода-вывода при ограниченном числе линий связи, а также для преобразования вводимых в ЭВМ чисел в двоичный или двоично-десятичный код. Рассмотрим шифратор "из 10 в 4" (рис. 2.13.1). Его схема представляет собой матрицу диодов — систему вертикальных и горизонтальных проводов, связанных между собой диодами. При нажатии, например, на ключ S_5 соответствующий горизонтальный провод оказывается соединенным с общим, через подключенные к нему диоды текут токи, на выходах A_3, A_2, A_1, A_0 появляется двоичное число 0101.

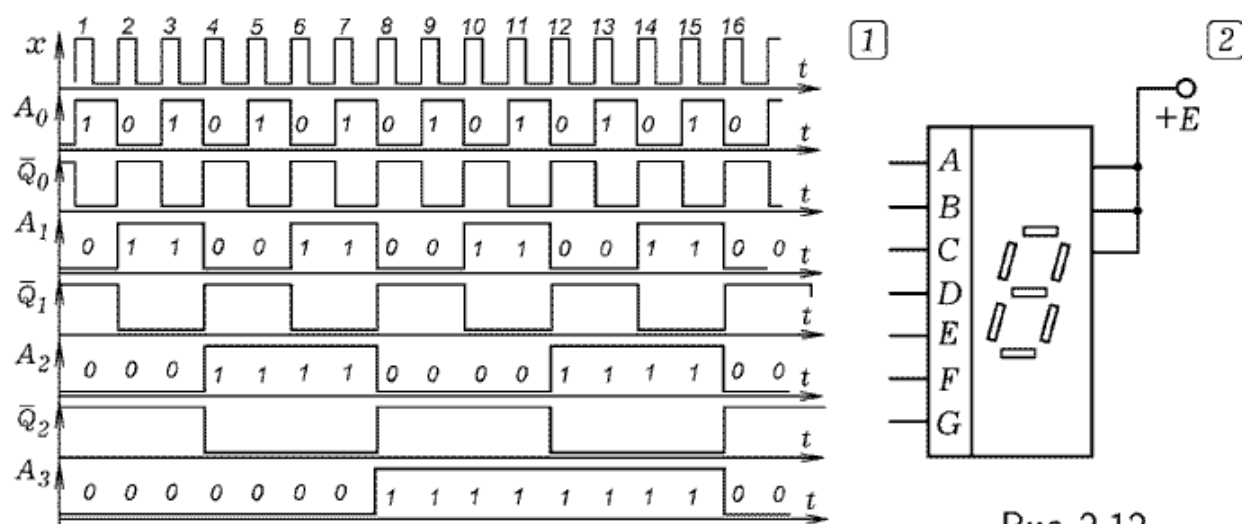


Рис. 2.12.

Дешифратор осуществляет обратное преобразование и используется в устройствах вывода информации. Дешифратор "из 3 в 8" состоит из параллельного регистра из трех RS-триггеров и матрицы диодов (рис. 2.13.2). Пусть на вход дешифратора поступает число 110. В этом случае на выходах Q триггеров T_2, T_3 и \bar{Q} триггера T_1 — лог. 1, подключенные к ним диоды закрыты. На остальных выходах триггеров — лог. 0, подключенные к ним диоды открыты, на соответствующих горизонтальных проводах — лог. 0. Лишь один горизонтальный провод B_6 не соединен с землей, на нем — лог. 1, что означает число 6.

6. Многосегментный индикатор. В устройствах визуальной индикации десятичных и шестнадцатеричных цифр используются многосегментные знаковосинтезирующие индикаторы на светодиодах или жидких кристаллах. Большое распространение получили семисегментные индикаторы, состоящие из 7 сегментов, которые излучают свет (светодиоды) или изменяют свою прозрачность (элементы на жидких кристаллах). Условное обозначение семисегментного индикатора изображено на рис. 2.12.2. Для его согласования со счетчиком используется дешифратор, преобразующий двоичный код в семисегментный. Это хорошо видно из схемы, представленной на рис. 2.20. На вход такого дешифратора поступает четырехразрядное двоичное число $\overline{a_3 a_2 a_1 a_0}$, при этом на выходах появляется комбинация лог. 0 и 1. Появление лог. 0 вызывает свечение соответствующего элемента индикатора (см. таблицу на рис. 2.14).

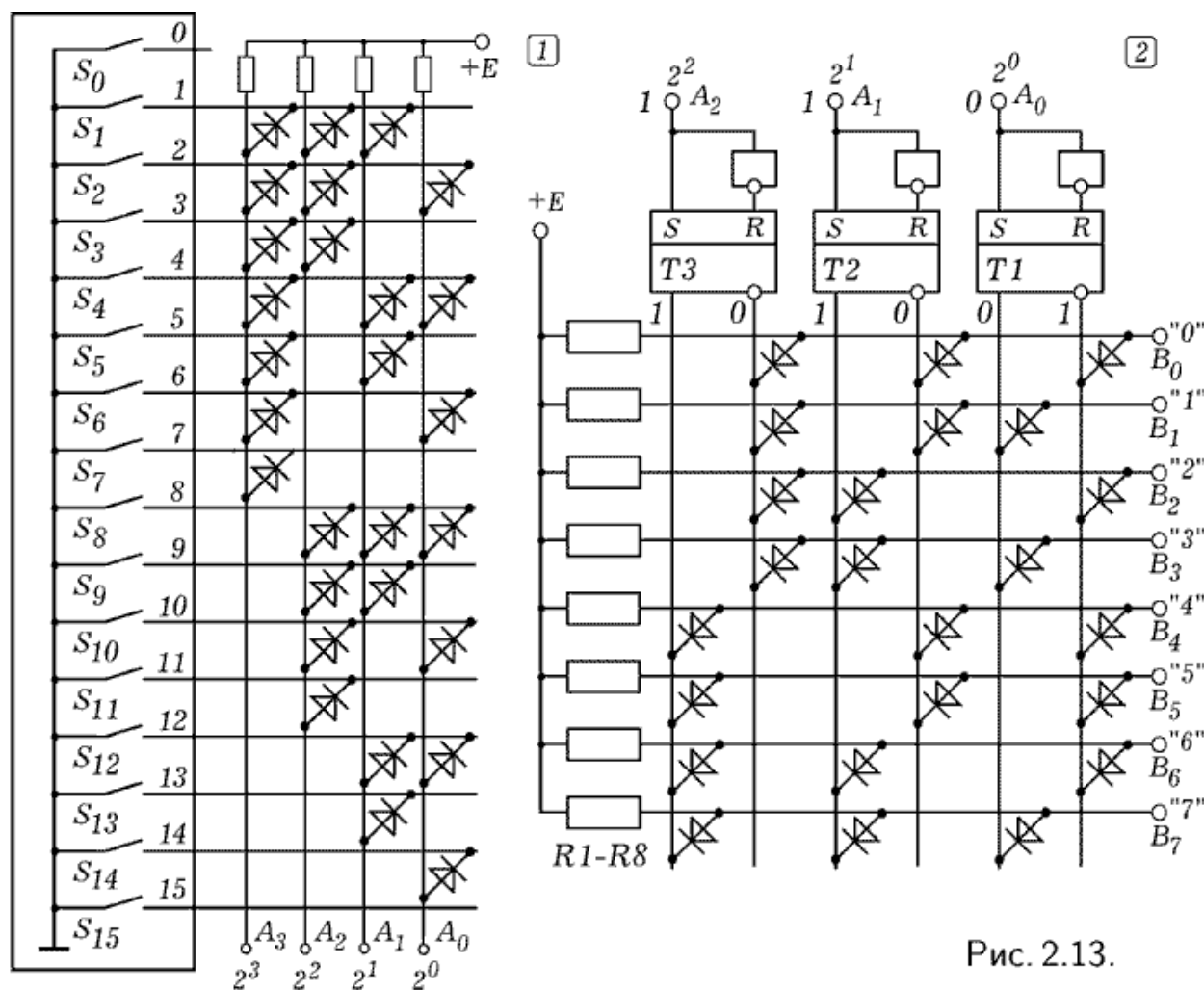


Рис. 2.13.

| C | a_3 | a_2 | a_1 | a_0 | F_0 | F_1 | F_2 | F_3 | F_4 | F_5 | F_6 |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 7 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

Рис. 2.14.

7. Одноразрядный и многоразрядный сумматор. Сумматор выполняет сложение двух двоичных чисел. Одноразрядным сумматором является логический элемент исключающее ИЛИ (рис. 2.15.1), выполняющий операцию сложения по модулю 2 (операция *xor*, рис. 2.16.1). Его отличие от элемента ИЛИ в том, что $1 \text{ xor } 1 = 0$, а у элемента ИЛИ $1 \text{ or } 1 = 1$. Параллельные сумматоры n -разрядных чисел состоят из n одноразрядных сумматоров, которые соединены цепями переноса из младшего разряда в старший. На рис. 2.15.2 и 2.15.3 изображены схемы одноразрядных

сумматоров. Таблица истинности для схемы на рис. 2.15.3 приведена на рис. 2.16.2. На входы сумматора поступают две двоичные цифры A_i и B_i и цифра переноса C_i из младшего разряда. На выходе — сумма S_i и цифра переноса C_{i+1} в старший разряд. Многоразрядный сумматор состоит из нескольких одnorазрядных, соединенных между собой цепью переноса разряда.

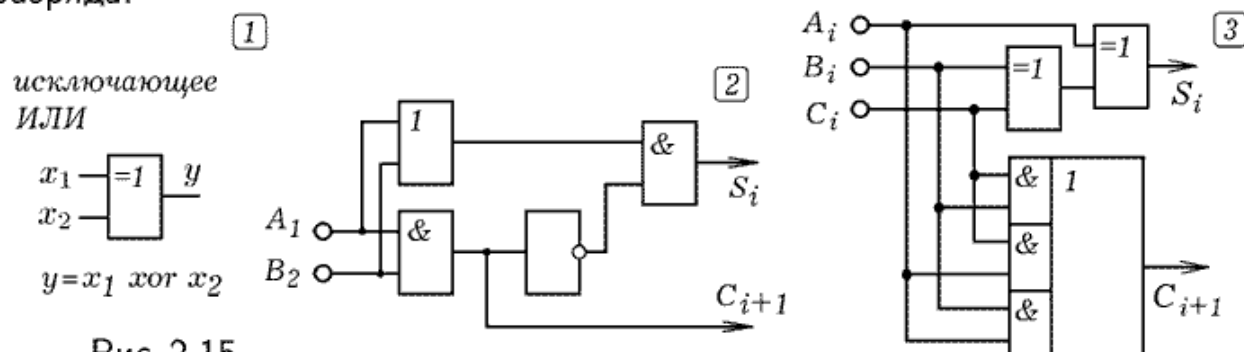


Рис. 2.15.

Операция сложения позволяет реализовать все остальные арифметические действия. Чтобы из двоичного числа a вычесть b , необходимо их перевести в дополнительный код, а затем сложить. Умножение или деление можно представить как многократное сложение или вычитание. В состав микропроцессора входит арифметико-логическое устройство, содержащее комбинационный сумматор (рис. 2.16.3). Он состоит из двух регистров 1 и 2 для хранения операндов, собственно сумматор и регистр 3, в который записывается результат сложения. При поступлении сигнала от устройства управления происходит сложение и результат записывается в регистр 3, откуда он может переслан в регистр 1.

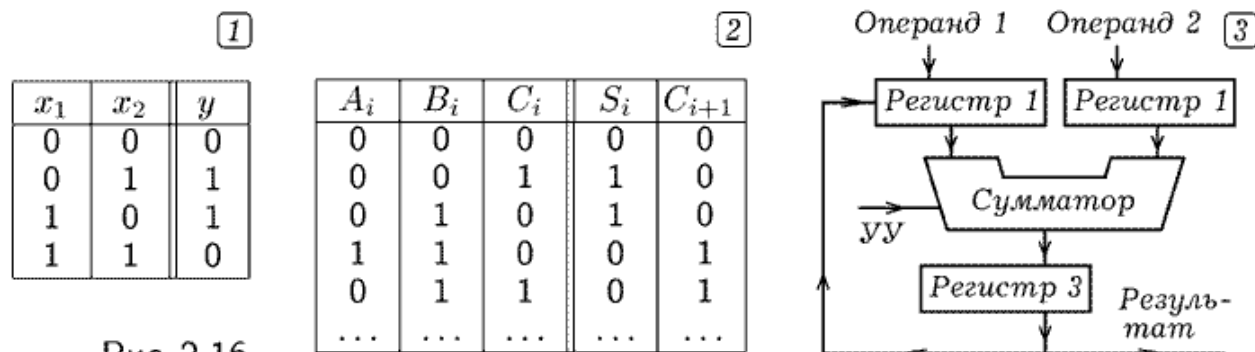


Рис. 2.16.

8. Мультиплексор и демультиплексор. А как сделать так, чтобы по одному каналу связи одновременно передавались бы сообщения от нескольких источников? Технически это можно осуществить двумя способами: 1) путем *частотного разделения*: каждое сообщение переносится колебаниями определенной частоты, которые выделяются приемным устройством с помощью резонансных фильтров; 2) путем *временного разделения*: в течение небольшого промежутка времени Δt (например, 1 мс) по каналу связи передается информация от источника 1, в течение следующего промежутка времени Δt — от источника 2 и т.д., после чего все повторяется снова с высокой частотой. Временное разделение сигналов также называют *мультиплексированием*.

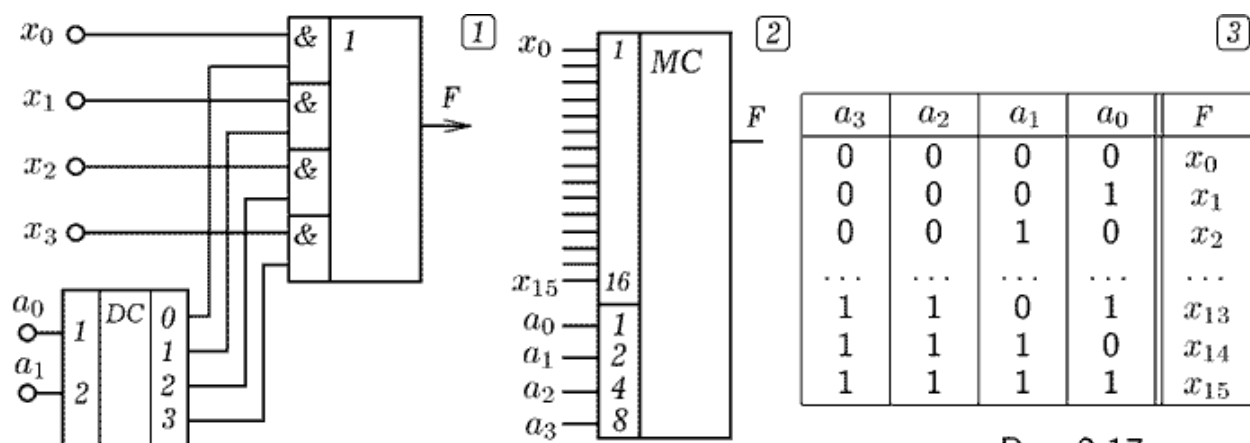


Рис. 2.17.

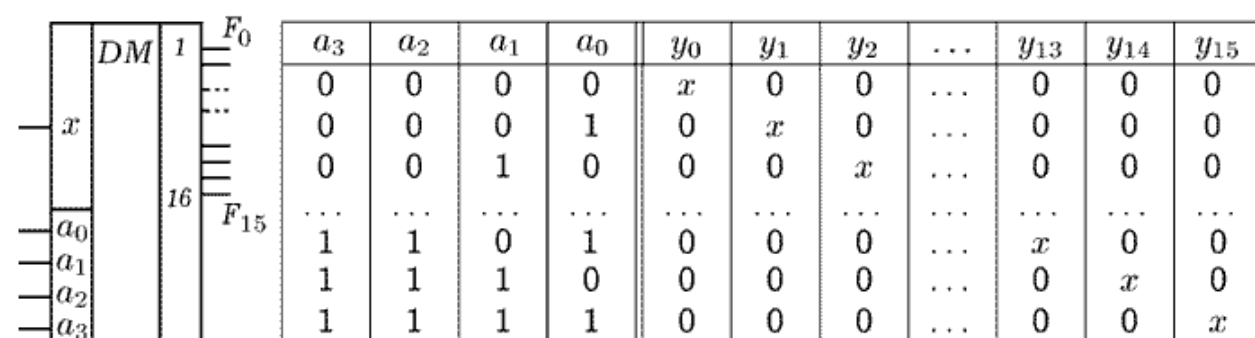


Рис. 2.18.

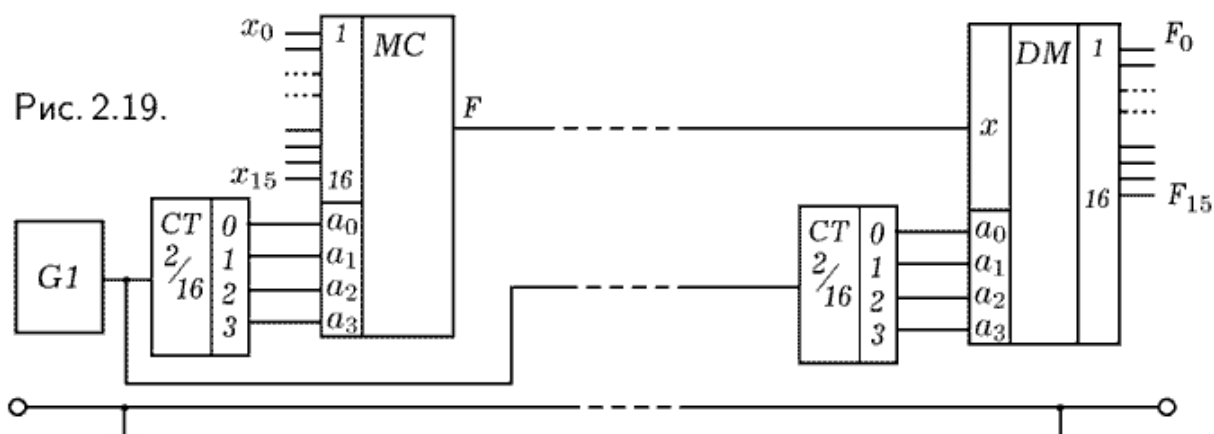


Рис. 2.19.

Мультиплексор — это цифровой многопозиционный переключатель с $n = 2^m$ информационными входами x_0, x_1, \dots, x_{n-1} , с m адресными входами a_0, a_1, \dots, a_{m-1} и одним выходом F (рис. 2.17.2). При подаче на адресные входы двоичного кода соответствующий информационный вход соединяется с выходом в соответствии с таблицей, приведенной на рис. 2.17.3. Принципиальная схема мультиплексора на 4 входа изображена на рис. 2.17.1. На адресные входы a_0, a_1 подается двоичное слово 00, 01, 10 или 11, при этом на 0, 1, 2 или 3 выходе дешифратора появляется лог. 1 и сигнал с информационного входа x_0, x_1, x_2, x_3 поступает на выход F .

Для восстановления мультиплексированной информации используется демultipлексор, имеющий один информационный вход x , m адресных



МАРКОВ Андрей Андреевич (1903–1979 гг.) Окончил Петроградский университет. Занимался разработкой проблем математики и логики, теорией множеств, теоретической физики. Предложил теорию нормальных алгорифмов. Марковские процессы, марковский источник информации, нормальная система подстановок Маркова — вот далеко не полный список терминов, которыми обогатилась наука. Член-корреспондент АН СССР.

входов a_0, a_1, \dots, a_{m-1} и $n = 2^m$ выходов F_0, F_1, \dots, F_{n-1} (рис. 2.18). На адресные входы подается двоичный код, например, $1010_2 = 10_{10}$, и информационный вход коммутируется с десятым выходом демультиплексора. Схема коммутации четырех каналов связи на один содержит мультиплексор, демультиплексор, генератор тактовых импульсов и два счетчика (рис. 2.19).

9. Технология изготовления интегральных схем. Центральным процессом в создании интегральных схем (ИС) является *фотолитография* — нанесение технологического шаблона на поверхность кремниевой пластины с помощью света. При этом используется светочувствительный материал — *фоторезист*, которым покрывают кремниевую пластину, служащую *подложкой*. После этого поверхность освещается через фотошаблон, содержащий прозрачные и непрозрачные участки. Объектив создает уменьшенное изображение фотошаблона, в слое фоторезиста формируется его скрытое изображение. Затем пластину обрабатывают химическим реактивом, который растворяет фоторезист с освещенных участков.

В слое фоторезиста образуются окна, через которые в пластину кремния вводят акцепторные или донорные примеси, формируя области *p* или *n*-типа. Это осуществляется методом диффузии: пластину вводят в камеру, наполненную парами той или иной примеси. Повторяя этот процесс с различными фотошаблонами и примесями, создают на пластине слой состоящий из большого количества транзисторов, диодов, конденсаторов и т.д. После этого формируют второй, третий и последующие слои т.д. В настоящее время на одной пластине изготавливают более десятка микросхем, а затем пластину разрезают, соответствующие участки металлизуют и к ним припаивают выводы.

2.2. КАК РАБОТАЮТ ЦИФРОВЫЕ ПРИБОРЫ

В настоящее время в технике получили широкое распространение различные цифровые измерители времени, напряжения, силы тока, сопротивления, температуры, частоты и других физических величин. В основе их работы лежит преобразование измеряемой величины в последовательность импульсов известной частоты и их подсчет. Также используются аналого-цифровой и цифро-аналоговый преобразователи. Для хранения цифровой информации применяются запоминающие устройства.

1. Счетчик импульсов. Измерение времени и частоты. Любой цифровой измерительный прибор, например, мультиметр или электронные часы, содержит счетчик импульсов. Как он устроен? Двухразрядный счетчик импульсов (рис. 2.20) состоит из двоично-десятичных счетчиков DD1 и DD2, дешифраторов DD3, DD4 и семисегментных индикаторов HG1 и HG2. При поступлении на вход x последовательности импульсов, на четырех выходах (выводы 12, 9, 8, 11) счетчика DD1 появляется двоичное число, равное количеству этих импульсов. Оно подается на дешифратор DD3, который преобразует двоичный код в семисегментный и вызывает свечение соответствующих сегментов индикатора HG1. Когда число поступивших импульсов достигает 10, счетчик обнуляется. Импульс с выхода $2^3 = 8$ (вывод 11) подается на счетчик DD2, соответствующий второму разряду. При нажатии на кнопку сброса S1 происходит обнуление счетчиков DD1 и DD2, на индикаторах HG1 и HG2 появляются нули.

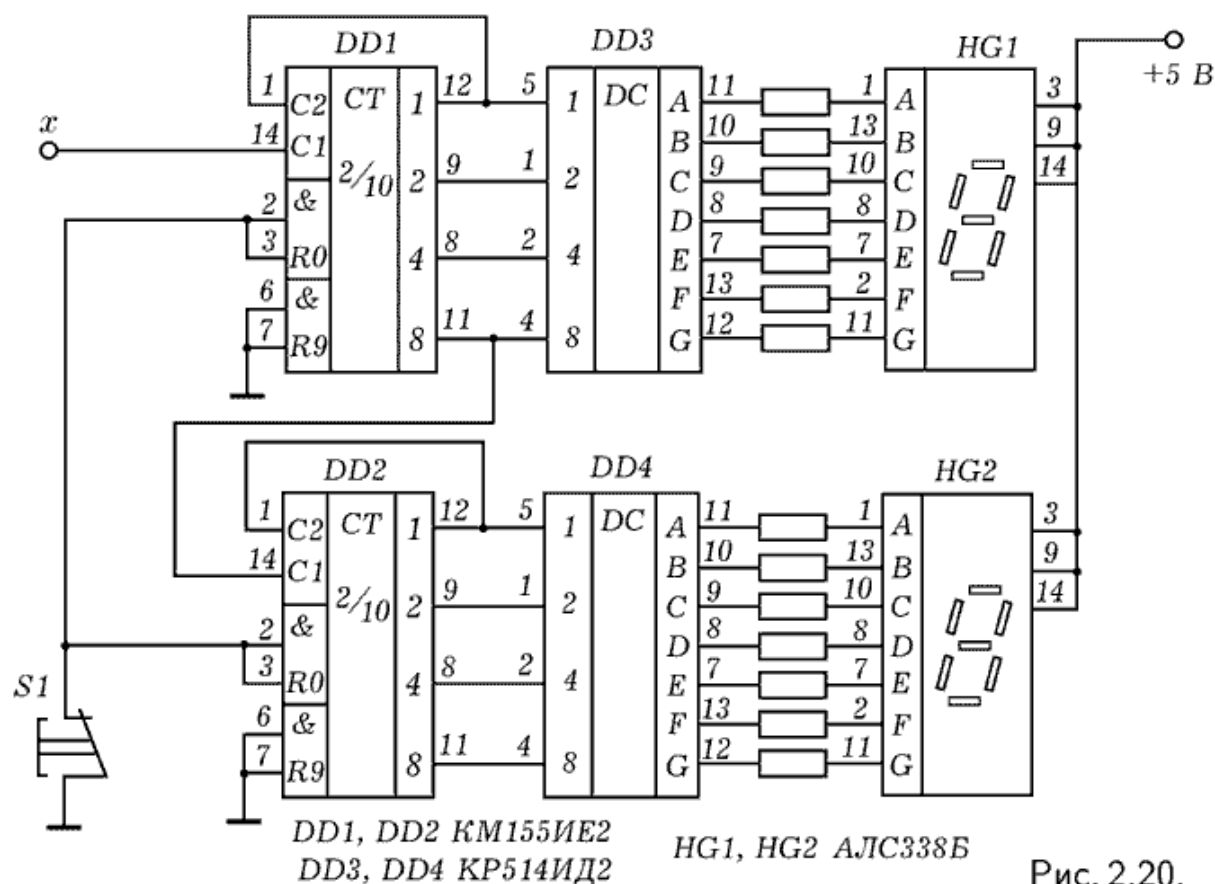


Рис. 2.20.

Электронный измеритель времени состоит из генератора счетных импульсов, ключевого элемента и счетчика импульсов (рис. 2.21.1). В простейшем случае ключевым элементом может являться элемент 2И. Генератор вырабатывает счетные импульсы известной частоты (например, $f=1$ кГц). К управляющим входам ключевого элемента подключены датчики, при срабатывании которых оно открывается на время, равное длительности исследуемого процесса Δt . Пока ключевой элемент открыт, импульсы с генератора проходят через него на вход счетчика, а тот их считает. Время равно $\Delta t = NT = N/f$, где N — число сосчитанных импульсов.

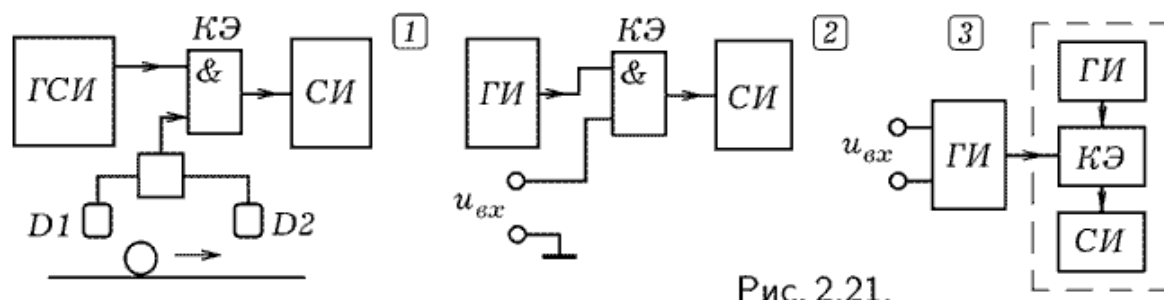


Рис. 2.21.

Электронный частотомер (рис. 2.21.2) используется для измерения частоты импульсов. Генератор импульсов вырабатывает управляющие импульсы длительностью, допустим 1 с, которые поступают на управляющий вход ключевого элемента. Пока он открыт, входные импульсы поступают в счетчик. Через 1 с ключевой элемент закрывается. Перед новым открыванием ключевого элемента происходит обнуление счетчика. На индикаторах периодически высвечивается число импульсов за 1 с, равное их частоте. Если вход частотомера соединить с выходом генератора, частота которого линейно зависит от входного напряжения $u_{вх}$ (рис. 2.21.3), то получится цифровой вольтметр.

2. Цифро-аналоговый преобразователь. Допустим, Вы захотели превратить персональный компьютер в генератор импульсов заданной формы. Для этого необходимо написать программу, которая через равные промежутки времени посылает в параллельный LPT-порт двоичные числа, соответствующие формируемому сигналу. Для автоматического преобразования цифрового кода в эквивалентное ему напряжение используется цифро-аналоговый преобразователь (ЦАП). В простейшем случае ЦАП может содержать конденсатор, который заряжается последовательно эталонных импульсов, соответствующих двоичному числу, подаваемому на входы ЦАП.

На рис. 2.22.1 представлена схема ЦАП на резисторах. Если на входы x_1, x_2, \dots, x_8 подается комбинация логических нулей и единиц, соответствующее некоторому двоичному числу $\overline{a_8 a_7 \dots a_1}$, то на выходе y получается напряжение, пропорциональное этому числу.

Пусть на вход x_3 подается напряжение U_0 , а на все остальные входы — лог. 0 (они заземлены). Сопротивление резисторов выше точки С и ниже точки С равно $2R$ (рис. 2.22.1). Поэтому потенциал точки С составляет $\varphi_C = U_0/3$. Потенциал точки D меньше в 2 раза: $\varphi_D = \varphi_C/2$, потенци-

ал точки Е меньше в 4 раза: $\varphi_E = \varphi_C/4$ и т.д., выходное напряжение меньше φ_C в 32 раза: $U_y = \varphi_H = \varphi_C/32$. Итак, при подаче на i -ый вход напряжения U_0 потенциал выхода увеличивается на $U_0 2^{-(8-i)}/3$. Если на входы x_1, x_2, \dots, x_8 подать комбинацию лог. 0 и 1, соответствующую числу $\overline{a_8 a_7 \dots a_1}$, то $U_i = U_0 a_i$ и на выходе

$$U_{\text{вых}} = \frac{U_0}{3} \sum_{i=1}^8 \frac{a_i}{2^{8-i}} = \frac{U_0}{3} \left(\frac{a_8}{1} + \frac{a_7}{2} + \frac{a_6}{4} + \dots + \frac{a_2}{2^6} + \frac{a_1}{2^7} \right).$$

Условное изображение ЦАП приведено на рис. 2.22.2. ЦАП широко используются в различных цифровых приборах: магнитофонах, МРЗ-плеерах, видеокамерах, звуковых платах компьютеров и т.д.

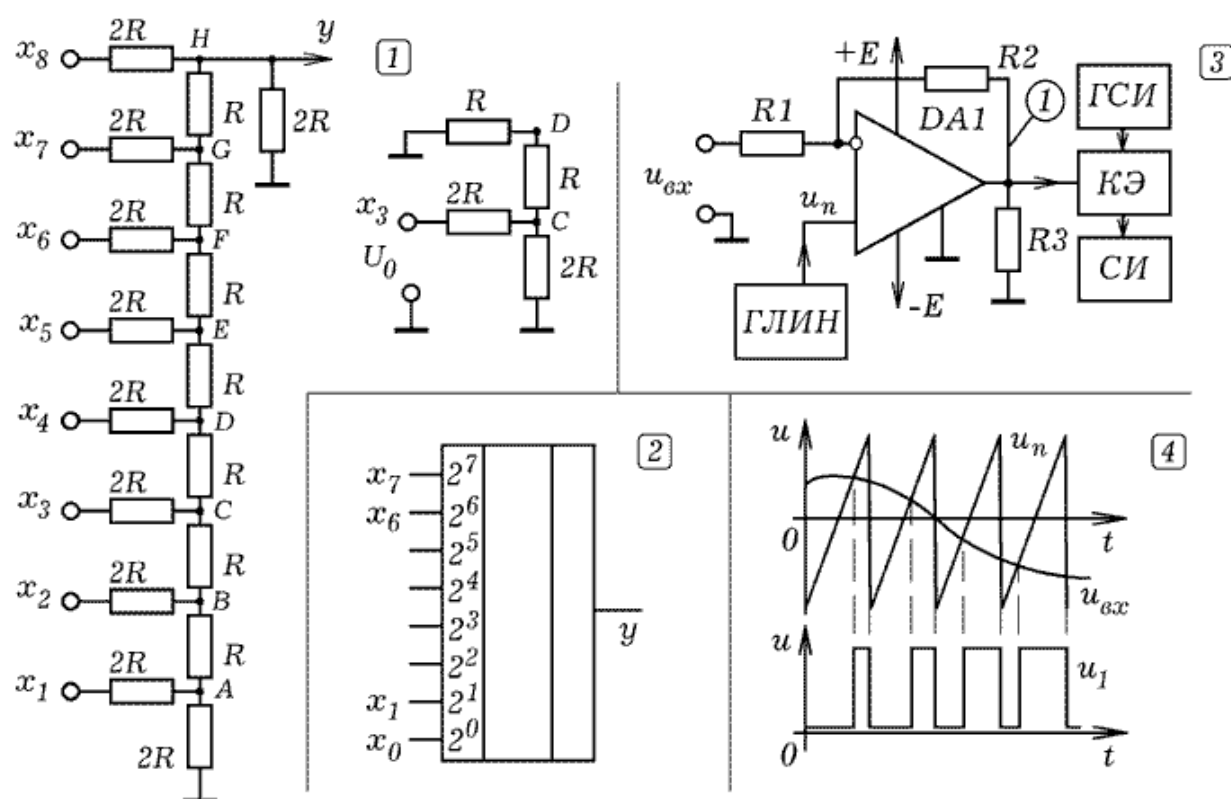


Рис. 2.22.

3. Аналого-цифровой преобразователь. А как решить обратную задачу, как преобразовать плавно изменяющееся напряжение, поступающее с датчика координаты или с микрофона, в цифровой код? Для этого используется *аналого-цифровой преобразователь (АЦП)*.

Простейший АЦП (рис. 2.21.3) состоит из генератора импульсов, частота которых зависит от входного напряжения, и цифрового частотомера, содержащего генератор импульсов, ключевой элемент и счетчик. Применяется *метод частотно-импульсного преобразования*: при изменении входного напряжения меняется частота импульсов, что регистрируется частотомером, отградуированным в вольтах (амперах, омах, градусах Цельсия и т.д.). В качестве частотомера может быть использована ПЭВМ, в этом случае результаты измерений записываются в файл.

Если необходимо оцифровать координату, то используют RC-генератор, частота которого зависит от сопротивления R резистивного датчика координаты, через который периодически разряжается или заряжается конденсатор C (рис. 1.20.3). Чтобы оцифровать плавные изменения температуры или освещенности в качестве датчика используют термо- или фоторезистор.

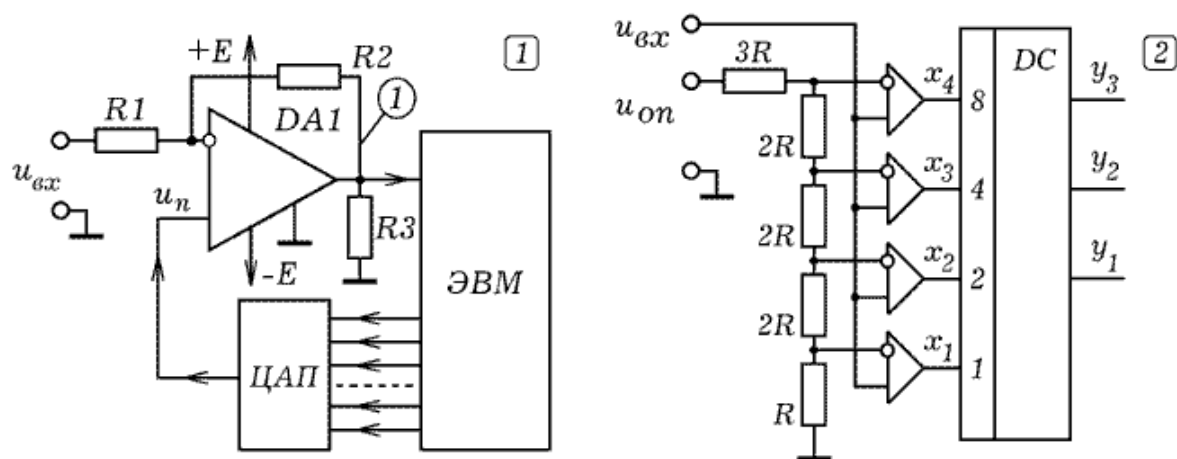


Рис. 2.23.

Рассмотрим последовательный АЦП, в котором осуществляется *время-импульсное преобразование* входного напряжения. Он состоит из генератора линейно-импульсного (пилообразного) напряжения ГЛИН, компаратора на базе ОУ, и электронно-цифрового измерителя времени, который включает в себя генератор счетных импульсов, ключевой элемент и счетчик импульсов (рис. 2.22.3). ГЛИН вырабатывает пилообразные импульсы u_n частотой f_n , которые поступают на один вход компаратора, а на другой вход подается исследуемое напряжение $u_{вх}$, изменяющееся с частотой $f \ll f_n$ (рис. 2.22.4).

ОУ многократно усиливает разность входных сигналов, поэтому пока $u_n < u_{вх}$ на его выходе низкое напряжение (лог. 0), а когда $u_n > u_{вх}$ — на выходе ОУ появляется напряжение высокого уровня (лог. 1). С появлением следующего пилообразного импульса все начинается снова. В результате на выходе ОУ возникает последовательность прямоугольных импульсов, длительность которых Δt прямо пропорциональна напряжению U . Они периодически открывают ключевое устройство электронно-цифрового измерителя времени, через которое счетные импульсы с генератора частотой $f \gg f_n$ поступают на счетчик. Подобные схемы используются в цифровых вольтметрах, амперметрах, мультиметрах. Длительность импульсов также может измеряться с помощью персонального компьютера, который запишет результаты в файл и построит график $u_{вх}(t)$.

Структурная схема АЦП, в котором в качестве блока подбора двоичного числа используется компьютер, изображена на рис. 2.23.1. На восьми выходах параллельного порта ПК появляется двоичное число, которое с каждым тактом увеличивается на 1: 00000000, 00000001, ..., 11111111, а затем все повторяется снова. Цифро-аналоговый преобразователь превращает эту последовательность двоичных сигналов в линейно-импульсное

(пилообразное) напряжение. Оно подается на один вход компаратора, а на другой его вход поступает исследуемое напряжение. В момент их равенства лог.0 сменяется лог.1, что регистрируется компьютером. Результаты измерений записываются в файл, строится график.

Недостатком рассмотренных выше АЦП является сравнительно невысокое быстродействие. Параллельный АЦП (рис. 2.23.2) осуществляет оцифровку аналогового сигнала с большей скоростью. Он включает в себя делитель напряжения, составленный из резисторов $3R - 2R - 2R - 2R - R$, компараторов и шифратора. Оцифровываемый сигнал поступает на неинвертирующие входы компараторов, а на инвертирующие входы подаются опорные напряжения с делителя равные $0,1u_0, 0,3u_0, 0,5u_0, 0,7u_0$. Допустим, напряжение на входе составляет $0,65u_0$, то есть превышает опорные напряжения на первых трех компараторах. В этом случае на их выходах — лог.1 и на входы шифратора x_3, x_2, x_1, x_0 поступает комбинация 0111. На выходах y_1, y_2, y_3 шифратора получается соответствующий двоичный код.

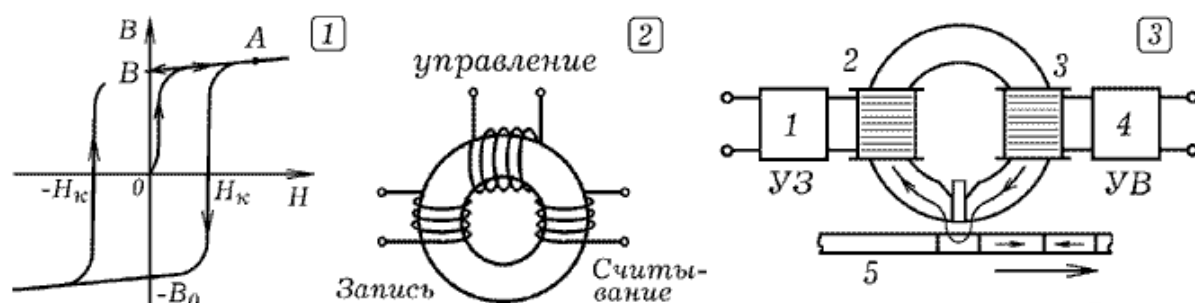


Рис. 2.24.

4. Магнитная запись цифровой информации. Многие современные цифровые приборы имеют память — запоминающие устройства (ЗУ) для хранения информации. Например, стиральная машина выполняет последовательность операций в соответствии с программой, хранящейся в ЗУ. Или телевизор после настройки каналов на определенные частоты вещания, регулировки яркости и громкости запоминает введенную информацию. Калькулятор, сотовый телефон, смартфон, не говоря уже о фото- и видеокамерах и компьютере, — все эти приборы содержат одно или несколько запоминающих устройств для хранения информации.

Магнитная запись информации основана на способности ферромагнетиков намагничиваться, то есть "запоминать" факт нахождения в магнитном поле. Если стальной гвоздь на некоторое время внести в магнитное поле, то он намагнитится, то есть превратится в постоянный магнит. На рис. 2.24.1 показана зависимость индукции магнитного поля B , характеризующей намагниченность ферромагнитного образца, от напряженности H , которая пропорциональна намагничивающей силе, равной произведению числа витков в намагничивающей обмотке и силы тока ($H \sim F = NI$). При создании вблизи ненамагниченого ферромагнетика (точка 0) магнитного поля происходит его намагничивание (точка А). Это обусловлено тем, что существенная часть доменов (областей спонтанной намагниченности) переориентируются по полю. После выключения по-

ля ферромагнетик сохраняет остаточную намагниченность (точка В). В случае, когда намагничивающая сила периодически изменяется, точка на плоскости B и H выписывает *кривую гистерезиса*.

На начальном этапе развития вычислительной техники использовались ЗУ на ферритовых кольцах. Это блок, состоящий из десятка двумерных матриц, содержащих элементарные ячейки памяти. Каждая ячейка представляет собой ферромагнитное кольцо (рис. 2.24.2), которое может находиться в двух намагниченных состояниях, соответствующих лог. 0 и 1. На кольце размещены обмотка управления, входная обмотка записи и выходная обмотка считывания. При пропускании через обмотку записи тока $i = +I$ ферритовое кольцо намагничивается до состояния $+B_r$, что соответствует лог. 1. Если пропускать ток в обратном направлении $i = -I$, то кольцо намагничивается до состояния $-B_r$, что означает лог. 0. Чтобы считать информацию, через обмотку управления пропускают импульс тока $-I$. Если в ячейке памяти записана лог. 1, то происходит перемагничивание кольца из $+B_r$ в $-B_r$, и в выходной обмотке возникает импульс напряжения. Если хранится лог. 0, то перемагничивания не происходит, в выходной обмотке напряжение остается равным 0.

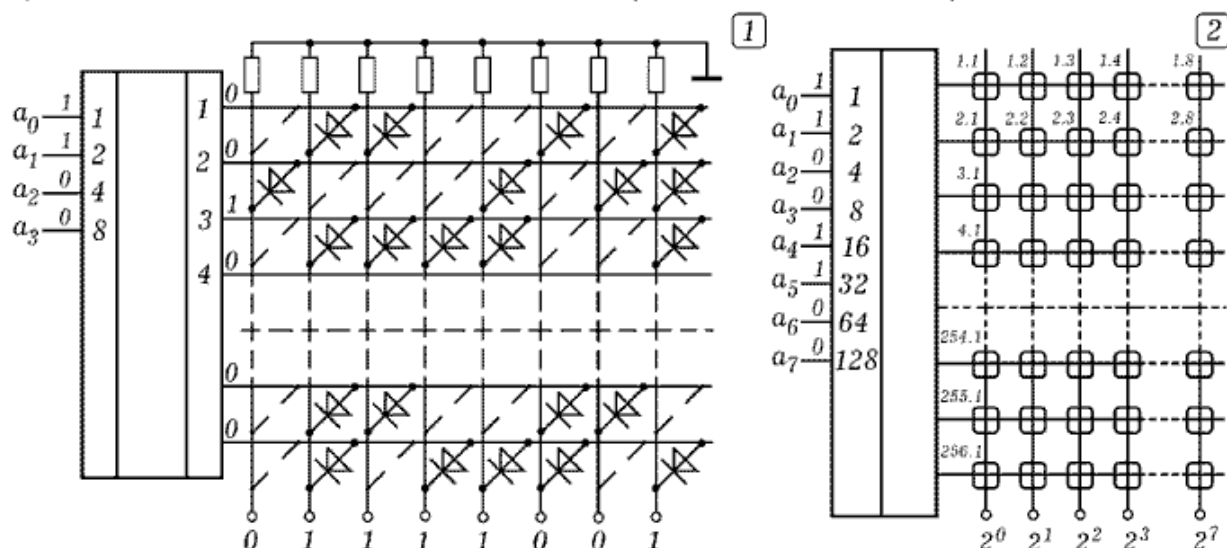


Рис. 2.25.

Рассмотрим принцип записи сигнала на магнитную ленту (диск, барабан). Электрические импульсы (колебания), несущие полезную информацию, после усиления поступают на магнитную головку, мимо которой движется магнитная лента, представляющая собой гибкую ленту с ферромагнитным покрытием (рис. 2.24.3). В результате происходит намагничивание отдельных участков ленты в ту или иную сторону. В режиме воспроизведения магнитная лента протягивается мимо магнитной головки так, что ее намагниченные участки создают в сердечнике переменное магнитное поле. Это приводит к возникновению переменного напряжения в обмотке, которое подается на усилитель воспроизведения и затем на динамик. Так работают накопители информации на гибком и жестком дисках, стримеры, цифровые магнитофоны, запоминающие устройства в некоторых видеокамерах т.д.

5. Постоянное и оперативное запоминающие устройства. При

записи информации на магнитную ленту или диск пользователь не может обратиться к тому или иному байту по отдельности, так как у них нет адреса. Это *неадресуемая память*. Адресуемая память реализуется на микросхемах ОЗУ и ПЗУ.

Микросхема ПЗУ является энергонезависимым запоминающим устройством. В нее вшита программа, которую начинает выполнять компьютер при включении: он осуществляет самотестирование и определение конфигурации входящих в него устройств, а затем запускает программу-загрузчик. Микросхему ПЗУ можно считать преобразователем n -разрядного кода адреса в m -разрядный код прочитанного слова.

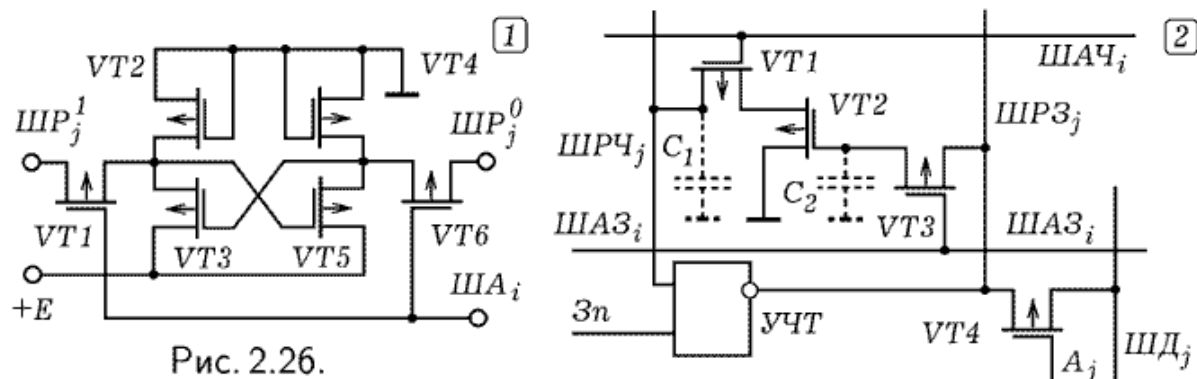


Рис. 2.26.

Однократно программируемая микросхема ПЗУ содержит дешифратор, адресные шины, разрядные шины и матрицу запоминающих элементов (рис. 2.25.1). Этими элементами являются диоды (или транзисторы) с плавкой перемычкой. Чтобы запрограммировать микросхему ПЗУ ее вставляют в программатор, — специальное устройство в котором на соответствующие выводы микросхемы подаются импульсы напряжения. В результате этого происходит сгорание некоторых перемычек, микросхема ПЗУ запоминает информацию. При считывании информации на адресные входы дешифратора подается двоичный код, например, $0011_2 = 3_{10}$. На выводе 3 дешифратора появляется лог. 1, а на разрядных шинах — содержимое ячейки памяти с адресом 0011.

Микросхема ОЗУ включает в себя дешифратор адреса и двумерную матрицу запоминающих элементов, состоящей из системы горизонтальных и вертикальных проводов, в местах пересечения которых расположены ячейки памяти емкостью 1 бит (рис. 2.25.2). Если на адресные входы микросхемы (входы дешифратора) подать двоичный код адреса $\overline{a_7 a_6 \dots a_0}$ считываемого или записываемого байта, то на соответствующем горизонтальном проводе появится лог. 1 и активизируются 8 элементарных ячеек ОЗУ с адресом $\overline{a_7 a_6 \dots a_0}$. При считывании на информационных выходах появляется комбинация 0 и 1, хранящаяся в байте с указанным адресом. При записи на информационные входы подается комбинация лог. 0 и 1, которая сохраняется в соответствующих ячейках памяти. Если адресная шина имеет n проводов, то объем такой памяти равен 2^n байт.

Различают статическую память SRAM и динамическую DRAM. Ячейка статического ОЗУ состоит из 8 триггеров, каждый из которых хранит

1 бит информации. Схема одного триггера представлена на рис. 2.26.1. В состоянии хранения разрядных шин $ШР_j^1$, $ШР_j^0$ лог. 0, а на адресной шине $ША_i$ лог. 1, транзисторы $VT1$, $VT6$ закрыты. Допустим, триггер находится в одном из устойчивых состояний: транзистор $VT3$ закрыт, $VT5$ открыт.

Рассмотрим процесс записи лог. 1. На адресную шину $ША_i$ подается лог. 0, а на разрядную шину $ШР_j^1$ лог. 1. Это вызывает открывание $VT1$ и лог. 1 поступает на сток $VT3$ и затвор $VT5$. Транзистор $VT5$ закрывается, а $VT3$ открывается. Чтобы записать в триггер 0, необходимо подать лог. 0 на $ША_i$, а на разрядную шину $ШР_j^0$ — лог. 1. Чтобы считать записанный бит необходимо подать лог. 0 на адресную шину $ША_i$. При этом транзисторы $VT1$ и $VT6$ открываются и разрядные шины $ШР_j^0$ и $ШР_j^1$ соединяются со стоками транзисторов $VT3$ и $VT5$.

Динамическая память DRAM состоит из ячеек, содержащих емкость, которая заряжается до некоторого напряжения. Вследствие утечки, величина заряда уменьшается, поэтому предусмотрена автоматическая регенерация хранимой информации для восстановления заряда запоминающей емкости (через 2 мс). Схема элемента динамической памяти представлена на рис. 2.26.2. Для хранения бита информации используется емкость $C2$ затвор–подложка. Для записи на адресные шины записи $ШАЗ$ подается лог. 1, транзистор $VT3$ открывается, запоминающая емкость $C2$ оказывается подключенной к разрядной шине записи $ШРЗ_i$. Если на ней лог. 1, то емкость $C2$ заряжается, а если лог. 0 — то она разряжается.

Для того, чтобы считать информацию на разрядную шину чтения $ШРЧ_j$ подается лог. 1, происходит заряд паразитной емкости $C1$ шины $ШРЧ_j$. По адресной шине чтения $ШАЧ_i$ посылается сигнал опроса — лог. 1, что вызывает открывание $VT1$. При заряженной запоминающей емкости $C2$ (лог. 1) транзистор $VT2$ открыт. При этом выходная емкость $C1$ разряжается через $VT1$ и $VT2$ на общий, на $ШРЧ_j$ появляется лог. 0, что означает хранение лог. 1 в ячейке памяти. Если запоминающая емкость $C2$ разряжена (лог. 0), то $VT2$ закрыт, выходная емкость $C1$ не разряжается, лог. 1 на $ШР_j$ показывает, что хранится лог. 0.

Сигнал с $ШРЧ_j$ поступает к инвертирующему усилителю чтения УЧТ. Прочитанный бит инвертируется и подается на разрядную шину записи $ШРЗ_j$ для регенерации. Через транзистор $VT4$ выход УЧ сигналом A_j подключается к шине ввода–вывода данных $ШД_j$. При записи информации УЧ блокируется.

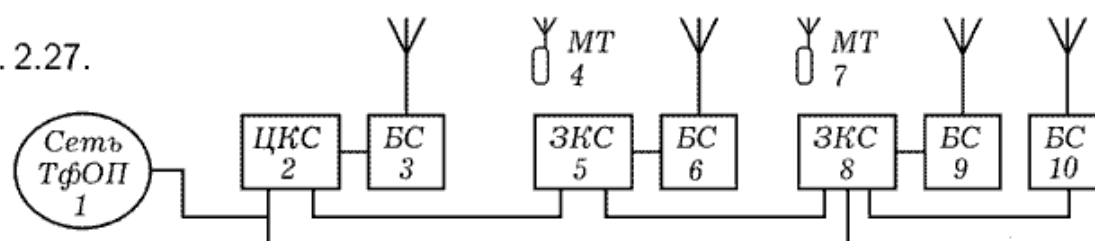
2.3. ЧТО ТАКОЕ ЦИФРОВАЯ СВЯЗЬ

Представьте себе, что Вы — супершпион, заброшенный на территорию противника. У Вас имеются архиважные сведения об оперативной обстановке, которые необходимо передать в Центр, пока они не потеряли своей актуальности. В вашем распоряжении — обычный ноутбук с подключенным к нему радиопередатчиком, способным связаться со спутником. Вы набираете текст сообщения и запускаете специальную программу, которая осуществляет кодирование и управляет передатчиком. Компьютер переводит сообщение на язык нулей и единиц, перемешивает биты в соответствии с заложенным алгоритмом, и радиопередатчик буквально выплевывает закодированное сообщение в эфир в течение секунды. Даже если его перехватят, на расшифровку потребуются годы...

1. Радиотелефонная цифровая связь. Различают следующие виды сетей связи: телефонные, видеотелефонные, сети звукового и телевизионного вещания, телеграфные, факсимильные. Имеется телефонная сеть общего пользования, представляющая собой *многоканальную систему передачи*. Так называется совокупность технических средств (устройств, линий связи), обеспечивающих одновременную и независимую передачу сообщений от нескольких источников по одной линии.

Впервые система радиотелефонной связи для всех желающих была создана в 1946 г. в США. На первых этапах использовались аналоговые системы с частотной или фазовой модуляцией (1 поколение). Основным недостатком состояли в том, что число фиксированных частот, на которых можно было осуществлять радиопереговоры, было не очень велико. Это ограничивало количество каналов связи, так как радиотелефоны, работающие на близких частотах, создавали взаимные помехи. Данная проблема была решена так: вся территория разбивалась на соты (ячейки); каждая сота обслуживалась передатчиком с ограниченным радиусом действия и фиксированной частотой; в соседних сотах используются различные частоты, а в удаленных сотах — одинаковые частоты (*сотовый принцип распределения частот*). В настоящее время используется цифровая связь (2 поколение), что позволяет применять временное разделение каналов, шифрование сообщений, использование блочного кодирования и т.д. Для передачи цифровой информации используется амплитудная, частотная или фазовая модуляции несущих колебаний.

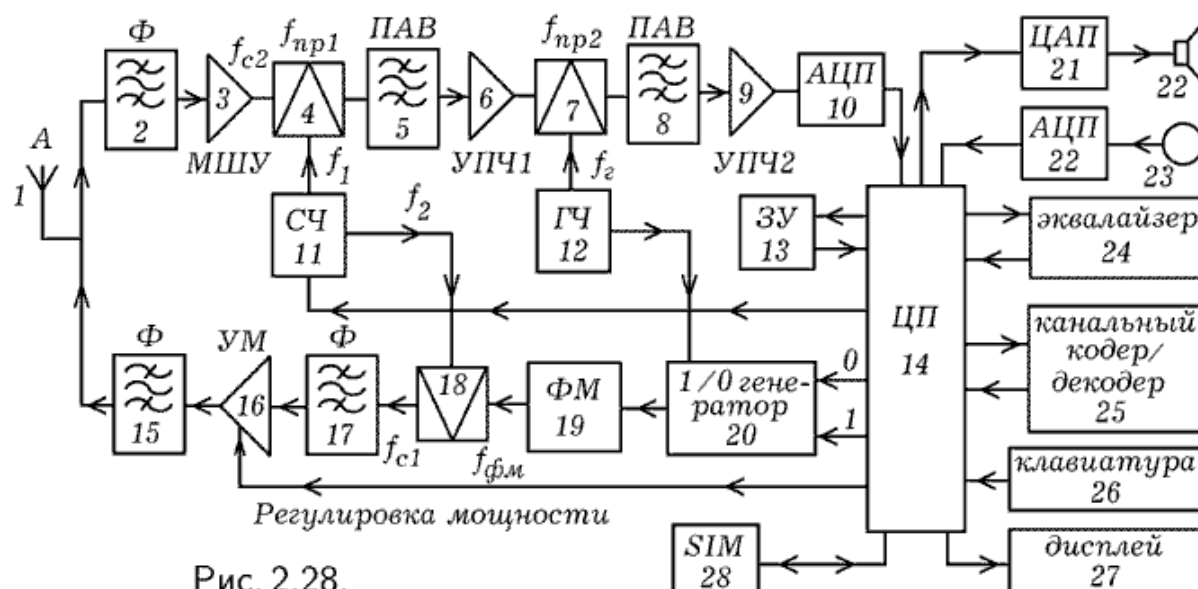
Рис. 2.27.



Сеть сотовой связи (рис. 2.27) состоит из центральной коммуникационной станции ЦКС 2, зональных коммуникационных станций ЗКС 5 и 8, базовых станций БС 3, 6, 9, 10 и мобильных телефонов МТ 4 и 7 (подвиж-

ных абонентских радиотелефонных аппаратов). Центральная коммуникационная станция соединена с телефонной сетью общего пользования 1. Базовые станции установлены на высоких зданиях в углах шестиугольников так, что каждый из них обслуживает все мобильные телефоны внутри своей шестиугольной области (ячейки или соты). При этом все эти соты покрывают территорию города. При перемещении абонента из одной соты в другую со включенным телефоном, он автоматически передается от одной базовой станции к другой.

Центр коммутации занимается маршрутизацией вызовов и управлением ими, постоянным слежением за работающими мобильными телефонами, "эстафетной передачей" абонента при его перемещении из одной соты в другую, соединяет подвижного абонента с обычной телефонной сетью, автоматически переключает каналы при наличии неисправности. При этом реализуется *транкинг* — метод равноправного доступа абонентов к данной совокупности каналов, при котором за каждым абонентом для текущего сеанса связи временно закрепляется конкретный канал связи. Процесс обслуживания "блуждающего" абонента, пересекающего границы сот и различных географических зон, называется *роумингом*.



В России используется цифровой общеевропейский стандарт GSM. Передатчики мобильных станций (телефонов) работают в диапазоне 890–915 МГц, а принимают сигнал с базовых станций в диапазоне 935–960 МГц. Таким образом между частотами приема и передачи имеется разнос 45 МГц, за счет чего может быть реализована двухсторонняя (дуплексная) связь. Каждый из диапазонов разбит на 124 частотных канала с шагом 200 кГц. Кроме того осуществляется временное разделение каналов так, что на одной частоте с некоторой базовой станцией могут связаться 8 мобильных телефонов. Один канал — управляющий, используется для вызова абонента. Общее число каналов связи 992, скорость передачи информации по одному каналу — 270 Кбит/с, радиус соты 0,5–35 км.

В состав сотового телефона входят передающее и приемное устрой-

ства, устройство преобразования и воспроизведения речи, устройство контроля и управления, клавиатура, дисплей, звонок, антенна. Рассмотрим принцип действия цифрового телефона, поддерживающего стандарт GSM (рис. 2.28). Передатчик и приемник работают попеременно, включаясь и выключаясь с достаточно большой частотой. В режиме приема сигнал с антенны 1 проходит через входной полосовой фильтр 2, усиливается маломощным усилителем 3, поступает в смеситель 4. На второй вход смесителя поступает сигнал с синтезатора частот 11 (первого гетеродина). Сигнал после смесителя 4 проходит через фильтр 5 на поверхностных акустических волнах ПАВ и усилитель промежуточной частоты 6 (УПЧ1). После этого он поступает во второй смеситель 7, где смешивается с сигналом от генератора частоты 12 (второго гетеродина). После прохождения через фильтр 8 и второй усилитель промежуточной частоты 9 (УПЧ2) сигнал подается на АЦП 10. АЦП осуществляет оцифровку сигнала и посылает его в сигнальный процессор 14, в котором сигнал декодируется и подается на ЦАП 21 с динамиком 22 на выходе.

В режиме передачи сигнал с микрофона 23 оцифровывается АЦП 22 и поступает в сигнальный процессор 14, который формирует информационный цифровой сигнал. Он управляет работой генератора 20, который посылает модулирующий сигнал в фазовый модулятор 19. В смесителе 18 сигнал с фазового модулятора смешивается с сигналом, вырабатываемым синтезатором частот 11. Получающийся высокочастотный сигнал проходит через фильтры 17 и 15, усилитель мощности 16 и подается на антенну 1. Цифровая часть телефона осуществляет демодуляцию, кодирование речевого сигнала, шифровку данных, обработку информации, вводимой с клавиатуры 26, вывод информации на дисплей 27, работает с SIM-картой 28, осуществляет аутентификацию абонента. Она также включает в себя эквалайзер 24 и каналный кодер/декодер 25.

2. Оцифровка изображений с помощью ФПЗС. В настоящее время для ввода и оцифровки графической и видеоинформации используются фоточувствительные приборы с зарядовой связью (ФПЗС). ФПЗС — интегральная микросхема, в корпусе которой имеется окошко, расположенное напротив матрицы светочувствительных элементов. Для создания этой матрицы на полупроводниковой подложке (кремниевой пластине) создается тонкий слой оксида (SiO_2) формируют электроды в виде узких полосок, образующих регулярную структуру. Эти электроды выполнены из алюминия или сильнолегированного кремния, имеют ширину 3–7 мкм, между ними зазор 0,2–3 мкм. На пластине площадью 1 см² располагается 500–2000 элементов в линейном и $10^4 - 10^6$ в мозаичном ПЗС. Вследствие близости электроды оказывают взаимное влияние друг на друга через электрическое поле.

Фоточувствительный ПЗС (рис. 2.29.1) используется для преобразования оптического изображения в последовательность электрических импульсов (видеосигнал), а также для хранения и обработки информации. Фотографируемый объект 1 проецируется объективом 2 на ФПЗС-матрицу. При освещении ФПЗС за счет внутреннего фотоэффекта в объ-

еме кристалла кремния 5 образуются избыточные электронно-дырочные пары, электроны собираются в потенциальные ямы, расположенные под электродами 4. Так образуются сигнальные зарядовые пакеты, причем величина заряда в каждом из них пропорциональна экспозиции (произведению освещенности на время) поверхности вблизи электрода.

Электроды 4 соединены с помощью системы проводников 3 с электронным устройством 6, которое последовательно опрашивает все элементы данной строки ПЗС-матрицы. В результате получаются импульсы напряжения, пропорциональные величине заряда. После этого переходят к следующей строке и т.д., пробегаая всю ПЗС-матрицу. В результате формируется видеосигнал, несущий информацию о распределении освещенности в различных точках кадра. Он создает изображение на мониторе 7. ФПЗС имеет преимущества над видеоконвертером — небольшие размеры, жесткий геометрический растр, долговечность, меньшая потребляемая мощность, нечувствительность к электрическим и магнитным полям.

В цифровом фотоаппарате этот сигнал оцифровывается и записывается на флеш-память (емкостная ЗУ). В цифровой видеокамере имеется генератор кадровой развертки, работающий с частотой 25–30 Гц, который обеспечивает периодический опрос всех элементов ФПЗС с последующей записью информации на магнитный или лазерный диск. Для получения цветных изображений используются следующие два метода: 1) с помощью призмы световой поток разделяется на красную, зеленую и синюю составляющие (RGB), каждая из которых проецируется на свой ФПЗС; 2) на поверхности ФПЗС создают специальный пленочный светофильтр, который образует растр из триад RGB. Светочувствительные элементы, находящиеся под красными светофильтрами, формируют видеосигнал, несущий информацию о красной составляющей изображения и т.д.

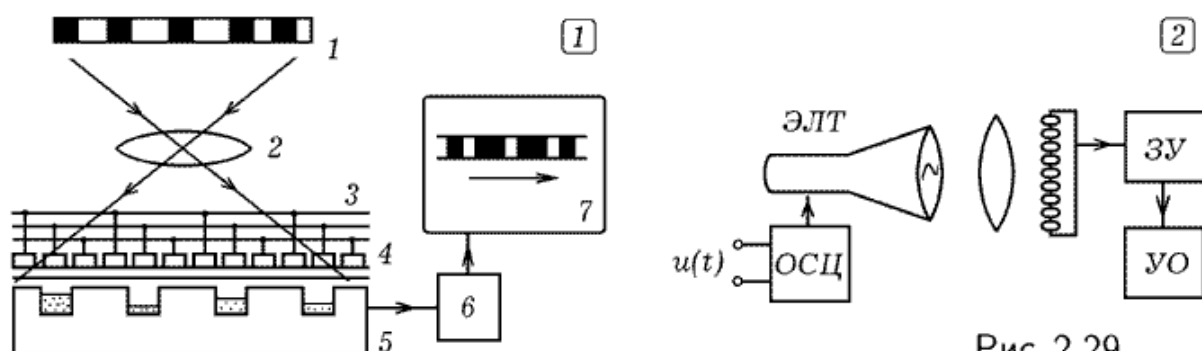


Рис. 2.29.

На основе ПЗС может быть создан кодировщик осциллограмм — это аналого-цифровой преобразователь, который превращает осциллограмму на экране ЭЛТ в последовательность цифровых сигналов (рис. 2.29.2). С помощью объектива осциллограмма с экрана ЭЛТ проецируется на матрицу ФПЗС, которая дискретизирует и квантует ее изображение. Получающийся поток двоичных сигналов сохраняется в ЗУ, откуда потом считывается и выводится на устройство отображения УО. Эти процессы синхронизированы сигналами, поступающими из блока управления. Кодировщик осциллограмм позволяет оцифровывать с экрана ЭЛТ однократные сигналы длительностью около 10^{-9} нс.

3. Принципы цифрового телевидения. Важнейшим технологическим прорывом является перевод российского телевидения из аналогового в цифровое. Это стало возможным в результате решения двух важных проблем: 1) создание алгоритма сжатия звука и изображения в стандарт MPEG2, что позволяет уменьшить скорость передачи (битрейт) видеоданных с 250 Мбит/с до приемлемых 3–6 Мбит/с; 2) использование амплитудно-фазовой модуляции несущей с эффективностью до 8 бит на 1 Гц полосы радиосигнала. Все это позволило передавать в стандартном ТВ-канале одновременно 8 телепрограмм. Было решено, что цифровое телевидение будет подчиняться правилу "6+7+8": телевизионные сигналы должны передаваться в стандартных полосах 6 МГц (Америка и Япония), 7 МГц (Западная Европа) и 8 МГц (Россия, Франция). Это позволит плавно перейти аналогового телевидения к цифровому.

Основные принципы спутникового телевидения (DVB-S) состоят в следующем. На телевизионном центре несколько цифровых программ, сжатых по алгоритму MPEG2, используя мультиплексор, объединяют в единый транспортный поток. Его разбивают на блоки или пакеты фиксированной длины (188 байт), в них добавляют проверочные биты (16 байт). Может использоваться сверточное кодирование, требующее введения дополнительных проверочных битов.

Возникла проблема, как получающийся широкополосный цифровой сигнал передавать внутри стандартного радиоканала. Амплитудная или частотная модуляции не позволяют передать этот сигнал даже через спутниковый ТВ-канал, полоса пропускания которого 27 МГц. Несколько телепрограмм по одному каналу передавать бессмысленно. Поэтому применяется квадратурно-фазовая модуляция QPSK (Quadrature Phase-Shift Keying), при использовании которой изменяется фаза несущих колебаний, принимая дискретные значения 45^0 , 135^0 , 225^0 и 315^0 .

В кабельном телевидении (DVB-C) используется полоса частот шириной в 8 МГц на канал. В кабельных сетях низкий уровень помех. Поэтому применяется модуляция QAM (Quadrature Amplitude Modulation) являющейся гибридом амплитудной и фазовой модуляции. У несущих высокочастотных колебаний одновременно квантуются и фаза, и амплитуда. Используется 64 дискретных значений. При этом в стандартной полосе шириной 8 МГц удастся передать до 8 телевизионных программ.

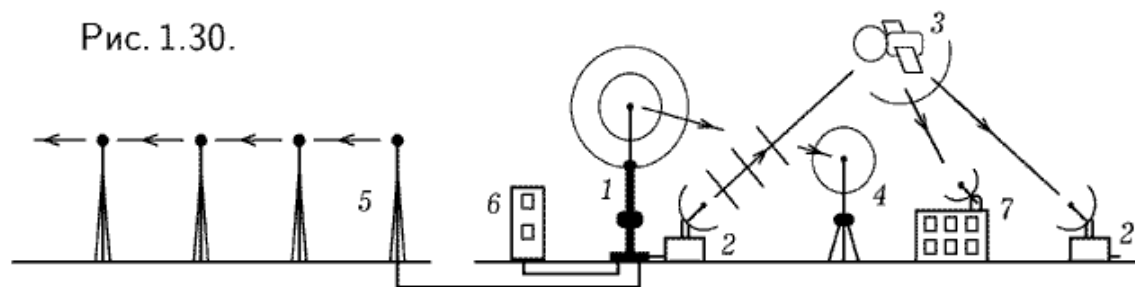
Самое сложное — разработка эфирного цифрового ТВ (DVB-T). Одна из трудностей в состоит в том, что при распространении радиоволны происходят многократные отражения от различных объектов (зданий), которые в аналоговом телевизионном приемнике приводят к раздвоению контуров изображения, а в цифровом могут вызвать полный срыв синхронизации. Другая проблема — в том, что широкополосный цифровой сигнал необходимо передать по стандартному каналу шириной 6–8 МГц. Поэтому используется самый сложный способ модуляции — COFDM (Coded Orthogonal Frequency Division Multiplex). Он состоит в том, что для передачи цифрового сигнала одновременно используются две тысячи (1705)

или 8 тысяч (6817) несущих (режимы "2К" и "8К"), которые расположены с интервалом в несколько килогерц и модулируются по частоте и фазе. Для осуществления COFDM используется цифровой процессор (СБИС), создающий сигнал, эквивалентный сумме нескольких тысяч несущих.

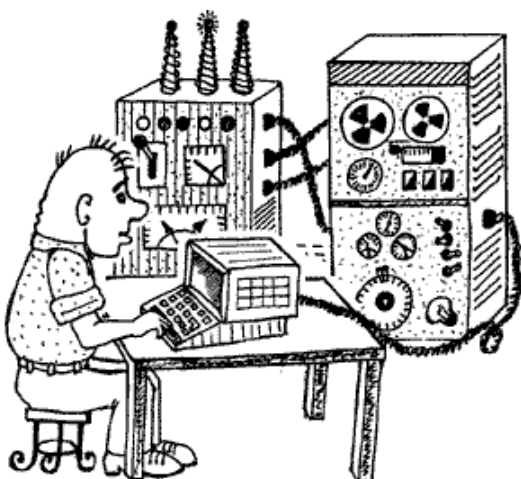
Чтобы исключить влияние отражений на качество приема, сигнал излучается не непрерывно, а небольшими порциями, причем в промежутках между порциями прием сигнала блокируется. Радиосигналы, отраженные от зданий, приходят в приемную антенну во время пауз и в приемный тракт телевизора не поступают. Этот способ модуляции также применяется в беспроводных интерфейсах Bluetooth и Wi-Fi.

Для приема сигнала используются цифровые приставки к аналоговым телевизорам, которые принимают радиосигнал в заданной полосе частот, демодулируют его, выделяя транспортный поток, корректируют ошибки, фильтруют сигнал, выделяя нужную программу, отделяют сигналы, несущие информацию о звуке и изображении, декодируют MPEG2 и преобразовывают их в аналоговый сигнал. При этом используются СБИС; так MPEG2-декодер содержит 2 млн. транзисторов, а COFDM-демодулятор — 10 млн. Аналого-цифровые iDTV (integrated Digital TV) телевизоры работают таким же образом. Наличие независимых аналогового и цифрового трактов усложняет устройство телевизора. iDTV телевизоры 2-го поколения созданы на основе процессора, который имеет два MPEG2-декодера и цифровой тракт аналогового телевизора. Если в схему добавить селектор каналов, то получится универсальный однокристалльный телевизор, который способен принимать эфирные и кабельного программы, как аналоговые, так и цифровые.

Рис. 1.30.



Система телевидения состоит из следующих компонентов (рис. 1.30): программный телецентр 1, наземные пункты связи 2, спутник связи 3 на геостационарной орбите, телепередатчики-ретрансляторы 4, радиорелейные линии прямой видимости 5, аппаратура кабельного телевидения 6, индивидуальные спутниковые приемники 7. Программный телецентр излучает мощный телевизионный радиосигнал в эфир. По кабельной магистрали сигнал также подается на наземные пункты спутниковой связи и на специальную аппаратуры для приема кабельного телевидения. На границах зоны уверенного приема установлены телепередатчики-ретрансляторы, которые принимают, усиливают и передают сигнал дальше. Также используют радиорелейные линии, состоящие из ретрансляторов, установленных на башнях в пределах прямой видимости.



ГЛАВА 3 КАК УСТРОЕНА ЭЛЕКТРОННО- ВЫЧИСЛИТЕЛЬНАЯ МАШИНА

Теперь, после изучения основ электроники мы уже готовы к тому, чтобы понять, как работает вычислительная машина. Следует помнить, что все вычислительные устройства делятся на механические (счеты, арифмометры, логарифмические линейки) и электрические (калькуляторы, чипы, управляющие АСУ, сотовые телефоны, ЦЭВМ). Все электрические вычислительные машины подразделяются на аналоговые, цифровые и гибридные, то есть работающие с информацией, которая представлена как в цифровой, так и в аналоговой форме. Современные вычислительные машины — это многопроцессорные комплексы с большим объемом памяти, высоким быстродействием, элементами искусственного интеллекта, возможностью передавать информацию по сети.

3.1. УСТРОЙСТВО И ПРИНЦИП ДЕЙСТВИЯ ЭВМ

1. История развития вычислительной техники. Более 1500 лет назад были изобретены счеты, которые в некоторых странах применяются и по сей день. В 1614 г. Дж. Непер предложил таблицы логарифмов, а чуть позже была изобретена логарифмическая линейка. Первая механическая суммирующая машина была предложена Б. Паскалем в 1642 г. Чуть позже появилась машина Лейбница (1673 г.), выполнявшая все четыре арифметических действия. Ч. Бэббидж в 1822–1850 гг. сконструировал первую универсальную программно-управляемую счетную машину механического типа, которая содержала устройство управления, устройство для выполнения арифметических операций, запоминающее устройство и устройство ввода информации и печати. Программы вводились с помощью перфокарт.

Паскалина — так называлась суммирующая машина Паскаля, — явилась прототипом арифмометра, созданного П. Чебышевым в 1876 г. В 1890 г. В. Однер получил патент на производство арифмометров и наладил их производство. Поздние модели арифмометров Однера имели электропривод и клавишный ввод информации.

Цифровая вычислительная машина, работающая в двоичной системе счисления, была предложена Дж. Атанасовым. Г. Айкен в 1937 г. сконструировал счетную машину на электромеханических реле, которая была создана в 1944 г. В 1938 г. К. Цузе завершил постройку электромеханической программируемой цифровой машины релейного типа, названной Z1. Позже были созданы Z2 и Z3, превосходившие известные в то время аналоги по ряду параметров.

Первая электронно-вычислительная машина ENIAC содержала 18 тысяч электронных ламп и была собрана Дж. Мочли и Пр. Экертом в 1945 г. ЭВМ ENIAC предназначалась для расчета дальности полета снаряда методами имитационного моделирования. Принципы работы цифровой ЭВМ были сформулированы Дж. Нейманом и независимо от него С.А. Лебедевым в конце 50-х годов, работавшим над созданием Малой электронно-счетной машины (МЭСМ) и Большой электронно-счетной машины (БЭСМ) на транзисторах.

В 1964 г. фирма IBM (Information Bussines Mashines) наладила выпуск ЭВМ, содержащих 32-разрядный процессор, оперативную память, магнитные запоминающие устройства, устройство печати и видеотерминал (монитор и клавиатура). В 1974 г. фирмой Intel был создан первый 8-разрядный процессор 8080, представляющий собой большую интегральную схему с 4500 транзисторами. На его базе Э. Робертс собрал компьютер Альтаир. Для него Пол Аллен и Билл Гейтс адаптировали язык Бейсик. Позже ими была организована фирма Microsoft, занимающаяся созданием программного обеспечения персональных ЭВМ. В 1976 г. Стив Возняк и Стив Джобс создали компьютер Apple-1, а позже организовали корпорацию Apple.

Изобретение электронных ламп, транзисторов, интегральных схем (ИС) предопределили появление и развитие ЭВМ. Принципиальные отличия ЦЭВМ от арифмометров состоит в возможности проверки истинности условий, организации циклов и проведения рекурсивных вычислений. Поколения ЭВМ:

1 поколение (до 1955 г.): элементная база — электронные лампы, ОЗУ на ЭЛТ емкостью ≈ 100 байт, быстродействие $10^3 - 10^4$ оп/с, применяются пульт управления и перфокарты.

2 поколение (1955–1965 г.): использовались транзисторы, ОЗУ на ферритовых сердечниках емкостью 1000 байт, быстродействие 10^6 оп/с, для ввода информации применяются перфокарты и перфоленты.

3 поколение (1965–1975): программно-совместимые ЭВМ на интегральных схемах (ИС), ОЗУ на ферритовых сердечниках, ее емкость $\approx 10^4$ байт, быстродействие 10^7 оп/с, для ввода и вывода информации используется алфавитно-цифровой терминал.

4 поколение (1975–1985): большие и сверхбольшие ИС, ОЗУ на БИС и СБИС, емкость ОЗУ $\approx 10^5 - 10^7$ байт, быстродействие — $10^8 - 10^9$ оп/с, монохроматический и цветной графический дисплей, клавиатура, мышь.

5 поколение (1985– ...): опто- и криоэлектроника, ОЗУ на СБИС,

емкостью $\approx 10^8$ байт, быстродействие 10^{12} оп/с, цветной дисплей, клавиатура, мышь, мультимедиа, устройство голосовой связи с ЭВМ и др.

В настоящее время выделяют следующие группы цифровых ЭВМ: 1. СуперЭВМ: производительность 1 трлн. оп/с (в 10^5 раз выше чем у ПЭВМ), используются для решения сверхсложных задач. 2. Большие вычислительные комплексы БВК (мейнфреймы): быстродействие до $5 \cdot 10^6$ оп/с, занимаемая площадь 10 – 100 м². К одному БВК можно подключить до 1000 терминалов. 3. Мини-ЭВМ: быстродействие $1,5 \cdot 10^6$ оп/с, используются в качестве серверов в ЛВС с числом рабочих станций до 300. 4. Рабочие станции — однопользовательские миниЭВМ, производительность выше, чем у ПК. 5. Микро ЭВМ, персональные компьютеры: производительность более 10^6 оп/с, ОЗУ ≈ 1 Гбайт, НЖМД до 100 – 1000 Гбайт. 6. Ноутбуки, карманные компьютеры, электронные записные книжки.

2. Аналоговая вычислительная машина. Допустим, необходимо разработать метод быстрого решения уравнения $a = e^{-bx}$ без использования таблиц и вычислительной техники. Как быть? Возьмем конденсатор емкостью C , зарядим его до напряжения U и будем разряжать через резистор R , причем подберем R и C так, чтобы $b = 1/RC$. Тогда время t , в течение которого конденсатор разрядится до напряжения aU и будет равно искомому x .

Или другая задача. Необходимо решить систему из трех линейных алгебраических уравнений $a_i x_1 + b_i x_2 + c_i x_3 = d_i$, $i = 1, 2, 3$. Любой системе уравнений соответствует электрическая цепь. В данном случае эта цепь будет содержать три ветви, текущие по ним токи — и есть неизвестные x_1, x_2, x_3 . Подберем ЭДС источников и сопротивления ветвей требуемым образом. Достаточно измерить токи в ветвях — и ответ найден!

Рассмотренные примеры поясняют общие принципы функционирования *аналоговой вычислительной машины*, которая используется для обработки информации, представленной в аналоговой (непрерывной) форме. При этом каждой математической переменной величине, входящей в решаемую систему уравнений, ставится в соответствие машинная переменная (линейная или угловая координата, давление газа или жидкости, сила тока, напряжение). Над машинными переменными осуществляются математические операции с помощью решающих элементов, работающих в соответствии с теми или иными физическими законами.

АВМ часто представляет собой устройство, состоящее из сумматоров, дифференциаторов, интеграторов и т.д., моделирующее исследуемый объект. При этом различают механические, гидравлические, пневматические, электромеханические, электронные АВМ. К первым аналоговым вычислительным машинам относят логарифмическую линейку (1620 г.), механический интегратор Дж.-Дж. Томсона (1876 г.), позже появились АВМ на операционных усилителях (1949 г.).

Сейчас применяются структурные АВМ, состоящие из набора решающих элементов на ОУ, соединенных друг с другом в соответствии с решаемой задачей. Рассмотрим порядок решения задач на подобных АВМ: 1)

исходя из рассматриваемой системы уравнений составляется блок схема-модели из решающих устройств АВМ; 2) рассчитывают масштабные коэффициенты, позволяющие перейти от математических переменных, входящих в систему уравнений, к соответствующим машинным переменным; 3) вычисляют параметры схемы: сопротивления, емкости, параметры нелинейных элементов; 4) соединяют решающие элементы так, как требует структурная схема, и устанавливают параметры элементов; 5) настраивают схему и решают задачу: подают на ее входы требуемые напряжения и регистрируют решение с помощью осциллографа или самописца в виде функции от времени.

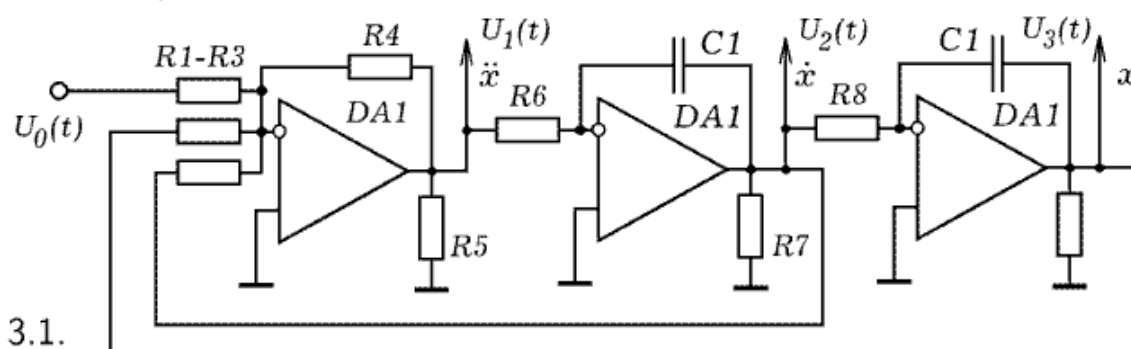


Рис. 3.1.

Рассмотрим структурную схему АВМ (рис. 3.1) для решения линейного дифференциального уравнения $a_0\ddot{x}(t) + a_1\dot{x}(t) + a_2x(t) = y(t)$ методом понижения порядка производной. АВМ состоит из трех ОУ, выполняющих роль сумматора, и двух интеграторов. При этом выполняются соотношения: $U_1(t) = M_1\ddot{x}(t)$, $U_2(t) = M_2\dot{x}(t)$, $U_3(t) = M_3x(t)$, $U_0(t) = M_y y(t)$. Сумматор складывает сигналы U_0, U_3, U_2 , поступающих на вход, что соответствует сумме $y(t) - a_2x(t) - a_1\dot{x}(t)$. На его выходе — напряжение $U_1(t)$, пропорциональное $\ddot{x}(t)$. Чтобы получить решение $x = x(t)$, необходимо подать напряжения U_3 на вход осциллографа или самописца.

К АВМ также относят устройства, в которых один объект моделируется другим, например, электростатическое или постоянное магнитное поле моделируется стационарным электрическим полем в жидкой среде, или на электропроводной бумаге. Для решения уравнения в частных производных используются сеточные АВМ. При этом область, внутри которой ищут решение, разбивается на элементарные объемы, строится сетка. Каждый элементарный объем заменяется электрической схемой замещения из R, L, C -элементов. На отдельные точки получившейся цепи подаются напряжения, соответствующие краевым условиям задачи. С помощью измерительных приборов определяются значения напряжения в тех или иных узлах сетки, после чего осуществляют переход от машинных переменных к математическим переменным решаемой задачи.

К достоинствам АВМ относят высокое быстродействие, низкая стоимость, возможность решения задачи в режиме реального времени. Недостатком является довольно высокая погрешность, которая в то же время сопоставима с погрешностью измеряемых величин.



фон НЕЙМАН Джон (Янош) (1903–1957 гг.)
 Окончив Будапештский университет, занимался математикой, экономикой, физикой, кибернетикой, лингвистикой. Входил в Манхэттенский проект по созданию атомной бомбы, принимал участие в работе над водородной бомбой. Разрабатывал логическую структуру ЭВМ, сформулировал основные принципы. Создал теорию самовоспроизводящихся автоматов.

3. Принципы работы ЭВМ. Структура ЭВМ. Представьте себе, что Вы решили создать компьютер — программируемое устройство для осуществления вычислений. Из каких блоков оно будет состоять? Во-первых, компьютер должен включать в себя центральный процессор — устройство, способное выполнять вычисления в соответствии с заданной программой. Во-вторых, в него должна входить память, в которой будут храниться исходные данные, исполняемая программа и результаты вычислений. И в-третьих, — устройства ввода-вывода, с помощью которого ЭВМ будет обмениваться информацией с программистом.

Рассуждая подобным образом, Джон фон Нейман разработал структуру ЭВМ (1945 г.), которая должна включать в себя арифметико-логическое устройство (АЛУ), оперативную память (ОЗУ), устройства ввода-вывода (УВВ), устройство управления (УУ). В ОЗУ размещается программа. УУ считывает команду из первой ячейки ОЗУ и обеспечивает ее выполнение, осуществляя арифметические и логические действия, ввод и вывод информации. После этого выполняется следующая команда. Сформулируем принципы фон Неймана:

1. Принцип программного управления: ЭВМ работает по программе, которая находится в оперативной памяти и выполняется автоматически; программы дискретны и представляют собой последовательность команд, каждая из которых осуществляет отдельный акт преобразования информации; все разновидности команд образуют систему команд машины.

2. Принцип условного перехода: При выполнении программы возможен переход к той или иной команде в зависимости от промежуточных результатов вычислений; это допускает создание циклов.

3. Принцип хранимой информации: Команды как и операнды представляются в машинном коде и хранятся в оперативной памяти. При работе команды обрабатываются устройством управления процессора, а операнды — арифметико-логическим устройством.

4. Принцип использования двоичной системы счисления: Информация кодируется в двоичной форме и разделяется на элементы, называемые словами. В двоичной системе используются две цифры 0 и 1, что соответствует двум состояниям двустабильной системы (кнопка нажата-отпущена, транзистор открыт-закрыт, ...)

5. Принцип иерархичности ЗУ: Компромиссом между необходимыми

большой емкостью памяти, быстрым доступом к данным, дешевизной и надежностью является иерархия запоминающих устройств: 1) быстродействующее ОЗУ, имеющее небольшую емкость для операндов и команд, участвующих в вычислениях; 2) инерционное ВЗУ, имеющее большую емкость для информации, не участвующей в данный момент в работе ЭВМ.

В современных ЭВМ реализован *магистрально-модульный принцип построения*: ЭВМ состоит из модулей: центрального процессора (ЦП), постоянного запоминающего устройства (ПЗУ), оперативного запоминающего устройства (ОЗУ), внешнего запоминающего устройства (ВЗУ), устройств ввода и вывода (УВВ), подключенных к информационной магистрали (рис. 3.2 и 3.3). Эта магистраль состоит из шин управления: кодовой шины команд, адресов и данных. При такой архитектуре сокращается аппаратура, стандартизируется процедура обмена информацией, но исключается одновременный обмен между несколькими устройствами. ЦП состоит из устройства управления, арифметико-логического устройства, микропроцессорной памяти.

Внутренняя память ЭВМ включает в себя ПЗУ (самотестирование и загрузка ОС) и ОЗУ (хранение оперативной информации). Внешняя память: НЖМД, НГМД, CD-ROM, DVD-ROM, Zip-диск, стример (хранение больших объемов информации). Устройства ввода: клавиатура, мышь, трекбол, сканер, цифровая фото- и видеокамера. Устройства вывода: монитор, ЖК-дисплей, звуковые колонки, принтер, ЖК-проектор. Сопряжение внешних устройств с системной шиной осуществляется с помощью микроконтроллеров — устройств с одной или несколькими БИС, содержащих процессор, память и систему связи с датчиками и исполнительными органами, воздействующими на объект управления (клавиатуру, монитор, накопитель НЖМД и т.д.).

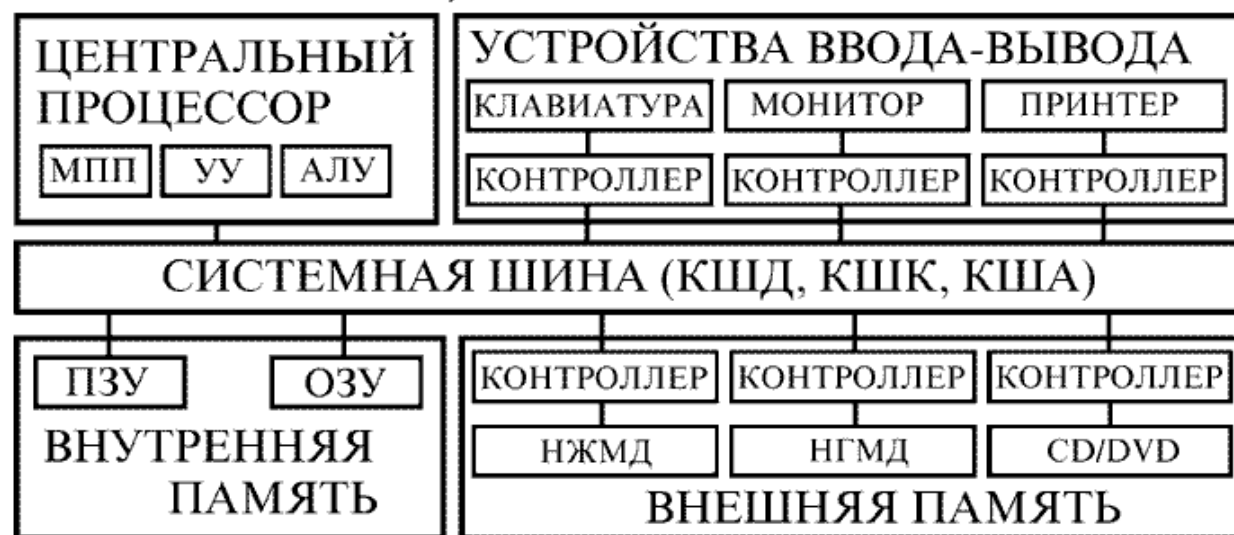


Рис. 3.2.

IBM-совместимые ПЭВМ построены в соответствии с *принципом открытой архитектуры*: компьютер не является неразъемным устройством, он может быть собран из независимых частей. Системный блок содержит системную плату, на которой находятся микропроцессор и микросхе-

мы ОЗУ, контроллеры различных устройств, накопители для жесткого, гибкого и компакт дисков, соединительные шлейфы, блок питания. На системной плате размещены слоты расширения (стандартные разъемы), в которые вставляются видео, звуковая, сетевая платы и т.д., обеспечивающие работу соответствующих внешних устройств.

Обсудим работу ЭВМ на примере персонального компьютера, архитектура которого представлена на рис. 3.3.1 и 3.3.2. Для того чтобы записать число в ячейку ОЗУ с некоторым адресом, процессор подает в шину команд код операции считывания, в шину адресов — адрес ячейки памяти, а в шину данных — двоичный код записываемого операнда. При считывании содержимого ячейки памяти ОЗУ или ПЗУ в шину команд подается код операции считывания, а шину адресов — адрес ячейки памяти. При этом данные перегоняются по шине данных из ПЗУ или ОЗУ в процессор. При вводе-выводе информации устройства ввода-вывода обмениваются информацией с ОЗУ по шине данных в режиме прямого доступа к памяти, а процессор управляет их работой по шине команд и адресов.

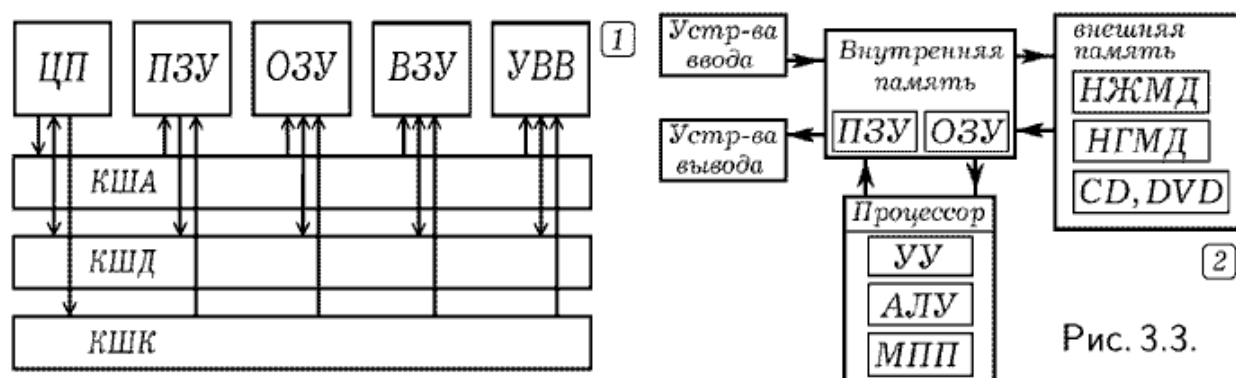


Рис. 3.3.

4. Центральный процессор ЭВМ. Центральный процессор (ЦП) — программно-управляемое устройство обработки информации, предназначенное для управления работой всех блоков ЭВМ, выполнения арифметических и логических операций. Функции процессора: чтение команд из ОЗУ; декодирование команд, то есть определение их назначения, способа выполнения и адресов операндов; исполнение команд; управление обменом информацией между микропроцессорной памятью (МПП), ОЗУ и периферийными устройствами; обработка прерываний; управление устройствами, составляющими ЭВМ. ЦП состоит из устройства управления, арифметико-логического устройства, микропроцессорной памяти, интерфейсной системы.

Устройство управления (УУ) формирует и подает во все блоки машины управляющие импульсы; выдает адреса требуемых ячеек памяти, и передает их в другие блоки ЭВМ. **Арифметико-логическое устройство АЛУ** состоит из регистров памяти, сумматора и схем управления; используется для выполнения арифметических и логических операций над числовой и символьной информацией. Для увеличения скорости работы АЛУ подключают математический сопроцессор.

Сумматор — электрическая схема, складывающая поступающие на вход двоичные машинные слова (по 2 или 4 байта). Включает в себя

два регистра быстродействующей памяти, в которые из шины данных помещают два слагаемых. После суммирования в одном из регистров памяти записывается результат, который и передается в шину данных. *Микропроцессорная память МПП* — память небольшой емкости, но чрезвычайно высокого быстродействия (время обращения к МПП примерно 1 нс). Состоит из регистров с разрядностью не менее машинного слова.

Интерфейс — совокупность схем сопряжения и связи устройств компьютера, обеспечивающая обмен информацией между ними. Интерфейсная система микропроцессора — внутренний интерфейс МП, буферные запоминающие регистры, схемы управления портами ввода-вывода и системной шиной. Она реализует сопряжение и связь с другими устройствами ЭВМ. Порт ввода-вывода — аппаратура сопряжения, позволяющая подключить к микропроцессору другое внешнее устройство.

Известно 5 типов микрокоманд процессора: 1) логические операции И, ИЛИ, НЕ, операция логического сравнения; 2) арифметическое сложение, вычитание, сравнение; 3) сложение, вычитание, умножение и деление чисел с фиксированной запятой; 4) операции управления, условный и безусловный переход; 5) операции записи и считывания информации при работе с внешними устройствами.

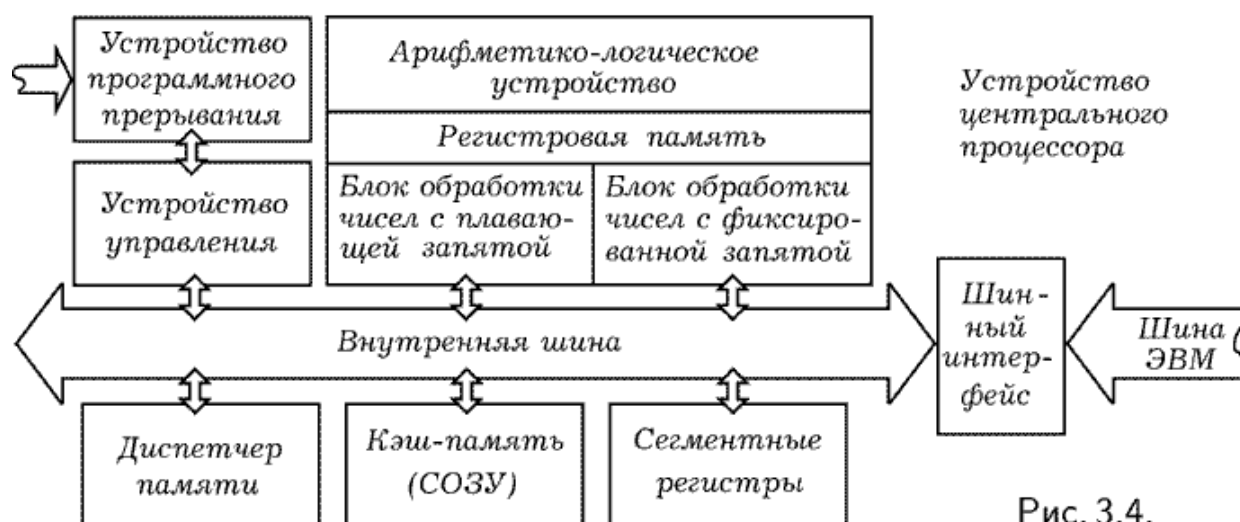


Рис. 3.4.

Обработка команды разбивается на несколько микрокоманд, которые могут выполняться последовательно друг за другом, либо параллельно. Например, выражение $5 \cdot a - 7/b$ в результате распараллеливания может быть вычислено так: одновременно двумя АЛУ определяются значения $5 \cdot a$ и $7/b$, после чего находится их разность. Многопоточная параллельная обработка данных и команд называется *конвейеризацией*. За один тактовый импульс осуществляется несколько элементарных операций, что позволяет достичь высокой производительности системы.

5. Работа процессора. Центральный процессор как и ЭВМ имеет магистральную структуру, то есть содержит информационную магистраль, через которую различные блоки ЦП обмениваются информацией. Этот обмен осуществляется в мультиплексном режиме, то есть попеременно, в

соответствии с принципом временного разделения.

Рассмотрим структуру 32-х разрядного процессора, произведенного фирмой Intel в 1985 г. (рис. 3.4). Основное отличие от предшествовавшего 16-ти разрядного процессора состоит в том, что шина адреса и шина данных имеют по 32 разряда. Это связано с увеличением разрядности регистров блока обработки чисел с фиксированной запятой от 16 до 32. Прямо на кристалле микропроцессора расположена внутренняя кэш-память — быстродействующая буферная память объемом от 8 до 512 Кбайт, необходимая для хранения часто используемой информации. Микропроцессор поддерживает мультипрограммный режим работы ЭВМ на аппаратном уровне. Это достигнуто за счет того, что в МПП предусмотрена возможность хранения нескольких готовых программ, запускаемых операционной системой (ОС) по прерываниям.

В МП входит: 1) устройство управления (буфер команд, дешифратор кода операции, управление защитой, управление переключением задач); 2) микропроцессорная память (диспетчер памяти, сегментные регистры памяти, кэш-память); 3) арифметико-логическое устройство (блоки обработки чисел с фиксированной и плавающей запятой); 4) интерфейсная система, состоящая из внутренней шины и шинного интерфейса, через который МП обменивается информацией с системной шиной ЭВМ. МПП имеет регистры, используемые для кратковременного хранения данных небольшого размера. Различают регистры общего назначения (РОН), регистр команды (РК), счетчик адреса команды (САК), регистр адреса (РА), регистр числа (РЧ), регистр результата (РР).

Порядок выполнения команды процессором: 1. Исходя из содержимого счетчика адреса команды (САК) УУ извлекает из ОЗУ соответствующую команду и помещает ее в регистр команд (РК). 2. Команда декодируется, входящие в нее адреса помещаются в регистры адреса (РА). 3. По адресам, размещенным в РА, производится чтение операндов и их запись в регистр чисел (РЧ). 4. Выполняется арифметическая или логическая операция, результат записывается в регистр результата (РР). 5. Результат пересылается в соответствующую ячейку ОЗУ. 6. Счетчик адреса команды (САК) увеличивается на 1, после чего осуществляется переход к следующей команде.

Взаимодействие ЦП с ОЗУ происходит так (рис. 3.3.1). Процессор по КША адрес требуемой ячейки памяти в двоичном виде, а по КШК — двоичный код команды считывания или записи. Для записи числа в соответствующую ячейку ОЗУ ЦП в КШД помещает двоичный код этого числа. При считывании числа из ОЗУ по КШД передается двоичный код числа в ЦП. Если ЦП имеет n -разрядную шину адреса, то он может непосредственно обратиться к 2^n ячейкам ОЗУ емкостью 1 байт. При этом реализуется прямая адресация: адресное слово в команде, обрабатываемой ЦП, является адресом ячейки памяти.

Микропроцессоры используются в различных устройствах автоматики: в станках с ЧПУ, в автоматическом карбюраторе, в бортовых ЭВМ

на самолетах, ракетах, космических аппаратах.

6. Параллелизм и конвейеризация. Параллельное выполнение нескольких программ достигается путем использования нескольких ЭВМ или нескольких процессоров, работающих одновременно и решающих различные части одной задачи. Существуют различные мультипроцессорные системы с одним или несколькими потоками команд и одним или несколькими потоками данных. Примером *параллельной вычислительной системы* является матричный процессор, состоящий из матрицы 64×64 процессорных элементов и рассчитанный на обработку числовых массивов. За один такт машинного времени матричный процессор параллельно обрабатывает все элементы массива (например, возводит в квадрат), что существенно повышает производительность системы.

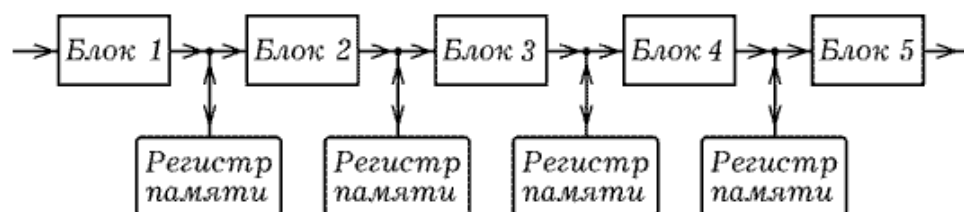


Рис. 3.5.

Еще одним способом распараллеливания вычислительного процесса является *конвейеризация*. Процесс выполнения команды можно разделить на несколько этапов или операций: 1) выборка команды (операция О1); 2) декодирование команды (операция О2); 3) определение адреса и выборка операндов из ОЗУ (операция О3); 4) выполнение команды (операция О4); 5) запись результата в ОЗУ (операция О5). При отсутствии конвейера каждая операция выполняется отдельным блоком, входящим в состав процессора. Получается, что для выполнения команды требуется 5 тактов машинного времени, причем в каждый данный момент загружен только один блок.

| К | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
|----|------|------|------|------|------|------|------|------|------|
| K1 | O1-1 | O1-2 | O1-3 | O1-4 | O1-5 | | | | |
| K2 | | O2-1 | O2-2 | O2-3 | O2-4 | O2-5 | | | |
| K3 | | | O3-1 | O3-2 | O3-3 | O3-4 | O3-5 | | |
| K4 | | | | O4-1 | O4-2 | O4-3 | O4-4 | O4-5 | |
| K5 | | | | | O5-1 | O5-2 | O5-3 | O5-4 | O5-5 |
| K6 | | | | | | O6-1 | O6-2 | O6-3 | O6-4 |
| K7 | | | | | | | O7-1 | O7-2 | O7-3 |
| K8 | | | | | | | | O8-1 | O8-2 |
| K9 | | | | | | | | | O9-1 |

Рис. 3.6.

Создание *конвейерной структуры* (pipeline) предполагает разделение исполняемой вычислительной операции на мелкие части, называемые ступенями или этапами, каждый из которых выполняется отдельным блоком аппаратуры. Так как команда раскладывается на 5 последовательных операций, то для ее выполнения требуется 5 последовательно расположенных

блоков аппаратуры (рис. 3.5). При завершении первого этапа результаты передаются следующему блоку для выполнения второго этапа и т.д. В это время устройство, используемое на первом этапе не простаивает, а применяется для обработки новых данных. Для сохранения промежуточных результатов они записываются в регистры памяти. Все это требует использования сложного управляющего устройства, обеспечивающего согласованную работу различных блоков вычислительной системы.

Состояния вычислительной системы в течение первого, второго и последующих тактов машинного времени представлены на рис. 3.6 и 3.7. В течение первого такта Т1 выполняется операция О1-1 команды К1, на втором такте Т2 — операция О1-2 команды К1 и операция О2-1 команды К2, на третьем такте Т3 — операция О1-3 команды К1, операция О2-2 команды К2, операция О3-1 команды К3 и т.д. (рис. 3.6). Из рис. 3.7 видно, что на первом такте Т1 блок 1 занят выполнением команды К1, на втором такте Т2 блок 1 занят выполнением команды К2, а блок 2 выполняет команду К1, на третьем такте Т3 блок 1 выполняет команду К3, блок 2 выполняет команду К2, а блок 3 выполняет команду К1 и т.д. Конвейеризация не сокращает время выполнения отдельной команды и даже наоборот за счет записи и считывания информации в регистры памяти оно незначительно увеличивается. Повышение производительности вызвано тем, что на различных ступенях получившегося конвейера в течение каждого такта машинного времени выполняются несколько операций. Наибольший эффект достигается тогда, когда новые операнды поступают с оптимальной скоростью и конвейер загружен полностью.

Рис. 3.7.

| Такт | Блок 1 | Блок 2 | Блок 3 | Блок 4 | Блок 5 |
|------|--------|--------|--------|--------|--------|
| T1 | К-1 | | | | |
| T2 | К-2 | К-1 | | | |
| T3 | К-3 | К-2 | К-1 | | |
| T4 | К-4 | К-3 | К-2 | К-1 | |
| T5 | К-5 | К-4 | К-3 | К-2 | К-1 |
| T6 | К-6 | К-5 | К-4 | К-3 | К-2 |
| T7 | К-7 | К-6 | К-5 | К-4 | К-3 |
| T8 | К-8 | К-7 | К-6 | К-5 | К-4 |
| T9 | К-9 | К-8 | К-7 | К-6 | К-5 |

7. Параметры процессоров ПЭВМ. Основные характеристики микропроцессора: 1. Разрядность шины данных, то есть количество битовых разрядов, обрабатываемых за один такт и пересылаемых в ОЗУ. 2. Разрядность шины адреса, определяющий максимальный объем адресуемой ОЗУ. 3. Тактовая частота. От них зависит быстродействие вычислительной системы, измеряемое в MIPS (Million Instructions Per Second — миллион операций в секунду). В таблице на рис. 3.8 представлены некоторые процессоры фирмы Intel (INTEgrated ELectionics), используемые в ПЭВМ, указаны год выпуска, частота шины, частота МП, разрядность шины данных и шины адреса, объем адресуемой памяти. В современных ПК также используются процессоры М-1 и М-2 фирмы Cyrix, К-5, К-6,

К-7 фирмы AMD (American Micro-Device).

Сейчас в персональных ЭВМ применяются 32- и 64-разрядные процессоры. Разрядность шины данных микропроцессора определяет разрядность ЭВМ в целом. Разрядность шины адреса процессора задает его адресное пространство, то есть максимальное количество ячеек ОЗУ, которое может непосредственно адресовано микропроцессором. Если шина имеет n разрядов, то адресное пространство — 2^n ячеек емкостью в 1 байт. Если шина адреса имеет 16 или 32 разряда, то объем адресного пространства МП равен 2^{16} байт = 64 Кбайта или 2^{32} байт = 4 Гбайта.

Если первое поколение процессоров (8088 и 8086) изготавливались по 3 мкм и 1,5 мкм технологии и имели 0,1 млн. транзисторов, то современные процессоры (AMD Athlon 64) — по 0,13–0,9 мкм технологии и содержат около 100 млн. транзисторов.

8. Иерархия памяти ЭВМ. Память ЭВМ должна иметь большую информационную емкость V , малое время обращения t (высокое быстродействие), высокую надежность и низкую стоимость. Но с увеличением емкости снижается быстродействие и растет стоимость. Чем больше быстродействие ЗУ, тем выше стоимость хранения 1 байта и меньше емкость. Деление памяти на ОЗУ и ВЗУ не снимает это противоречие полностью, так как различие в быстродействии процессора, ОЗУ и ВЗУ очень велико. Поэтому обмен информацией производится через дополнительные буферные устройства, память ЭВМ имеет многоуровневую структуру.

| Тип МП | Год | Частота шины, МГц | Частота МП, МГц | Шина КШД | Шина КША | Объем ОЗУ |
|-------------|------|-------------------|-----------------|----------|----------|----------------|
| 8086 | 1978 | 4,77–8 | 4,77–8 | 16 | 20 | 1 Мбайт |
| 80286 | 1982 | 6–20 | 6–20 | 16 | 24 | 16 Мбайт |
| 80386 | 1985 | 16–33 | 16–33 | 16/32 | 32 | 4 Гбайт |
| 80486 | 1989 | 25–40 | 25–120 | 32 | 32 | 4 Гбайт |
| Pentium | 1993 | 60–66 | 60–200 | 64 | 32 | 4 Гбайт |
| Pentium II | 1997 | 66 | 233–300 | 64 | 32 | 4 Гбайт |
| Pentium II | 1999 | 66–100 | 266–533 | 64 | 32 | 4 Гбайт |
| Pentium III | 1999 | 100 | 450–1200 | 64 | 32 | 4 Гбайт |
| AMD Athlon | 1999 | 266 | 500–2200 | 64 | 32 | 4 Гбайт |
| Pentium IV | 2000 | 400 | 1,4–3,4 ГГц | 64 | 32 | 4 Гбайт |
| Athlon 64 | 2003 | 400 | 2 ГГц | 64 | 64 | 2^{34} Гбайт |

Рис. 3.8.

Микропроцессорная память — высокоскоростная память небольшой емкости, входящая в ЦП и используемая АЛУ для хранения операндов и промежуточных результатов вычислений. *КЭШ-память* — это буферная, не доступная для пользователя память, автоматически используемая компьютером для ускорения операций с информацией, хранящейся в медленно действующих запоминающих устройствах. В ПЭВМ для ускорения операций с основной памятью организуется регистровая КЭШ-память внутри микропроцессора (КЭШ-память первого уровня) или вне микропроцессора (КЭШ-память второго уровня) на системной плате. Для ускорения операций с дисковой памятью (НЖМД, НГМД) организуется буферная

КЭШ-память на ячейках электронной памяти.

Внутренняя память состоит из ПЗУ (ROM — Read Only Memory — память только для чтения) и ОЗУ (RAM — Random Access Memory — память произвольного доступа). Микросхемы ПЗУ размещены на системной плате и используются для хранения неизменяемой информации: загрузочных программ операционной системы (ОС), программ тестирования устройств компьютера и некоторых драйверов базовой системы ввода-вывода (BIOS — Base Input-Output System) и др. Из ПЗУ можно только считывать информацию, емкость ПЗУ — сотни Кбайт. Это энергонезависимая память, — при отключении ЭВМ информация сохраняется.

Внешняя память относится к внешним устройствам ЭВМ и используется для долговременного хранения любой информации, которая может потребоваться. В ВЗУ хранится программное обеспечение ЭВМ. К устройствам внешней памяти относятся НЖМД и ЖМД, НГМД и ГМД (магнитный диск), стример (НМЛ — накопитель на магнитной ленте), оптические накопители для CD-ROM и DVD-дисков.

Виды памяти изображены на рис. 3.9.1: 1) регистры микропроцессорной памяти, а также кэш-память первого и второго уровня (время доступа $t = 10^{-9} - 10^{-6}$ с, объем $V = 10^2 - 10^4$ бит); 2) внутренняя память ПЗУ, ОЗУ ($t = 10^{-6} - 10^{-3}$ с, $V = 10^4 - 10^7$ бит); 3) внешняя память ($t = 10^{-3} - 1$ с, $V = 10^7 - 10^9$ бит); 4) массовая или архивная память ($t = 1 - 10$ с, $V = 10^9 - 10^{10}$ бит). Эта система запоминающих устройств работает как единое ЗУ с большой емкостью (за счет внешних ЗУ) и высоким быстродействием (за счет внутренних ЗУ).

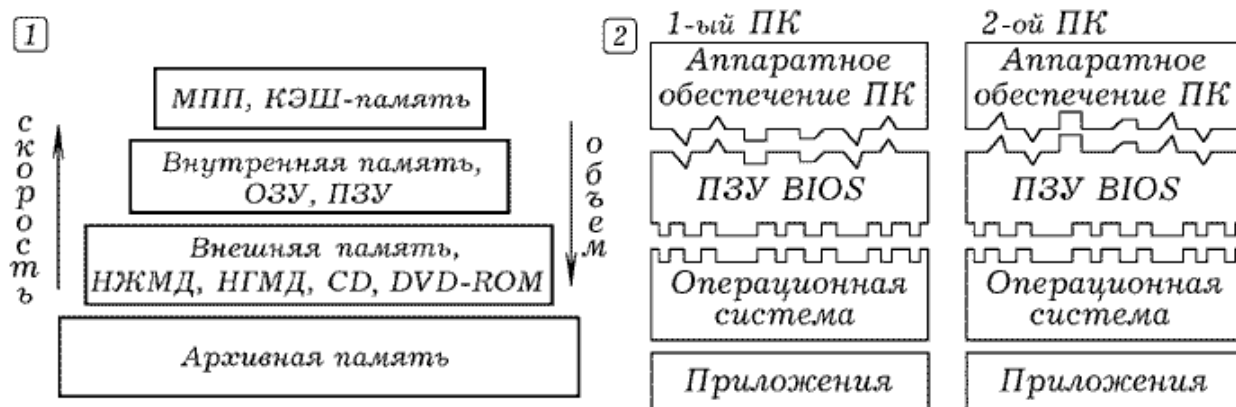


Рис. 3.9.

Информационная структура внешней памяти — файловая. Наименьшей именуемой единицей памяти, к которой может обратиться операционная система, является кластер. Файл занимает целое число кластеров, каждый из которых имеет свой адрес, указанный в таблице FAT. Информация в файле состоит из битов и байтов, но они не имеют адресов, так как носитель (магнитный диск) не дискретный.

9. Организация внутренней памяти. ОЗУ предназначено для хранения информации (программ и данных), непосредственно участвующей в работе ЭВМ в текущий или в последующие моменты времени. ОЗУ

— энергозависимая память, то есть при отключении питания записанная в нем информация теряется. Она состоит из нескольких больших интегральных схем (БИС), содержащих матрицу ячеек памяти, каждая из которых способна находиться в двух устойчивых состояниях и хранить 1 бит информации.

Внутренняя память дискретна, ее информационная структура представляет собой матрицу двоичных ячеек, в каждой из которых хранится по 1 биту информации. Она адресуема: каждый байт (8 ячеек по 1 биту) имеет свой адрес — порядковый номер. Доступ к байтам ОЗУ происходит по адресам. Так как ОЗУ позволяет обратиться к произвольному байту, то эта память называется *памятью произвольного доступа* (RAM).

Оперативная память ЭВМ подразделяется на две области: 1) непосредственно адресуемая память емкостью 1024 Кбайт, занимающая ячейки с адресами от 0 до 1024 Кбайт; 2) расширенная память с адресами 1024 Кбайт и выше, доступ к которой возможен при использовании специальных программ (драйверов). Стандартная память — непосредственно адресуемая память от 0 до 640 Кбайт. Верхняя память — непосредственно адресуемая память от 640 до 1024 Кбайт. Она зарезервирована для видеопамати и работы ПЗУ.

| Адрес | Содержимое байта | | | | | | | |
|-------|------------------|---|---|---|---|---|---|---|
| 0001h | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0002h | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| FFFFh | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

Преимущества ОЗУ: высокое быстродействие и прямой адресный доступ к ячейке. Недостаток ОЗУ: небольшая емкость (64-128-256-512-1024 Мбайт), энергозависимость.

С целью увеличения объема адресуемой памяти используется *относительная адресация*. Адресное пространство разделено на блоки смежных ячеек памяти размером 64 Кбайт, называемые *сегментами*. Внутри каждого сегмента организована линейная адресация: ЭВМ может обратиться к байту 0, к байту 1 и т.д. относительно начала сегмента, которое имеет *базовый адрес*. *Исполнительный адрес* определяется как сумма базового адреса и *адреса смещения Offset*, который определяет номер байта внутри сегмента.

В ЦП i8086 каждый базовый адрес имеет длину 16 бит или 4 шестнадцатиричных разряда, поэтому физический адрес ячейки ОЗУ находится так: $Address = Segment * 16 + Offset$. Базовый адрес и адрес смещения записываются в регистры памяти, затем базовый адрес смещается на 4 бита (1 шестнадцатиричный разряд) и суммируются. Допустим, регистр *DS* содержит базовый адрес $Segment = 2134_{16}$, а регистр *SI* — адрес $Offset = 4563_{16}$, тогда физический адрес ячейки памяти равен:

$$Address = Segment * 16 + Offset = 2134_{16} \cdot 10_{16} + 4563_{16} = 258A3_{16}.$$

Оперативная память включает в себя сравнительно медленную динамическую память DRAM и быструю статическую память SRAM. Динамическая память DRAM состоит из запоминающих ячеек (рис. 2.26.2), выполненных в виде конденсаторов, собранных в ИС и образующих двумерную матрицу. При записи логической 1 соответствующий конденсатор

заряжается, а при записи 0 — разряжается. Схема считывания разряжает через себя конденсатор, и чтобы записанная информация сохранилась, подзаряжает его до прежнего уровня. Со временем конденсатор разряжается, информация теряется, такая память требует периодической подзарядки (регенерации) и может работать только в динамическом режиме.

Статическая память SRAM при наличии питания хранит информацию сколько угодно долго. Состоит из триггеров — элементов с двумя устойчивыми состояниями (рис. 2.26.1). Статическая память SRAM имеет время доступа 1–10 нс, и поэтому может работать на частоте системной шины ЭВМ. Центральный процессор работает быстрее DRAM, поэтому ОЗУ большого объема на DRAM используют совместно с небольшой кэш-памятью на SRAM. Кэш-память 1 уровня находится внутри процессора, а 2 уровня — вне процессора на системной плате.

ПЗУ состоит из ИС, программируемых в процессе изготовления или после него. Они имеют матричную структуру и представляют собой систему параллельных проводников, которые в узлах соединены переключками — полупроводниковыми диодами (рис. 2.25.1). Наличие или отсутствие переключки соответствует логическому 0 или 1 в данной ячейке памяти. Программирование ПЗУ состоит в прожигании переключек по требуемым адресам.

Различают: 1) масочные ПЗУ, их содержимое определяется рисунком технологического шаблона (быстродействие 30–70 нс); 2) однократно программируемые ПЗУ, запись информации в которые осуществляется путем прожигания ячеек памяти в специальных устройствах — программаторах; 3) репрограммируемые ПЗУ, которые могут быть перепрограммированы. Наиболее распространены ПЗУ, информация в которых стирается ультрафиолетовыми лучами.

10. Системная шина и другие устройства ЭВМ. *Системная шина* — основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой. Она включает в себя провода, разъемы и схемы сопряжения для параллельной передачи всех разрядов числового кода операнда, адреса ячейки памяти или команды. Системная шина включает в себя: кодовую шину данных (КШД); кодовую шину адреса (КША); кодовую шину команд (КШК); шину питания, имеющую провода и схемы подключения для блоков ПК к системе энергоснабжения.

Системная шина обеспечивает три направления передачи информации: 1) между центральным процессором и ОЗУ; 2) между центральным процессором и портами ввода-вывода внешних устройств (клавиатура, монитор); 3) между ОЗУ и портами ввода-вывода внешних устройств (в режиме прямого доступа к памяти). Шины ЭВМ характеризуются разрядностью и скоростью передачи информации (байт/с). *Разрядность шины* — число проводов, по которым передаются двоичные сигналы. Разрядность шины данных обычно такая же как у микропроцессора.

Прямой доступ к памяти (DMA — Direct Memory Access) — способ обмена данными, при котором их передача между основной памятью и

внешними устройствами осуществляется минуя процессор. *Контроллер прямого доступа к памяти* освобождает ЦП от прямого управления накопителями на магнитных дисках, что повышает быстродействие ЭВМ. Без него обмен данными между ВЗУ и ОЗУ осуществляется через ЦП, а при его наличии данные непосредственно передаются между ВЗУ и ОЗУ, минуя ЦП.

Внутримашинный системный интерфейс — система связи узлов и блоков ЭВМ, представляющий собой совокупность шин, проводов и схем сопряжения с компонентами компьютера. В качестве системного интерфейса используются шины расширения и локальные шины. Шины расширения используются для большого числа разнообразных устройств и предназначены для расширения функциональных возможностей ЭВМ (шины ISA, EISA, MCA). Локальные шины обеспечивают связь процессора с ОЗУ, ВЗУ, видеосистемой и т.д. (шины VLB и PCI).

Математический сопроцессор используется для ускоренного выполнения операций над двоичными числами с плавающей запятой, над двоично-кодированными числами, а также для вычисления некоторых трансцендентных функций типа e^x , $x^{2/3}$, $\log_3 x$ и т.д.

Сопроцессор ввода-вывода за счет параллельной работы с ЦП значительно ускоряет выполнение процедур ввода-вывода при обслуживании нескольких внешних устройств (дисплей, принтер, НЖМД, НГМД и др.); освобождает ЦП от обработки процедур ввода-вывода в том числе реализует и режим прямого доступа к памяти. Если вычислительная система имеет несколько внешних устройств, обмен информацией осуществляется следующими способами: 1) последовательный опрос всех устройств; 2) внешнее устройство посылает сигнал прерывания, ЦП прекращает выполнять текущую программу и начинает выполнять программу обслуживания прерывания из ОЗУ; 3) внешнее устройство посылает сигнал и работает в режиме прямого доступа к памяти.

Контроллер прерываний обслуживает процедуры прерывания, то есть временный останов выполнения одной программы в целях оперативного выполнения другой, в данный момент более важной (приоритетной) программы. Аппаратные прерывания инициируются аппаратурой (сигналом от принтера, таймера, нажатием клавиши). Логические прерывания возникают при нестандартной ситуации ЦП (деление на нуль, переполнение регистров). Программные прерывания появляются, когда одна из программ должна получить доступ к аппаратным средствам.

Процедуры ввода-вывода информации выполняются по прерываниям. Таймер более десятка раз в секунду вырабатывает импульсы, опрашивающие устройства ввода и вывода (клавиатуру, мышь и т.д.). Перемещение мыши, нажатие кнопок клавиатуры регистрируется ЭВМ. Контроллер прерываний принимает запрос на прерывание от внешних устройств, определяет уровень его приоритета и выдает соответствующий сигнал в центральный процессор. Процессор приостанавливает выполнение исполняемой программы и переходит к выполнению программы обслуживания прерывания, запрошенного внешним устройством. По окончании програм-

мы обслуживания возобновляется выполнение прерванной программы.

Генератор тактовых импульсов ГТИ генерирует последовательность электрических импульсов. С каждым тактом ЭВМ совершает одну операцию: пересылает число, выполняет команду, осуществляет математическую операцию. В качестве *источника питания* используется сетевой блок питания, состоящий из трансформатора, выпрямителя и фильтра, и аккумулятор. *Таймер* — электронный секундомер, обеспечивающий отсчет текущего момента времени (год, месяц, часы, минуты, секунды, доли секунд). Таймер подключен к аккумулятору и продолжает работать при выключении ЭВМ.

Системный блок ПЭВМ обычно включает в себя системную плату, блок питания, накопители на дисках, разъемы для дополнительных устройств и платы расширения с контроллерами (адаптерами) внешних устройств. На системной плате размещаются: микропроцессор, математический сопроцессор, генератор тактовых импульсов, блоки (микросхемы) ОЗУ и ПЗУ, контроллеры (адаптеры) клавиатуры, НЖМД, НГМД, контроллер прерываний, таймер и др. Внешние устройства (монитор, принтер, клавиатура и др.) подключаются к системному блоку.

11. Операционная система ЭВМ. *Операционная система (ОС)* — неотъемлемая часть программного обеспечения ЭВМ, управляющая всеми ее аппаратными компонентами. Она является буфером (посредником) между аппаратными ресурсами ЭВМ, системой BIOS с одной стороны, и программным обеспечением с другой (рис. 3.9.2). ОС делает возможной работу стандартных программ на ПЭВМ с различной конфигурацией. В состав ОС входят следующие модули:

1. Базовый модуль или ядро ОС: управление работой программ и файловой системой, доступ к файлам, информационный обмен между периферийными устройствами.
2. Командный процессор: исполнение команд пользователя, поступающих через устройства ввода.
3. Драйверы периферийных устройств: обеспечение их согласованной работы с процессором.
4. Дополнительные сервисные программы или утилиты: облегчение работы пользователя с компьютером.

Файлы, составляющие ОС, хранятся на жестком диске, поэтому ОС называется дисковой ОС (ДОС). После включения компьютера происходит его поэтапная загрузка — файлы ОС загружаются в ОЗУ.

Перечислим этапы загрузки ОС на современных ПЭВМ:

1. В ПЗУ содержатся программы самотестирования POST (Power-On Self Test) аппаратной части компьютера, которые начинают выполняться сразу при включении ЭВМ.
2. Инициализируются загрузочные устройства: клавиатура, позволяющая управлять загрузкой; монитор, на который выводятся сообщения загрузчика; устройство начальной загрузки: НГМД, НЖМД, CD-ROM.
3. МП обращается к первому (загрузочному) сектору НЖМД, находит программу-загрузчик, считывает ее в ОЗУ, и передает ей управление.



ВИНЕР Норберт (1894–1964 гг.)

Известный американский математик, написавший в 1948 г. книгу "Кибернетика или Управление и связь в животном и машине", с которой принято отсчитывать появление новой науки — кибернетики. Занимался теорией вероятности, рядами и интегралами Фурье, решал прикладные задачи, разрабатывал основы теории информации. Автор книг "Кибернетика и общество", "Творец и робот".

4. Программа-загрузчик ищет на диске базовый модуль ОС, загружает его в ОЗУ и передает ему управление.

5. Базовый модуль загружает остальные модули ОС в ОЗУ. После окончания загрузки ОС управление передается командному процессору (программе) и на экране появляется командная строка (ОС MS DOS), либо графический интерфейс (Windows, Linux, Unix).

Во время работы ПК в ОЗУ находится базовый модуль ОС и командный процессор; драйверы устройств и утилиты подгружаются по мере необходимости.

В ПЭВМ используются различные ОС (MS DOS, Windows 3.11, Windows 95, Windows NT, Unix, Linux и др.), каждая из которых ориентирована на определенную разрядность процессора, его тип и емкость ОЗУ. Они имеют гибкую файловую систему, основанную на иерархической структуре каталогов; позволяют подключать периферийные устройства, работать в дружественном пользовательском интерфейсе.

12. Программа BIOS. Чтобы одна и та же ОС работала на различных ПК, имеющих уникальную конфигурацию, используется программа BIOS (Base Input-Output System), хранящаяся в ПЗУ. Она является буфером между аппаратными ресурсами ПЭВМ и ОС, предназначена для изоляции ОС от специфики аппаратуры (рис. 3.9.2). BIOS обеспечивает программную поддержку устройств ПЭВМ, конфигурирование аппаратных средств, их диагностику и вызов загрузчика операционной системы.

При запуске ПЭВМ тестируется процессор, память, системные средства ввода-вывода, а также конфигурирует все программно управляемые аппаратные средства системной платы. В BIOS хранятся настройки аппаратных средств, осуществляется автоматическое распределение системных ресурсов. BIOS обслуживает аппаратные прерывания от системных устройств (таймера, клавиатуры, дисков), управляет энергопотреблением и автоматическим конфигурированием.

BIOS управляет питанием компьютера, периферийных устройств, процессора, их производительностью, конфигурирует устройства на системной плате, обрабатывает системные события (перегрев устройств, их установка или снятие), управляет батареями, осуществляет термоконтроль ЦП.

3.2. ВНЕШНИЕ УСТРОЙСТВА ЦИФРОВЫХ ЭВМ

1. Магнитные запоминающие устройства. *Носитель информации* — материальный объект, используемый для хранения информации. Различают бумажные носители (перфокарты, перфоленты), магнитные носители (ленты, диски, барабаны) и оптические носители (лазерные диски). *Накопитель* — механическое устройство, управляющее записью, хранением и считыванием данных. В ЭВМ используют накопители на гибких магнитных дисках НГМД и накопители на жестких магнитных дисках НЖМД, накопители на оптических и магнитооптических дисках НОД.

Накопитель на жестком магнитном диске НЖМД состоит из нескольких магнитных дисков МД, установленных на валу двигателя, вблизи которых расположены магнитные головки, связанные с механическим приводом. Информация на МД записывается и считывается магнитными головками вдоль концентрических окружностей — *дорожек (треков)*. *Цилиндр* — совокупность дорожек МД, равноудаленных от его центра. Каждая дорожка МД разбита на *секторы* — области емкостью 512 байт, определяемые идентификационными метками и номером. Сектор — минимальный объем данных, с которым могут работать программы в обход ОС.

Обмен данными между НМД и ОЗУ осуществляется последовательно целым числом секторов. *Кластер* — минимальный объем размещения информации на диске, воспринимаемый ОС, он состоит из одного или нескольких смежных секторов дорожки. *Форматирование диска* — разметка на диске дорожек (треков) и секторов, маркировка дефектных секторов, запись служебной информации.

Файл — область внешней памяти (НГМД, НЖМД, НОД), используемая для хранения массива однотипных данных (текстовых, графических, звуковых и т.д.). Каждому файлу выделяется целое число кластеров, которые могут находиться в любом месте диска, то есть необязательно быть смежными. Файлы, хранящиеся в разбросанных по диску кластерах, называют *фрагментированными*. Процедура перезаписи информации, при которой файлы размещаются в последовательных секторах на смежных дорожках, называется *дефрагментацией* диска.

На каждом магнитном диске имеется таблица размещения файлов FAT16 или FAT32, в которой каждый кластер имеет свой двоичный код (адрес) из 16 или 32 разрядов. Так как в 32 разрядах можно записать 2^{32} различных значений, то число кластеров (а значит и записанных файлов) на диске или логическом разделе не может превышать 2^{32} . Чем больше МД, тем больше размер кластера. Для рационального использования МД его разбивают на логические разделы *C, D, E...*

НГМД с форм-фактором 3,5" имеют по 80 дорожек на каждой стороне, 18 секторов по 512 байт на каждой дорожке, общая емкость дискеты $2 \cdot 80 \cdot 18 \cdot 512 = 1474560$ байт = 1,44 Мбайт, доступ к информации 0,1–1 с, скорость чтения/записи 50 кбайт/с. НЖМД имеет емкость 100–200 Гбайт, время доступа 1–100 мс, скорость чтения/записи 1 Мбайт/с, скорость вращения 3600 об/мин. Емкость Zip-дисков — 100 Мбайт и выше.

Емкость компакт-диска CD-ROM — 700 Мбайт.

В машинах-серверах и суперЭВМ применяются *дисковые массивы RAID* (Redundant Array of Independent Disks — матрица с резервируемыми независимыми дисками), в которых несколько НЖМД объединены и образуют один большой диск.

Накопитель на магнитной ленте (стример) состоит из лентопротяжного механизма, магнитной ленты и магнитной головки. Кассеты с магнитной лентой (картриджи) имеют емкость более 2000 Мбайт. *Стримеры* имеют высокую инерционность, используются для резервного копирования и архивирования информации.

Кроме магнитных накопителей в настоящее время применяются *флэш-накопители* — портативные носители информации, подключаемые к ПЭВМ по USB. Они состоят из микросхемы флэш-ПЗУ, спецконтроллера и интерфейса. Преимущества: малое время доступа, надежность, компактность. Емкость — 1–2 Гбайт, скорость чтения/записи 500–1000 Кбайт/с, масса 15 г.

2. Оптические и магнитооптические запоминающие устройства.

В *оптических ЗУ* запись и считывание осуществляется с помощью источника света. Накопители на оптических дисках (НОД) включают в себя источник (полупроводниковый лазер) и приемник света (фотодиод), оптическую запоминающую среду, модулятор светового пучка, поляризационную призму. Компакт-диск состоит из жесткой прозрачной основы, на которую нанесен рабочий и защитный слой. При записи (воспроизведении, стирании) диск вращается, а луч лазера, сфокусированный на дорожку, перемещается вдоль радиуса вращающегося диска.

Неперезаписываемые лазерно-оптические диски или компакт-диски ПЗУ CD-ROM (Compact Disk Read Only Memory) поставляются фирмой-изготовителем с уже записанной информацией. Для их изготовления создается первичный мастер-диск: в специальном устройстве лазерным лучом большой мощности выжигают на рабочем слое диска след — дорожку с микроскопическими впадинами (питами). Тиражирование CD-ROM осуществляется путем литья под давлением по мастер-диску. В НОД записанная на CD-ROM информация считывается лазерным лучом меньшей мощности, который отражаясь от углублений изменяет свою интенсивность. Это регистрируется фотодиодом.

Также используются перезаписываемые лазерно-оптические диски с однократной (CR-R) и многократной (CD-RW) записью. Устройство накопителя на оптических CD-дисках показано на рис. 3.10.1 и 3.11.1. Он состоит из лазера 1, модулятора 2, поляризационной призмы 3, линзы 4, собственно диска 5 и фотоприемника 6. Сигнал, несущий информацию, подается на модулятор, который изменяет свою прозрачность в такт с изменениями напряжения. В процессе записи лазерный луч в прожигает микроуглубления под защитным слоем диска, либо изменяет оптические свойства рабочего слоя.

CD-диск разделен на секторы по 3234 байта, разбитых на 98 фреймов (кадров) по 33 байта. Из них 24 байта — информационные, 1 байт — та-

блица содержания, 8 байт используются для контроля четности. Так как повреждения поверхности диска могут разрушить целые фреймы, отдельные байты разделяются и перемешиваются с байтами из других фреймов (интерливинг). Все байты из одного фрейма растягиваются на 108 фреймов. При считывании осуществляется деинтерливинг.

Преимущества компакт-дисков CD-ROM состоят в следующем: высокая плотность записи, отсутствие механического контакта при работе, долговечность записи, надежность, небольшие размеры. CD-ROM имеют емкость от 50 Мбайт до 1,5 Гбайт, время доступа от 30 до 300 мс, скорость считывания информации от 150 до 1500 Кбайт/с. Применяемые компакт-диски имеют диаметр 3,5" и 5,25" (1"=1 дюйм=2,53 см). Переход на стандарт *Blue-Ray* предусматривает запись и чтение информации сине-фиолетовым лазером с длиной волны 405 нм, при этом длина пита (одного углубления) 0,160–0,138 мкм.

Магнитооптические диски с однократной записью отличаются от обычных тем, что на их контрольные дорожки наносятся специальные метки, запрещающие стирание и повторную запись. Емкость магнитооптических дисков достигает нескольких Гбайт, время доступа от 15 до 150 мс, скорость считывания до 2000 Кбайт/с. Основным недостатком является высокая цена.

При записи или стирании информации на перезаписываемые магнитооптические диски лазерный луч осуществляет местный разогрев поверхности диска. Записывающая магнитная головка намагничивает разогретый участок. Считывание информации производится лазерным лучом меньшей мощности, при отражении которого от намагниченного участка изменяется ориентация плоскости поляризации. Это регистрируется с помощью анализатора и фотоприемника.

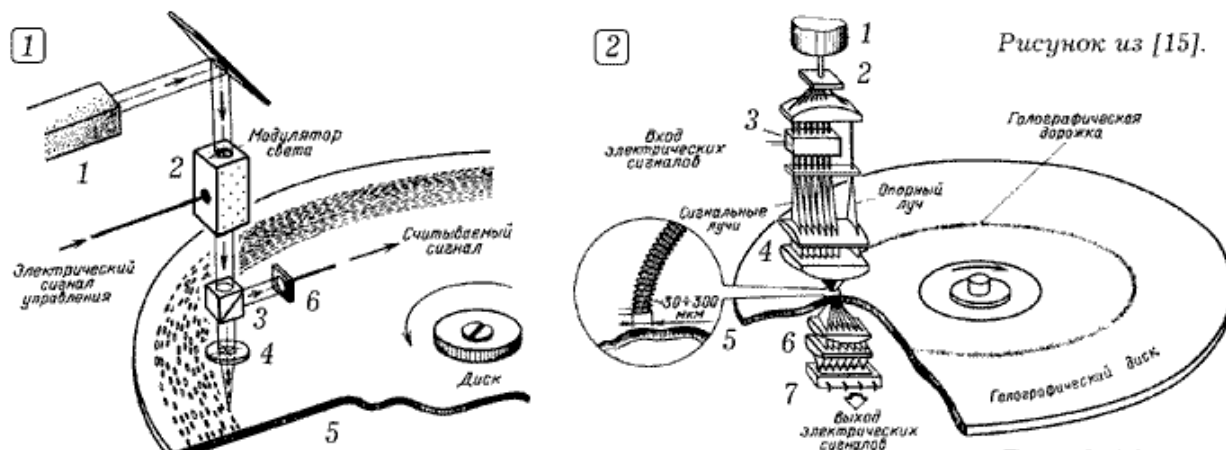


Рис. 3.10.

DVD-диск (Digital Versatile Disk) — цифровой многофункциональный диск. Носителем информации является диск диаметром 120 мм толщиной 1,2 мм. Внешне он похож на компакт диск. Бывают DVD-диски односторонние, двухсторонние, с одним и двумя рабочими слоями с каждой стороны. Однослойный односторонний DVD-диск имеет емкость 4,7 Гбайт, двухслойный односторонний — 8,5 Гбайт, двухслойный двухсторонний диск — 17 Гбайт.

Основным элементом *голографических ЗУ* является запоминающая голографическая матрица, состоящая из небольших голограмм (диаметром 2–5 мм), на каждой из которых может быть записано до 10^4 бит информации. Считывание осуществляется многоэлементным фотоприемным устройством. Дисковая система памяти на одномерных голограммах состоит из лазера 1, голографического расщепителя 2, многоканального модулятора 3 света, системы линз 4, голографического диска 5, системы линз 6 и многоканального фотоприемника 7 (рис. 3.10.2, 3.11.2). На диск нанесена голографическая дорожка, состоящая из последовательности голограмм. Расщепитель (дифракционная решетка) расщепляет лазерный луч на несколько лучей. Модулятор света последовательно пропускает лучи, которые вместе с опорным лучом создают интерференционную картину, регистрируемую фотослоем диска. При считывании голограмма освещается лазером, за ней возникает дифракционная картина, на соответствующие фотоприемники падает свет.

3. Устройства ввода. К устройствам ввода информации относятся: клавиатура, мышь, трекбол, трекпоинт, джойстик, графические планшеты, световое перо, сенсорные экраны, сканер, аудио- и видео магнитофон, микрофон, цифровой фотоаппарат, видеокамера, телевизионный тюнер, ресивер, музыкальный инструмент, АЦП, различные датчики, игровые устройства, киберперчатки и киберкостюм.

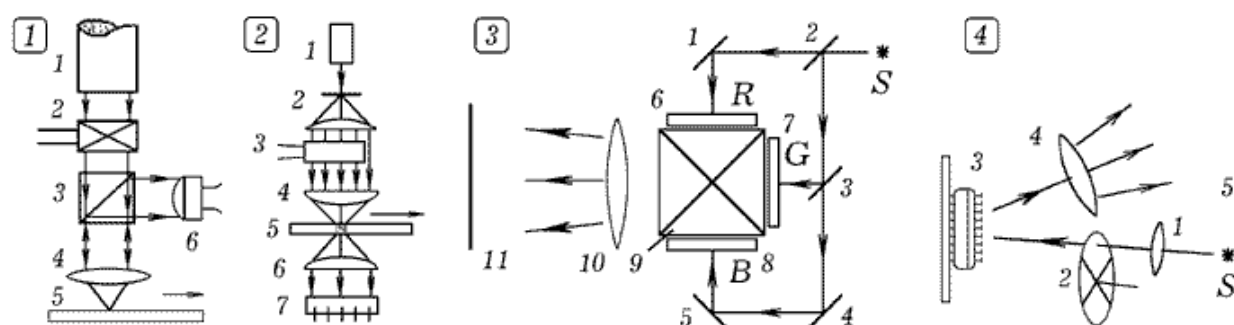


Рис. 3.11.

Клавиатура — устройство ручного ввода информации в ЭВМ, состоящее из совокупности клавиш различного назначения и схемы сопряжения. **Курсор** — символ (прямоугольник или жирная черта), указывающий позицию на экране дисплея, в которой будет отображаться очередной выведенный на экран символ. **Драйвер клавиатуры** — специальная программа, обеспечивающая отображение на экране монитора символа, набранного на клавиатуре. **Контроллер клавиатуры** — устройство сопряжения клавиатуры с ЭВМ. Он тестирует клавиатуру при включении ЭВМ; опрашивает состояния клавиш; запоминает до 20 отдельных скан-кодов клавиш; преобразует скан-коды нажатых клавиш в коды ASCII. При нажатии (отпускании) клавиши контроллер запоминает код нажатия (отпускания). Одновременно поступает запрос на соответствующее аппаратное прерывание. При выполнении прерывания скан-код преобразуется в код ASCII, и оба кода (скан-код и ASCII-код) пересылаются в соответствующее поле ОЗУ машины. Если клавиша нажата более 0,5 с, то генерируются повторный

код нажатия, на экране появляются еще один символ.

Манипуляторы (устройства указания): джойстик — рычаг, мышь, трекбол — шар в оправе, световое перо, геймпад и др.

Мышь — устройство ввода, представляющее собой коробку с кнопками, при движении которой по поверхности стола происходит перемещение указателя на экране. Если разрешение мыши 900 dpi (dots per inch — точек на дюйм), то при ее перемещении на 1 дюйм влево микроконтроллер мыши выдает сигнал о смещении на 900 единиц влево. Драйвер мыши обеспечивает соответствующее смещение курсора.

На нижней стороне оптико-механической мыши имеется отверстие, в котором находится шарик диаметром 1,5–2 см. Шарик касается двух взаимно перпендикулярных валиков горизонтального и вертикального перемещения. Каждый валик связан с диском, имеющим растровые прорези. По обе стороны каждого диска напротив друг друга расположены по два светодиода и два фотодиода. При перемещении мыши по коврику шарик поворачивает соответствующий валик с диском, фотодиоды периодически освещаются и затемняются, на их выходах появляются импульсы. Они преобразуются микроконтроллером в совместимые с ЭВМ данные и передаются на системную плату. Существуют мыши, подключаемые к системной шине, оптические, инфракрасные мыши и радиомыши.

Трекбол представляет собой перевернутую мышь. В нем обычно используется оптико-механический принцип ввода данных. Применяется в ноутбуках. *Джойстик* — манипулятор выполненный в виде ручки с кнопкой, укрепленной на шарнире. Используется в играх. Цифровой джойстик регистрирует поворот ручки управления влево, вправо, вверх, вниз и состояние кнопки "огонь". Аналоговый джойстик реагирует на небольшие движения управляющей ручки.

Световое перо — содержит светодиод, регистрирующий изменения яркости в том месте экрана, куда им указывает пользователь. По запаздыванию сигнала с фотодиода по отношению к пилообразным колебаниям, формируемым генераторами кадровой и строковой разверток, определяется точка, на которую указывает перо. В эту точку ставится курсор. При нажатии на кнопку, находящуюся на световом пере, нажимается "кнопка" на экране монитора, компьютер начинает выполнять программу.

Графический планшет — устройство для ввода контурных изображений. На рабочую поверхность кладут лист бумаги и на ней рисуют изображение. Планшет имеет большое число микропереключателей, срабатывающих под давлением карандаша. Изображение оцифровывается, записывается в память и может быть выведено на экран.

Сканер — это устройство ввода в ЭВМ графической информации непосредственно с бумажного документа. Содержит линейку ФПЗС-датчиков, которая перемещается вдоль документа. Черно-белые сканеры могут считывать штриховые и полутоновые изображения. Цветные сканеры работают и с черно-белыми, и с цветными оригиналами. В цветных сканерах используется цветовая модель RGB (красный-зеленый-синий): сканируемое изображение освещается от последовательно зажигаемых трехцвет-

ных ламп; сигнал, соответствующий каждому основному цвету, обрабатывается отдельно. Число передаваемых цветов лежит в интервале от 256 до 65536 (стандарт High Color) и даже до 16,7 млн. (стандарт True Color). Разрешение сканеров — от 75 до 1600 dpi (точек на дюйм).

Файл, создаваемый сканером в памяти машины, называется *битовой картой*. Существуют два формата представления графической информации в файлах компьютера: *растровый* формат и *векторный*. В растровом формате графическое изображение запоминается в файле в виде мозаичного набора множества точек (нулей и единиц), соответствующих пикселям отображения этого изображения на экране дисплея. В векторном формате задаются координаты или радиус-вектора точек и их цвета. При необходимости координаты X, Y умножаются на коэффициент и рисунок меняет размеры. Текстовые файлы кроме самого текста содержат коды шрифтов, специальных символов, абзацев и т.п.

Для превращения графического файла, получившегося в результате сканирования страницы, в текстовый файл используется специальная программа, распознающая буквы. Она сопоставляет битовые (мозаичные) контуры символов (букв и цифр) и кодирует их ASCII-кодами, переводя в удобный для текстовых редакторов векторный формат.

Цифровая фотокамера содержит ПЗС-матрицу (ПЗС — прибор с зарядовой связью), состоящую из большого количества фотоэлементов (300–900 тыс.), на которую с помощью объектива фокусируют изображение. Цифровая фотокамера имеет ЗУ для хранения файлов-фотографий и жидко-кристаллический дисплей, который является видеоискателем и позволяет просматривать содержимое памяти. *Цифровая видеокамера (видеокодак)* получает последовательность фотографий с частотой 25–30 кадров/с и записывает их в видеофайл. Параллельно идет запись звука.

4. Устройства вывода: мониторы, проекторы. К устройствам вывода информации относятся монитор, проектор, принтер, плоттер, сектор Брайля клавиатуры для слепых, акустические системы, устройство выдачи запахов и вкуса, устройство передачи тактильных импульсов. Видеосистема состоит из монитора и видеоконтроллера (видеоадаптера). Видеоконтроллер устанавливается на системную плату и содержит видеопроцессор (графический ускоритель и 3D-ускоритель), видеопамять и интерфейс (устройства сопряжения с монитором).

Мониторы на ЭЛТ содержат электронно-лучевую трубку, генераторы строчной и кадровой разверток, блок питания. Размер экрана характеризуется длиной его диагонали в дюймах: от 10 до 21 дюйма (обычно 15–17 дюймов). Частота кадровой развертки — 70–80 Гц; частота строчной развертки — 40–50 кГц. *Разрешение* монитора: 320 × 200, 640 × 480, 800 × 600, 1024 × 768 пикселей. Качество изображения также зависит от размера зерен люминофора, которые образуют ряд: 0,42 мм, 0,39 мм, 0,31 мм, 0,28 мм, 0,26 мм. Различают монохромные и цветные мониторы.

Плазменные мониторы состоят из трех прозрачных пластин, наложенных друг на друга. На первую и третью пластины нанесены вертикальные и горизонтальные прозрачные проводящие полосы — шины (2–4 полосы

на 1 мм). Во второй пластине, расположенной между ними сделаны отверстия, заполненные инертным газом. Вертикальные и горизонтальные проводники образуют координатную сетку, при подаче на них напряжения элементы изображения (пиксели) начинают светиться. Разрешение монитора составляет 512 x 512 или 1024 x 1024 пиксель.

Электролюминесцентные мониторы имеют координатную сетку и пластину покрытую люминофором. При подаче напряжения на вертикальные и горизонтальные координатные шины наблюдается свечение люминофора под воздействием электрического поля.

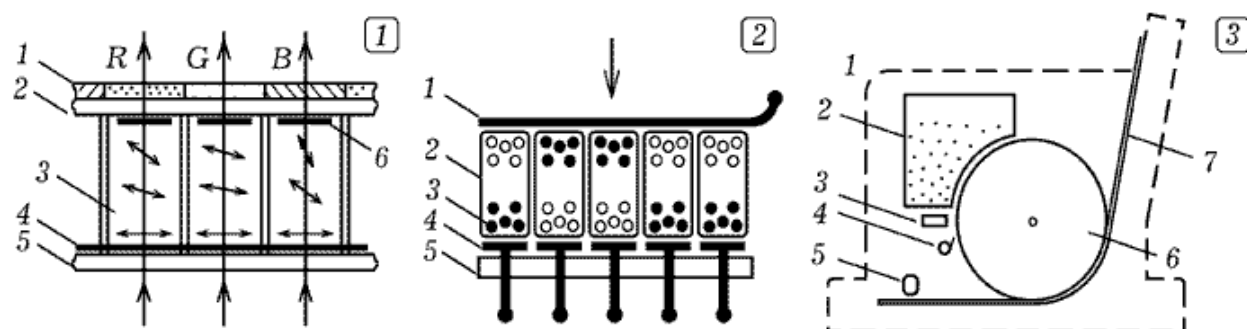


Рис. 3.12.

Жидкокристаллические мониторы состоят из элементов на жидких кристаллах, которые изменяют свои оптические свойства при подаче напряжения. Каждый такой элемент состоит из пленочного светофильтра 1, прозрачных пластин 2 и 5, системы полупрозрачных металлических электродов 4 и 6, и собственно жидкого кристалла 3 (рис. 13.12.1). Жидкие кристаллы — особое состояние органических веществ, в котором они обладают текучестью, но сохраняют упорядоченность молекул и оптическую анизотропию, что характерно для кристаллов. Когда на соответствующую пару электродов подают напряжение, молекулы жидкого кристалла располагаются более упорядоченно и происходит поворот плоскости поляризации проходящего пучка света. Это приводит к изменению интенсивности светового пучка, выходящего из ЖК-ячейки. Каждый пиксель экрана ЖК-монитора состоит из трех ячеек с красным, зеленым и синим светофильтрами. ЖК мониторы пассивные, работают либо в проходящем, либо в отраженном свете. Имеют следующие преимущества: небольшие габариты, изображение плоское и не мерцает, излучение отсутствует, потребляемая мощность мала.

Гибкие дисплеи (электронная бумага) построены на технологии электронных чернил. Они состоят из большого числа микрокапсул 2, заполненных прозрачной жидкостью, в которой взвешены белые и черные пигментные частицы 3, заряженные соответственно положительно и отрицательно (рис. 3.12.2). Микрокапсулы образуют матрицу, находящуюся между двумя рядами взаимно перпендикулярных электродов 1 и 4, составляющих адресную сетку. Сверху электроды прозрачны, снизу — непрозрачны и расположены на гибкой основе 5. Если на два электрода подать напряжение, то в точке их пересечения возникает электрическое поле, частицы с одним зарядом поднимутся в верхнюю часть микрокапсулы, а

частицы с другим зарядом — опустятся в нижнюю. На дисплее появится белая или черная точка. Используются для чтения электронных книг. Также разрабатываются светоизлучающие полимерные дисплеи.

Жидкокристаллический проектор содержит три матрицы, состоящих из жидкокристаллических элементов: красную, зеленую и голубую (RGB), расположенные друг над другом и подключенные к специальному устройству, связанному с ЭВМ. Под матрицами находится мощный источник света с коллиматором, — системой линз, обеспечивающей равномерную освещенность. Над матрицами расположен объектив. В зависимости от поступающего из ЭВМ сигнала изменяется прозрачность тех или иных жидкокристаллических элементов матриц. В результате формируется цветная картина, проецируемая объективом на экран.

LCD-проектор (Liquid-Crystal Display — технология) содержит мощную лампу, систему зеркал 1 — 5, жидко-кристаллические панели 6, 7, 8, призматическую систему 9, объектив 10, проецирующий изображение на экран (рис. 3.11.3). Зеркала 2 и 3 дихроические, они позволяют расщепить пучок белого света на красную, зеленую и синюю составляющие. Каждая составляющая проходит через соответствующую ЖК-панель, формирующую изображение данного цветового слоя. В призматической системе происходит наложение трех цветовых слоев.

DLP-проектор (Digital Light Processing — технология) содержит DMD-матрицу 3 (Digital Micro Device) — миниатюрный чип, на поверхности которого расположен массив микроскопических зеркал на микроприводах. Она освещается лампой 5 через линзу 1 (рис. 3.11.4). С помощью микроприводов можно поворачивать микроскопические зеркала в одно из двух устойчивых состояний, изменяя направление отраженного пучка и посылая его в объектив 4. Если часто переключать зеркала из одного состояния в другое (дизеринг), то можно сформировать полутона. Так создается монохромное полутонное изображение.

Чтобы получить цветное изображение, используют быстровращающийся светофильтр 2, который выполнен в виде диска, с тремя (красный, зеленый, синий) или четырьмя (красный, зеленый, синий, белый) секторами. Вращение светофильтра должно быть синхронизировано с работой матрицы, которая последовательно выводит изображение соответствующих цветовых слоев. Объектив 4 проецирует изображение на экран 5.

5. Устройства вывода: принтеры. *Принтеры* — это устройства вывода данных из ЭВМ, преобразующие ASCII-коды в соответствующие им графические изображения букв, цифр и печатающие их на бумаге. Принтеры различаются по следующим признакам: 1) цветность: черно-белые и цветные; 2) способ формирования символов: знакопечатающие и знакосинтезирующие; 3) принцип действия: матричные, термические, струйные, лазерные; 4) способы печати: ударные, безударные; 5) способы формирования строк: последовательные, параллельные; 6) ширина каретки; 7) длина печатной строки (80 и 132–136 символов); 8) набор символов; 9) скорость печати; 10) разрешение в точках на дюйм или миллиметр.

Печать у принтеров может быть *посимвольная, построчная, постра-*

ничная. Скорость печати варьируется от 10–300 зн/с (ударные принтеры) до 500–1000 зн/с и даже до нескольких десятков (до 20) страниц в минуту (безударные лазерные принтеры); разрешение — от 3–5 точек/мм до 30–40 точек/мм (лазерные принтеры).

В *игольчатых (ударных) матричных* принтерах печать точек осуществляется тонкими иглами, ударяющими бумагу через красящую ленту. Каждая игла управляется собственным электромагнитом. Печатающий узел перемещается в горизонтальном направлении, и знаки в строке печатаются последовательно. Многие принтеры выполняют печать как при прямом, так и при обратном ходе. Количество игловок в печатающей головке определяет качество печати. Недорогие принтеры имеют 9 игл. Матрица символов в таких принтерах имеет размерность 7х9 или 9х9 точек. Более совершенные матричные принтеры имеют 18 игл и даже 24.

Термопринтеры оснащены печатающей головкой с термоматрицей и требуют использования при печати специальной термобумаги или термокопирки, что является недостатком.

Струйные принтеры в своей печатающей головке содержат тонкие трубочки – сопла (от 12 до 64), через которые на бумагу выбрасываются мельчайшие капельки красителя. Современные струйные принтеры имеют разрешение до 20 точек/мм и скорость печати до 500 зн/с. Имеются цветные струйные принтеры.

В *лазерных принтерах* применяется электрографический способ формирования изображений, используемый в ксероксах. Принтер состоит из корпуса 1, контейнера с тонером 2, линейки светодиодов (полупроводниковых лазеров) 3, электрода 4 для заряда барабана, нагревательного элемента 5, светочувствительного барабана 6, бумаги 7 (рис. 13.12.3). Светочувствительный барабан из селена заряжается от электрода 4 отрицательным зарядом и освещается линейкой светодиодов, яркость которых управляется микропроцессором. В некоторых случаях используется один лазер, посылающий световой пучок на колеблющееся зеркало так, что отраженный пучок движется вдоль образующей барабана. Интенсивность пучка меняется в соответствии с сигналом с компьютера. В результате внешнего фотоэффекта (выбивания электронов с поверхности проводника под действием света) на поверхности барабана формируется электростатическое изображение. На барабан наносится краситель (тонер), налипающий на заряженные участки, и выполняется печать – перенос тонера с барабана на бумагу. Закрепление изображения на бумаге осуществляется путем разогрева тонера до плавления.

Лазерные принтеры обеспечивают качественную печать с разрешением до 50 точек/мм (1200 dpi) и скоростью до 1000 зн./с. Широко используются цветные лазерные принтеры.

6. Звуковая и сетевая платы, модем. Первые ПК были оснащены встроенным динамиком, который мог выдавать примитивные звуки. Современный компьютер имеет звуковую плату (Sound Card), — устройство, связывающее системную плату с микрофоном, динамиком и джойстиком. Она используется для звукового сопровождения мультимедийных про-

грамм и компьютерных игр. Звуковая плата (адаптер) состоит из блока цифровой записи воспроизведения и обработки звука и частотного синтезатора звука. Ее основными составляющими являются аналого-цифровой преобразователь (АЦП), цифро-аналоговый преобразователь (ЦАП), усилитель, интерфейс для микрофона, колонок и джойстика.

Аналого-цифровой преобразователь (АЦП) — схема, преобразующая аналоговый (непрерывный) сигнал в цифровой. Аналоговый сигнал, поступающий с микрофона на вход АЦП нормируется по амплитуде, квантуется по уровню и кодируется. На выходе получается дискретный сигнал. Чем выше частота дискретизации, тем точнее записывается, а затем и воспроизводится звуковой сигнал. *Разрешающая способность* АЦП определяется наименьшим изменением аналогового сигнала, приводящим к увеличению или уменьшению цифрового кода на единицу. Восемьразрядный АЦП квантует сигнал по величине на 256 уровней, 16-разрядный — на 65536 уровней. Преимущество цифровой записи сигнала в том, что сигнал записывается в виде последовательности двоичных чисел, сохранение и копирование которых производится без потери качества.

Цифро-аналоговый преобразователь (ЦАП) — устройство, преобразующее цифровой сигнал в аналоговый. В звуковой карте ЦАП используется для воспроизведения оцифрованного звука. Чтобы сгладить ступеньки напряжения на выходе ЦАП применяют специальные фильтры.

Модем (модулятор-демодулятор) — устройство для передачи информации по телефонной линии. Модулятор преобразует посылаемый от ЭВМ двоичный сигнал в аналоговый с частотной или фазовой модуляцией. Демодулятор осуществляет обратное преобразование поступающего сигнала, извлекая из него двоичную информацию и передавая ее в принимающую ЭВМ. *Факс-модем* передает и принимает факсимильные изображения. Он сканирует и оцифровывает изображение, сжимает данные и через модем передает их в телефонную линию. На приемной стороне осуществляются обратные преобразования. Голосовой модем оцифровывает звуковой сигнал и передает его по линии связи.

Сетевой адаптер — специальная плата, устанавливаемая в шину расширения на системной плате и используемая для подключения ЭВМ к вычислительной сети. Функции сетевого адаптера: синхронизация, кодирование и декодирование сигналов, расчет контрольной суммы для проверки правильности передачи данных.

7. Передача данных по вычислительной сети. В компьютерных системах используются два способа связи: параллельный и последовательный. *Параллельный способ* передачи данных предполагает одновременную передачу всех битов m машинного слова и требует использования *шины*. Шина представляет собой линию связи, состоящую из проводников, количество которых равно числу битов m (разрядность шины). Между блоками компьютера используются 16 и 32 разрядные шины. *Пропускная способность* шины в бит/с равна $C = fm/N$, где f — тактовая частота, m — разрядность шины, N — число тактов, в течение которых осуществляется передача машинного слова. При $f = 500$ МГц $N = 2$ и

$m = 32$ скорость передачи $C = 32 \cdot 500 \cdot 10^6 / 2 = 8 \cdot 10^9$ бит/с.

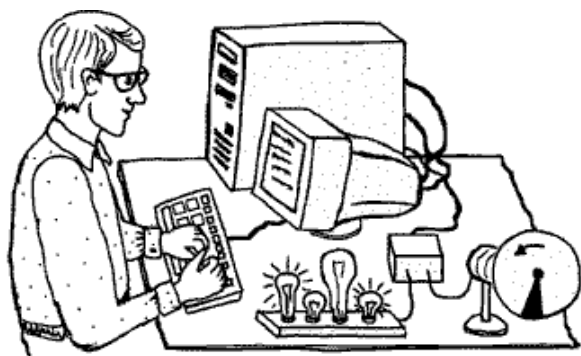
Синхронная передача параллельным кодом: каждый бит передается по отдельному проводу, одновременно передаются синхроимпульсы. Этот способ используется для обмена информацией между блоками ЭВМ и для передачи данных на небольшие расстояния. Он имеет плохую помехозащищенность. *Последовательный способ* медленнее, но экономически более выгоден при передаче на большие расстояния. В случае *синхронной* передачи одновременно с передаваемым битом посылается синхроимпульс, который управляет приемом информации. Линия связи содержит три провода: для данных, для синхроимпульсов и общий. Для передачи информации *асинхронным* способом не требуется синхронизация источника и приемника, линия связи содержит два провода. Перед передачей информационных битов передатчик генерирует *стартовый бит*, имеющий заданную длительность. После всех информационных битов посылается *контрольный бит четности*, затем следует *стоповый бит*. Эта последовательность сигналов называется *кадром*. Если кадр содержит четное число единиц, то бит четности 0, иначе — 1. При наличии ошибки приемник, сравнивая число единиц в кадре с битом четности, требует повторной передачи.

Возможны три режима передачи данных: *симплексный*, (только в одном направлении), *полудуплексный* (попеременно то в одном, то в другом направлении), и *дуплексный* (одновременно в обоих направлениях).

Для организации вычислительных сетей часто используется сетевая архитектура Ethernet. ЛВС имеет шинную топологию, пропускная способность составляет 10 Мбит/с. Передаваемые данные разделены на *кадры* — пакеты длиной 64–1518 байт. Каждый кадр кроме полезных данных несет управляющую информацию: код начала кадра, адреса источника и приемника, тип протокола, поле для проверки ошибок.

Развиваются беспроводные способы передачи данных. Беспроводная сетевая плата содержит передатчик с антенной и вставляется в разъем на системной плате ПЭВМ. Вблизи с компьютерами, образующими локальную сеть должна быть точка доступа — специальное устройство с антеннами, через которое осуществляется передача данных от одной ПЭВМ к другой со скоростью до 10 Мбит/с.

Другое направление — передача информации по проводам электрической сети (PowerLine Communications — PLC-технология). Сигналы с ПЭВМ с помощью модулятора преобразуются в модулированный сигнал с частотой несущей в несколько МГц и посылаются в электроосветительную сеть. Принимающая ПЭВМ осуществляет демодуляцию. Рассмотренные выше технологии позволяют создать *цифровой (умный) дом*, в котором все электроприборы связаны в одну сеть и управляются центральным компьютером или по сети Интернет.



ГЛАВА 4 ДЕЛАЕМ ЭКСПЕРИМЕНТЫ С КОМПЬЮТЕРОМ

Проблема ввода и вывода сигналов из ЭВМ возникла в самом начале развития вычислительной техники: работа программиста требовала ввода-вывода информации, для чего использовались различные внешние устройства: клавиатура, мышь, монитор и т.д. С технической точки зрения проблема информационного обмена с внешними устройствами решается сравнительно просто: в ряде случаев для подключения к ПЭВМ датчика, индикатора или микродвигателя достаточно двух транзисторов или микросхемы. Так что если Вы окончательно решили связать свою жизнь с компьютерами, то пора браться за паяльник! Рассказывают, что два студента Стив Возняк и Стив Джобс тоже начинали с этого, а потом создали компьютер Apple...

4.1. ВВОД ЦИФРОВОГО СИГНАЛА В ПЭВМ

1. Сопряжение ПЭВМ с внешними устройствами. Персональный компьютер — многофункциональный прибор, который может быть использован в экспериментах в качестве генератора сигналов и измерителя различных физических величин. Для этого к нему через последовательный СОМ-порт или параллельный LPT-порт, подключают различные устройства сопряжения.

Порт LPT состоит из 25 выводов и применяется для подключения принтера. При этом используется стандарт Centronix. Назначение каждого вывода: 1 вывод — стробирующий, то есть передающий сигнал, синхронизирующий передачу информации, 2–9 выводы — 8 разрядная шина, используемая для передачи 1 байта от ПК к внешнему устройству, 10–17 — шина передачи данных о состоянии принтера (подтверждение приема данных, сигнал "занят", конец бумаги, перевод строки, ошибка, готовность к работе и т.д.). Выводы 18–25 образуют общую шину заземления (корпус ПЭВМ). Считывание 1 байта информации от внешнего устройства производится через 10–17 выводы LPT-порта, при этом передаваемое восьмиразрядное двоичное число записывается в ячейку ОЗУ с адресом $379_{16} = 889_{10}$. Чтобы переслать 1 байт информации от ПК к внешнему устройству необходимо записать требуемое восьмиразрядное

двоичное число в ячейку памяти ОЗУ с адресом $378_{16} = 888_{10}$. В этом случае на выводах 2–9 появится соответствующая комбинация лог. 0 и 1.

Для написания программ, представленных ниже, используются среды Borland Pascal 7.0 и MS-DOS QBasic 1.0. В языке QBasic считывание числа из ячейки ОЗУ $379_{16} = 889_{10}$ осуществляется с помощью команды $x = INP(&H379)$ или $x = INP(889)$. Для записи числа в порт используется оператор $OUT &H378, 123$ или $OUT 888, 63$. В языке Pascal им соответствуют операторы: $Port[888] := 212$; $Port[\$378] := 12$; — запись числа в ячейку ОЗУ с адресом 888_{10} ; $x := Port[889]$; или $x := Port[\$379]$; — считывание числа из ячейки 889_{10} и его присвоение переменной x . Более подробную информацию любознательный читатель может получить в книге Г. Г. Матаева "Компьютерная лаборатория в школе и вузе" или в статьях Р. В. Акатова.

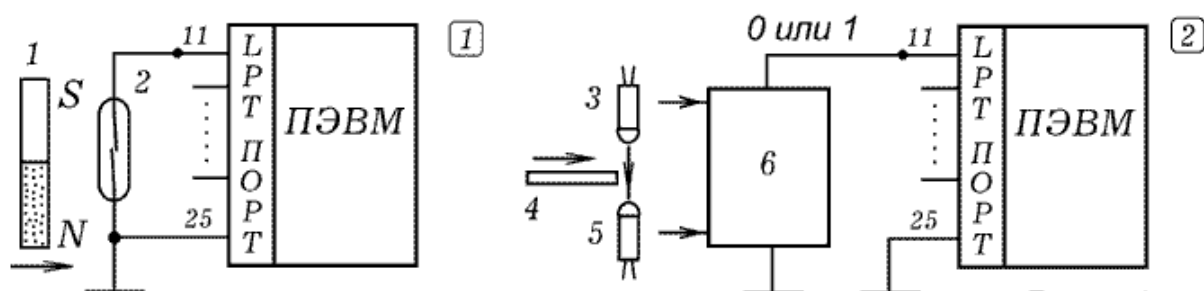


Рис. 4.1.

2. Изучение магнито-контактного датчика. Геркон (герметичный контакт) — стеклянный баллон с двумя ферромагнитными контактами, которые замыкаются при поднесении магнита. Подключим геркон 2 к параллельному порту LPT (выводы 11 и 25) и поднесем магнит 1 (рис. 4.1.1). Вдали от магнита геркон незамкнут, на всех выводах порта логическая 1, поэтому в ячейке памяти с адресом H379 находится число 11111111 (255). При поднесении магнита соответствующий вывод LPT порта замыкается на общий, на нем появляется логический 0. В ячейку памяти H379 помещается число 01111111 (127), которое может быть считано командой $x = INP(&H379)$.

```
CLS : x = INP(&H379)                                     'Пр-1
WHILE INKEY$ = ""                                       'QBASIC
WHILE x = 255: x = INP(&H379): PRINT "1"; : WEND 'Цикл 1.1
WHILE x = 127: x = INP(&H379): PRINT "0"; : WEND 'Цикл 1.2
WEND : END
```

1. Тестирование геркона. Наберите программу 1 (Пр-1), к LPT-порту подключите геркон. Запустите программу и поднесите к геркону магнит. Пронаблюдайте, как на экране компьютера 0 сменяется на 1 и наоборот. Пока геркон замкнут, x равно 255, программа возвращается в цикл 1.1, печатая на экране "1". Когда геркон разомкнут, x равно 127, программа возвращается в цикл 1.2, печатая на экране "0". При нажатии на пробел компьютер выходит из программы.

```
CLS : x = INP(&H379)                                     'QBASIC                                     'Пр-2
WHILE x = 127: x = INP(&H379): PRINT "ОЖИДАНИЕ "       'Цикл 1
```

```
WEND : T0 = TIMER
WHILE x = 255: x = INP(&H379): PRINT "ГЕРКОН ЗАМКНУТ" 'Цикл 2
WEND: T = TIMER: PRINT "ВРЕМЯ ", T - T0
END
```

2. *Измерение времени замыкания геркона.* Наберите программу Пр-2 и запустите ее. Сначала ПЭВМ находится в режиме ожидания (цикл 1), на экране – сообщение "ОЖИДАНИЕ". При поднесении магнита ПЭВМ выходит из цикла 1, переменной $T0$ присваивается текущее время, программа начинает вращаться в цикле 2, печатая на экране сообщение "ГЕРКОН ЗАМКНУТ". Удалите магнит, – переменной T присваивается время по системным часам компьютера, соответствующее моменту размыкания геркона. После этого вычисляется время замыкания $T - T0$, результат выводится на экран.

Чтобы убедиться в правильности работы программы, измерьте время замыкания геркона с помощью секундомера и компьютера, сравните результаты. Напишите аналогичную программу на языке Pascal, которая рисует на экране монитора несколько семисегментных индикаторов большого размера и выводит результат измерения времени.

3. *Измерение времени между последовательными замыканиями.* Самостоятельно напишите программу и запустите ее. К геркону поднесите магнит, удалите его и через время 5–20 с, отсчитанное по секундомеру, снова поднесите. На экран компьютера должно выводиться время между последовательными замыканиями.

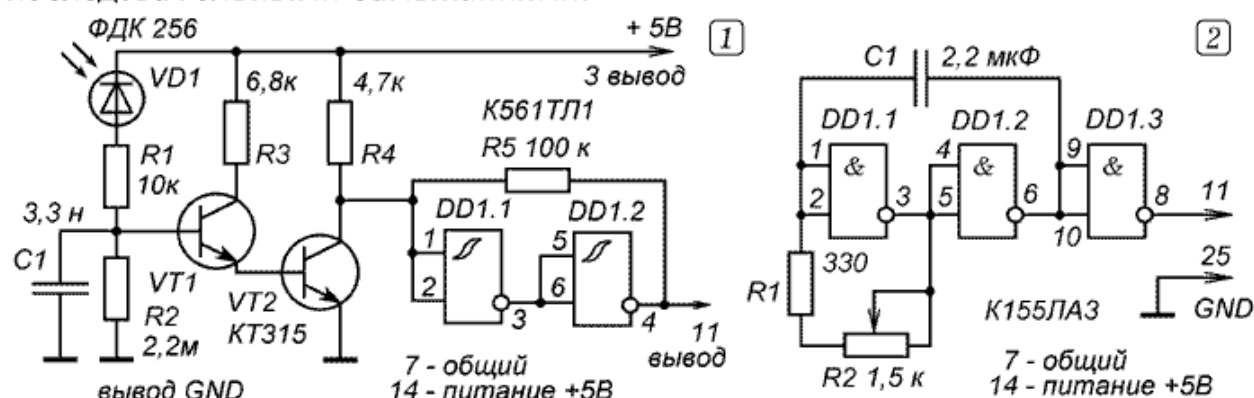


Рис. 4.2.

3. **Подключение и программирование оптодатчика.** Для измерения скорости тела к нему прикрепляют флажок длиной Δl и измеряют время пересечения Δt светового пучка, идущего от светодиода (лампочки) к фотодиоду оптодатчика. Скорость будет равна $v = \Delta l / \Delta t$.

Оптодатчик состоит из инфракрасного светодиода 3, напротив которого установлен фотодиод 5, подключенный к ПК через схему сопряжения 6 (рис. 4.1.2). При пересечении оптодатчика флажком 4 на выходе схемы резко меняется напряжение: логическая 1 сменяется логическим 0 или наоборот. Зная время пересечения светового пучка или время между двумя последовательными пересечениями, можно определить скорость тела. Схема оптодатчика изображена на рис. 4.2.1. Его питание осуществляется от 3 и 25 выводов порта LPT. Для этого по адресу H378 записывают

число 255, что приводит к появлению лог. 1 на 2–9 выводах.

1. *Тестирование оптодатчика.* Подключите оптодатчик, наберите программу Пр-3 и запустите ее. Программа содержит цикл 1, в котором находятся вложенные циклы 1.1 и 1.2. Если в порт LPT поступает число $11111111_2 = 225$, то программа возвращается в цикл 1.1 и на экране печатается "1". Если в порт LPT поступает число $01111111_2 = 127$, то программа возвращается в цикл 1.2, на экран выводится "0". Освещая и затемняя фотодиод, убедитесь в том, что программа работает описанным выше образом.

```
OUT &H378, 255                                'Пр-3
WHILE INKEY$ = "" : x = INP(&H379)             'цикл 1      'QBASIC
WHILE x = 255: x = INP(&H379): PRINT "1";       'цикл 1.1
WEND : t0 = TIMER
WHILE x = 127: x = INP(&H379): PRINT "0";      'цикл 1.2
WEND: t = TIMER: PRINT t - t0
WEND: END
```

2. *Измерение времени пересечения светового пучка.* Загрузите программу Пр-3, закомментируйте операторы PRINT "0" и PRINT "1". Когда при перекрывании светового пучка программа выйдет из цикла 1.1, переменной $T0$ будет присвоено текущее время ПЭВМ. Пока фотодиод затемнен, программа возвращается в цикл 1.2. При освещении фотодиода программа выходит из цикла 1.2 и теперь переменной T присваивается текущее время ПЭВМ. Разность $T - T0$ выводится на экран ПЭВМ, после чего все повторяется снова (цикл 1) до нажатия на пробел. Измерьте время затемнения фотодиода с помощью секундомера и сравните его с результатом, выдаваемым ПЭВМ.

3. *Измерение времени между двумя пересечениями светового пучка.* Самостоятельно напишите программу, которая ждет первого затемнения оптодатчика, после чего присваивает переменной $T0$ текущее время по системным часам ПЭВМ, затем ждет второго пересечения оптодатчика и соответствующее время присваивает переменной T . Разность $T - T0$ должна выводиться на экран. Запустите программу, измерьте время между последовательными пересечениями оптодатчика с помощью секундомера и ПЭВМ, сравните результаты.

4. **Использование ПЭВМ в опытах по механике.** Персональный компьютер, соединенный с одним из рассмотренных выше датчиков, при наличии соответствующего программного обеспечения становится универсальным прибором, позволяющим измерять длительность исследуемого процесса, подсчитывать количество импульсов за заданный промежуток времени, вычислять скорость движения и координату, строить графики. Рассмотрим некоторые примеры использования ПЭВМ в учебном эксперименте по механике.

1. *Изучение скатывания шара с помощью ПЭВМ.* Экспериментальная установка для изучения скатывания шара по наклонной плоскости состоит из компьютера 1, шарика 2, двух металлических наклонных направляющих 3 длиной 1,2 м (рис. 4.3.1). Направляющие подключены к LPT-порту

ПЭВМ (11 и 25 выводы). На одной направляющей лежит полоска бумаги 3, на которую ставят шарик перед запуском. При качении направляющие замыкаются, и компьютер определяет время скатывания. Чтобы исключить дребезг контактов, используют конденсатор C емкостью 2 мкФ и резистор R сопротивлением 430 ом; кроме того в программе за одно прохождение цикла производится не один, а несколько опросов LPT-порта.

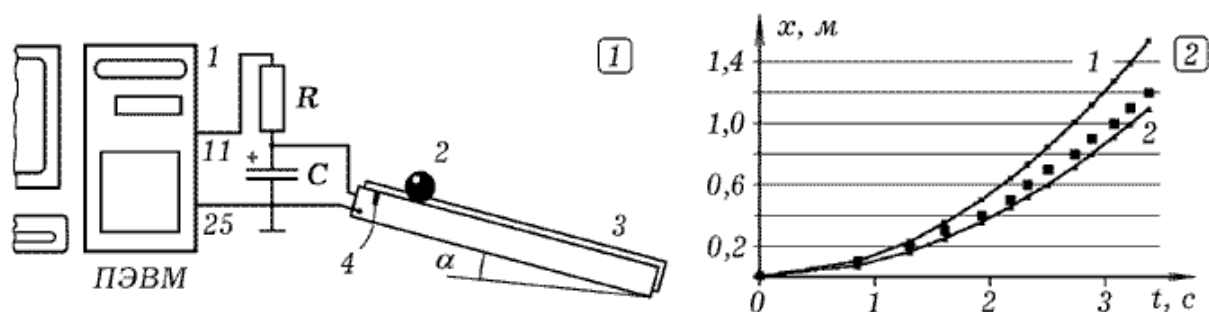


Рис. 4.3.

Нами использовался компьютер Intel Celeron, 366 МГц, с операционной системой Windows98SE. Программа Пр-4 для измерения времени замыкания направляющих написана на языке Pascal, ее текст приведен ниже.

```
uses crt; var time,x1,x2,x3,x4,x5,x6,x7,x8,t : integer;      'Пр-4
BEGIN clrscr; writeln('Ожидание'); time:=10;              'PASCAL
  Repeat until (KeyPressed)or(port[889]=255);
  Repeat x1:=port[889]; delay(time); x2:=port[889]; delay(time);
        x3:=port[889]; delay(time); x4:=port[889]; delay(time);
        x5:=port[889]; delay(time); x6:=port[889]; delay(time);
        x7:=port[889]; delay(time); x8:=port[889];
  if (x1<>127)or(x2<>127)or(x3<>127)or(x4<>127)or(x5<>127)or
  (x6<>127)or(x7<>127)or(x8<>127) then begin t:=t+1;
  writeln('Время замыкания t=', t); end; until KeyPressed; ReadKey;
END.
```

Запускают программу, на полоску бумаги устанавливают шар, затем его отпускают. Время замыкания направляющих выводится на экран в условных единицах времени. Чтобы перевести результат в секунды, необходимо отградуировать получившийся электронный секундомер с помощью механического секундомера.

На рис. 4.3.2 представлены две теоретические кривые, соответствующие скольжению (график 1) и скатыванию (график 2) шарика, а также проставлены экспериментальные точки, полученные в результате измерений. При угле наклона $1,6^\circ$ ускорение скатывания шарика $0,19 \text{ м/с}^2$, а ускорение скольжения $0,27 \text{ м/с}^2$. За счет вращения время скатывания шара больше времени скольжения. Рассмотренный эксперимент может быть использован на лабораторных занятиях по физике.

2. Изучение вращения сегнерова колеса с помощью геркона. В нижней части пластиковой бутылки на 2 л сделаем два диаметрально противоположных отверстия и вставим в них две изогнутые трубки так, чтобы вытекающая из них жидкость раскручивала бутылку в одну сторону. Подвесим получившееся сегнерово колесо на нити длиной $\approx 0,2 \text{ м}$. Чтобы

бутылка не раскачивалась, нить пропустим через трубку, удерживаемую вблизи нижнего конца нити в специальном держателе. К боковой поверхности бутылки приклеим два постоянных магнита, а рядом с ней расположим геркон так, чтобы при вращении бутылки происходило его срабатывание. Геркон подключают к ПЭВМ и запускают программу, периодически определяющую время замыкания геркона и скорость вращения сегнерова колеса. Результаты измерений могут выводиться на экран в текстовом и графическом виде или сохраняться в файле. Под действием реактивной силы сегнерово колесо набирает скорость, закручивая нить, останавливается и начинает вращаться в противоположную сторону, совершая затухающие крутильные колебания.

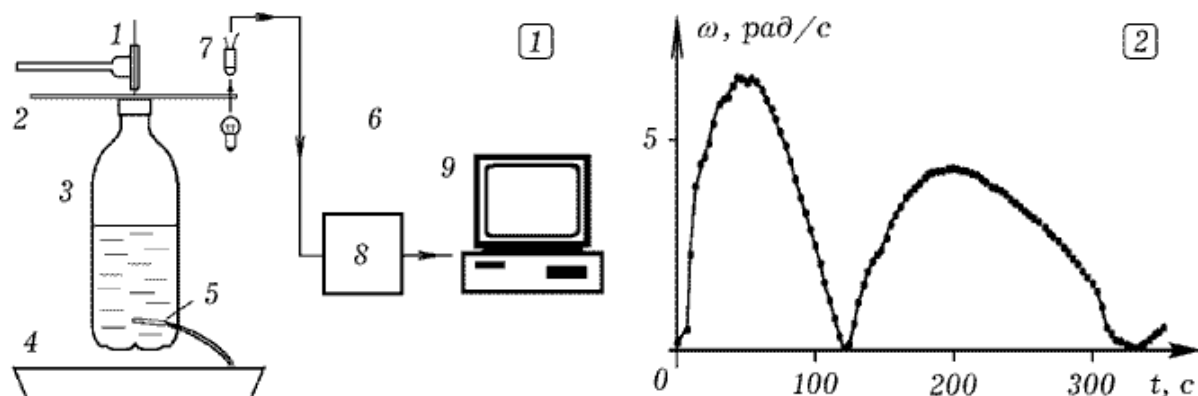


Рис. 4.4.

3. Изучение реактивного движения сегнерова колеса с помощью оптодатчика. Соберите установку, состоящую из подвешенной на нити 1 пластиковой бутылки 3 на 2 л с двумя изогнутыми трубками 5, к горлышку или крышке которой прикреплен диск 2 с 96 прорезями по краю (рис. 4.4.1). В верхней части бутылки имеется отверстие для воздуха. Вблизи края диска установите оптодатчик (лампочка 6, фотодиод 7), подключенный через схему сопряжения 8 (рис. 4.2.1) к параллельному LPT-порту ПЭВМ 9. Чтобы уменьшить раскачивание бутылки, нить пропускают через трубку диаметром 1–2 мм, которую фиксируют с помощью держателя вблизи ее нижнего конца. При вращении бутылки с диском происходит периодическое освещение и затемнение фотодиода, в результате чего в компьютер поступает последовательность логических 0 и 1, которая обрабатывается соответствующей программой написанной на языке Pascal или Qbasic. Результаты измерений скорости вращения в последовательные моменты времени выводятся на экран в цифровом или графическом виде. Запустите сегнерово колесо и получите на экране компьютера график зависимости модуля скорости от времени (рис. 4.4.2). Скорость сначала увеличивается до некоторого значения, затем бутылка останавливается и начинает вращаться в противоположном направлении. После вытекания всей жидкости система совершает затухающие колебания.

4. Ускоренное движение тела под действием вращающего момента. К неподвижному блоку прикрепите диск с прорезями, а рядом установите оптодатчик так, чтобы прорези при вращении диска пересекали свето-

вой пучок оптодатчика, периодически открывая и закрывая фотодиод. На блок намотайте нить, к концу которой привяжите тело. На ПЭВМ запустите программу, считающую количество импульсов в единицу времени с выхода оптодатчика. Отпустите груз, система придет во вращение и на экране монитора получится график зависимости угловой скорости диска от времени. Проведите серию опытов при различных массах груза.

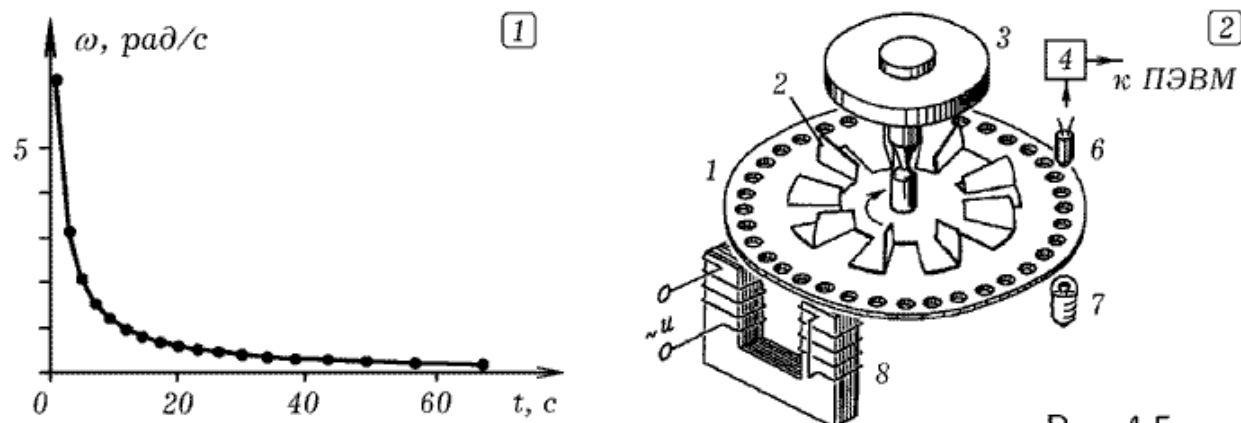


Рис. 4.5.

Для изучения торможения диска силами трения используется диск с подшипником, способный вращаться вокруг вертикальной оси, на которую кладут постоянный магнит. Рядом с диском устанавливают геркон так, чтобы при вращении магнит вызывал замыкание контактов геркона. Геркон подключают к ПЭВМ, рукой раскручивают диск и запускают программу, измеряющую время замыкания геркона (или время между последовательными замыканиями). Из-за сил трения скорость вращения ω уменьшается. Исходя из экспериментальной зависимости $\omega = \omega(t)$ (рис. 4.5.1) методами численного интегрирования и дифференцирования можно рассчитать угловое перемещение и ускорение в требуемые моменты времени, построить графики $\varphi = \varphi(t)$ и $\varepsilon = \varepsilon(t)$.

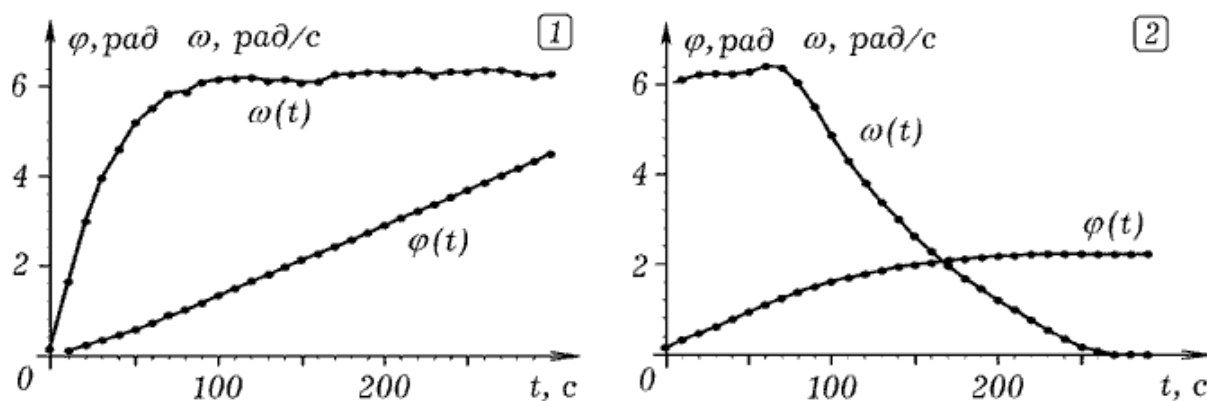


Рис. 4.6.

5. Вращение тела в вязкой среде. К центру алюминиевого диска 1 с отверстиями или прорезями по краю прикрутите стальной болт, за который подвесьте диск к постоянному магниту 3 так, чтобы он вращался вокруг вертикальной оси, оставаясь в горизонтальной плоскости (рис. 4.5.2). На диск наденьте крыльчатку 2, увеличивающую силу сопротивления возду-

ха. Под диском расположите магнитный движитель 8, состоящий из двух обмоток на U-образном сердечнике, одна из которых замкнута накоротко, а другая подключена к ЛАТРу. Вблизи края диска установите соединенный фотодиод 6 и лампочку 7 так, чтобы при вращении диска прорезы пересекали световой пучок (рис. 4.5.2).

Фотодиод 6 подключите через схему сопряжения 4 (рис. 4.2.1) к ПЭВМ и запустите программу, определяющую количество прорезей, прошедших мимо оптодатчика за 1 с. В результате получают графики $\omega = \omega(t)$, $\varphi = \varphi(t)$ при разгоне диска (рис. 4.6.1) и его торможении (рис. 4.6.2).

В качестве движителя может использоваться однофазный асинхронный двигатель АД (рис. 4.7.1). В этом случае диск 2 с 16 прорезями и крыльчаткой (на рис. она не показана) закрепляется на валу двигателя 1. Вязкой средой является масло в подшипниках и окружающий воздух. Измерение скорости вращения по прежнему осуществляется с помощью оптодатчика 3, подключенного через схему 4 к компьютеру. Получающиеся графики $\omega = \omega(t)$, $\varphi = \varphi(t)$ представлены на рис. 4.7.2.

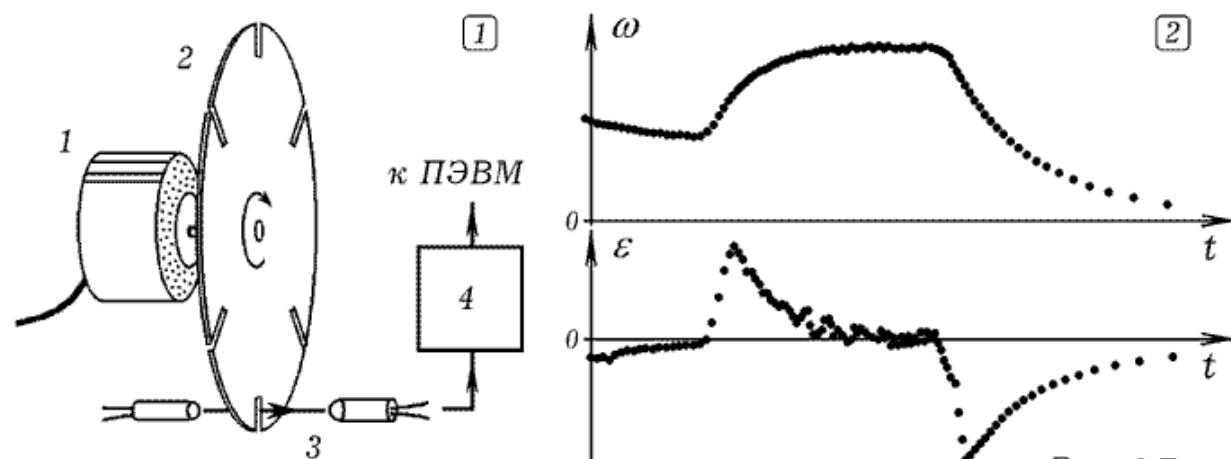


Рис. 4.7.

Установка позволяет провести различные эксперименты: изучить процесс перехода двигателя из одного установившегося режима работы в другой, исследовать зависимость скорости вращения АД от напряжения питания и т. д. Для обработки сигналов с оптодатчика и построения графика $\omega = \omega(t)$ используется программа Пр-5.

```
uses crt, Graph; var i,n,Gd,Gm : integer; F: text;
v,vv,aaa,aa,a,t,time: real;
BEGIN Gd := Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
  if GraphResult <> grOk then Halt(1); Port[888]:=255;
  n:=0; Assign(F,'c:\bp\dat.txt'); Rewrite(F);
Repeat t:=0; for i:=1 to 16 do begin
  Repeat until port[889]=127;
  Repeat t:=t+1; {delay(1)} until port[889]=255; end;
  v:=10000/t; time:=time+1/v; a:=v*v-vv*vv; n:=n+1;
  WriteLn(F, n, ' ', time, ' ', t, ' ', v, ' ', a);
  circle(10+round(time),250-round(500*v),2);
  circle(10+round(time),350-round(5000*(a+aa+aaa)),2);
  aaa:=aa; aa:=a; vv:=v;
  if time>640 then begin time:=0; cleardevice;
```

'Пр-5
'PASCAL


```
line(10,250,620,250); line(10,350,620,350); end;
until KeyPressed; Close(F); CloseGraph;
END.
```

6. *Затухающие колебания физического маятника.* Установка включает в себя физический маятник, выполненный в виде металлической пластины 1 с флажком 3 на конце, и оптодатчик, состоящий, например, из лазерной указки 4, фотодиода 5 и схемы сопряжения 6 (рис. 4.8.1). Оптодатчик подключите к LPT-порту ПЭВМ и расположите его под точкой подвеса так, чтобы, проходя положение равновесия, маятник пересекал световой пучок. На языке Pascal напишите программу, определяющую скорость прохождения маятником положения равновесия. Результаты измерений амплитуды скорости можно сохранять в файле и выводить на экран в текстовом или графическом виде (рис. 4.8.2). Графиком зависимости амплитуды скорости от времени в случае затухающих колебаний является экспонента. По экспериментальным результатам можно оценить логарифмический декремент затухания для маятника без флажка и с флажком (кривые 1 и 2, рис. 4.8.2).

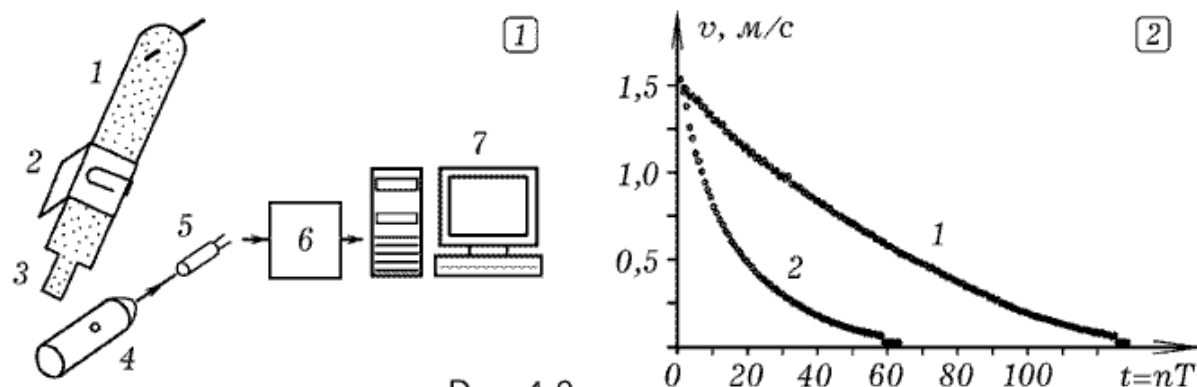


Рис. 4.8.

5. *Анализатор цифрового сигнала на ПЭВМ.* Для того, чтобы исследовать более или менее сложную цепь на логических элементах часто недостаточно одно- или двухлучевого осциллографа, так как необходимо иметь возможность сравнивать фазы и время запаздывания сразу нескольких цифровых сигналов. Это может быть сделано с помощью анализатора сигналов, представляющего собой ПЭВМ, у которой выводы 10–17 LPT-порта через резисторы сопротивлением 120 ом подключены к различным точкам исследуемой цепи. Такой цепью может быть, например, последовательный регистр, состоящий из нескольких триггеров. Программа, обрабатывающая сигналы, должна периодически опрашивать LPT-порт, выделять из получающегося двоичного слова отдельные биты и строить соответствующие осциллограммы на экране.

4.2. ВВОД АНАЛОГОВОГО СИГНАЛА В ПЭВМ

1. Генераторный датчик координаты. Так как ПЭВМ работает с цифровыми сигналами, то ввод аналоговых сигналов требует их оцифровки. Это можно осуществить с помощью внешнего аналого-цифрового преобразователя (АЦП), либо с помощью АЦП, имеющегося на звуковой плате. Для измерения медленно изменяющихся величин может быть использован аналого-цифровой преобразователь, состоящий из генератора прямоугольных импульсов 2, частота которых зависит от сопротивления резистора 1, являющегося, например, датчиком координаты (рис. 4.9.1). Сигнал с генератора подается в ПК. Чем меньше сопротивление резистора, тем быстрее происходит заряд или разряд конденсатора, соответственно выше частота импульсов. Генератор собран на микросхеме К155ЛА3 (рис. 4.2.2), и запитан от параллельного LPT-порта.

ПЭВМ в течение заданного промежутка времени dt (например, 1 с) считает количество поступивших импульсов, и определяет их частоту. Результаты подсчета импульсов могут выводиться на экран в графическом виде, в числовом формате, либо записываться в файл.

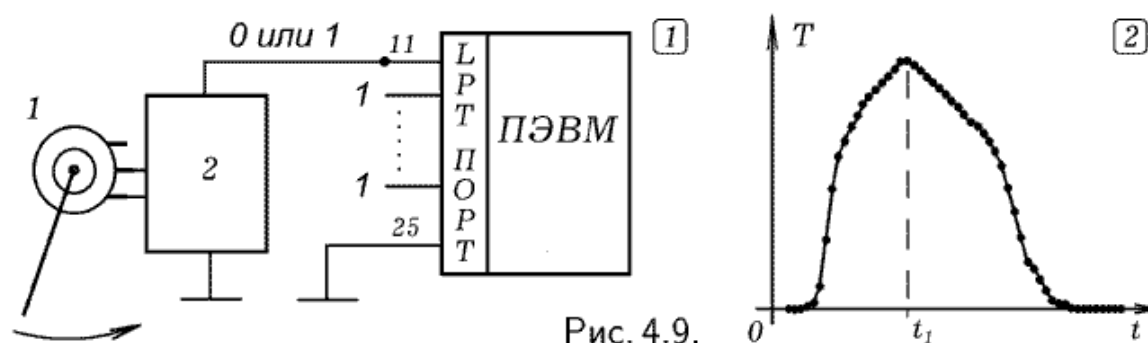


Рис. 4.9.

1. Изучение датчика координаты. К ПЭВМ подключите АЦП с датчиком координаты. Наберите программу Пр-6 и запустите ее. Поворачивая подвижный контакт резистора, наблюдайте получающуюся кривую зависимости координаты от времени. Повторите предыдущий опыт, задав другие время счета импульсов dt и масштаб M . Измените программу так, чтобы она через заданное время dt выводила координату в числовом виде.

```
SCREEN 2
LINE (10,180)-(640,180): LINE (10,0)-(10,480)
OUT (888), 255: M = .2
WHILE INKEY$ = ""
n = 0: dt = 1: t0 = TIMER: t = t0 + .01
WHILE t - t0 < dt
x = INP(889): 'PRINT " x= ", x;
IF (y = 127) AND (x = 255) THEN n = n + 1
y = x: t = TIMER
WEND: n = M * (n - 80): tt = tt + dt
LINE (10+tt*5, 180-n/dt)-(10+(tt-dt)*5, 180-nn/dt)
nn = n: 'PRINT "КООРДИНАТА", n / dt;
WEND: END
```

'Пр-6
'QBASIC

2. **Подключение терморезистора к ПЭВМ.** К ПЭВМ подключите генератор импульсов (рис. 4.2.2) с терморезистором (типа ММТ-12), включенным вместо резистора $R2$. Запустите программу и сделайте так, чтобы количество сосчитанных компьютером импульсов выводилось на экран в числовом формате. Добейтесь построения графика зависимости температуры от времени. Погрузите терморезистор в сосуд с горячей водой на 15–20 с, затем достаньте его из сосуда. Температура терморезистора сначала плавно повышается, затем медленно возвращается к исходному значению (рис. 4.9.2). Так как частота импульсов зависит от температуры терморезистора нелинейно, то получающийся график не совсем точно соответствует реальной зависимости $T = T(t)$.

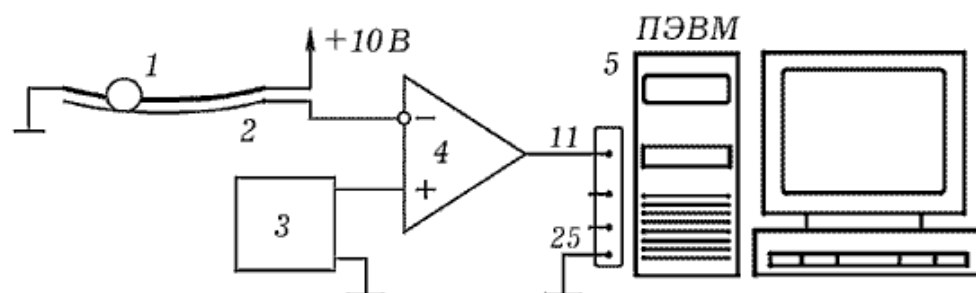


Рис. 4.10.

2. **Использование АЦП последовательного счета.** Рассмотрим функционирование АЦП и его использование в физических опытах на примере установки для изучения затухающих колебаний (рис. 4.10). Она состоит из стального шарика 1, катающегося по вогнутым направляющим 2, являющимися резистивным датчиком координаты. На одну из направляющих надета нихромовая спираль, на концы которой подано напряжение 10 В. Напряжение, снимаемое со второй металлической направляющей, оказывается пропорционально координате шарика. Оно подается на аналого-цифровой преобразователь (АЦП), состоящий из генератора пилообразного напряжения 3, компаратора 4 и персонального компьютера 5 (см. пп. 2.2, рис. 2.22.3). Частота пилообразного напряжения f_r в десятки раз превышает частоту f_c сигнала с датчика. При этом на выходе АЦП возникает последовательность прямоугольных импульсов частотой f_r , длительность которых зависит от входного напряжения. Эти импульсы подаются на 11 вывод LPT-порта ПЭВМ (рис. 4.10).

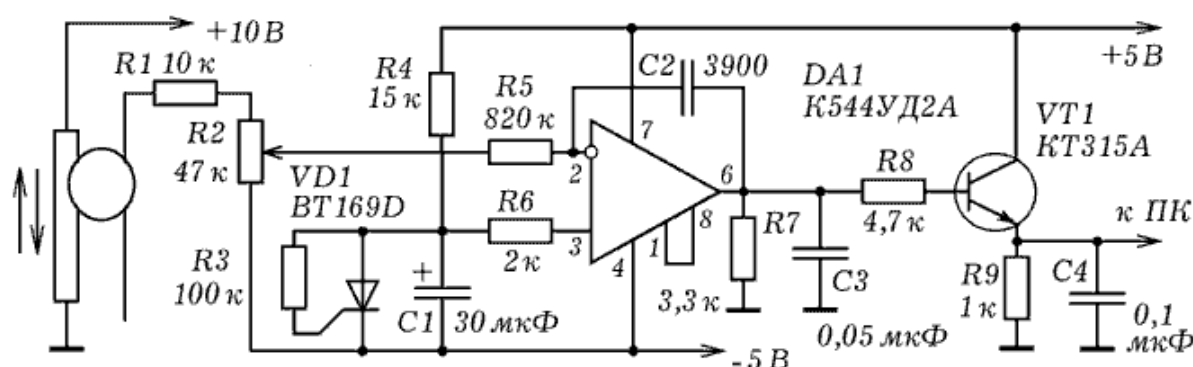


Рис. 4.11.

Принципиальная схема АЦП изображена на рис.4.11. На тиристоре VD1, резисторе R4 и конденсаторе C1 собран релаксационный генератор, вырабатывающий пилообразные импульсы с частотой 2–4 Гц. Они поступают на неинвертирующий вход 3 операционного усилителя DA1, используемого в качестве компаратора. На инвертирующий вход 2 подается плавно изменяющееся напряжение с датчика координаты. На выходе ОУ возникают прямоугольные импульсы, длительность которых пропорциональна напряжению с датчика (то есть координате шарика).

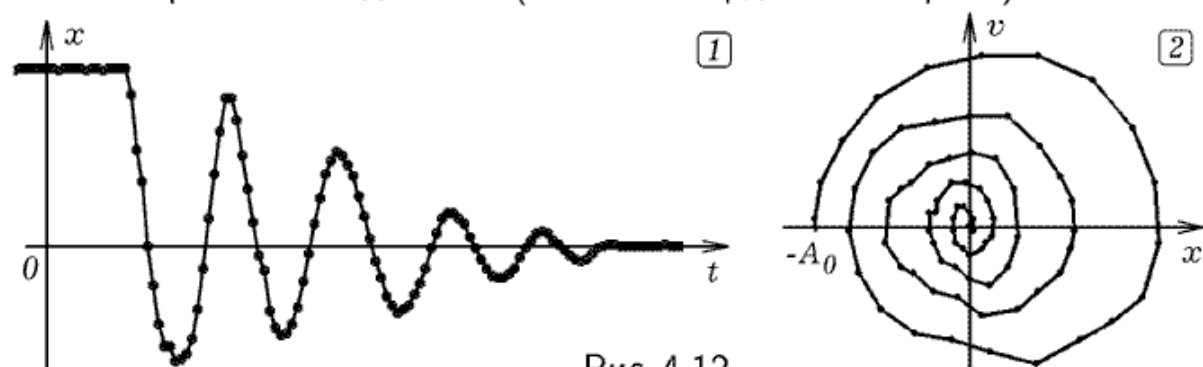


Рис. 4.12.

Для обработки сигнала с АЦП используется программа Пр-7. Она содержит цикл по времени, в который вложен цикл по переменной x . Пока на выходе компаратора логическая 1, значение x увеличивается на 1. Результаты измерения координаты x выводятся на экран в виде графика (рис.4.12.1) и, при необходимости, сохраняются в файле. Для получения фазовой кривой (рис.4.12.2) необходимо, чтобы программа вычисляла скорость шарика. Кривая оказывается больше похожей на ломанную вследствие погрешностей, неизбежно возникающих при измерениях.

```
uses crt, graph;                                     'Пр-7
var DV, MV, EC, u, uu, x, time: integer; F: text;    'PASCAL
Procedure GraphInit;
begin DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
      EC:=GraphResult; if EC<>grOK then Halt(1); end;
BEGIN Assign(F,'c:\bp\data.txt'); Rewrite(F); GraphInit;
      repeat x:=0; repeat uu:=u; u:=port[889]; delay(15);
        if u<>127 then x:=x+1;
        until (uu=127)and(u<>127); time:=time+1;
        circle(3*time,400-x*1,1); circle(3*time,400-x*1,2);
        WriteLn(F, x); if 3*time>640 then begin time:=0; cleardevice;
        end; until Keypressed; CloseGraph;Close(F);
End.
```

Рассмотренный выше АЦП позволяет провести целую серию экспериментов, требующих оцифровку и запись аналогового сигнала, изменяющегося в интервале 0–10 В, с частотой отсчетов ≈ 2 –4 Гц. Например, довольно просто получить кривые заряда и разряда конденсатора. Для этого нами использовался электролитический конденсатор емкостью 500 мкФ, подключаемый через резистор 10–100 кОм к источнику постоянного напряжения. Параллельно конденсатору включают вольтметр и АЦП, соединенный с компьютером, и запускают программу.

3. Цифровой осциллограф на базе ПЭВМ. Использование ПЭВМ

в учебном процессе в качестве осциллографа имеет ряд преимуществ: 1. Компьютеры многофункциональны, распространены и доступны, в то время как современный средний осциллограф имеет высокую стоимость и решает сравнительно узкий класс задач. 2. ПЭВМ имеет большой экран, что позволяет демонстрировать осциллограммы всей аудитории. 3. Оцифрованный сигнал можно записать в графический файл и затем его обработать. 4. Учащиеся осваивают программирование, понимают сущность оцифровки аналогового сигнала.

В цифровом осциллографе осуществляется преобразование аналогового сигнала в цифровой. Это может быть сделано, например, с помощью самодельного аналого-цифрового преобразователя на базе ПЭВМ, либо с помощью звуковой платы (см. работы Р. В. Акатова и Г. Г. Матаева).

Рассмотрим работу цифрового осциллографа, основанного на использовании последовательного АЦП (п. 2.2, рис. 2.23.1). Используется программа Пр-8. Компьютер перебирает все числа n от $0_{10} = 00000000_2$ до $255_{10} = 11111111_2$, выводя их в параллельный порт (ячейка ОЗУ с адресом 888), соединенный с 8 входами ЦАП. На выходе ЦАП возникает пилообразное напряжение $u_0(t) = n\Delta U$, где ΔU — шаг квантования по напряжению. Оно подается на вход операционного усилителя (ОУ), используемого в качестве компаратора (рис. 2.23.1). На другой вход ОУ подается исследуемое напряжение $u(t)$, изменяющееся значительно медленнее. Когда $u(t)$ оказывается равным $u_0(t)$ логический 0 на выходе ОУ сменяется логической 1. Это регистрируется компьютером (ячейка ОЗУ с адресом 889), который запоминает двоичное число x_i , соответствующее входному напряжению в момент отсчета t_i . Это число записывается в массив, после чего осуществляется следующий отсчет. Когда массив оказывается заполнен, программа стирает предыдущий график и, перебирая все элементы массива, строит новый график зависимости $u(t)$. Затем все повторяется снова.

```
uses Crt, Graph; {Цифровой осциллограф}                                'Пр-8
const M=2; Ms=2; y0=240;                                                'PASCAL
var Data : array [1..640] of integer; c,d,xx,x,i,j : integer;
procedure GrInit; var GD, GM, EC : Integer;
begin GD := Detect; InitGraph (GD, GM, '');
EC:=GraphResult; if EC <> grOk then begin Halt; end end;
Begin GrInit;
Repeat For i:=1 to 320 do begin
  For j:=0 to 255 do begin port[888]:=j; c:= port[889];
    if c=127{<>255} then Data[i]:=j-128; end;
    x:=round((Data[i]+Data[i-1]+Data[i-2])/3);
    line(M*i-1,y0-Ms*xx,M*i,y0-Ms*x); xx:=x; end; setcolor(8);
  for i:= 1 to 320 do begin
    x:=round((Data[i]+Data[i-1]+Data[i-2])/3);
    line(M*i-1,y0-Ms*xx,M*i,y0-Ms*x); xx:=x;
  end; setcolor(15); until KeyPressed; CloseGraph;
End.
```

Используемая программа Пр-8 работает не слишком быстро, так как для осуществления одного отсчета ПЭВМ должно перебрать все числа от 0 до 255. Чтобы увеличить частоту отсчетов следует использовать метод половинного деления, состоящий в последовательном делении интервала, содержащего неизвестную величину x на две равные части. Это позволит существенно повысить быстродействие программы, так что она будет воспринимать сигнал частотой до 5 кГц.

Для того, чтобы продемонстрировать работу цифрового осциллографа, подайте на вход АЦП медленно изменяющееся напряжение. Выведите результаты измерений в числовом или графическом виде. Подайте переменное напряжение от генератора звуковой частоты и получите осциллограмму. Измените амплитуду и частоту. Оцените чувствительность осциллографа и максимальную частоту входного сигнала, осциллограмма которого может быть получена на экране.

4. Опыты со звуковыми импульсами. Для изучения быстропротекающих процессов необходимо осуществлять запуск ждущей развертки. Для этого следует использовать еще один вход LPT-порта, на который подают запускающий сигнал. При появлении на нем логического 0 программа должна начать оцифровку сигнала на входе АЦП, заполнить массив, стереть предыдущую осциллограмму и построить новую, после чего перейти в режим ожидания следующего запускающего сигнала.

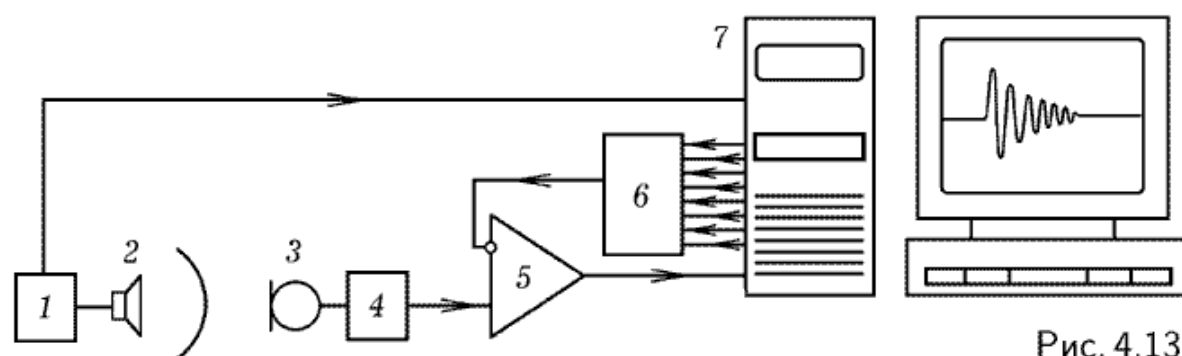


Рис. 4.13.

Пусть нам необходимо исследовать распространение звукового импульса в воздухе. Для этого будем использовать цифровой осциллограф со ждущей разверткой на базе ПЭВМ. Установка состоит из генератора импульсов 1 с динамиком 2 на выходе, микрофона 3, подключенного через усилитель 4 к аналого-цифровому преобразователю, состоящему из компаратора 5, цифро-аналогового преобразователя 6 и персонального компьютера 7 (рис. 4.13). Сигнал с генератора идет к порту LPT ПЭВМ для синхронизации развертки. Оцифровка сигнала с микрофона осуществляется с помощью программы, использующей метод половинного деления, ее текст не приводится.

При включении генератор импульсов вырабатывает электрические импульсы частотой около 1 Гц, которые преобразуются динамиком в звуковые и одновременно подаются в LPT-порт ПЭВМ. В результате компьютер начинает оцифровывать аналоговый сигнал с микрофона. Когда звуковой сигнал достигает микрофона, на его выходе возникает импульс

напряжения, который оцифровывается и сохраняется в массиве. После этого осуществляется очистка экрана и по записанным в массив значениям строится график зависимости напряжения на выходе усилителя от времени. С приходом следующего импульса все повторяется снова. На экране монитора возникает осциллограмма сигнала с микрофона, синхронизированная сигналом с генератора импульсов.

Для демонстрации распространения звуковых импульсов соберите установку, запустите ПЭВМ и генератор импульсов, получите на экране осциллограмму сигнала с микрофона. Приближая и удаляя динамик от микрофона на 0,5–1 м, измерьте соответствующее смещение осциллограммы и вычислите скорость звука. Пронаблюдайте интерференцию непосредственно падающего и отраженного импульсов. Для изучения модели эхолота и принципа эхолокации установите динамик и микрофон рядом, между ними поставьте экран, а перед ними — препятствие, от которого будут отражаться звуковые волны.

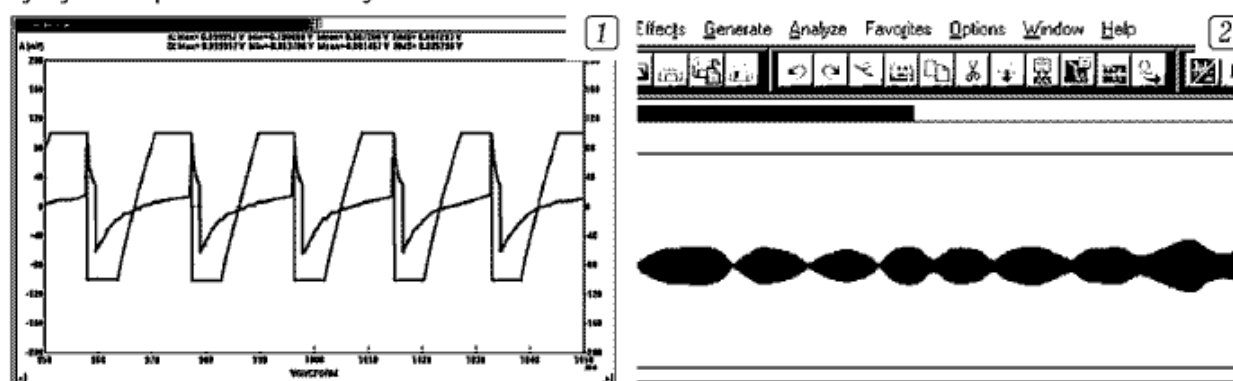


Рис. 4.14.

5. Экспериментируем со звуковой платой. Для превращения ПЭВМ в измерительный прибор или генератор сигналов применяются специальные платы расширения, стоимость которых сравнительно велика. Но даже обычная звуковая плата содержит АЦП, ЦАП и может осуществить оцифровку входного и генерацию выходного сигналов специальной формы. Для получения из ПЭВМ цифрового вольтметра, частотомера, осциллографа или спектроанализатора исследуемый сигнал подают на микрофонный вход, либо на линейный вход LINE-IN звуковой платы. Линейный вход позволяет заводить в ПЭВМ два сигнала амплитудой около 1–2 В и частотой до 20 кГц. Генерируемые компьютером сигналы снимаются с линейного выхода LINE-OUT.

Требуемое программное обеспечение (Oscilloscope2.51, Audio Tester, Marchand Function Generator или аналогичные программы) можно скачать из Интернета. На рис. 4.14.1 показаны осциллограммы двух сигналов с релаксационного генератора, полученные с помощью программы Virtins Sound Card Oscilloscope, эмулирующей двухлучевой осциллограф.

Существуют специальные аудиоредакторы, позволяющие генерировать звуковые колебания требуемой частоты, записывать сигнал с микрофона, осуществлять его спектральный анализ, строить сонограмму — зависимость амплитуды и частоты от времени. К ним относятся, напри-

мер, Cool Edit Pro и Sound Forge.

На рис. 4.14.2 показан результат сканирования звукового поля между двумя динамиками, полученный с помощью Cool Edit Pro. ПЭВМ работает в многозадачном режиме: под управлением одной программы колонки выработывают звуковой сигнал, а с помощью другой осуществляется запись сигнала с микрофона в файл. При равномерном перемещении микрофона между динамиками на экране получается последовательность минимумов и максимумов, соответствующая узлам и пучностям стоячей волны.

6. Программирование звуковой платы. Определенный интерес представляет собой самостоятельное создание программы, обрабатывающей сигналы со звуковой платы (см. книгу Матаева Г.Г. "Компьютерная лаборатория в вузе и школе").

Параметры используемой нами ПЭВМ: Intel Celeron, 416 МГц, операционная система Windows98 SE. Звуковая плата ES1868 Plug and Play AudioDrive, диапазон ввода-вывода: 0220-022F. Перед запуском программ требуется отключить драйвер звуковой платы: Мой компьютер → Панель управления → Система → Устройства → ES1868 Plug and Play AudioDrive (WDM). Выберите "Свойства", поставьте галочку в опции "Отключено в данной конфигурации". Интерфейс звуковой платы отключать не следует. После этого необходимо перезагрузить компьютер. К входу звуковой платы следует подключить микрофон, либо подать исследуемый сигнал амплитудой в несколько милливольт. Листинг программы на Basic представлен ниже (Пр-9).

```
SCREEN 11                                     'Пр-9
OUT &H226, 1: FOR t = 0 TO 100: NEXT: OUT &H226, 0      'QBASIC
FOR t = 0 TO 100: NEXT: n = 1000: DIM u(n)
100: CLS: i = 1: WHILE ABS(x(i) - 127) < 10: GOSUB 200: WEND
FOR i = 1 TO n: GOSUB 200: CIRCLE (i, (x(i) - 127) * 1 + 200), 1
LINE (i - 1, z) - (i, (x(i) - 127) + 200): z = (x(i) - 127) * 1 + 200
NEXT i: GOTO 100
END
'=====
200: OUT &H22C, &H20: x(i) = INP(&H22A): RETURN
```

Результат оцифровки входного сигнала считывается из ячейки памяти 22Ah и сохраняется в массиве $x(i)$. Когда на входе 0 В, $x(i) = 127$. После заполнения всего массива либо одновременно с этим процессом осуществляется построение графика на экране компьютера. Чтобы синхронизировать этот процесс с входным сигналом, создают цикл, вращаясь внутри которого, ПЭВМ опрашивает звуковую плату до тех пор, пока величина $x(i)$ отличается от 127 менее чем на 10 (уровень запуска "развертки"). При больших отклонениях $x(i)$ от 127 программа выходит из цикла и происходит запись данных в массив, построение осциллограммы.

Программирование звуковой платы в QBasic состоит в записи команды 20h (команда ввода 1 байта данных) в ячейку 22Ch с помощью оператора $OUT \&H22C, \&H20$ (в Pascal: $port[\$22C] := \20). Операторы $OUT \&H226, 1$ и $OUT \&H226, 0$ (или $port[\$226] := 1; port[\$226] := 0$;

используются для сброса цифрового сигнального процессора и обеспечения правильной работы звуковой платы (Пр-10). Результат оцифровки входного сигнала считывается оператором $x(i) = INP(\&H22A)$ ($x[i] := port[\$22A];$).

```
uses crt, graph; const n=1000;                                'Пр-10
var x: array [0..N] of integer; Gd, Gm, i, z: integer;          'PASCAL
BEGIN Gd:=Detect; InitGraph(Gd,Gm, 'c:\bp\bgi');
if graphResult <> grOk then Halt(1);
port[$226]:=1; delay(10); port[$226]:=0;
Repeat Repeat port[$22C]:=$20; until port[$22A]-127>10;
cleardevice;
for i:=1 to n do begin port[$22C]:=$20; x[i]:=port[$22A]-127;
circle(i,240-x[i],2); line(i,240-x[i],i-1,240-z); z:=x[i];
delay(5); end; until KeyPressed;
END.
```

Для тестирования цифрового осциллографа к микрофонному или линейному входам звуковой платы подключают два провода через соответствующий разъем. Если коснуться провода рукой, то на экране появится осциллограмма напряжения, вызванная "наводками" на теле человека (частота 50 Гц). Чтобы получить осциллограмму напряжения с выхода двухполупериодного выпрямителя на 12 В, нами использовался делитель напряжения (переменный резистор на 4,7 кОм), подвижный контакт которого через резистор 22 кОм соединялся с входом звуковой платы. Было обнаружено, что при слабом входном сигнале коэффициент усиления платы плавно растет, но когда сигнал достигает некоторого уровня, происходит его резкое уменьшение. Это, очевидно, связано с желанием производителей звуковой платы регистрировать как тихие, так и громкие звуки.

4.3. ВЫВОД СИГНАЛОВ ИЗ ПЭВМ

Персональный компьютер может быть использован в экспериментах в качестве генератора сигналов заданной формы. Для этого удобно задействовать параллельный порт LPT, который позволяет выдавать 8 разрядное двоичное слово. Чтобы вывести требуемую совокупность лог.0 и лог.1 необходимо записать в ячейку памяти с адресом $378_{16} = 888_{10}$ соответствующее число. При этом используется команда *OUT {адрес}, {число}*. Например, команда

OUT &H378, 15

записывает в ячейку ОЗУ 378_{16} число 00001111_2 . На выводах 2, 3, 4, 5 LPT-порта появится лог.1, а на остальных — лог.0.



БЕББИДЖ Чарльз (1791–1871 гг.)

Разработал логическую структуру и программное обеспечение аналитической машины механического типа. Она содержала арифметическое устройство ("мельница"), оперативную память ("склад"), управление работой осуществлялось с помощью перфокарт. В машине было предусмотрено разветвление выполнения программы в зависимости от результата вычисления (принцип условного перехода).

1. Управление матрицей светодиодов через LPT-порт. Для того чтобы визуализировать состояние параллельного порта ПЭВМ к нему следует подключить 8 светодиодов (рис. 4.15.1). В случае записи в ячейку ОЗУ 378₁₆ числа $1_{10} = 00000001_2$ будет гореть первый светодиод, если записать число $64_{10} = 01000000_2$, загорит седьмой светодиод. Если записать число $15_{10} = 00001111_2$, загорят первые четыре светодиода и т.д.

1. Управление линейкой светодиодов. Напишите программу, вызывающую свечение а) пятого и восьмого светодиодов; б) всех светодиодов кроме третьего; в) восьмого, седьмого и шестого светодиодов; г) всех четных светодиодов; д) всех нечетных светодиодов.

2. Бегущие огни. Запустите программу Пр-11 и изучите ее работу. На ее основе напишите программу, работающую так: а) загорается первый светодиод, затем гаснет и одновременно с этим загорается второй светодиод, и т.д. (слева на право); б) загорается восьмой светодиод, затем гаснет и одновременно загорается седьмой светодиод, и т.д. (справа на лево); в) загорается первый и второй светодиоды; гаснет первый, загорается третий; гаснет второй, загорается четвертый и т.д.

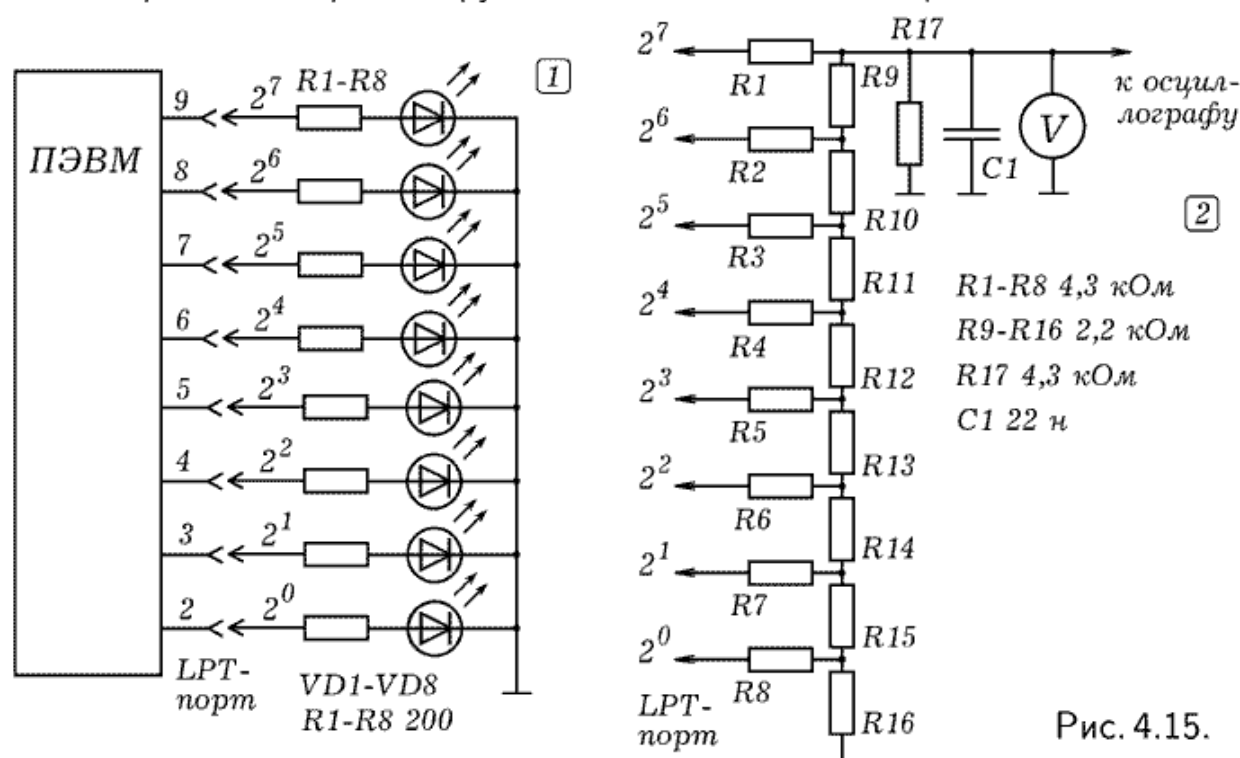
```

WHILE INKEY$ = ""                                'QBASIC                                'Пр-11
x = 0 : OUT &H378, x: GOSUB 10: x = 1 : OUT &H378, x: GOSUB 10
x = 2 : OUT &H378, x: GOSUB 10: x = 4 : OUT &H378, x: GOSUB 10
x = 8 : OUT &H378, x: GOSUB 10: x = 16: OUT &H378, x: GOSUB 10
x = 32: OUT &H378, x: GOSUB 10: x = 64: OUT &H378, x: GOSUB 10
x = 128: OUT &H378, x: GOSUB 10: WEND
10 FOR i = 1 TO 100: PRINT x: NEXT: RETURN
    
```

3. Мигание четных и нечетных светодиодов. Напишите программу, которая вызывает периодическое загорание четных светодиодов, их гашение с одновременным загоранием нечетных светодиодов и т.д. Повторите опыты изменяя время свечения светодиодов и частоту их мигания.

2. Изучение цифро-аналогового преобразователя. Чтобы получить формирователь сигналов заданной формы к ПЭВМ следует подключить цифро-аналоговый преобразователь. Простейший ЦАП может быть собран из резисторов (рис. 4.15.2). Если в ячейку памяти 378₁₆ записать число x от 0 до 255, то на восьми выводах LPT-порта, соединенных с входами ЦАП, появляется комбинация лог. 0 и 1, соответствующая

числу x в двоичном коде. Схема такова, что вклад лог.1, поданной на вход $k + 1$ разряда, в 2 раза больше, чем вклад лог.1, поданной на вход k -ого разряда. В результате напряжение на выходе ЦАП прямо пропорционально величине x , в чем можно убедиться с помощью вольтметра или осциллографа. Для сглаживания ступенек, получающихся на выходе ЦАП, параллельно резистору подключают конденсатор $C1$.



Для того, чтобы получить на выходе ЦАП прямоугольные импульсы максимальной амплитуды, в параллельный порт следует через равные промежутки времени записывать следующие числа: 0, 0, 0, 0, 255, 255, 255, 255, 0, 0, 0, 0, 255, 255, 255, 255, ... Для получения пилообразного напряжения, величина x должна в цикле равномерно увеличиваться от 0 до 255, затем обращаться в 0, после чего снова увеличиваться. Аналогичным образом формируются синусоидальные колебания.

1. Изучение ЦАП с помощью вольтметра. Подключите к выходу ЦАП вольтметр и получите на выходе ЦАП различные напряжения, соответствующие 0, 127, 255 и т.д. (программа Пр-12).

```

WHILE INKEY$ = ""
OUT &H378, 255 : FOR i = 1 TO 30: PRINT "255": NEXT i
OUT &H378, 127 : FOR i = 1 TO 30: PRINT "127": NEXT i
OUT &H378, 0 : FOR i = 1 TO 30: PRINT "0": NEXT i
WEND
    
```

Пр-12
QBASIC

2. Формирование прямоугольных импульсов. Подключите к выходу ЦАП осциллограф. Напишите программу, позволяющую получить на выходе ЦАП прямоугольные импульсы с периодом 0,1–1 с (Пр-13).

```
WHILE INKEY$ = ""
OUT &H378, 255 : FOR i = 1 TO 1000: PRINT "255": NEXT
OUT &H378, 0 : FOR i = 1 TO 1000: PRINT "0": NEXT
WEND
```

'Пр-13
'QBASIC

3. *Формирование пилообразных импульсов.* Получите на выходе ЦАП пилообразные импульсы (Пр-14). Измените их амплитуду и частоту.

```
uses crt, dos; var i: integer;
BEGIN Repeat for i:=1 to 255 do
begin port[888]:=i; write(i, ' ');
delay(500); end; until KeyPressed;
END.
```

'Пр-14
'PASCAL

4. *Получение импульсов заданной формы.* Напишите программу, которая позволяет получить на выходе ЦАП синусоидальные колебания (программа Пр-15). Научитесь изменять их частоту и амплитуду. Получите импульсы следующей формы: в течение Δt напряжение $U = 0$, затем оно возрастает пропорционально t , после чего убывает по экспоненциальному закону. С помощью осциллографа убедитесь в том, что формируемый сигнал является дискретным, то есть состоит из маленьких ступенек, которые могут быть сглажены емкостным фильтром.

```
uses crt, dos; var i: integer;
BEGIN Repeat
i:=i+1; port[888]:=round(127+127*sin(i/5000));
until KeyPressed;
END.
```

'Пр-15
'QBASIC

3. Передача информации по каналу связи. Для того, чтобы продемонстрировать одностороннюю передачу информации по проводам достаточно два одинаковых компьютера соединить двумя проводниками (рис. 4.16.1). Один провод — общий, соединяет 25 вывод LPT-порта ПЭВМ 1 с 25 выводом LPT-порта ПЭВМ 2. Второй провод — сигнальный, соединяет 3 вывод LPT-порта ПЭВМ 1 с 11 выводом LPT-порта ПЭВМ 2. На передающем компьютере следует запустить программу, кодирующую сообщение, а на принимающем — программу, декодирующую сообщение. Обе программы могут быть написаны на языках QBasic или Pascal.

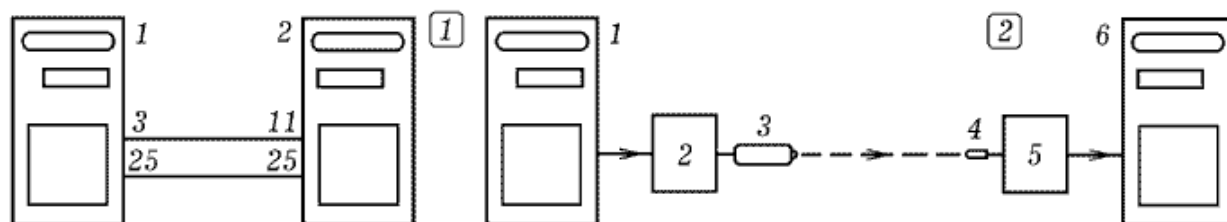


Рис. 4.16.

При запуске программы-кодера на ПЭВМ 1, она запрашивает передаваемое сообщение, состоящее из 0 и 1, затем последовательно перебирает символ за символом, осуществляя частотно-импульсное кодирование, так что на соответствующем выводе LPT-порта появляются импульсы напряжения изменяющейся частоты. Допустим, символу "1" соответствуют импульсы частотой f_1 , символу "0" — импульсы частотой f_2 , если

сообщение не передается, то на выходе — лог.0. Программу-декодер на ПЭВМ 2 запускают раньше начала сеанса связи, она должна осуществлять декодирование поступающих сигналов. После окончания передачи сообщения программа-декодер должна перейти в режим ожидания до начала следующего сеанса связи.

Для осуществления передачи сообщений по оптическому каналу связи к LPT-порту передающей ПЭВМ 1 через схему сопряжения 2 подключают полупроводниковый лазер-указку 3 (рис. 4.16.2). Его луч должен попадать на фотодиод 4, соединенный через формирователь сигнала 5 с LPT-портом принимающей ПЭВМ 6. Сначала запускают программу-декодер, принимающая ПЭВМ 2 находится в режиме ожидания. Затем запускают программу-кодер на ПЭВМ 1 и вводят сообщение в виде последовательности 0 и 1. ПЭВМ 1 осуществляет частотно-импульсное кодирование, лазер начинает мигать, периодически освещая фотодиод. ПЭВМ 6 декодирует сообщение и выводит его на экран. Можно предусмотреть передачу сообщений на русском языке, для этого программа-кодер сначала должна закодировать каждую из 32 букв пятью битами 0 или 1, а уже потом получившуюся последовательность 0 и 1 использовать для частотно-импульсной модуляции излучения лазера. Подключение к LPT-порту лазерной указки, лампочки, динамика, двигателя и т.д. осуществляется с помощью схемы, изображенной на рис. 4.17.2.

4.4. ЗАМКНУТЫЕ САУ НА БАЗЕ ПЭВМ

В современной технике широко используются различные системы автоматического управления (САУ). Разомкнутые САУ состоят из управляющего органа и связанного с ним исполнительного органа, который непосредственно воздействует на объект управления. Замкнутые САУ отличаются наличием одного или нескольких датчиков, контролирующих состояние объекта управления и посылающих соответствующие сигналы в управляющий орган (обратная связь). Создать простую систему автоматического управления можно на базе ПЭВМ. Для этого к выводам LPT-порта через специальные схемы сопряжения подключают те или иные исполнительные органы (лампочка, светодиод, электродвигатель, динамик, электромагнит) и датчики (фотодиод, фоторезистор, микрофон).

1. Электронно-механическая САУ. Соберем установку, позволяющую промоделировать функционирование разомкнутой и замкнутой САУ. Для этого к LPT-порту ПЭВМ подключим устройство сопряжения 2, выход которой соединен с маломощным электродвигателем 3 и оптодатчик, состоящий из светодиода 7, фотодиода 6 и формирователя импульсов 3 (рис. 4.17.1). Принципиальная схема устройства, позволяющего подключить двигатель к ПЭВМ, представлена на рис. 4.17.2. К валу двигателя прикрепим картонный диск 5 с одной или несколькими прорезями. Оптодатчик установим так, чтобы при вращении диска происходило затемнение и освещение фотодиода.



ГЛУШКОВ Виктор Михайлович (1923–1982 гг.)
Директор института кибернетики, организованного в 1962 г. на базе Вычислительного центра АН УССР (Киев). Развивал теорию автоматов, ее математические основы, занимался проблемой синтеза дискретных устройств, новыми методами в искусственном интеллекте. Разрабатывал принципы построения ЭВМ, которые были воплощены в вычислительных машинах КИЕВ, ДНЕПР-2, МИР. Один из выдающихся кибернетиков в СССР.

Для того, чтобы получить разомкнутую систему управления оптодатчик следует отодвинуть от диска и запустить программу, обрабатывающую сигналы с оптодатчика и управляющую работой двигателя. Эта программа достаточно проста и читатель может написать ее самостоятельно. Допустим, сначала компьютер находится в режиме ожидания, вращаясь в цикле и опрашивая при этом LPT-порт. При кратковременном пересечении светового пучка оптодатчика на его выходе формируется сигнал, который поступает в компьютер. Тот выполняет заданную последовательность действий, моделируя технологический процесс: двигатель включается на несколько секунд, затем выключается, после этого снова включается на чуть большее время, затем снова выключается и т.д. После окончания компьютер снова переходит в режим ожидания в котором находится до следующего пересечения оптодатчика.

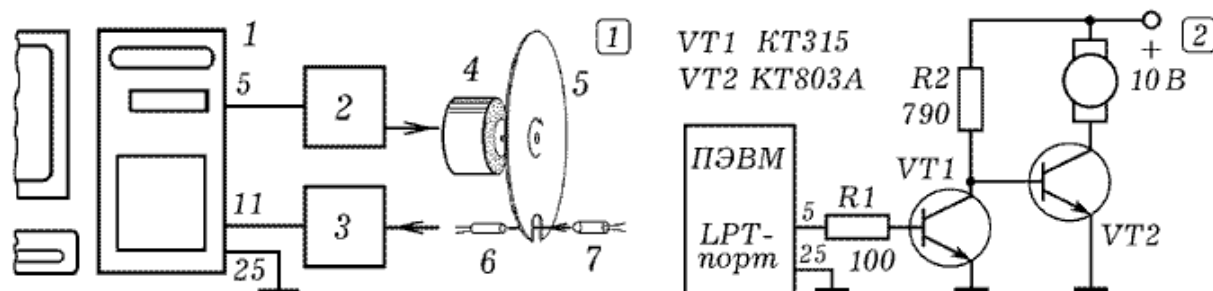


Рис. 4.17.

Установим оптодатчик так, чтобы диск находился между светодиодом и фотодиодом, тем самым замкнув САУ. Запустим программу Пр-16. Двигатель начнет вращать диск, на выходе оптодатчика появится последовательность импульсов, которые будут поступать в ПЭВМ. Когда их число достигнет 50, компьютер выключит двигатель, диск по инерции совершит несколько оборотов и остановится. Через некоторое время компьютер снова включит двигатель и после совершения 100 оборотов выключит его. Используемые программы могут быть составлены так, что при их запуске на экране ПЭВМ будет появляться список режимов функционирования разомкнутой или замкнутой САУ. При выборе того или иного режима система ведет себя соответствующим образом.

```
1 : CLS : OUT &H378, 4: k = 0
WHILE k < 120: x = INP(&H379): PRINT x;
```

'Пр-16
'QBASIC


```

IF (x = 119) AND (y = 127) THEN k = k + 1
y = x: PRINT k: WEND: OUT &H378, 5
FOR i = 1000 TO 500 STEP -20
SOUND i, i / 300: NEXT
k = 0: OUT &H378, 4
WHILE k < 250: x = INP(&H379): PRINT x;
IF (x = 119) AND (y = 127) THEN k = k + 1
y = x: PRINT k: WEND
OUT &H378, 5: SOUND 500, 10
FOR i = 1 TO 1500: PRINT "ОСТАНОВКА": NEXT: GOTO 1
END

```

2. Замкнутая электронно-акустическая система. В качестве еще одного примера замкнутой САУ рассмотрим систему, в которой компьютер одновременно используется в качестве генератора сигнала и регистрирующего устройства. К 5 выводу LPT-порта подключим активные колонки, а напротив них установим микрофон, соединенный со звуковой платой ПЭВМ (рис. 4.18.1). Можно написать программу так, чтобы колонки генерировали звуковые импульсы, а на экране монитора периодически получались осциллограммы сигнала с выхода микрофона.

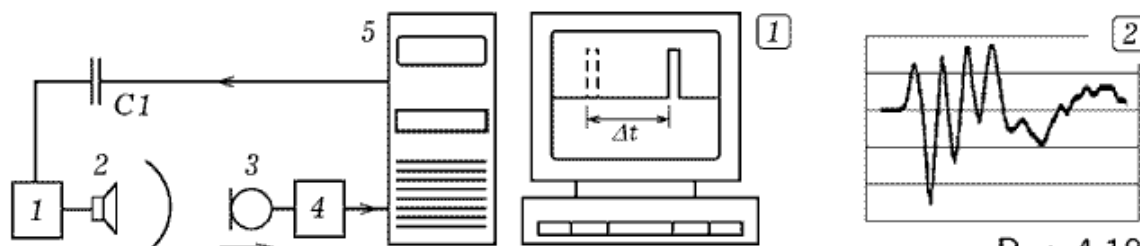


Рис. 4.18.

Текст используемой программы Пр-17 представлен ниже. С помощью оператора `port[888] := 255;` (или 0) осуществляется подача логической 1 (или 0) на 2–9 выводы LPT-порта. К одному из них через резистор 1 кОм или конденсатор 0,05 мкФ следует подключить вход активных колонок. С приходом импульса колонки выдают щелчок и одновременно начинается оцифровка сигнала с микрофона. В результате положение осциллограммы на экране (рис. 4.18.2) зависит от расстояния между микрофоном и динамиком. При удалении микрофона импульс на экране смещается вправо. Для стабилизации осциллограммы на экране монитора вычисляется среднее значение оцифрованного напряжения по всему массиву $x[i]$ (переменная `sredn`) и учитывается его изменение, при построении графика.

```

uses crt, graph; const n=200; m=5;
var x: array [0..N] of integer;
s, sredn, Gd, Gm, i, z: integer;
BEGIN Gd:=Detect;
  InitGraph(Gd,Gm, 'c:\bp\bgi');
  if graphResult <> grOk then Halt(1);
  port[$226]:=1; delay(10); port[$226]:=0;
  Repeat {Repeat port[$22C]:=$20; until port[$22A]-127>5;}
  port[888]:=255; delay(15); port[888]:=0; delay(15);
  port[888]:=255; delay(15); port[888]:=0; s:=0;
  for i:=1 to n do begin port[$22C]:=$20;

```

'Пр-17
'PASCAL

```
x[i]:=(port[$22A]-127); delay(1); s:=s+x[i]; end;
sredn:=round(s/n);
for i:=1 to n do begin circle(m*i,240-x[i]+sredn,2);
line(m*i,240-x[i]+sredn, m*(i-1),240-z+sredn); z:=x[i]; end;
delay(3000); cleardevice;
until KeyPressed;
END.
```

3. Акустический метод измерения координаты. Предыдущий эксперимент может быть положен в основу измерения координаты тела с достаточно высокой частотой отсчетов. Допустим, необходимо измерить координату колеблющегося маятника через каждые 0,2 с (с частотой 5 Гц) и построить график зависимости $x = x(t)$. Для этого на конце маятника устанавливают микрофон, соединенный через устройство сопряжения (усилитель и формирователь прямоугольных импульсов) с 11 выводом LPT-порта. Напротив колеблющегося маятника с микрофоном располагают активные колонки, соединенные через конденсатор с 3 выводом LPT-порта ПЭВМ. При запуске программы на колонки поступают электрические импульсы, динамик издает щелчки. Звуковые импульсы достигнув микрофона вызывают появление на 11 выводе LPT-порта импульса напряжения. По времени запаздывания сигнала с микрофона относительно сигнала, посылаемого на колонки, можно определить координату микрофона, результаты записать в файл, построить график зависимости координаты маятника от времени. Программа может быть написана так, что на экране ПЭВМ появится метка (изображение импульса с микрофона), смещающаяся вправо или влево в зависимости от положения микрофона, либо будет выводиться его координата.

Рассмотрим один из возможных вариантов программы (Пр-18). При записи в ячейку ОЗУ с адресом 888 числа $2_{10} = 00000010_2$ на 3 выводе LPT-порта появляется логическая 1, динамик вырабатывает звуковой импульс. Практически сразу же запускается цикл 1 в котором в массиве $xx[i]$ сохраняются состояния ячейки ОЗУ с адресом 889 в последовательные моменты времени. Когда звуковой импульс достигает микрофона, на выходе формирователя прямоугольных импульсов появляется логическая 1, в результате чего изменяется содержимое ячейки ОЗУ 889. Это также сохраняется в массиве $xx[i]$. После этого запускается цикл 2, в котором состояние массива $xx[i]$ выводится на экран в текстовом или графическом виде. Оператор `delay(100)`; обеспечивает задержку в 100 мс, после чего осуществляется очистка экрана (оператор `cleardevice`);. Все это находится в цикле `Repeat ... untill KeyPressed`; и поэтому повторяется до тех пор, пока не будет нажат пробел.

```
uses crt,graph;
var t,DV, MV, EC, x, y, k, time, i :integer;
xx : array[1..1200] of integer;
Begin
    DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
    EC:=GraphResult; if EC<>grOK then Halt(1); end;
    repeat
        {Начало общего цикла}
```

'Пр-18
'PASCAL

```

port[888]:=2; delay(2); port[888]:=0; {Формирование импульса}
for i:=1 to 1200 do xx[i]:=port[889]; {Цикл 1}
for i:=1 to 1200 do begin
    {Цикл 2}
    if xx[i]<120 then xx[i]:=200 else xx[i]:=0;
{ if i/50=round(i/50) then line(i,0,i,480); }
    circle(round(i/2),240-xx[i],2); end; {Конец цикла 2}
    delay(100); cleardevice; {Задержка и очистка экрана}
until keypressed; {Конец общего цикла}
CloseGraph;
End.

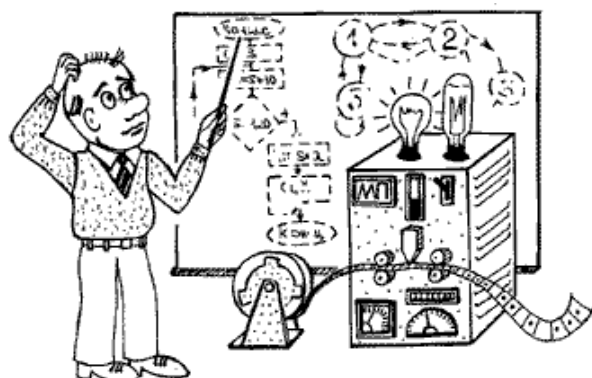
```

При запуске программы динамик начинает выдавать последовательность импульсов. Перемещая микрофон относительно динамика, наблюдайте смещение импульса на экране монитора. При приближении микрофона к динамику на распространение звука требуется меньшее время, поэтому изображение импульса на экране смещается влево. Сравнительно небольшие перемещения микрофона (2–5 мм) вызывают заметные смещения импульса на экране монитора.

4. Замкнутая электронно–оптическая система. Для того, чтобы продемонстрировать работу замкнутой САУ, достаточно подключить к ПЭВМ источник света, фотодатчик и написать соответствующую программу. В качестве источника света используется монитор, яркость которого можно изменять плавно, выводя на черный экран заданное количество случайно расположенных точек, или скачком, рисуя белый квадрат. В качестве фотодатчика следует использовать фоторезистор ФСК–1, так как фотодиод ФДК–256 не реагирует на свечение монитора. Схема сопряжения аналогична схеме на рис. 4.2.1.

В одном из вариантов опыта используется программа, которая на экране медленно рисует светлую полосу, что моделирует некоторый управляемый процесс. Вблизи экрана устанавливают фоторезистор, соединенный с LPT–портом. Когда полоса доходит до него, в ПЭВМ поступает сигнал, компьютер перестает рисовать полосу и выдает звук. Затем экран монитора очищается и все повторяется снова. Передвигая фоторезистор по экрану, можно получить полосы различной длины.

В другом опыте фотодатчик устанавливают напротив экрана и запускают программу, которая рисует на черном экране случайно расположенные окружности. Когда яркость экрана достигает порогового значения, фотодатчик вырабатывает сигнал, ПЭВМ издает звук и начинает стирать окружности с экрана. Когда яркость экрана понижается, фотодатчик срабатывает, и система автоматически реагирует на его сигнал. Если разомкнуть систему, отодвинув фотодатчик от экрана, и направить на него пучок света, то можно показать работу разомкнутой САУ.



ГЛАВА 5 ОСНОВЫ ТЕОРИИ КОДИРОВАНИЯ И ПЕРЕДАЧИ ИНФОРМАЦИИ

Пришла пора познакомиться с фундаментальными идеями теоретической информатики: что такое информация, как определить количество информации в сообщении, какие используются способы для кодирования различных видов информации и т.д. Будущий специалист в области вычислительной техники и телекоммуникационных систем должен знать, какие закономерности описывают процесс передачи информации по каналу связи, что такое помехоустойчивый и самокорректирующийся коды, какие задачи алгоритмически неразрешимы, как можно обучить компьютер...

5.1. ДАННЫЕ, ИНФОРМАЦИЯ, КОДИРОВАНИЕ

1. Энтропия как мера неопределенности знаний. Датчик, измеряющий физическую величину, создает электрический сигнал. *Данные* — это зарегистрированные сигналы. Методы регистрации сигналов предполагают перемещение тел, изменение формы, поверхности, электрических, магнитных и оптических характеристик носителя информации, изменение состояния электронной системы и т.д. К основным операциям с данными относятся сбор, формализация, фильтрация, сортировка, архивация, защита, передача, преобразование информации.

Пусть выход датчика в зависимости от измеряемой величины может принимать n различных состояний. Каждый акт измерения является опытом с n исходами, результат которого заранее не определен. Аналогично, при формировании сообщения каждый символ выбирается из алфавита, содержащего n букв, что может рассматриваться как опыт с n исходами.

Проведение опыта (получение сообщения) снижает неопределенность наших знаний об объекте. Если опыт имеет n исходов, то мерой неопределенности является функция $H(n)$, зависящая от числа исходов. Если $n = 1$, то неопределенность отсутствует и $H(n) = 0$, с ростом n функция $H(n)$ возрастает. Рассмотрим два независимых опыта α и β с числом равновероятных исходов n_α и n_β . Сложный опыт, состоящий в одновременном выполнении опытов α и β имеет $n_\alpha \cdot n_\beta$ равновероятных исходов (то же самое относится к сообщению из двух символов). Итак, мера

неопределенности сложного опыта должна быть равна сумме мер неопределенностей опытов α и β : $H(n_\alpha \cdot n_\beta) = H(n_\alpha) + H(n_\beta)$. Перечисленным требованиям удовлетворяет функция $H(n) = \log_a(n)$, которая называется *энтропией*. Если $a = 2$, то единицей измерения является *бит* (binary digit – двоичная цифра), если $a = 10$, то *дит*.

В опыте, имеющем n равновероятных исходов, вероятность каждого из них $p = 1/n$. На долю каждого из n исходов приходится энтропия $H_1 = (1/n)\log_2 n = -(1/n)\log_2(1/n) = -p\log_2 p$. Найдем количество информации, получаемой при выборе одного из двух равновероятных исходов: $H = (1/2)\log_2 2 + (1/2)\log_2 2 = 1$ (бит).

Энтропия опыта (сообщения) с n неравновероятными исходами вычисляется по формуле Шеннона (1948 г.):

$$H = \sum_{i=1}^N p_i \log_a(1/p_i) = - \sum_{i=1}^N p_i \log_a p_i.$$

где $p_i = 1/n$ — вероятность i -ого исхода. Например, для опыта с двумя исходами, вероятности которых $p_1 = 0,3$ и $p_2 = 0,7$ энтропия равна $H = -(0,3 \cdot \log_2 0,3 + 0,7 \cdot \log_2 0,7) = 0,88$ бит, то есть меньше 1 бит. В русском алфавите 34 символа (включая пробел), если бы они использовались с равными вероятностями $1/34$, то на каждый символ приходилось бы $H = \log_2 34 > 5$ бит. Учитывая вероятности использования каждого символа (рис. 5.5.2), получаем $H \approx 4,72$ бита.

Рассмотрим еще один пример: случайно загадали число x в интервале от 1 до 64. Необходимо определить количество информации в сообщении: а) $x = 9$; б) $x > 32$; в) $x < 23$.

Величина x может с равной вероятностью $p = 1/64$ принять одно из 64 значений. Следовательно энтропия (мера неопределенности наших знаний о системе) до прихода сообщения равна $H_0 = \log_2 64 = 8$ бит. С приходом сообщения $x = 9$ неопределенность уменьшилась до нуля: $H_1 = 0$. Значит количество информации в сообщении $I = H_0 - H_1 = 8$ бит.

Во втором случае неопределенность уменьшается в два раза: остается 32 равновероятных исхода опыта, энтропия становится равной $H_2 = \log_2 32 = 4$ бит. Информативность этого сообщения равна $I = H_0 - H_2 = 4$ бит. С приходом сообщения $x < 23$ остается 22 равновероятных исхода опыта, энтропия уменьшается до $H_3 = \log_2 22 = 4,46$ бит. Количество информации в третьем сообщении $I = H_0 - H_3 = 3,54$ бит.

Рассмотрим еще одну задачу. Пусть в алфавите имеется 4 буквы с вероятностями $p_1 = 0,47$, $p_2 = 0,26$, $p_3 = 0,15$, $p_4 = 0,12$. Требуется найти энтропию сообщения из 1000 букв.

Вычислим среднюю энтропию, приходящуюся на 1 букву:

$$H_1 = - \sum_{i=1}^4 p_i \log_2 p_i = -(0,47 \log_2 0,47 + 0,26 \log_2 0,26 + 0,15 \log_2 0,15 + 0,12 \log_2 0,12) = 1,795 \text{ бит.}$$

Если в сообщении $N = 1000$ букв, то в нем $H = NH_1 = 1795$ бит.

Энтропия термодинамической системы определяется по формуле, аналогичной формуле Шеннона. Пусть в сосуде объемом V находятся N молекул газа. Разобьем V на s одинаковых элементарных объемов ΔV и подсчитаем число молекул n_i в каждом объеме ΔV_i ($i = 1, 2, \dots, s$). Мерой беспорядка является энтропия

$$H = - \sum_{i=1}^s \frac{n_i}{N} \ln \frac{n_i}{N} = - \sum_{i=1}^s p_i \ln p_i, \quad \sum_{i=1}^s n_i = N, \quad p_i = \frac{n_i}{N}.$$

В самом деле, если молекулы распределены совершенно беспорядочно и однородно, то $N/n_i = s$ и энтропия равна $-\ln(n_i/N) = \ln s$. Если молекулы оказались внутри одного k -ого элементарного объема ($p_k = 1$, и $p_j = 0$ для $j \neq k$), то есть система упорядочилась максимальным образом, то энтропия $H = 0$.

2. Информация и ее свойства. Пусть до получения сообщения имеются некоторые предварительные сведения об объекте α . Мерой нашей неосведомленности является функция $H(\alpha)$, которая в то же время служит и мерой неопределенности состояния объекта. После получения некоторого сообщения β получатель приобретает дополнительную информацию $I_\beta(\alpha)$, уменьшающую его априорную неосведомленность так, что апостериорная (после получения сообщения β) неопределенность его знаний о состоянии объекта становится равной $H_\beta(\alpha)$. Тогда количество информации $I_\beta(\alpha)$ об объекте, содержащееся в сообщении β , равно убыли энтропии, то есть уменьшению неопределенности знаний о состоянии системы: $I_\beta(\alpha) = H(\alpha) - H_\beta(\alpha)$,

Если после сообщения неопределенность обратилась в ноль ($H_\beta(\alpha) = 0$), то первоначальное неполное знание заменится полным знанием и количество информации в сообщении равно начальной (априорной) энтропии $I_\beta(\alpha) = H(\alpha)$. Иными словами, энтропия объекта $H(\alpha)$ может рассматриваться как мера недостающей информации. Основными требованиями к информации являются достоверность, полнота, актуальность, полезность, понятность, адекватность.

Пусть в результате опыта датчик выдает дискретную случайную величину x . Энтропия дискретной случайной величины x равна минимальному количеству бит, которое необходимо передать по каналу связи (записать в ЗУ) о текущем значении этой величины x .

Свойства информации (энтропии): 1) информация (энтропия) сложного опыта (сообщения) равна сумме информации (энтропии) его частей; 2) при том же числе исходов наибольшую информацию (энтропию) имеет опыт (сообщение) с равновероятными исходами.

Докажем последнее утверждение. Энтропия опыта с двумя исходами равна: $H(p) = -p \ln p - (1-p) \ln(1-p)$ (рис. 5.1.1). Исследуем $H(p)$ на максимум, для чего продифференцируем по p и приравняем к 0: $dH/dp = -\ln p - 1 + \ln(1-p) + 1 = 0$. Имеем: $\ln p = \ln(1-p)$. Это выражение

превращается в истинное высказывание, когда $p = 1 - p$, $p = 0,5$, то есть оба исхода равновероятны.

3. Кодирование текстовой информации. Информативность сообщения в битах равна числу вопросов с равновероятными бинарными вариантами ответов, на которые необходимо ответить, чтобы передать данное сообщение (снять неопределенность знаний об объекте). Например, компьютер случайно "загадал" число x от 1 до 128. Пользователь, стараясь угадать x за меньшее число шагов, делит интервал от 1 до 128 пополам и вводит $y = 64$. Вероятности того, что $x < y$ и $x > y$ одинаковы, поэтому при любом ответе компьютер сообщит 1 бит информации. Так как $\log_2 128 = 7$ ($2^7 = 128$), потребуется 7 ответов информативностью 1 бит.

Рассмотрим алфавит из m символов, где p_i ($i = 1, 2, 3, \dots, m$) — вероятность выбора из алфавита i -ой буквы для кодирования сообщения. Каждый такой выбор уменьшает степень неопределенности сообщения, увеличивая количество информации. Среднее количество информации, сообщаемое при выборе одного символа, согласно формуле Шеннона, равно:

$$I_1 = - \sum_{i=1}^m p_i \log_2 p_i.$$

Если выбор каждой буквы равновероятен, то $p_i = 1/m$. Подставляя это значение, получим, что каждый символ несет информацию:

$$I_1 = - \log_2 \frac{1}{m} \cdot \sum_{i=1}^m \frac{1}{m} = - \log_2 \left(\frac{1}{m} \right) = \log_2 m.$$

В ЭВМ информация кодируется числовыми кодами. Если система счисления имеет основание P , то числу разрядов n соответствует количество $N = P^n$ различных комбинаций цифр, то есть чисел, которые могут быть записаны в n разрядах. При $P = 2, n = 4$, количестве чисел $N = 2^4 = 16$: 0000, 0001, 0010, 0011..., 1111. Значит в 4 двоичных разрядах может быть записано $I = \log_2 16 = 4$ бита.

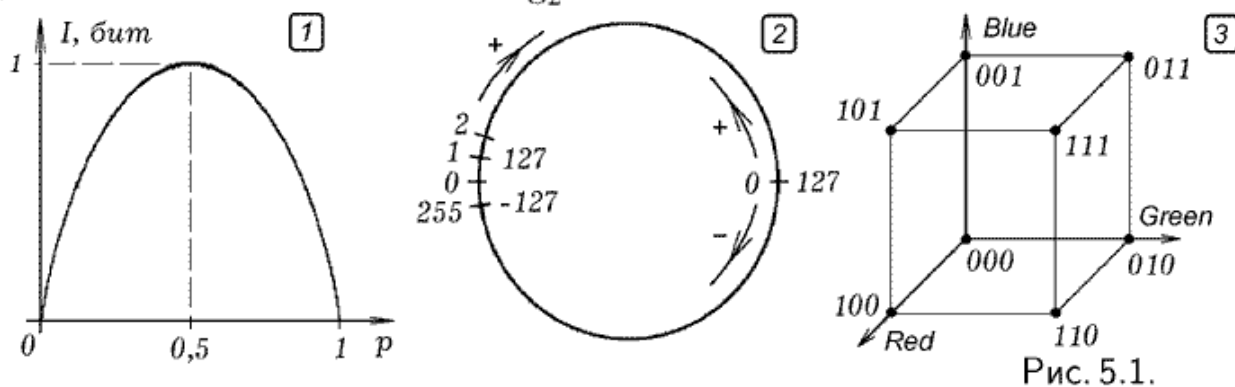


Рис. 5.1.

В случае представления информации в виде последовательности символов 0 и 1, декодирование удобно осуществлять при наличии соглашения о фиксированной длине последовательностей из 0 и 1, составляющих слово. Такой длиной принято считать 1 байт: восемь символов 0 или 1, то есть 8 бит. Существуют более крупные единицы измерения количества информации:

1 Кбайт=1024 байта= 2^{10} байт, 1 Мбайт=1024 Кбайта= 2^{20} байт,
1 Гбайт=1024 Мбайта= 2^{30} байт, 1 Тбайт=1024 Гбайта= 2^{40} байт
($2^{10} = 1024_{10} = 1000000000_2$).

В восьми разрядах (байте) можно записать $2^8 = 256$ различных целых двоичных чисел, что достаточно чтобы дать неповторяющееся обозначение каждой заглавной и строчной букве русского и английского алфавитов, цифрам, знакам препинания, всем символам на клавиатуре компьютера. Таблица кодирования символов 8-битовыми числами называется кодовой таблицей символов ASCII (American Standart Code for Information Interchange), она представлена на обложке. Если считать, что использование каждого символа равновероятно, то его информационный вес равен $I = \log_2 256 = 8 \text{ бит} = 1 \text{ байт}$. Информационный объем текстового сообщения в байтах численно равен количеству символов N (или $8N$ бит). Тексты хранятся в памяти компьютера в двоичном коде и программным способом преобразуются в изображение на экране.

4. Системы счисления. Числовая информация в ЭВМ кодируется в двоичной, двоично-десятичной или в шестнадцатиричной системе счисления. Система счисления — это способ наименования и изображения чисел с помощью цифр, то есть символов, имеющих определенные количественные значения.

В позиционной системе счисления количественное значение каждой цифры зависит от ее места (позиции) в числе. Количество P различных цифр, используемых для изображения числа в позиционной системе счисления, называется *основанием системы счисления*. Значения цифр лежат в пределах от 0 до $P-1$. Числа в системе счисления с основанием P равны: $a_{r-1}P^{r-1} + a_{r-2}P^{r-2} + \dots + a_1P^1 + a_0P^0 + a_{-1}P^{-1} + a_{-2}P^{-2} + \dots + a_{-s}P^{-s}$, где нижние индексы определяют месторасположение цифры в числе, r и s — количества разрядов для записи целой и дробной части числа соответственно.

Максимальное целое число, которое может быть представлено в r разрядах: $N_{max} = P^r - 1$. Минимальное значащее (не равное 0) число, которое можно записать в s разрядах дробной части: $N_{min} = P^{-s}$. Имея в целой части числа r , а в дробной s разрядов, можно записать всего P^{r+s} разрядных чисел. Диапазон значащих чисел N в системе счисления с основанием P при наличии r разрядов в целой части и s разрядов в дробной части (без учета знака числа) будет: $P^{-s} \leq N \leq P^r - P^{-s}$. При $P = 2, r = 10, s = 6$ имеем: $2^{-6} \leq N \leq 1024 - 2^{-6}$.

Можно доказать теорему: Чтобы перевести число X_P из системы счисления с основанием P в систему счисления с основанием Q , необходимо X_P последовательно делить с остатком на число Q , представленное в системе счисления P , пока частное не станет равно 0. Остатки от деления записанные в обратном порядке являются цифрами числа X_P в системе счисления с основанием Q .

Десятичная система счисления ($P = 10$) используется при ручном счете. При этом: $157,43_{10} = 1 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0 + 4 \cdot 10^{-1} + 3 \cdot 10^{-2}$.

В шестнадцатиричной системе счисления основание $P = 16$. Для записи чисел используются цифры от 0 до 9 и буквы $A=10$, $B=11$, $C=12$, $D=13$, $E=14$, $F=15$. Восемь двоичных разрядов (байт) могут быть представлены только двумя шестнадцатиричными разрядами (1 байт = 8 бит, то есть $2^8 = (16)^2 = 256$ комбинаций).

Двоично-десятичным называется такое представление десятичного числа, при котором каждая цифра отдельно кодируется четырьмя двоичными цифрами и в таком виде записываются последовательно друг за другом. Двоично-десятичная запись более длинная, но в ней удобнее производить арифметические действия и выводить числа на экран.

Представление чисел в различных системах счисления.

| P=10 | P=2 | P=4 | P=8 | P=16 | P=10 | P=2 | P=4 | P=8 | P=16 |
|------|------|-----|-----|------|------|------|-----|-----|------|
| 0 | 0000 | 0 | 0 | 0 | 8 | 1000 | 20 | 10 | 8 |
| 1 | 0001 | 1 | 1 | 1 | 9 | 1001 | 21 | 11 | 9 |
| 2 | 0010 | 2 | 2 | 2 | 10 | 1010 | 22 | 12 | A |
| 3 | 0011 | 3 | 3 | 3 | 11 | 1011 | 23 | 13 | B |
| 4 | 0100 | 10 | 4 | 4 | 12 | 1100 | 30 | 14 | C |
| 5 | 0101 | 11 | 5 | 5 | 13 | 1101 | 31 | 15 | D |
| 6 | 0110 | 12 | 6 | 6 | 14 | 1110 | 32 | 16 | E |
| 7 | 0111 | 13 | 7 | 7 | 15 | 1111 | 33 | 17 | F |

Рассмотрим несколько примеров: 1. Переведем двоичное число в десятичное: $101110_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 46$. 2. Запишем шестнадцатиричное число в двоичной системе счисления: $F17B_{16} = 1111000101111011_2$. 3. Переведем десятичное число в двоично-десятичное: $9703_{10} = 1001011100000011_{2-10}$.

5. Кодирование целых положительных чисел. Кодирование — преобразование состояния X_i системы в символ Y_j посредством оператора P , соответствующего некоторому набору правил: $Y_j = P(X_i)$, где $X = (X_1, X_2, \dots, X_n)$, $Y = (Y_1, Y_2, \dots, Y_m)$ — конечные множества состояний системы и соответствующих им символов.

Система кодирования — правила кодового обозначения объектов, используемые для удобной и эффективной обработки информации. Вся информация (данные и команды) представлена в виде двоичных кодов. Ячейка памяти ОЗУ, к которой можно обратиться, имеет длину 1 байт. Каждый байт ОЗУ имеет свой адрес. Поле — последовательность нескольких бит, записанная в определенном формате и имеющая некоторый смысл. Машинное слово — наибольшая последовательность разрядов (бит), которую ЭВМ может обрабатывать как единое целое. Его длина зависит от разрядности процессора (16, 32, 64 бита). Различают слово, полуслово, двойное слово.

Если длина машинного слова 16 бит, то наибольшее целое число, которое может быть в них записано равно $1111111111111111_2 = 2^{16} - 1 = 65535_{10}$, а наименьшее — 0. В языке Pascal такие числа имеют тип Word и занимают 2 байта. Числа типа Longint занимают в памяти ЭВМ 4 байта.

Арифметические действия в двоичной системе счисления осуществляются по тем же правилам, что и в десятичной. Читателю предлагается самостоятельно рассмотреть операции $6 + 7 = 13$, $14 - 8 = 6$, $5 \cdot 3 = 15$, $12 : 4 = 3$ в двоичной системе счисления.

6. Кодирование целых чисел со знаком. Различают прямой, обратный и дополнительный коды числа. *Прямой код* двоичного числа включает в себя код знака ("+" — 0 или "—" — 1) и абсолютное значение числа:

$$X_{10} = -13, X_2 = -1101, [X_2]_{\text{пр}} = 1|1101,$$

$$X_{10} = 9, X_2 = 1001, [X_2]_{\text{пр}} = 0|1001.$$

Вертикальная линия отделяет знаковый разряд от абсолютного значения числа. *Обратный код* положительных чисел такой же как и прямой код, в знаковом разряде — 0. Для получения обратного кода отрицательного числа его разряды инвертируются (0 заменяется на 1 и наоборот), в знаковом разряде — цифра 1:

$$X_{10} = 12, X_2 = 1100, [X_2]_{\text{пр}} = [X_2]_{\text{об}} = 0|1100,$$

$$X_{10} = -7, X_2 = -0111, [X_2]_{\text{об}} = 1|1000.$$

Дополнительный код положительных чисел такой же как и прямой код. У отрицательных чисел дополнительный код равен результату суммирования обратного кода числа с единицей младшего разряда:

$$X_{10} = 18, X_2 = 10010, [X_2]_{\text{пр}} = [X_2]_{\text{об}} = [X_2]_{\text{дк}} = 0|10011,$$

$$X_{10} = -11, X_2 = -1011, [X_2]_{\text{дк}} = [X_2]_{\text{об}} + 1 = 1|0100 + 1 = 1|0101.$$

В ЭВМ разность $x_1 - x_2$ находится так: оба числа переводятся в дополнительный код, затем складываются и единица в самом старшем разряде отбрасывается. То есть вычитание сводится к сложению, осуществляемому с помощью сумматора. Например, в двоичной системе: $0101 - 0011 = 0010$; дополнительный код для 0011 равен 1101 ($0011 + 1101 = 10000$). Итак $0101 + 1101 = 10010$. Старший разряд отбрасывается, получается $0010 = 2_{10}$. Тот же пример в десятичной системе: $5 - 3 = 2$. Дополнительный код для 3 равен 7, потому что $3 + 7 = 10$. Так как $5 + 7 = 12$, то отбрасывая старший разряд получаем 2.

Целые числа отображаются в ЭВМ с помощью знакового и беззнакового представления. В одном байте может представлено 256 различных комбинаций 0 и 1: 1) беззнаковое представление: от 0 до 255; 2) знаковое представление: от -127 до 127. Числовая прямая при этом как бы замкнута (рис. 5.1.2): если к числу 11111111 прибавить 1, то за счет отбрасывания 9-ого бита снова получится 00000000.

Рассмотрим знаковое представление чисел в 2 байтах. Сместим положение нуля на середину интервала 0 — 65535 и будем считать, что числа из первой половины 0—32767 положительные, а из второй половины 32769 — 65535 — отрицательные. Тогда $100000000000001_2 = 32771_{10}$ — код отрицательного числа -3_{10} , а $000000000000001_2 = 1_{10}$ — код положительного числа. Это соответствует типу *Integer* на языке Pascal, при размещении чисел из интервала $[-32768; 32767]$ в 2-х байтном машинном слове.



АЙЗЕРМАН Марк Аронович (1913–1992 гг.)
Один из выдающихся представителей первого поколения кибернетиков в СССР. Внес значительный вклад в становление информатики и развитие теории автоматического управления. Занимался теорией автоматов, проблемой распознавания образов. Написал книги "Теория автоматического регулирования", "Логика, автоматы, алгоритмы", "Выбор вариантов. Основы теории".

7. Кодирование действительных чисел. Дискретность машинного представления чисел приводит к потере информации, к погрешностям, возникающим при округлении на стадии записи числа и выполнении арифметических действий. Действительные числа образуют континуум — непрерывное множество, между любыми двумя элементами которого можно найти хотя бы еще один элемент. В ЭВМ для записи числа x , принадлежащего бесконечному множеству действительных чисел, отводится конечное число разрядов. При этом множество возможных записей (отображений) X конечно. Например: 0.0000; 0.0001; 0.0002; ...; 0.3960; 0.3961; 0.3962; ... В общем случае машинные числа лишь приближенно соответствуют отображаемым числам: числа 0.0000145 и 0.396131 в данной ситуации будут восприниматься ЭВМ как 0.0000 и 0.3961, то есть приближенно. Это приводит к следующему:

1. Строгие соотношения между действительными числами x_i оказываются нестрогими для их отображений X_i : если $x_1 < x_2$, то $X_1 \leq X_2$.

2. Так как отображения действительных чисел являются приближенными, то математическая обработка действительных чисел в ЭВМ ведется с погрешностью.

3. Математическое понятие нуля заменяется понятием "машинный ноль", то есть значения, которое достаточно близко к нулю.

Число называется *нормализованным* в системе счисления P , если оно представлено в виде $x = \pm m_P \cdot 10^{\pm k_P}$, где m_P — мантисса, лежащая в интервале $P^{-1} \leq m_P < 1$, k_P — порядок числа. В качестве примера, представим числа $2376,89_{10}$ и $-1011,01_2$ в нормализованном виде (в формате с плавающей запятой) $2376,89_{10} = 0,237689_{10} \cdot 10^4$, и $-1011,01_2 = -0,101101_2 \cdot 2^{10}$.

Мантисса двоичного числа лежит в интервале $0,1_2 = 0,5_{10} \leq M < 1$ и содержит только цифры дробной части, которая всегда начинается с 1. Поэтому ноль в разряде целых, запятую, отделяющую дробную часть, и первую цифру дробной части мантиссы можно не сохранять, но восстанавливать при вычислениях. Пример размещения числа $-0,10110101 \dots 1110_2 \cdot 2^{-1101011_2}$ в 4 байтах (32 разрядах) показан на рис. 5.2. Числа типа *Real* на языке *Pascal* размещаются 6 байтах, числа типа *Extended* занимают 10 байт.

Дискретность машинного представления чисел приводит к потере информации, к погрешностям, возникающим при округлении на стадии записи числа в ЭВМ и выполнении арифметических действий. Перед сложением и вычитанием чисел $\overline{0, m_a} \cdot 2^{k_a}$ и $\overline{0, m_b} \cdot 2^{k_b}$ осуществляется выравнивание порядков: $a - b = (\overline{0, m_a} - \overline{0, m_b} \cdot 2^{k_b - k_a}) \cdot 2^{k_a}$. При умножении (делении) мантиссы умножаются (делятся), порядки складываются (вычитаются): $a : b = (\overline{0, m_a} : \overline{0, m_b}) \cdot 2^{k_a - k_b}$.

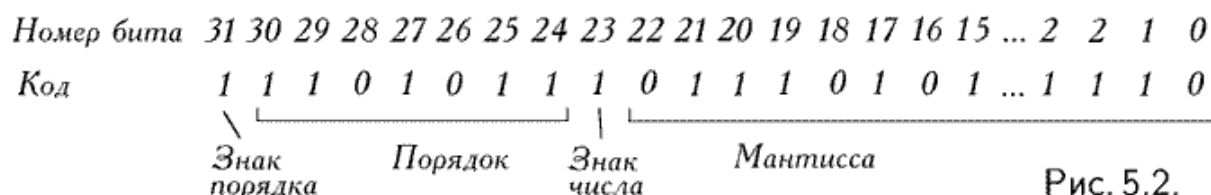


Рис. 5.2.

8. Кодирование двоично-десятичных чисел. Для вывода чисел на экран используется двоично-десятичное представление чисел. При выполнении сложения и вычитания двоично-десятичных чисел используется упакованный формат. Для каждой десятичной цифры отводится по 4 двоичных разряда (полбайта), при этом знак числа кодируется в крайнем правом полубайте числа (1100 — знак "+" и 1101 — знак "-"):

|Цифра|Цифра|Цифра|...|Цифра|Знак|.

В распакованном формате для каждой десятичной цифры отводится по целому байту, при этом старшие полбайта (зона) каждого байта кроме самого младшего заполняются кодом 0011, а в младших полбайтах обычным образом кодируются десятичные цифры. Старший полубайт (зона) самого младшего (правого) байта используется для кодирования знака числа. Структура поля распакованного формата:

|Зона|Цифра|Зона|...|Знак|Цифра|.

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

Например, число $-193_{10} = -000110010011_{2-10}$ в упакованном формате имеет следующий вид: |0001|1001|0011|1101|. В распакованном формате оно запишется так: |0011|0001|0011|1001|1101|0011|. Сложение чисел в двоично-десятичном представлении выглядит так:

$$123 + 345 = 468, \quad 000100100011 + 001101000101 = 010001101000.$$

9. Кодирование команд. Машинная команда — это элементарная инструкция, выполняемая ЭВМ автоматически без дополнительных указаний. Она состоит из двух частей: операционной и адресной. В операционной части команды записан ее код, а в адресной — один или несколько адресов ячеек памяти, в которых хранятся операнды (числа, участвующие в операции). Различают безадресные, одно-, двух- и трехадресные команды.

Структура трехадресной команды: |КОП|A1|A2|A3|, где КОП — код операции, A1, A2 — адреса ячеек, содержащих первый и второй опе-

ранды, $A3$ — адрес ячейки, куда следует поместить результат выполнения операции. Структура двух- и одноадресной команды имеет вид: $[КОП|A1|A2]$, $[КОП|A1]$, где $A1, A2$ — адреса ячеек ОЗУ, где хранятся операнды, и куда должен быть записан результат операции. Безадресная команда содержит только код операции, а информация для нее должна быть заранее помещена в определенные ячейки памяти. Команда $[СЛ|0103|5102]$ расшифровывается так: "сложить число, записанное в ячейке 0103 ОЗУ, с числом, записанным в ячейке 5102, а затем результат (то есть сумму) поместить в ячейку 0103."

Команды могут кодироваться в двоичной, шестнадцатиричной и мнемонической (символической) формах. ЭВМ понимает только двоичную форму, при этом средняя по сложности программа состоит из тысяч бит. Для сокращения записи используют шестнадцатиричный код. Но это тоже неудобно: программист должен знать коды всех команд (около 100). Поэтому применяется мнемонический (символический) код: каждая команда представлена совокупностью букв — сокращений от английских названий команд. Символическая программа транслируется ЭВМ в двоичный код и выполняется. На языке Assembler команды *mov ax, bx* и *add ax, bx* помещают в регистр микропроцессорной памяти *ax* содержимое ячейки *bx*, после чего складывают содержимое регистров *ax* и *bx* с последующей записью суммы в *ax*.

10. Кодирование графической и видео- информации. Форму представления на экране дисплея графического изображения, состоящего из отдельных точек (пикселей), называют *растровой*. Минимальным объектом в растровом графическом редакторе является точка (пиксель — picture element). Разрешение монитора (количество пикселей по горизонтали и вертикали), а также число возможных цветов каждой точки определяются типом монитора. Например: $640 \cdot 480 = 307\,200$ точек, $800 \cdot 600 = 480\,000$ точек. Один пиксель черно-белого изображения кодируется одним битом информации. Количество различных цветов K и битовая глубина (число разрядов, используемых для кодировки цвета) b связаны формулой: $K = 2^b$.

Согласно *трехкомпонентной теории цвета*, предложенной М. В. Ломоносовым, существуют три линейно-независимых цвета, ни один из которых не может быть получен путем смешивания остальных. Сформулируем *закон трехмерности*: с помощью трех линейно-независимых цветов можно получить любой цвет.

В модели RGB основные цвета: красный, зеленый, синий. Если основным цветом кодируется 1 битом (0 или 1), то всего получается 8 оттенков: (000, 100, ..., 111), каждому из которых соответствует вершина единичного трехмерного куба (рис. 5.1.3). В видеорежиме True Color каждый цвет кодируется 1 байтом, поэтому битовая глубина — 24 бита = 3 байта. При печати используется модель CMYK (Cyan — Magenta — Yellow — Black), основные цвета: голубой, пурпурный, желтый, черный. В видеозаписи используется модель HSB (Hue — Saturation — Brightness).

Цветовая палитра монитора зависит от информационной емкости (битовой глубины) одного пикселя: 4 бита — 16 цветов, 8 бит — 256 цветов. Объем памяти, необходимой для хранения графического изображения, занимающего весь экран, равен произведению количества пикселей (разрешения экрана) $N \cdot M$ на число бит C , кодирующих одну точку. То есть объем графического файла в битах составляет: $V = N \cdot M \cdot C$.

Например, при разрешении $640 \cdot 480$ и количестве цветов 16 (4 бита) объем памяти равен: $640 \cdot 480 \cdot 4 = 1228800$ бит или $1228800/8/1024 = 150$ Кбайт. Объемы видеопамати для мониторов с различным разрешением и цветовой палитрой представлены ниже.

| Бит/пиксель (число цветов) | 4 бита ($2^4 = 16$) | 8 бит ($2^8 = 256$) | 16 бит (2^{16} = 65536) | 24 бита (2^{24} = 16777216) |
|-------------------------------|--------------------------|--------------------------|-------------------------------|-----------------------------------|
| 640*480 | 150 Кбайт | 300 Кбайт | 600 Кбайт | 900 Кбайт |
| 800*600 | 234,4 Кбайт | 468,8 Кбайт | 937,6 Кбайт | 1,4 Мбайт |
| 1024*768 | 384 Кбайт | 768 Кбайт | 1,5 Мбайт | 2,25 Мбайт |
| 1280*1024 | 640 Кбайт | 1,25 Мбайт | 2,5 Мбайт | 3,75 Мбайт |

Ввод и хранение в ЭВМ технических чертежей, состоящих из отрезков, дуг, окружностей, осуществляется в *векторной форме*. Минимальной единицей, обрабатываемой векторным графическим редактором, является объект: отрезок, прямоугольник, круг, дуга и т.д. Для каждого объекта указываются его параметры и тип линии: тонкая, штрихпунктирная и т.д. Хранение информации в векторной форме на несколько порядков сокращает необходимый объем памяти по сравнению с растровой.

Видеокадр представляет собой последовательность кадров. Число кадров вычисляется как произведение длительности видеоклипа Δt на скорость кадров v , то есть их количество в 1 с: $V = N \cdot M \cdot C \cdot v \cdot \Delta t$. При разрешении $800 \cdot 600$ точек, разрядности цвета $C=16$, скорости кадров $v = 25$ кадров/с, видеоклип длительностью 30 с будет иметь объем: $V = 800 \cdot 600 \cdot 16 \cdot 25 \cdot 30 = 576 \cdot 10^7$ (бит) = $72 \cdot 10^7$ (байт) = 687 (Мбайт).

11. Кодирование дискретных и аналоговых сигналов. При *оцифровке* или *дискретизации* аналогового сигнала $X(t)$ происходит замена непрерывной функции $u = u(t)$, выражающей зависимость напряжения от времени в интервале $[t_1, t_2]$, ломанной с большим количеством маленьких звеньев (ступенек). Это осуществляется путем развертки по времени и квантования по величине. Длительность ступенек равна шагу квантования по времени $\Delta t = 1/f$, где f — частота отсчетов. Высота ступенек кратна шагу квантования по величине ΔU . Каждой ступеньке напряжения ставится в соответствие двоичный код, в результате чего получается массив (x_1, x_2, \dots, x_n) .

По *теореме Фурье* любая периодическая функция $u = u(t)$ может быть представлена в виде суммы бесконечного числа гармонических колебаний (гармоник), с частотами $\omega, 2\omega, 3\omega, \dots$ кратными основной частоте $\omega = 2\pi/T$: $u(t) = a_0 + A_1 \sin(\omega t + \varphi_1) + A_2 \sin(2\omega t + \varphi_2) + A_3 \sin(3\omega t + \varphi_3) + \dots$ Так, сигнал с микрофона, имеющий основную частоту 1 кГц, расклады-

вается на гармоники с частотами 1 кГц, 2 кГц, 3 кГц и т.д.

Можно доказать справедливость следующих теорем: *Теорема отсчетов Котельникова*: Непрерывный сигнал может быть оцифрован и воссоздан без потерь, если шаг развертки по времени Δt не превышает половину периода наивысшей гармоники сигнала: $\Delta t \leq T_{min}/2 = 1/(2f_{max})$. *Теорема отсчетов Найквиста – Шеннона*: Сигнал $u(t)$, наивысшая гармоника которого имеет частоту f_{max} , после дискретизации с частотой отсчетов $F = 1/\Delta t > 2f_{max}$ может быть точно восстановлен из дискретных значений по формуле:

$$u(t) = \sum_{k=-\infty}^{\infty} \frac{u(k\Delta t)}{2\pi F} \frac{\sin(2\pi F \cdot (t - k\Delta t))}{t - k\Delta t}.$$

Чтобы понять суть этой формулы, проанализируем график функции

$$\xi_k(t) = \frac{\sin(2\pi F(t - k\Delta t))}{2\pi F(t - k\Delta t)},$$

приведенный на рис. 5.3.1. Видно, что он представляет собой последовательность острых максимумов, соответствующих моментам $t = i\Delta t$, $i = 0, \pm 1, \pm 2, \dots$. Про других t значение $\xi_k(t)$ близко к 0.

Из рассмотренных выше теорем следует, что для оцифровки речевого сигнала с частотой до 5 кГц, частота отсчетов должна быть не менее 10 кГц. Рис. 5.3.2 соответствует ситуации, когда частота отсчетов $F = 1/\Delta t$ больше удвоенной частоты сигнала $2f$, при этом сигнал может быть точно восстановлен по форме. Если частота отсчетов меньше $2f$, то отсчеты происходят слишком редко, сигнал восстановить невозможно (рис. 5.3.3).

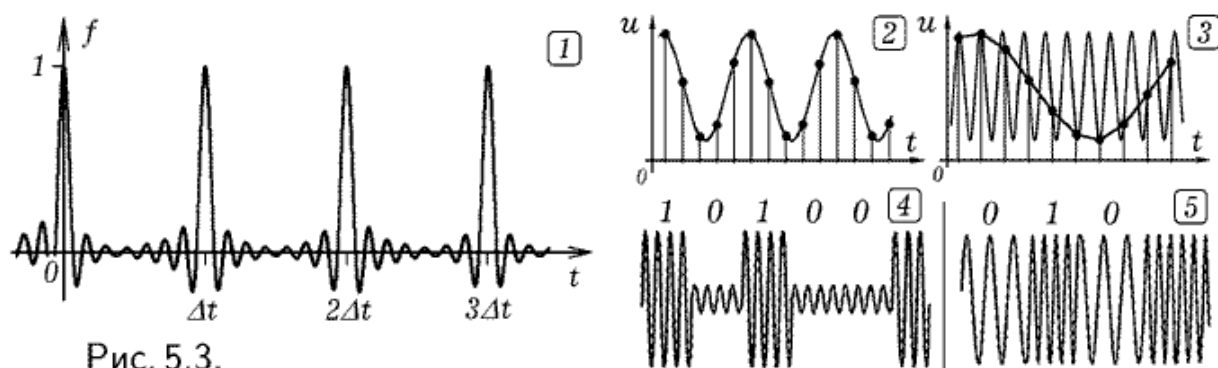


Рис. 5.3.

Теорема квантования по величине: Для качественной оцифровки и воспроизведения непрерывного сигнала достаточно, чтобы шаг квантования по величине ΔU был меньше чувствительности приемного устройства δU . Если глаз человека (приемник) различает 16 миллионов цветовых оттенков, то при квантовании цвета не имеет смысла делать больше градаций.

Оцифровка аналогового сигнала используется в цифровых магнитофонах, фото- и видеокамерах, различных устройствах для хранения и передачи информации. Преимущества цифрового представления сигнала: высокая помехоустойчивость, универсальность, простота и надежность устройств, обрабатывающих информацию. Копия аудио- или видеофайла

всегда идентична оригиналу (несет то же количество информации). При копировании аналогового сигнала часть информации теряется.

Для записи и обработки аналогового сигнала с выхода микрофона осуществляется его оцифровка с помощью аналого-цифрового преобразователя. При этом они заменяются цифровыми значениями отдельных выборок, взятых через достаточно короткие промежутки времени Δt . То есть непрерывный сигнал квантуется по величине и по времени, в результате чего получается поток логических 0 и 1, которые обрабатывает процессор. При 16-разрядном амплитудном кодировании непрерывный сигнал разбивается на $2^{16} = 65536$ ступеней квантования. В соответствии с теоремой отсчетов, частота квантования 44,1 кГц обеспечивает воспроизводимость всего звукового диапазона частот (до 20 кГц). При этом объем файла в битах равен $V = f \cdot r \cdot t$, где f_k — частота квантования, r_k — разрядность квантования по величине (число ступенек), t — время записи. Оцифрованный сигнал может быть преобразован в аналоговый с помощью цифро-аналогового преобразователя.

Аналоговое кодирование цифрового сигнала применяется при передаче данных по телефонным линиям связи и использует три способа модуляции:

1. Амплитудная модуляция: в зависимости от передаваемых битов изменяется амплитуда синусоидальных колебаний, например, лог. 1 соответствует большая амплитуда, а лог. 0 отвечает маленькая амплитуда или отсутствие колебаний (рис. 5.3.4).

2. Частотная модуляция: в соответствии с передаваемыми битами изменяется частота колебаний: лог. 1 — высокая частота, лог. 0 — низкая (рис. 5.3.5).

3. Фазовая модуляция: при переходе от логической 1 к логическому 0 и наоборот фаза колебаний изменяется на 180° .

5.2. ЭЛЕМЕНТЫ ТЕОРИИ КОДИРОВАНИЯ

1. Первая теорема Шеннона. Информация от источника кодируется кодером и превращается преобразователем "коды-сигналы" в последовательность сигналов, поступающих в канал связи (рис. 5.4). Преобразователь "сигналы-коды" и декодер осуществляют обратное преобразование. На канал связи действуют *шумы* — помехи, искажающие передаваемый сигнал, что компенсируется защитой от шумов.

Первичным называется алфавит, используемый в сообщении, которое выдает источник. Кодер преобразует сообщение, используя *вторичный* алфавит (сигналы). Пусть первичный алфавит A содержит N знаков, а вторичный B — M знаков (сигналов). На каждый знак первичного и вторичного алфавита приходится средняя информация I_A и I_B .

Исходное сообщение содержит n знаков, а закодированное — m знаков (сигналов), они несут количество информации $I(A) = nI_A$ и $I(B) = mI_B$. Кодирование будет обратимым, при условии неисчезновения информации, когда $I(A) \leq I(B)$. Можно сказать, что количество информации при обратимом кодировании не уменьшается. Тогда $I_A \leq (m/n)I_B = K(A, B)I_B$, где $K(A, B) = m/n$ — средняя длина кода, то есть среднее число знаков вторичного алфавита B , используемых для кодирования одного знака первичного алфавита A . Итак, длина кода $K(A, B) \geq I_A/I_B$, а ее минимальное значение $K^{min}(A, B) = I_A/I_B$. Обычно $I_A > I_B$, поэтому $K(A, B) > 1$, то есть одному знаку первичного алфавита соответствует несколько знаков вторичного. Относительная избыточность кода равна

$$Q(A, B) = \frac{K(A, B) - K_{min}(A, B)}{K_{min}(A, B)} = \frac{K(A, B)}{K_{min}(A, B)} - 1.$$

Проблема оптимизации кода состоит в нахождении метода кодирования, при котором средняя длина кода стремится к своему пределу: $K(A, B) \rightarrow K^{min}(A, B)$. При этом избыточность $Q \rightarrow 0$, операция кодирования становится более эффективной. Решение этой проблемы позволит затратить на передачу сообщения меньше энергии и времени, а при хранении информации использовать меньше поверхности носителя.

Первая теорема Шеннона о кодировании при отсутствии помех: При отсутствии помех возможно кодирование, при котором избыточность кода будет сколь угодно близка к нулю. (Возможно кодирование, при котором среднее число знаков вторичного алфавита B , приходящееся на 1 знак первичного алфавита A , будет сколь угодно близким к отношению количеств информации, приходящихся на знаки первичного и вторичного алфавитов: $m/n \rightarrow I_A/I_B$).

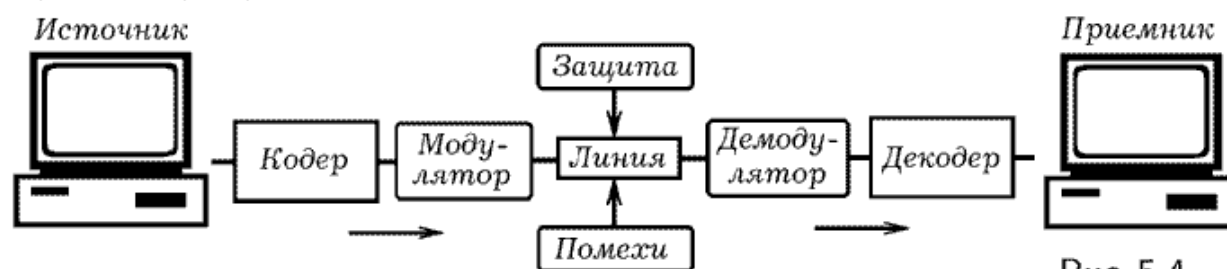


Рис. 5.4.

Из теоремы Шеннона для канала связи без помех вытекают следствия:

1. При передаче информации по сети, кодировании, сжатии без потерь среднее число бит, соответствующее одному символу первичного алфавита, не может быть меньше энтропии одного символа этого алфавита. Если вторичный алфавит имеет 2 знака (двоичный код), то в случае оптимального кодирования средняя длина двоичного кода должна быть равна среднему информационному содержанию знака первичного алфавита.
2. Возможен такой способ кодирования, что среднее число сигналов (символов алфавита B), затрачиваемых на передачу одной буквы сообщения в алфавите A , было бы сколь угодно близко к $s = \log(n)/\log(m) = \log_m(n)$, но не меньше этой величины.

3. При использовании любого метода кодирования m -ичным кодом среднее число s электрических сигналов, приходящихся на 1 букву сообщения не может быть меньше $s_{min} = H / \log_2 m$, где H — энтропия или информация в 1 букве. Если кодировать сразу длинные блоки из N букв величина s может быть сколь угодно близка к s_{min} .

Перечислим характеристики канала связи: амплитудно-частотная характеристика (АЧХ — зависимость коэффициента передачи от частоты), полоса пропускания (диапазон частот, на границах которого коэффициент передачи в $\sqrt{2}$ меньше максимального), коэффициент передачи $K = U_{вых} / U_{вх}$, затухание, пропускная способность, достоверность передачи данных (вероятность ошибки), помехоустойчивость.

2. Вторая теорема Шеннона. На реальный канал связи действуют помехи, в результате чего при передаче информации возникают ошибки. *Уровень достоверности* передаваемого сообщения может быть определен как отношение числа ошибок к общему количеству знаков: $D = n_{ош} / n$.

Пусть L — скорость передачи сигналов по каналу связи (в сигнал/с). Если источник вырабатывает m различных сигналов (уровней напряжения) с равными вероятностями, то каждый сигнал несет $\log_2 m$ бит информации. Пропускная способность (емкость) канала, то есть наибольшее количество информации, которая проходит по каналу за 1 с, равна $C = L \log m$. Если H — энтропия одной буквы, то максимальная скорость передачи сообщения $v_{max} = L \log m / H$ (букв/с). Если в канале связи различимы m состояний (колебаний, уровней напряжения) длительностью T , то скорость передачи равна $\log_2 m / T$ в бит/с.

Пусть вторичный алфавит B имеет m символов (сигналов), а первичный алфавит A — $n = m^s$ букв. Тогда для передачи одной буквы первичного алфавита достаточно $s = \log(n) / \log(m) = \log_m(n)$ символов (сигналов) вторичного алфавита. Например, в алфавите A $n = 16$ символов, встречающихся с равными вероятностями, а в алфавите B — $m = 2$. Тогда один символ алфавита A следует кодировать 4 символами из алфавита B $s = \log(16) / \log(2)$. Такой код используется при двоичном кодировании цифр $0, 1, \dots, E, F$ шестнадцатеричной системы счисления.

Вторая теорема Шеннона о кодировании при наличии помех: Если скорость передачи не превышает пропускной способности канала связи с шумом, то всегда найдется способ кодирования, при котором сообщение будет передаваться с требуемой достоверностью.

Сформулируем *обратную теорему кодирования*: если производительность источника R (в бит/с) превышает емкость канала связи C , то не существует никакого метода кодирования, позволяющего передать информацию с какой угодно малой вероятностью ошибки.

Пропускная способность канала с шумом зависит от действующих помех. Пусть имеется двоичный симметричный канал с вероятностью правильной передачи p и вероятностью ошибки $q = 1 - p$ (рис. 5.6.1). Пропускная способность канала связи при передаче одного символа в секунду равна $C = 1 + p \log_2 p + (1 - p) \log_2 (1 - p)$. График этой зависимости пред-

ставлен на рис. 5.5.1. Если вероятность ошибки равна $p = 0,5$, то есть в каждом втором символе ошибка, то пропускная способность равна $C = 0$.

Допустим, R — производительность источника сообщения, то есть среднее количество информации (в бит/с), вырабатываемое источником в единицу времени, а C — пропускная способность канала с шумом (в бит/с). Если $R > C$, то не существует системы кодирования, позволяющей передать сообщение по каналу связи с шумом, так как источник будет вырабатывать информацию быстрее, чем ее способен передать канал связи. При $R = C$ передача возможна при отсутствии помех.

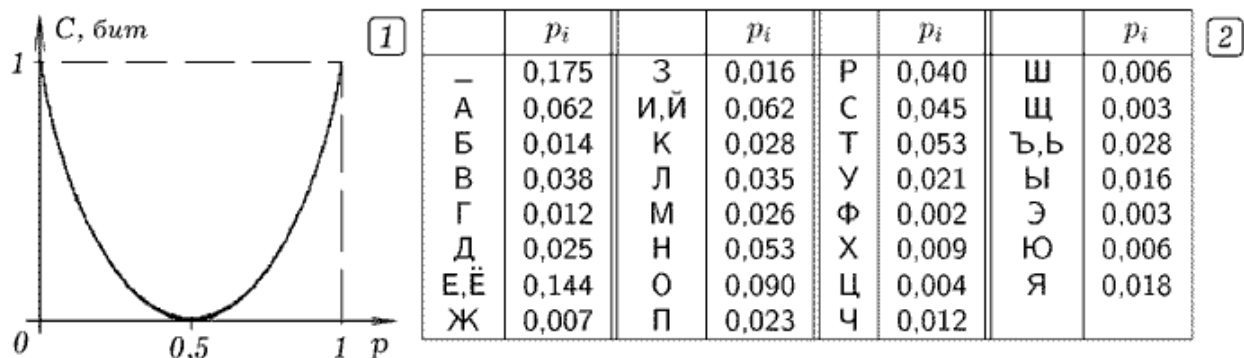


Рис. 5.5.

3. Формула Хартли–Шеннона. Максимальная скорость передачи информации по каналу связи в бит/с называется *емкостью канала* $C = \Delta I_{\max} / \Delta t$. Пропускная способность зависит от емкости канала и способа физического кодирования (то есть какие последовательности сигналов отвечают тому или иному символу). Если источник без памяти выдает последовательность символов из алфавита, содержащего m букв, причем все символы появляются с равной вероятностью, то на один символ приходится информация $I_1 = \log_2 m$. Пусть длина сообщения равна N , тогда в нем содержится информация $\Delta I = N \log_2 m$. Это справедливо и в случае, когда аналоговый сигнал оцифровывается так, что получается N сигналов с m равновероятными дискретными состояниями.

При передаче сигнала, частота которого находится в диапазоне от 0 до f , необходимо, чтобы частота дискретизации была равна $2f$. Тогда за время Δt произойдет $N = 2f \Delta t$ отсчетов.

При отсутствии шума число дискретных состояний стремится к бесконечности. При наличии помех различие в мощности (амплитуде) полезного сигнала должна быть сравнима с амплитудой (мощностью) шума. Поэтому число различных по мощности уровней сигнала равно $(P_c + P_{\text{ш}}) / P_{\text{ш}}$, а по амплитуде $m = \sqrt{(P_c + P_{\text{ш}}) / P_{\text{ш}}}$. Емкость канала

$$C = \frac{\Delta I}{\Delta t} = \frac{N \log_2 m}{\Delta t} = \frac{1}{\Delta t} 2f \Delta t \log_2 \sqrt{\frac{P_c + P_{\text{ш}}}{P_{\text{ш}}}} = f \log \left(1 + \frac{P_c}{P_{\text{ш}}} \right).$$

Эта формула Хартли–Шеннона. Из нее следует, что уменьшение полосы пропускания f и увеличение мощности помех $P_{\text{ш}}$ приводят к уменьшению емкости канала, снижению его пропускной способности.

Допустим, аналоговый сигнал дискретизируется с частотой отсчетов $f_1 = 12$ кГц. Каждый отсчет квантуется на 256 уровней и кодируется символами из $N = 8$ бит, которые встречаются одинаково часто. Возможно ли передать закодированный сигнал по каналу связи с белым шумом с полосой пропускания $f_2 = 8,0$ кГц и отношением сигнал/шум $\beta = 15$ дБ со сколь угодно малой вероятностью ошибки? Какова минимальная ширина полосы пропускания, при которой возможна передача информации со сколь угодно высокой достоверностью?

Информационный поток равен $I = Nf_1 = 8 \cdot 12000 = 96000 = 9,6 \cdot 10^4$ бит/с. По условию задачи $\beta = 10 \log_{10}(P_c/P_{\text{ш}}) = 15$ дБ, следовательно $P_c/P_{\text{ш}} = 31,6$. Пропускная способность канала равна

$$C = f_2 \log_2 \left(1 + \frac{P_c}{P_{\text{ш}}} \right) = 8000 \log_2(1 + 31,6) = 4,02 \cdot 10^4 \text{ бит/с.}$$

Так как информационный поток I превышает емкость C канала связи, то не существуют способа кодирования, который позволил бы передать информацию со сколь угодно высокой достоверностью.

Найдем минимальную ширину полосы пропускания f_{\min} , при которой в принципе возможна безошибочная передача информации при соответствующем способе кодирования: $C = I = f_{\min} \log_2(1 + P_c/P_{\text{ш}})$, $f_{\min} = I / \log_2(1 + P_c/P_{\text{ш}}) = 9,6 \cdot 10^4 / \log_2(32,6) = 19,2 \cdot 10^3$ Гц.

4. Передача информации по каналу с шумом. Допустим, на выходе линии связи декодер воспринимает сигналы $X = (x_1, x_2, \dots, x_m)$. По ним он должен восстановить сигналы, поступающие на вход линии связи и образующие множество $Y = (y_1, y_2, \dots, y_n)$. Условная вероятность $p(y_i/x_j)$ (события y_i при x_j) — это вероятность поступления сигнала y_i в линию связи, если на ее конце появляется x_j . Совместная вероятность одновременного наступления двух событий $X = x_j$ и $Y = y_i$ равна $p(x_j, y_i) = p(x_j)p(y_i/x_j)$. Чтобы найти безусловную вероятность $p(y_i)$ необходимо условную вероятность усреднить по всевозможным значениям x_1, x_2, \dots, x_m :

$$p(y_i) = \sum_{j=1}^m p(x_j)p(y_i/x_j) = \sum_{j=1}^m p(x_j, y_i).$$

По формуле Шеннона априорная (первоначальная) неопределенность сигнала Y характеризуется энтропией $H(Y)$:

$$H(Y) = - \sum_{i=1}^n p(y_i) \log_2 p(y_i) = - \sum_{i=1}^n \sum_{j=1}^m p(x_j)p(y_i/x_j) \log_2 p(y_i).$$

Пусть на выходе канала связи появился сигнал x_j . Это позволяет утверждать, что с вероятностью $p(y_i/x_j)$ на вход канала связи поступил символ y_i , где $i = 1, 2, \dots, n$. Апостериорная неопределенность (энтропия) сигнала Y после приема сигнала x_j :

$$H_{x_j}(Y) = - \sum_{i=1}^n p(y_i/x_j) \log_2 p(y_i/x_j),$$

где $p(y_1/x_j), p(y_2/x_j), \dots, p(y_n/x_j)$ — вероятности поступления сигналов y_1, y_2, \dots, y_n на вход линии, если на ее выходе сигнал x_j . Если бы принятый сигнал X однозначно соответствовал посланному Y , то есть всегда при $X = x_j$ на входе линии $Y = y_i$, то условная вероятность $p(y_i/x_j) = 1$ и энтропия $H_{x_j}(Y) = 0$. Принятый сигнал X может иметь значения x_1, x_2, \dots, x_m с вероятностью $p(x_1), p(x_2), \dots, p(x_m)$, поэтому средняя условная энтропия посланного сигнала Y при известном принятом X равна:

$$H(Y/X) = \sum_{j=1}^m p(x_j) H_{x_j}(Y) = - \sum_{j=1}^m \sum_{i=1}^n p(x_j) p(y_i/x_j) \log_2 p(y_i/x_j).$$

Условная энтропия не превышает безусловную: $H(Y/X) \leq H(Y)$. Равенство $H(Y/X) = H(Y)$ соответствует статистически независимым событиям X, Y , когда получение сигнала x_j совершенно не означает, что был послан сигнал y_i и $p(y_i/x_j) = p(y_i)$. Для того, чтобы вычислить среднее количество информации, содержащееся в принимаемых сигналах X о посылаемых сигналах Y , необходимо определить разность между безусловной $H(Y)$ и условной $H(Y/X)$ энтропиями:

$$\begin{aligned} I(X, Y) &= H(Y) - H(Y/X) = \\ &= - \sum_{j=1}^m \sum_{i=1}^n p(x_j) p(y_i/x_j) \log_2 p(y_i) + \sum_{j=1}^m \sum_{i=1}^n p(x_j) p(y_i/x_j) \log_2 p(y_i/x_j) = \\ &= \sum_{j=1}^m \sum_{i=1}^n p(x_j, y_i) (\log_2 \frac{p(x_j, y_i)}{p(x_j)} - \log_2 p(y_i)) = \\ &= \sum_{j=1}^m \sum_{i=1}^n p(x_j, y_i) \log_2 \frac{p(x_j, y_i)}{p(x_j) p(y_i)}. \end{aligned}$$

Из формулы видно, что количество информации неотрицательно и симметрично по отношению к перестановке X и Y . Если эти сигналы независимы, то $p(x_j, y_i) = 0$ и количество информации $I(X, Y) = 0$. В случае, когда принятый сигнал однозначно соответствует посланному, энтропия посланного сигнала при известном принятом $H(Y/X) = 0$ и количество информации достигает максимума $I(X, Y) = H(Y)$.

Рассмотрим симметричный двоичный канал на вход которого поступают двоичные сигналы $y_1 = 0$ и $y_2 = 1$ с равными вероятностями $p(y_1) = p(y_2) = 0,5$. Если вероятность правильной передачи 0,8, а вероятность ошибки 0,2, то $p(x_1) = p(x_2) = 0,5$, $p(y_1/x_1) = p(y_2/x_2) = 0,8$ и $p(y_2/x_1) = p(y_1/x_2) = 0,2$. Энтропия посланного сигнала $H(Y) = 1$ бит. Условная энтропия сигнала Y при принятом сигнале X :

$$H(Y/X) = -2(0,5 \cdot 0,8 \cdot \log_2 0,8 + 0,5 \cdot 0,2 \cdot \log_2 0,2) = 0,722 \text{ бит.}$$

Поэтому среднее количество информации, содержащееся в одном символе выходного сигнала, равно: $I = H(Y) - H(Y/X) = 1 - 0,722 = 0,278$ бит.

5. Помехоустойчивое кодирование. Рассмотрим равномерный код, в котором каждая из 32 букв кодируется 5 битами ($2^5 = 32$). Если в результате ошибки инвертируется один или несколько бит, то приемник вместо одной разрешенной кодовой последовательности получит другую, соответствующую другой букве, поэтому ошибка обнаружена не будет. Чтобы создать помехоустойчивый код, необходимо в передаваемую кодовую последовательность внести избыточные биты в соответствии с правилами, известными источнику и приемнику сообщения. В результате ошибки должен получаться неразрешенный код, отступление от правила, что обнаруживается приемником. Он может потребовать повторной передачи, либо найти и автоматически исправить ошибку. Возможно сразу предусмотреть передачу сообщения по нескольким каналам связи с последующим сравнением.

Повысить помехоустойчивость возможно за счет избыточности кода. Если продублировать каждый бит $1 \rightarrow 11$, $0 \rightarrow 00$, то можно выявить ошибку, но нельзя ее исправить. Для исправления одной ошибки, необходимо утроить каждый бит: $1 \rightarrow 111$, $0 \rightarrow 000$. Чтобы декодер автоматически исправлял две ошибки, каждый бит следует повторить 5 раз и т.д. Итак, для исправления m ошибок каждый бит необходимо повторить $2m + 1$ раз.

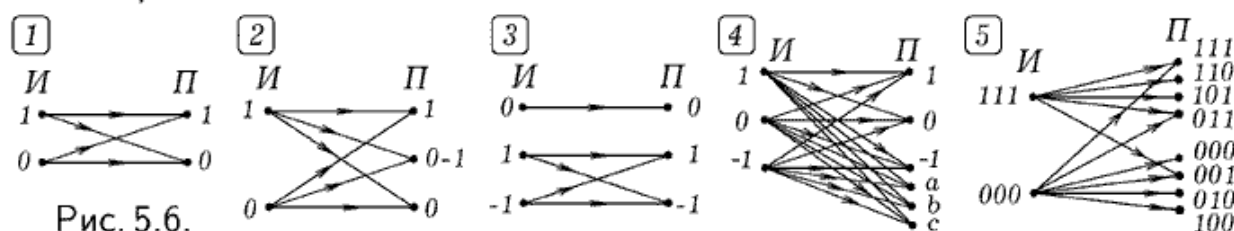


Рис. 5.6.

Перечислим наиболее типичные случаи передачи информации по каналу связи с шумом (рис. 5.6):

1. Двоичная симметричная линия. По линии передаются сигналы 0 и 1, вероятность ошибки одинакова и равна p (рис. 5.6.1).

2. Двоичная симметричная линия со стиранием. На конце линии связи принятый сигнал с вероятностью q так искажается, что нельзя определить это логический 0 или 1. Обычно q больше вероятности ошибки p , при котором 0 инвертируется в 1 или наоборот (рис. 5.6.2).

3. По линии связи можно передавать три сигнала (0, 1, -1), причем один из них может быть передан без ошибок (рис. 5.6.3).

4. В общем случае имеется разрешенная совокупность сигналов, каждому из которых соответствует символ 1, 0 или -1. За счет помех с некоторой вероятностью p из одного разрешенного сигнала может получиться другой разрешенный сигнал (1, 0, -1) и с вероятностью q — неразрешенный сигнал (a, b, c), которому не соответствует ни один символ (рис. 5.6.4). В первом случае система не распознает ошибки, а во втором — распознает.

5. Используется самокорректирующийся код, исправляющий одну ошибку, в котором каждый символ 0 и 1 повторяется 3 раза. Кодовые последовательности 001, 010, 100 воспринимаются как 000, а последовательности 110, 101, 011 — как 111 (рис. 5.6.5).

Рассмотрим код с проверкой четности. Кодирование состоит в том, что поток 0 и 1 разделяется на кадры по n бит (a_1, a_2, \dots, a_n) и после каждого кадра вставляется $n + 1$ бит в соответствии с правилом:

$$K(a_1, a_2, \dots, a_n) \rightarrow (a_1, a_2, \dots, a_n, a_{n+1}),$$

$$\text{где } a_{n+1} = \begin{cases} 0, & \text{если } a_1 + a_2 + \dots + a_n \text{ четно;} \\ 1, & \text{если } a_1 + a_2 + \dots + a_n \text{ нечетно.} \end{cases}$$

При этом $a_1 + a_2 + \dots + a_n + a_{n+1}$ четно. В конце канала связи происходит декодирование в соответствии с правилом:

$$D(a_1, a_2, \dots, a_n, a_{n+1}) = \begin{cases} (a_1, a_2, \dots, a_n), & \text{если } a_1 + \dots + a_{n+1} \text{ четно;} \\ \text{ошибка,} & \text{если } a_1 + \dots + a_{n+1} \text{ нечетно.} \end{cases}$$

Если ошибки взаимно независимые, то вероятность инвертирования любых r битов в n -разрядном двоичном слове равна: $p_r = C_n^r p^r (1-q)^{n-r}$, где C_n^r — число сочетаний r из n . Например, при кодировании рассмотренным выше методом в случае $n = 2$ получается так: $00 \rightarrow 000$, $01 \rightarrow 011$, $10 \rightarrow 101$, $11 \rightarrow 001$. При передаче сообщения, закодированного этим кодом, по двоичному симметричному каналу связи с вероятностью правильной передачи p и вероятностью инверсии бита q возможны следующие варианты: 1) все три бита инвертированы, — вероятность $p_3 = q^3$; 2) два любых бита инвертированы, один бит передан правильно, — вероятность $p_2 = 3pq^2$; 3) два бита переданы верно, в одном бите ошибка, — вероятность $p_1 = 3p^2q$. Коэффициент 3 в последнем случае появляется из-за того, что с вероятностью p^2q ошибка может быть в первом бите, с той же вероятностью — во втором бите, и с той же вероятностью — в третьем бите. Совершенно аналогично во втором случае. Декодер не заметит ошибки в двух битах, так как они не изменяют четности; вероятность такого события $3pq^2$.

Если не утраивать каждый бит, то вероятность передачи сообщения из двух бит с одной ошибкой равна $2pq$, а с двумя ошибками — q^2 . Если канал связи достаточно хорош, то $q \ll 1$, и $3pq^2 \ll 2pq + q^2$. Например, при $p = 0,99$ и $q = 1 - p = 0,01$, получается $3pq^2 = 0,000297$ и $2pq + q^2 = 0,0199$. Утроение каждого бита снижает вероятность ошибки в ≈ 70 раз.

6. Поиск оптимального кода. Расстояние Хэмминга. Проблема кодирования состоит в нахождении помехоустойчивого кода, при использовании которого длина закодированного сообщения минимальна, что позволяет затратить на передачу сообщения меньше энергии и времени, а при хранении требует меньших объемов памяти. Проблема декодирования: приемник получает кодовые последовательности символов (сигналов), некоторые из которых не совпадают ни с одной из разрешенных последовательностей; необходимо раскодировать сообщение.

Геометрической моделью равномерного двоичного кода длины n является n -мерный единичный куб E^n . Совокупность двоичных последовательностей длины n ($101 \dots 1, \dots, 111 \dots 1$) можно представить как множество из 2^n вершин единичного куба E^n в n -мерном пространстве. Степень близости двух кодовых последовательностей $\bar{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ и $\bar{\beta} = (\beta_1, \beta_2, \dots, \beta_n)$, где α_i и β_i равны 0 или 1, измеряется расстоянием Хемминга:

$$\rho(\bar{\alpha}, \bar{\beta}) = \sum_{i=1}^n |\alpha_i - \beta_i|.$$

Например, если $\bar{\alpha} = (1, 1, 0, 1, 0, 1)$, $\bar{\beta} = (1, 0, 0, 1, 1, 1)$, то $\rho(\bar{\alpha}, \bar{\beta}) = 2$. Расстояние между любыми вершинами E^n не меньше 1 не больше n .

Шар $S_{\bar{\alpha}}^t$ радиуса t с центром в точке $\bar{\alpha}$ из куба E^n — это множество точек из E^n , находящихся от α на расстоянии не превышающем t . Если $\bar{\alpha} = (0, 0, 0)$ и $t = 1$, то шар $S_{\bar{\alpha}}^t$ образован точками $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$. Общее количество точек в шаре равно $S(n, t) = 1 + C_n^1 + C_n^2 + \dots + C_n^t$, где $C_n^k = n!/(k!(n-k)!)$ — число сочетаний k из n . Множество точек n -мерного куба E^n , равноудаленных на расстояние t от центра $\bar{\alpha}$ называется сферой $O_{\bar{\alpha}}^t$.

Целесообразность кода $Q = (\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_m)$ для передачи по каналу с шумом зависит от возможности этого кода обнаруживать и исправлять ошибки. Код обнаруживает l ошибок, если изменение в любом кодовом слове $\bar{\alpha}_i$ не более чем l символов приводит к слову $\bar{\beta}_i$, которое не входит в множество Q . После этого приемник требует повтора сообщения (передача по каналу с шумом при наличии обратной связи).

Теорема: Код Q способен обнаружить l ошибок лишь тогда, когда расстояние Хемминга между любыми двумя кодовыми последовательностями из Q не меньше чем $l + 1$.

Чтобы построить n -разрядный код, обнаруживающий l ошибок, в n -мерном пространстве выбирают точку $\bar{\alpha}_1$, окружают ее шаром радиусом l , а точку $\bar{\alpha}_2$ выбирают за пределами этого шара. Аналогично находят остальные коды $\bar{\alpha}_3, \bar{\alpha}_4, \dots$.

Можно представить себе множество шаров радиусом r , наполняющих n -мерное пространство без самопересечений (рис. 5.7.1). Центрами этих шаров и соответствуют искомым кодам, расстояние между ними не меньше $2r + 1$. Каждый из шаров радиусом r содержит $S(n, r) = 1 + C_n^1 + C_n^2 + \dots + C_n^r$ точек. Для покрытия всех 2^n вершин куба E^n требуется не более чем $N = 2^n / S(n, r)$ шаров. Отсюда вытекают два следствия: 1. Число двоичных кодовых последовательностей длиной n , способных исправлять r ошибок, не может быть больше верхней границы Варшамова-Гильберта $N = 2^n / (1 + C_n^1 + C_n^2 + \dots + C_n^r)$ (выполняется при больших n). 2. Нижняя граница Хемминга определяет минимальное количество двоичных разрядов $r \geq \log_2(C_n^k + C_n^{k-1} + \dots + C_n^1 + 1)$, при которой расстояние между кодами составляет $2k + 1$.

Пусть расстояние между кодами равно $r = 2k$, например, 1100 и 0011,

$r = 4$. Если вследствие ошибки придет код 0110, удаленный от каждого из разрешенных кодов на 2, то приемник не сможет исправить ошибку. Для того чтобы код корректировал ошибку при инвертировании 1 или 2 бит ($k = 2$), необходимо чтобы расстояние между кодовыми последовательностями равнялось $2k + 1 = 5$. Например: 11000 и 00111.

Групповыми называются коды, которые содержат вместе с любыми двумя точками α_i и α_j их сумму, то есть точку $\bar{\alpha}_i \oplus \bar{\alpha}_j$. Например, код $Q = (000, 001, 010, 100, \dots, 011, 111)$. Их преимущество в том, что для задания группового кода нет необходимости перечислять все элементы, достаточно указать несколько из них. *Эквидистантные* коды — это коды, у которых все попарные расстояния между точками одинаковы. Например: (10000), (01000), (00100), ..., — расстояние равно 2. Или код (110000), (011000), (001100), Каждый код получается сдвигом единиц в соседний разряд. *Совершенным* называется код, соответствующий двум противоположным вершинам куба E^n : (00000), (11111). Если из-за помехи получается точка (10010), принадлежащая шару с центром (00000) и радиусом 2, то после корректировки приемник воспринимает ее как код (00000). Этот код исправляет 2 ошибки.

7. Равномерное и неравномерное кодирование. 1. *Равномерное кодирование*: каждой букве первичного алфавита A соответствует одинаковое количество символов (сигналов) алфавита B . Например, кодирование цифр 0–15 двоичным кодом 0000, 0001, ..., 1110, 1111; кодирование 32 букв алфавита пятибитовыми двоичными числами 00000, 00001, 00010, 00011, ..., 11111; байтовое кодирование всех 256 состояний клавиатуры (различные комбинации нажатых клавиш). Иногда входные буквы объединяются в блоки по k символов, каждый из которых преобразуется в блок по n символов. Получается (n, k) -код, скорость передачи которого $v_B = (k/n)v_A$. Для декодирования сообщения достаточно отсекать требуемое количество символов и сопоставлять их с алфавитом B . Равномерный код не является оптимальным, так как разные буквы первичного алфавита A встречаются с различной вероятностью (рис. 5.5.2) и кодировать их одинаковым числом бит неэкономично.

2. *Неравномерный код с разделителем знаков*. Код будет близок к оптимальному, если часто встречающиеся буквы алфавита A будут иметь короткие кодовые последовательности, а буквы, вероятность использования которых мала, — длинные. Для разделения знаков каждый код алфавита A должен оканчиваться, например, на 00. Тогда коды букв должны начинаться на 1 и не содержать 00 в середине. Например, закодируем сообщение "АБРАКАДАБРА" с помощью кода: А - 100, Р - 1100, Б - 0100, К - 11100, Д - 10100. Получим: 100 0100 1100 100 11100 100 10100 100 0100 1100 100.

3. *Префиксный код* — это неравномерный код, в котором ни одна из кодовых последовательностей не совпадает с началом (префиксом) кода другой буквы алфавита A . Можно сформулировать условие *Фано*: неравномерный код может быть однозначно декодирован, если он является префиксным. Например, если одна из букв алфавита A имеет код 001, то

нельзя использовать 0, 00, 0010, 00101 и т.д.

8. Код Шеннона-Фано. Код Хаффмана. Метод Шеннона-Фано позволяет построить префиксный код, в котором длина кодовой последовательности данной буквы алфавита A тем меньше, чем выше вероятность p_i использования этой буквы. Алгоритм кодирования состоит в следующем: 1) буквы a_1, a_2, \dots, a_6 располагают в порядке убывания вероятности p_i , $i = 1, 2, \dots, 6$; 2) разбивают буквы на две группы так, чтобы вероятности использования букв из каждой группы были бы одинаковы; 3) одной группе дают код 1, а другой — 0; 4) снова разбивают каждую группу на две подгруппы, присваивая коды 11, 10, 01, 00; 5) повторяют эти операцию, пока не закодируют все буквы.

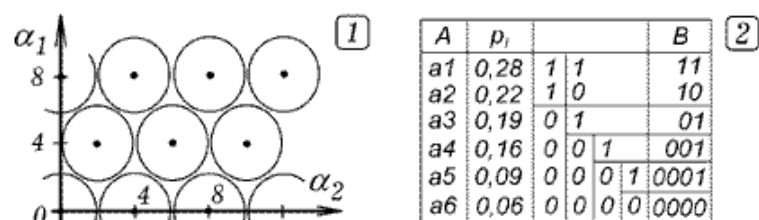
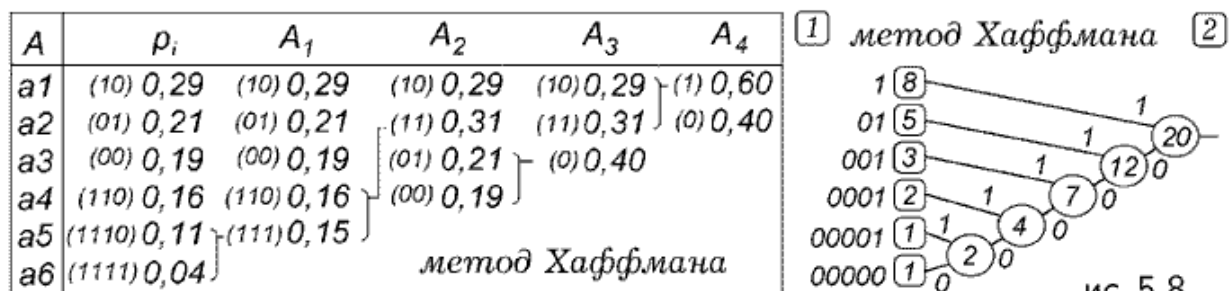


Рис. 5.7.

Пример построения префиксного кода Шеннона-Фано — на рис. 5.7.2. Оптимальная длина кода равна информации, приходящейся на одну букву: $K_{min} = I = -(0,28 \log_2 0,28 + 0,22 \log_2 0,22 + 0,19 \log_2 0,19 + 0,16 \log_2 0,16 + 0,09 \log_2 0,09 + 0,06 \log_2 0,06) = 2,429$ (бит). Поэтому в идеале каждая буква кодируется в среднем 2,429 двоичными разрядами. В этом случае средняя длина кода $K = \sum p_i k_i = 2 \cdot 0,28 + 2 \cdot 0,22 + 2 \cdot 0,19 + 3 \cdot 0,16 + 4 \cdot 0,09 + 4 \cdot 0,06 = 2,46$ (бит), избыточность кода $Q = (2,46 - 2,429)/2,429 = 0,013$. Вероятность появления 1 в закодированном сообщении равна $P_1 = (2 \cdot 0,28 + 1 \cdot 0,22 + 1 \cdot 0,19 + 1 \cdot 0,16 + 1 \cdot 0,09 + 0 \cdot 0,06)/2,46 = 1,31/2,46 = 0,496$. Вероятность 0 равна $P_0 = 1 - P_1 = 0,504$.

Префиксный код Хаффмана строится так (рис. 5.8.1): 1) располагают буквы a_1, a_2, \dots, a_6 в порядке возрастания вероятности p_i ; 2) объединяют редко встречающиеся буквы a_5, a_6 в одну, образуя промежуточный алфавит A_1 и складывая соответствующие вероятности p_5 и p_6 ; 3) повторяют эту процедуру, каждый раз объединяя редко встречающиеся буквы в одну и получая промежуточные алфавиты A_2, A_3, A_4 ; 4) последний алфавит A_4 имеет две буквы с примерно равными вероятностями, им присваивают коды 0 и 1; 5) движутся в обратном направлении, каждый раз при появлении из одной буквы двух кодируют их числами 0 и 1.





ШЕННОН Клод Эдвуд (1912–1954 гг.)

Занимался проблемой проектирования контактно-релейных схем, создал теорию кодирования и передачи информации, работал над основами теории информации и теории автоматов. Обосновал формулу для расчета энтропии, сформулировал теоремы о передаче информации по каналу связи. Установил связь между способом кодирования, скоростью передачи и вероятностью искажений.

В рассмотренном случае на одну букву приходится информация $I = -(0,29 \log_2 0,29 + 0,21 \log_2 0,21 + 0,19 \log_2 0,19 + 0,16 \log_2 0,16 + 0,11 \log_2 0,11 + 0,04 \log_2 0,04) = 2,405$ (бит), то есть при идеальном методе кодирования средняя длина кода равна $K_{min} = 2,405$ (бит). После кодирования методом Хаффмана средняя длина кода равна: $K = \sum p_i k_i = 2 \cdot 0,29 + 2 \cdot 0,21 + 2 \cdot 0,19 + 3 \cdot 0,16 + 4 \cdot 0,11 + 4 \cdot 0,04 = 2,46$ (бит), где k_i — длина кода i -ой буквы. Избыточность кода Хаффмана $Q = (2,46 - 2,405)/2,405 = 0,0229$.

Если кодирование было бы равномерным, то средняя длина кода (количество бит на 1 букву) $\log_2(6) = 2,5849$ бит, избыточность кода $Q = (2,5849 - 2,405)/2,405 = 0,0748$, то есть выше. Таким образом, код Хаффмана является более экономичным. Еще одно преимущество этого кода состоит в равновероятности 0 и 1. В коде буквы a_1 вероятность которой $p_1 = 0,29$ единица встречается 1 раз, кода буквы a_2 вероятность которой $p_2 = 0,21$ единица встречается тоже 1 раз и т.д., поэтому вероятность появления единицы в закодированном сообщении $P_1 = (1 \cdot 0,29 + 1 \cdot 0,21 + 0 \cdot 0,19 + 2 \cdot 0,16 + 3 \cdot 0,11 + 4 \cdot 0,04)/2,46 = 1,31/2,46 = 0,53$, где $K = 2,46$ — средняя длина кода.

Рассмотрим еще один пример. Пусть имеется сообщение *ABACBA EBDABABDACFACA* из 20 символов. Определяют сколько раз в данном сообщении встречаются буквы: А - 8, В - 5, С - 3, D - 2, Е - 1, F - 1. Строят двоичное дерево Хаффмана (рис. 5.8.2), и, двигаясь слева направо, на каждом шаге объединяют две редко встречающиеся буквы в одну. Двигаясь в обратном направлении, осуществляют двоичное кодирование. Закодированное сообщение 1 01 1 001 01 1 00001 01 0001 1 01 1 01 0001 1 001 000001 1 001 1 содержит $8 \cdot 1 + 5 \cdot 2 + 3 \cdot 3 + 2 \cdot 4 + 1 \cdot 5 + 1 \cdot 6 = 46$ символов "0" или "1", из них 20 "1" и 26 "0". Итак, после кодирования на одну букву алфавита А приходится $46/20 = 2,3$ бита.

По формуле Шеннона средняя длина кода (информативность одной буквы) равна: $I = -((8/20) \log_2(8/20) + (5/20) \log_2(5/20) + (3/20) \log_2(3/20) + (2/20) \log_2(2/20) + (1/20) \log_2(1/20) + (1/20) \log_2(1/20)) = 2,204$ (бит). Избыточность кода Хаффмана составляет $Q = (2,3 - 2,204)/2,204 = 0,044$. При равномерном кодировании 6 букв средняя длина кода (количество бит на одну букву) $\log_2(6) = 2,5849$ бит, избыточность кода равна

$Q = (2, 5849 - 2, 204)/2, 204 = 0, 17$, что существенно больше.

9. Блочное кодирование. Код Хэмминга. При блочном кодировании кодируются не отдельные буквы, а их сочетания и даже слова. Допустим в словаре, используемом для передачи сообщений между морскими судами, $16384 = 2^{14}$ слова. Если их кодировать равномерным кодом, то достаточно 14 бит. Так как средняя длина слова русского языка 6,3 буквы (включая пробел), то на одну букву приходится информация $I_A = 14/6,3 = 2,22$ бита. При равномерном кодировании букв на каждую букву приходится $I_A = \log_2 32 = 5$ бит. Если установить относительную частоту появления слов и использовать неравномерный код, то можно еще повысить экономичность кода.

Хэмминг предложил в кодовую последовательность добавлять биты, позволяющие в случае ошибки восстановить переданное сообщение. Повысить надежность кода можно с помощью контроля четности. К информационным битам s_1, \dots, s_n добавляются избыточные биты p_1, \dots, p_k , контролирующие четность их сумм. Например, к битам s_1, s_2, s_3, s_4 добавляются три бита p_1, p_2, p_3 , так, что $s_1 + s_2 + s_3 + p_1, s_2 + s_3 + s_4 + p_2, s_1 + s_3 + s_4 + p_3$ — четные числа (при делении по модулю 2 дают 0). Декодер осуществляет контроль четности и в случае инвертирования 1 бита выявляет ошибку.

Пусть x — входной символ, a_i — символ на выходе кодера, поступающий в канал связи, b_i — символ после канала связи на входе декодера, y — символ на выходе декодера. Рассмотрим три кода Хэмминга:

1. (3,1)-код: кодирующая функция $a_1 = x, a_2 = x, a_3 = x$, то есть кодер утраивает входной символ. Декодирующая функция: $y = b_1 b_2 \oplus b_2 b_3 \oplus b_1 b_3$. Если сообщение $x = 1$, то после кодера в канал связи поступает 111. Допустим, из-за помехи на вход декодера приходит 110, тогда все равно на его выходе $y = 1$.

2. (5,2)-код: блок из 2 символов кодируется блоком из 5 символов. Кодирующая функция $a_1 = x_1, a_2 = x_1, a_3 = x_1 + x_2, a_4 = x_2, a_5 = x_2$, декодирующая: $b' = b_1 \oplus b_3 \oplus b_5, y_1 = b_1 \oplus (b_1 \oplus b_2)b', y_2 = b_5 \oplus (b_5 \oplus b_4)b'$. Если сообщение $(x_1, x_2) = (0, 1)$, то после кодера получается 00111. Самостоятельно убедитесь, в том, что декодирующая функция позволит декодировать сообщение даже если 1 бит из-за помех будет инвертирован.

3. (7,4)-код: к 4 информационным битам i_1, i_2, i_3, i_4 добавляются еще 3 проверочных $p_1 = i_1 \oplus i_2 \oplus i_3, p_2 = i_2 \oplus i_3 \oplus i_4, p_3 = i_3 \oplus i_2 \oplus i_4$. При этом возможны 16 различных кодовых последовательностей:

| | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|
| 0000000 | 0001011 | 0010011 | 0010110 | 0011101 | 0100111 | 0101100 |
| ... | ... | ... | ... | ... | 1110100 | 1111111 |

На конце канала связи получаются 7-ми битовые кодовые слова $x = (i'_1, i'_2, i'_3, i'_4, p'_1, p'_2, p'_3)$. Декодер выделяет биты $s_1 = p'_1 \oplus i'_1 \oplus i'_2 \oplus i'_3, s_2 = p'_2 \oplus i'_2 \oplus i'_3 \oplus i'_4, s_3 = p'_3 \oplus i'_1 \oplus i'_2 \oplus i'_4$. Получающаяся трехбитовая последовательность называется *синдромом*. Данный код исправляет 1 ошибку в 7 битах. Всего возможно $2^3 = 8$ синдромов, которые соответствуют следующим 8 вариантам: одному безошибочному и семи вариан-

там с ошибками в первом, втором, ..., седьмом битах.

(7,4)-код Хэмминга, может быть задан порождающей матрицей:

$$G_{4 \times 7} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Для того, чтобы из информационного слова a получить кодовое слово b , используют правило: $b = a \odot G$. Знак \odot означает умножение по $\text{mod } 2$: строка одномерной матрицы a почленно умножаются на столбец, результаты суммируются и находится остаток от деления на 2. Допустим, $a = (1101)$, тогда

$$b = a \odot G = (1101) \odot \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} = (0001101).$$

Покажем, как получен первый бит в кодовом слове b : $1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 = 2$, остаток от деления на 2 равен 0. Самостоятельно убедитесь в том, что если $a = 1010$, то $b = 0011010$; если $a = 0110$, то $b = 1000110$; если $a = 1110$, то $b = 0101110$. Видно, что последние 4 бита кодового слова b совпадают с информационным словом a . Код, позволяющий выделять информационные и проверочные символы из соответствующего ему кодового слова, называется *систематическим*. При этом структура кодовой последовательности такова: $x = (x_1, x_2, \dots, x_{n-k} || x_{n-k+1}, x_{n-k+2}, \dots, x_n)$, где первые $(n - k)$ символов *информационные*, а последние k символов — *проверочные* или наоборот.

10. Архивирование, сжатие и шифрование. *Сжатие файла* — процесс кодирования, в результате которого из одного файла получается файл меньшего объема. Различают два типа алгоритмов сжатия: *обратимые* — позволяют точно восстановить исходный файл (архивирование); *необратимые* или *с регулируемой потерей информации* — сохраняют существенную информацию, несущественную — теряют.

К обратимым относится метод упаковки, который применяется, когда в сжимаемом массиве присутствует небольшая часть используемого алфавита. Он состоит в уменьшении числа бит, отводимых для кодирования 1 символа. Например, в сообщении "abcdefghigklmnop" 16 букв, поэтому для кодирования 1 буквы достаточно 4 бит вместо 8. Другой пример — BCD-формат (Binary Coded Decimal), то есть двоично-десятичная запись числа, используемая в базах данных для хранения целых чисел: кодирование одного символа требует не восьми, а четырех бит.

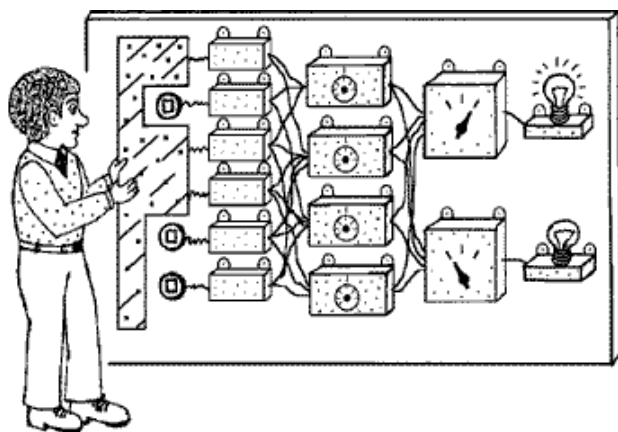
Обратимым является алгоритм RLE (Run Length Encoding), предусматривающий кодирование путем учета числа повторений. Он выявляет повторяющуюся последовательность данных, число повторений и заменяет их простой структурой, содержащей повторяющийся фрагмент и коэффициент повтора. Используется для кодирования графических файлов из *.bmp (битовая карта) в *.pcx.

Алгоритмы Лемпеля и Зива LZ77, LZ78 также являются обратимыми. Они кодируют повторяющиеся цепочки байтов в текстовых файлах и даже целые слова, которых в 7–8 раз меньше, чем букв. Алгоритм LZ77 для цепочки первых байтов незакодированной части ищет длинное совпадение в прочитанной части. Если оно найдено, определяется смещение и длина совпадения. Когда информация "смещение–длина" меньше числа байт, соответствующих совпадающей части текста, она записывается в выходной массив, в противном случае этого не происходит. Алгоритм LZ78 имеет совокупность повторяющихся цепочек букв, каждой из которых соответствует короткий код. Для последовательности кодируемых байтов ищется наиболее длинное совпадение, код записывается в выходной файл. При необходимости в совокупность цепочек добавляется новое сочетание букв. Возможно сверточное кодирование: кодер содержит запоминающее устройство, в котором сохраняется входная последовательность бит. Затем осуществляется поблочное кодирование.

Теорема. Универсального архиватора, обратимо сжимающего любой файл хотя бы на 1 бит не существует. *Доказательство:* Предположим, что такая программа есть. Тогда многократно применяя ее, можно было бы обратимо сжать любой файл до объема 1 бит. Но это абсурд, так как из такого файла нельзя восстановить исходный.

Методы сжатия с регулируемой потерей информации применяются для уменьшения объема графических, аудио и видеофайлов. Глаз человека наиболее чувствителен к зеленому цвету, а к красному и синему — в 4 и 10 раз хуже. Поэтому на красный и синий части спектра можно отводить меньшее число бит. Это используется в кодировании методом JPEG (Joint Photographic Experts Group), получаются файлы *.jpg. Кодирование звуковых файлов осуществляется по алгоритму MP3: звуковой файл разбивают на фреймы (кадры, участки), в каждом из которых звук разлагается на гармоники. Гармоники высокого порядка удаляют, остальные сжимают по методу Хаффмана. Алгоритм MPEG (Motion Picture Experts Group) используют для сжатия видео методом опорного кадра. Сохраняют опорный кадр и его небольшие изменения. Изображения движущихся объектов кодируют низким качеством. Сам фильм, титры и логотип сохраняют в различных потоках данных (слоях) по отдельности. Титры можно хранить в виде букв, логотип — в 1 экземпляре и копировать в каждый кадр и т.д.

Шифрованием называется специальное кодирование сообщения с помощью ключа (правила шифрования), которое препятствует его чтению лицом, не имеющим ключ. Некоторые виды шифров: 1) шифр замены: буквам одного алфавита ставятся в соответствие другие буквы того же или другого алфавита; 2) шифр перестановки: сообщение записывают в строки таблицы (матрицы), результат кодирования считывают по столбцам; 3) шифрование функцией: зашифрованное сообщение $y = y_1, y_2, \dots, y_m$ представляет собой последовательность чисел (кодов букв), которые найдены по формуле $y_i = a_1 x_i^1 + a_2 x_i^2 + \dots + a_n x_i^n$, где $x = x_1, x_2, \dots, x_m$ — исходное сообщение, а a_1, a_2, \dots, a_n — ключ.



ГЛАВА 6 АЛГОРИТМЫ, ПЕРЦЕПТРОНЫ И ПРОЧИЕ ИГРЫ

Если мы хотим научить компьютер решать ту или иную задачу, то ему необходимо сообщить алгоритм ее решения, например, введя соответствующую программу. Слово "алгоритм" произошло от имени узбекского математика Аль-Хорезми, который еще в 9 веке сформулировал правила выполнения арифметических действий. Оно означает жестко детерминированную последовательность операций, приводящую к преобразованию информации, энергии или материальных объектов. В этой главе Вы узнаете, что такое алгоритм, познакомитесь с абстрактными машинами Поста и Тьюринга, цепями Маркова, рекурсивными функциями, детерминированными и вероятностными автоматами, нейронными сетями, выясните, что такое графы, получите представление о теории игр, линейном программировании ...

6.1. ОБ АЛГОРИТМАХ И АВТОМАТАХ

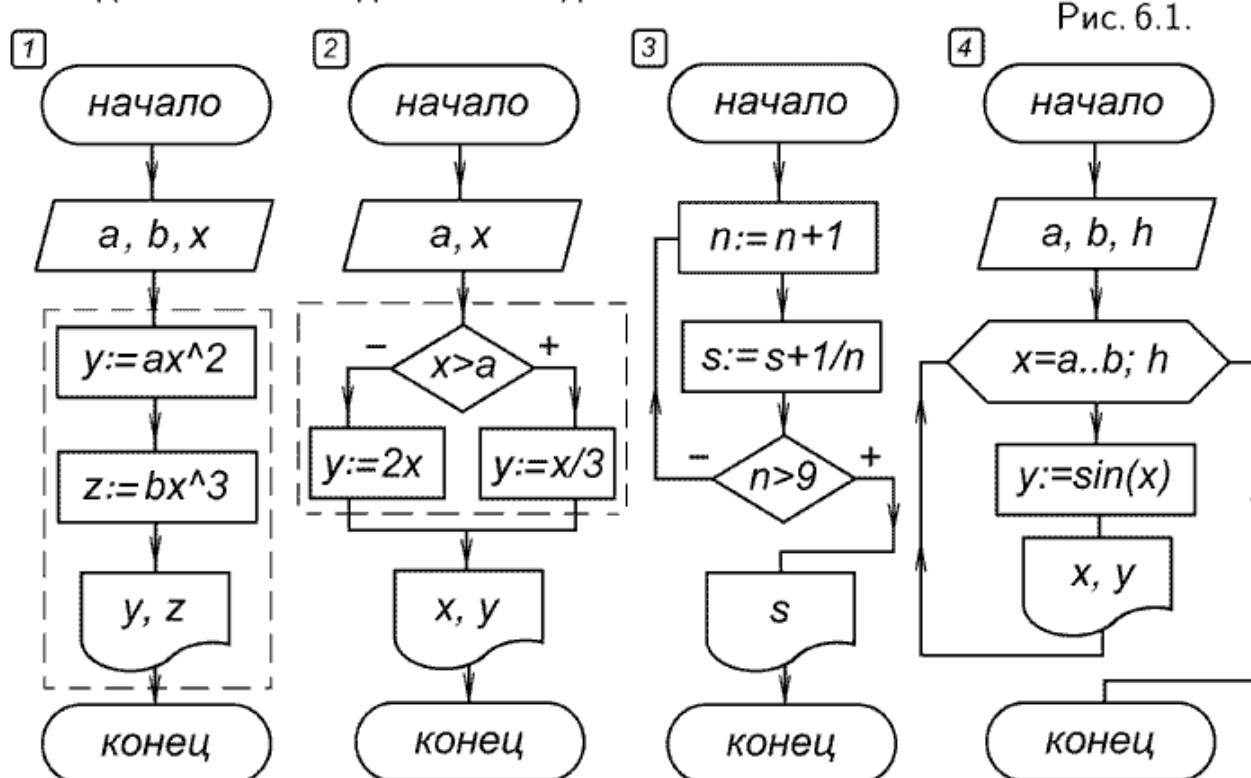
1. Понятие алгоритма. Все задачи делятся на два класса: вычислительные функции и распознавание образов (классификация). Функция *вычислима* если возможно построение алгоритма для ее вычисления. Множество *разрешимо* если существует алгоритм, позволяющий определить принадлежит данный объект к этому множеству или нет. Теория алгоритмов занимается изучением различных видов алгоритмов и их свойств, поиском эффективных алгоритмов и алгоритмически неразрешимых задач.

Решение задачи на ЭВМ состоит из следующих этапов: 1) постановка задачи; 2) математическое описание задачи; 3) разработка алгоритма решения; 4) написание программы на языке программирования; 5) подготовка исходных данных; 6) ввод программы и данных в ЭВМ; 7) отладка и тестирование программы; 8) решение задачи; 9) обработка и интерпретация результатов. Таким образом, перед написанием программы необходимо разработать алгоритм решения задачи.

Формальный исполнитель — устройство, которое читает программу, изменяет в соответствии с ней свое внутреннее состояние, и исполняет

команды, осуществляя пошаговую обработку информации. Примерами таких устройств являются ЭВМ, программируемый калькулятор, станок с числовым программным управлением (ЧПУ) т.д. Чтобы задать исполнитель достаточно определить: а) систему команд; б) форму представления входной и выходной информации; в) множество внутренних состояний; г) язык программирования.

Будем исходить из того, что *алгоритм* — последовательность арифметических и логических действий над числами, приводящая к получению результата и решению задачи за конечное число шагов. Иначе говоря, алгоритмом называется любая конечная система правил преобразования информации. Свойства алгоритма: 1) детерминированность, — применение алгоритма к одним и тем же данным должно приводить к одному результату; 2) массовость, — дает результат при различных исходных данных; 3) результативность, — дает результат через конечное число шагов. Исполнение алгоритма порождает *вычислительный процесс*. В широком смысле алгоритмом называется жестко определенная последовательность действий, приводящая к преобразованию информации, материи и энергии. Если все выполняемые операции пронумеровать, то получится последовательность действий над числами.



Существуют алгоритмы: 1) *линейной* структуры: вычислительные действия выполняются последовательно друг за другом, алгоритм не содержит условий (рис. 6.1.1); 2) *разветвляющейся* структуры: в зависимости от выполнения некоторого условия вычислительный процесс осуществляется по одной или по другой ветви (рис. 6.1.2); 3) *циклической* структуры: алгоритм может содержать циклы — многократно повторяющиеся участки вычислительного процесса (рис. 6.1.3 – 6.1.4).

Алгоритм может быть изображен с помощью блок-схемы — ориентированного графа, вершинами которого являются команды, а ребрами — переходы от одной команды к другой (рис. 6.1). А. А. Ляпунов предложил иной способ описания алгоритмов — с помощью специальных операторных схем. Рассмотрим, например, алгоритм вычисления функции $y = x^2 + 3x$ при $x > a$ и $y = 2x$ при $x \leq a$. Он может быть записан в виде, представленном на рис. 6.2.1. При этом Φ — оператор ввода, P — условный переход, Ω — безусловный переход, O — вывод результатов вычислений, S — стоп, остановка программы. Чтобы упростить запись, стрелки соответствующие условному и безусловному переходам не рисуются, а изображаются лишь их начало и конец, отмеченные одинаковыми номерами (рис. 6.2.2). Циклы с параметром i , то есть многократное повторение последовательности операторов A, B, C , некоторые из которых зависят от i , обозначают буквой Π (рис. 6.2.3), указывая число повторений n .

Функциональным блоком называется часть алгоритма, имеющая один вход (выполнение всегда начинается с одного и того же оператора) и один выход (всегда заканчивается одним и тем же оператором). Совокупность операторов, выделенная на рис. 6.1.1 пунктиром, является функциональным блоком, а на рис. 6.1.2 — нет. **Структурным** называется алгоритм, являющийся комбинацией линейного, разветвляющегося и циклического алгоритмов.

Теорема Бома-Джакопини: Любой неструктурный алгоритм может быть сведен к эквивалентному ему структурному алгоритму.

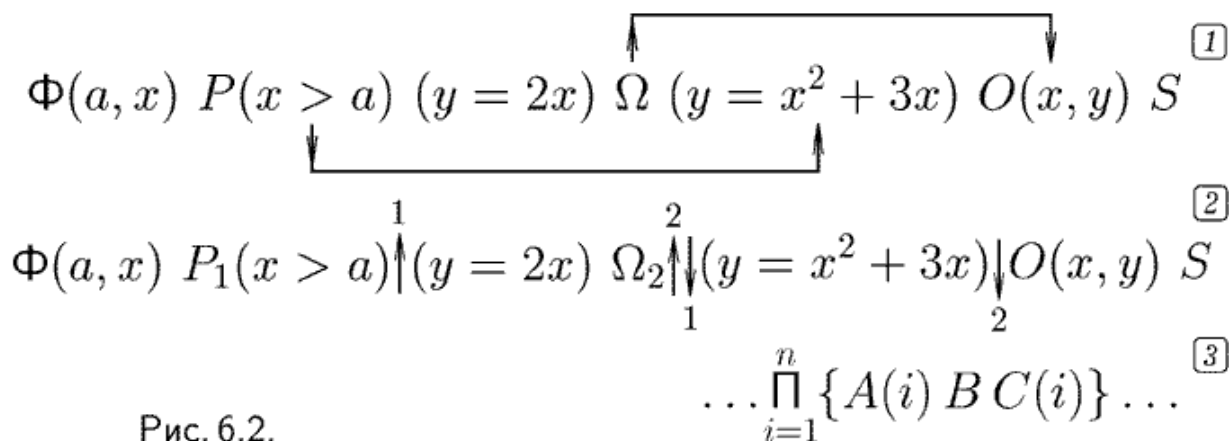


Рис. 6.2.

Сложность объекта определяется длиной двоичного слова $101 \dots 1$, которое дает его точное описание, или длиной программы, способной сгенерировать это слово. **Вычислительная сложность** алгоритма характеризуется числом элементарных шагов выполнения программы или временем ее выполнения. Так как алгоритм может иметь разветвления, то его сложность зависит от исходных данных. **Эффективным** называется алгоритм, число шагов которого пропорционально полиному от характерного размера задачи (количества исходных данных).

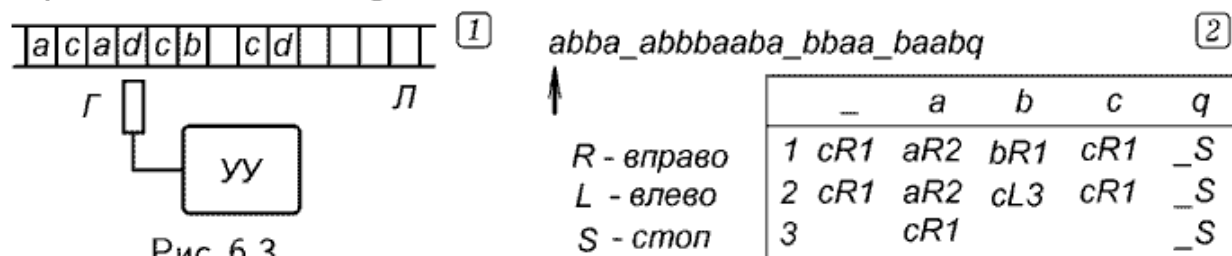
Алгоритмически неразрешимой называется задача, для решения которой невозможно построить алгоритм. К таким задачам относятся: 1. **Проблема остановки:** по описанию алгоритма A над заданным числом x нельзя ответить (создать универсальный алгоритм отладки) остановится

алгоритм A или нет. 2. *Проблема распознавания эквивалентности алгоритмов*: невозможно построить универсальный алгоритм, который бы для любых программ определял вычисляют они одну и ту же функцию или нет. Для того, чтобы решить проблему алгоритмической разрешимости той или иной задачи были предложены следующие универсальные алгоритмические модели: 1. Абстрактные машина Тьюринга и Поста; 2. Система подстановок; 3. Арифметизация алгоритма с помощью рекурсии.

2. Абстрактные машины Тьюринга и Поста. *Машина Тьюринга (МТ)* — абстрактная машина, предложенная для обоснования понятия алгоритма и доказательства алгоритмической разрешимости задачи.

МТ состоит из бесконечной подвижной ленты Π , разделенной на ячейки, головки чтения/записи Γ и управляющего устройства УУ (рис. 6.3.1). Головка чтения/записи, которая, может считывать содержимое обозреваемой ячейки, стирать, либо записывать в ячейку один символ из алфавита $X = (x_1, x_2, \dots, x_n)$. УУ в каждый такт находится в одном из множества состояний $Q = (q_1, q_2, \dots, q_m)$.

За один такт МТ головка считывает символ из обозреваемой ячейки, устройство управления переходит в новое состояние и выполняет команду, перемещая головку относительно ленты на требуемое число ячеек влево или право и записывая необходимый символ. Программа МТ состоит из команд записанных в виде: $q_i a_j \rightarrow q'_i a'_j d_k$. Это означает следующее: если в обозреваемой ячейке a_j , а УУ в состоянии q_i , то УУ переходит в состояние q'_i , в данную ячейку записывается a'_j . Если d_k равно -1, 1 или 0, то головка смещается влево, вправо или остается на месте соответственно. Перейдя в состояние q_z МТ останавливается.



На рис. 6.3.2 представлена программа, которая в слове, составленном из последовательности символов "а", "b" и "_" заменяет: 1) подслово "ab" на "cc"; 2) пробел "_" на "c"; 3) символ "q" означает остановку.

Многоленточная МТ имеет несколько входных лент и одну выходную для записи результата вычислений. За один шаг головка считывает символы со всех входных лент, УУ переходит в другое состояние, затем осуществляется запись новых символов и смещение всех лент.

С помощью МТ можно показать, что любой сложный алгоритм разложим на простые операции. Если для решения задачи можно построить машину Тьюринга, то задача алгоритмически разрешима. Тьюринг сформулировал положение, которое было принято без доказательства и называется *тезисом Тьюринга*: Для любой вычислимой функции можно построить машину Тьюринга, которая ее вычисляет.

Машина Поста — абстрактная машина, состоящая из каретки (счи-

тывающей и записывающей головки) и бесконечной ленты, разбитой на ячейки. В каждой ячейке либо стоит метка, либо она остается пустой. В МП используется унарная система счисления: числу 4 соответствует 4 ячейки с метками, расположенные по порядку. Программой МП является конечный список следующих команд (рис. 6.4): сдвинуть каретку вправо/влево и перейти к команде N ; если в обозреваемой кареткой ячейке метка, то перейти к команде N , иначе — к M ; стереть метку и перейти к команде N ; остановиться. Тут же представлена программа для МП, которая вычитает из левого числа правое. Исходное положение каретки — напротив пустой ячейки между числами.

3. Нормальные алгоритмы Маркова. Алгоритм в алфавите A — это функция, преобразующая входную последовательность символов из алфавита A в выходную. Нормальный алгоритм Маркова задается алфавитом $A = (a_1, a_2, \dots, a_n)$, содержащим конечное непустое множество элементов, нормальной схемой подстановок и порядком их применения. Нормальной схемой подстановок называется набор правил (формул) вида $P_k \rightarrow P_n$, согласно которым левое подслово P_k исходного слова P заменяется на слово P_n .

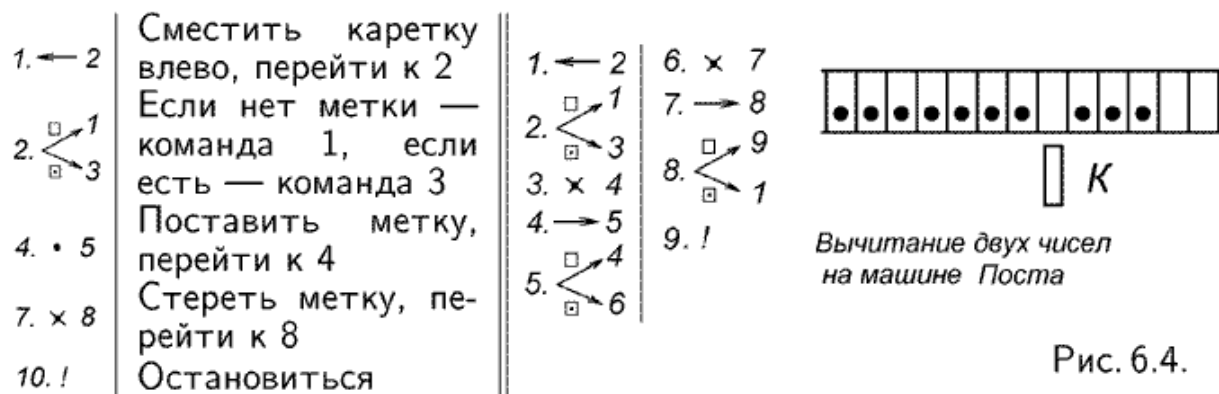


Рис. 6.4.

К исходному слову P применяют по порядку каждую пару из схемы подстановок. Если подстановка возможна, то ее осуществляют и все начинают сначала. Выполнение алгоритма прекращается, когда нет ни одной допустимой подстановки. В результате получается конечное слово Q . Например, нормальная схема: $cb \rightarrow bc$, $ca \rightarrow cb$, $cc \rightarrow ba$. Если входное слово $cacbbc$, то: $cacbbc \rightarrow cabcbc \rightarrow cabbcc \rightarrow cbbbcc \rightarrow bcbbcc \rightarrow bbbcc \rightarrow bbbcc \rightarrow bbbcc$.

Рассмотрим другой пример, иллюстрирующий возможность использования цепей Маркова для сложения и вычитания чисел в унарной системе счисления. Допустим, дана нормальная схема: 1) $1+ \rightarrow 1$; 2) $1-1 \rightarrow -$; 3) $-+ \rightarrow +$; 4) $+- \rightarrow -$. Если применить ее к входному слову $111+11-1+11$, то получится 111111 , то есть $3+2-1+2=6$.

Может быть сформулирован принцип нормализации (тезис Маркова): всякий алгоритм, определяющий произвольный потенциально осуществимый процесс переработки слов в некотором алфавите, который из различных исходных данных позволяет получить соответствующий результат, эквивалентен нормальному алгоритму.

Теорема. Нормальные алгоритмы эквивалентны машинам Тьюринга.

Если доказано, что нельзя построить нормальную схему подстановок, позволяющую перейти от P к Q , то данная задача алгоритмически неразрешима. Множества алгоритмически разрешимых задач для нормальных алгоритмов Маркова, машин Тьюринга и Поста совпадают.

4. Рекурсивные функции. Любой алгоритм на входе имеет исходные данные, а на выходе — конечные, и поэтому может рассматриваться как метод вычисления сложной функции. Существует два принципиально различных способа задания сложной функции: 1. *Суперпозиция* — подстановка функции в функцию: $x \cdot y + x/z - y$, содержащая фиксированное число операций. 2. *Рекурсия* — определение очередного значения функции f_{i+1} через ранее вычисленное f_i . Структура примитивно-рекурсивного определения функции предполагает задание значения функции для начального значения аргумента и правила определения f_{i+1} через f_i .

Функция $f(x)$ называется *частичной*, если она определена не для каждого x , а, например, только для натуральных x . Простейшими называются функции: 1) одноместная функция непосредственного следования: $S^1(x) = x + 1$; 2) n -местная функция тождественного равенства 0: $0^n(x_1, x_2, \dots, x_n) = 0$; 3) n -местная проектирующая функция тождественного повторения значения одного из своих аргументов: $I_m^n(x_1, \dots, x_m, \dots, x_n) = x_m$, где m — целое, в интервале от 1 до n .

Примитивно-рекурсивные функции получаются с помощью операторов: 1. Суперпозиция частичных функций $S(g, f_1, \dots, f_n)$: Имеются n m -местных функции $f_1(x_1, \dots, x_m)$, $f_2(x_1, \dots, x_m)$, ..., $f_n(x_1, \dots, x_m)$, их подставляют в n -местную функцию $g(x_1, x_2, \dots, x_n)$. В результате получается n -местная функция:

$$h(x_1, x_2, \dots, x_m) = g(f_1(x_1, \dots, x_m), f_2(x_1, \dots, x_m), \dots, f_n(x_1, \dots, x_m)).$$

2. Примитивная рекурсия: Функция f образуется из частичных функций $g(x_1, \dots, x_n)$ и $h(x_1, \dots, x_n, m, k)$ посредством примитивной рекурсии, если для всех натуральных x_1, \dots, x_n справедливо:

$$f(x_1, \dots, x_n, 0) = g(x_1, \dots, x_n),$$

$$f(x_1, \dots, x_n, m + 1) = h(x_1, \dots, x_n, m + 1, f(x_1, \dots, x_n, m)).$$

Эти условия задают последовательность нахождения функции f :

$$f(x_1, \dots, x_n, 0) = g(x_1, \dots, x_n),$$

$$f(x_1, \dots, x_n, 1) = h(x_1, \dots, x_n, 1, f(x_1, \dots, x_n, 0)),$$

$$f(x_1, \dots, x_n, 2) = h(x_1, \dots, x_n, 2, f(x_1, \dots, x_n, 1)),$$

.....

Примитивно-рекурсивной называется такая частичная функция, которая может быть получена конечным числом операций суперпозиции и примитивной рекурсии, исходя из элементарных функций $S^1, 0^n, I_m^n$. Такие функции всюду определены.

В качестве примера, покажем, что функция $f(x, y) = x + y$ примитивно-рекурсивная. В самом деле: $f(x, 0) = x + 0 = x = I_1^1(x)$, $f(x, y + 1) = x + (y + 1) = s^1(x + y) = s^1(f(x, y))$. Итак, функция $f(x, y)$ строится из примитивно-рекурсивных функций I_1^1, s^1 . При этом $f(2, 3) = s^1(f(2, 2)) =$

$$s^1(s^1(f(2,1))) = s^1(s^1(s^1(f(2,0)))) = s^1(s^1(s^1(I_1^1(2)))) = s^1(s^1(s^1(2))) = s^1(s^1(3)) = s^1(4) = 5.$$

Кроме суперпозиции и примитивной рекурсии, может быть введена *мю-операция*: отыскание наименьшего натурального числа a , для которого значение $(n+1)$ -местной функции $g(x_1, x_2, \dots, x_n, a)$ равно 0. μ – оператор строит n -местную функцию $f(x_1, x_2, \dots, x_n)$ равную x_n . Если такого числа для набора x_1, x_2, \dots, x_n не существует, то функция f не определена. Это может быть записано так:

$$\mu(f(x_1, x_2, \dots, x_n)) = \min_{a \in \mathbb{N}} \{f(x_1, x_2, \dots, x_{n-1}, a) = x_n\}.$$

Тезис Черча: Любая вычислимая функция является рекурсивной, то есть может быть построена из целых чисел и арифметических операций с помощью суперпозиции и рекурсии. (Множество алгоритмически вычислимых функций совпадает с классом всех рекурсивных функций. Понятие вычислимости универсально и не зависит от выбора формализации.)

Можно доказать следующие *теоремы*: 1. Любая рекурсивная функция может быть вычислена на соответствующей машине Тьюринга. 2. Любая задача, решаемая на машине Тьюринга может быть решена с помощью нормального алгоритма Маркова. 3. Любой конечный вычислительный алгоритм может быть реализован на цифровой ЭВМ с достаточно большой памятью за конечное время. Итак, ЭВМ-вычислимость эквивалентна рекурсивной вычислимости, арифметической вычислимости по Маркову и вычислимости по Тьюрингу.

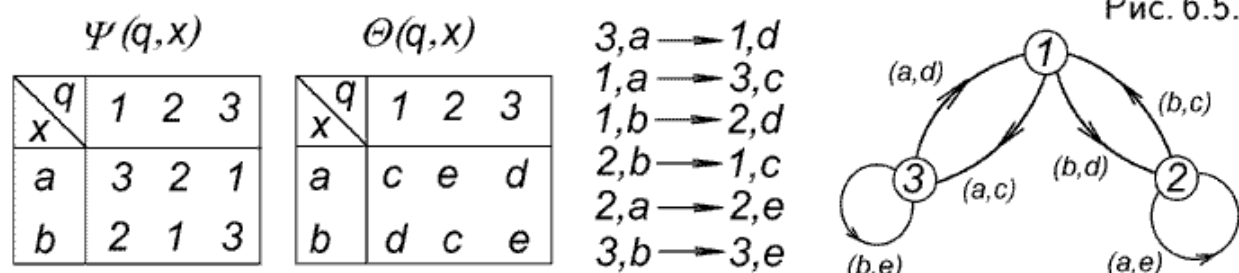
ЭВМ может выполнять операцию подстановки и рекурсивные вычисления, например, по схеме: $f(0) = a$, $f(1) = b$, $f(i) = h(i, f(i-1), f(i-2)) = 2i - 3f(i-1) - f^2(i-2)$. Классическим примером рекурсивного задания функции является последовательность Фибоначи: $f(0) = 1$, $f(1) = 2$, $f(x+2) = f(x) + f(x+1)$. При этом получаются значения: 1, 2, 3, 5, 8, 13, Рекурсивная процедура применяется для получения на экране ЭВМ изображения фракталов — самоподобных геометрических фигур. Рекурсивный способ вычисления функций используется, например, при расчете движения частиц в силовом поле численными методами. Для нахождения их координат и проекции скоростей в следующий момент времени t_{i+1} необходимо знать координаты и скорости частиц в предыдущий момент времени t_i . Можно механизировать мю-операцию, которая состоит в отыскании первого натурального числа, для которого $g(x) = 0$.

5. Основные понятия теории автоматов. Автомат — дискретное устройство, выполняющее заданную последовательность действий, в результате чего происходит преобразование информации, материальных объектов или энергии. Автомат имеет n входов и m выходов, и характеризуется множеством внутренних состояний Q , а также функцией перехода Ψ и функцией выхода θ .

Функция перехода Ψ связывает внутреннее состояние устройства q_{t+1} в момент времени $t+1$ с внутренним состоянием q_t и входным символом x_t (сигналом) в предыдущий момент времени t : $q_{t+1} = \Psi(q_t, x_t)$. Функция выходов θ связывает внутреннее состояние q_t устройства и входной

сигнал x_t в момент t с выходным сигналом y_t в тот же момент времени: $y_t = \theta(q_t, x_t)$. Компоненты $\{X, Y, Q, \Psi, \theta\}$ задают автомат, преобразующий входные сигналы X в выходные Y .

Множество внутренних состояний Q является внутренней памятью автомата. Различают автоматы без памяти, с конечной памятью, с бесконечной памятью. *Автоматы без памяти* — это автоматы, имеющие всегда одно внутреннее состояние, и задаваемые тройкой компонентов $\{X, Y, \theta\}$. Они не меняют своего поведения: выходной сигнал зависит только от входного и не зависит от ранее поступивших сигналов: $y_t = \theta(x_t)$. При работе автомата без памяти осуществляется побуквенный перевод входного сообщения. Если X и Y — двоичные сигналы, то автомат без памяти состоит из логических элементов И, ИЛИ, НЕ. Схемы, в которых выходы одних логических элементов подсоединяются к входам других называются *комбинационными*.



Пусть входной алфавит конечного автомата $X = (a, b)$, выходной алфавит $Y = (c, d, e)$, множество внутренних состояний $Q = 1, 2, 3$. Автоматные функции переходов $\Psi(q, x)$ и выходная функция $\Theta(q, x)$ заданы таблицами (рис. 6.5) или с помощью ориентированного графа (диаграммы Мура). Вершины графа изображают состояния автомата $Q = (1, 2, 3)$. Каждой команде $q_i x_r \rightarrow q_j x_s$ соответствует ребро, идущее из q_i в q_j называемое (x_r, y_s) . Для диаграмм Мура выполняются требования: 1) множества вершин и ребер конечны; 2) из одной вершины не выходят два ребра с одним входным символом; 3) для каждой входной комбинации (q_i, x_r) существует ребро, идущее из q_i с символом x_r .

Примем без доказательства следующие утверждения: 1. Для любого конечного автомата можно построить единственный эквивалентный ему минимальный автомат. 2. Цифровая ЭВМ, содержащая устройство управления, арифметико-логическое устройство, запоминающее устройство и устройство ввода-вывода, является конечным автоматом.

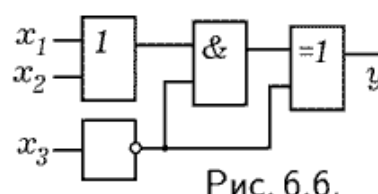
6. Логические функции и комбинационные схемы. К основным логическим операциям относятся (рис. 6.6): 1) логическое сложение ИЛИ (дизъюнкция "+", $A \text{ or } B$); 2) логическое умножение И (конъюнкция "·", $A \text{ and } B$); 3) логическое отрицание НЕ (инверсия, $\text{not } A$). Существуют и другие логические операции, например, логическое сложение по $\text{mod } 2$ (исключающее ИЛИ " \oplus ", $A \text{ xor } B$) и т.д.

Логическим высказыванием называется предложение, которое может быть истинным (лог. 1, true) или ложным (лог. 0, false). Сложение и умножение производятся по правилам: $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$,

$1 \oplus 1 = 0$, $0 * 0 = 0$, $0 * 1 = 0$, $1 * 0 = 0$, $1 * 1 = 1$. Перечисленные выше операции позволяют составить логические функции, которые в зависимости от значений аргументов A , B , C принимают два значения: истина (1) или ложь (0). Например: $f(A, B, C) = (\text{not } A) \text{ or } (B \text{ xor } C)$.

Для логических высказываний справедливы законы булевой алгебры (предложена Дж. Булем): 1. Ассоциативность дизъюнкции и конъюнкции: $A \text{ and } (B \text{ and } C) = (A \text{ and } B) \text{ and } C = A \text{ and } B \text{ and } C$, $A \text{ or } (B \text{ or } C) = (A \text{ or } B) \text{ or } C = A \text{ or } B \text{ or } C$. 2. Коммутативность дизъюнкции и конъюнкции: $A \text{ and } B = B \text{ and } A$, $A \text{ or } B = B \text{ or } A$. 3. Дистрибутивность конъюнкции относительно дизъюнкции и дизъюнкции относительно конъюнкции: $A \text{ and } (B \text{ or } C) = (A \text{ and } B) \text{ or } (A \text{ and } C)$, $A \text{ or } (B \text{ and } C) = (A \text{ or } B) \text{ and } (A \text{ or } C)$. 4. Идемпотентность: $A \text{ and } A = A$, $A \text{ or } A = A$. 5. Закон двойного отрицания: $\text{not}(\text{not } A) = A$. 6. Закон противоречия: $\text{not}(A) \text{ and } A = 0$. 7. Закон исключенного третьего: $A \text{ or } (\text{not}(A)) = 1$. 8. Правила де Моргана: $\text{not}(A \text{ and } B) = \text{not}(A) \text{ or } (\text{not}(B))$, $\text{not}(A \text{ or } B) = \text{not}(A) \text{ and } (\text{not}(B))$. 9. Свойства 0 и 1: $A \text{ and } 1 = A$, $A \text{ and } 0 = 0$, $A \text{ or } 1 = 1$, $A \text{ or } 0 = A$, $\text{not}(1) = 0$, $\text{not}(0) = 1$.

| A | B | A AND B | A OR B | A XOR B | NOT A |
|---|---|---------|--------|---------|-------|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |



Каждой логической функции соответствует комбинационная схема из логических элементов и наоборот. Например, схеме, изображенной на рис. 6.6, соответствует логическая функция

$$y(x_1, x_2, x_3) = ((x_1 \text{ or } x_2) \text{ and } (\text{not } x_3)) \text{ xor } (\text{not } x_3).$$

На основе логических элементов И, ИЛИ, НЕ построен процессор, осуществляющий математические и логические операции и управляющий работой ЭВМ, память ОЗУ, контроллеры и другие блоки ЭВМ.

7. Автоматическая классификация объектов. Классификацией называют упорядочение объектов, осуществляемое исходя из степени сходства или различия между ними. При этом степень сходства между объектами, включенными в один класс, должна быть больше, чем степень сходства между объектами из различных классов. Важно, чтобы классификация учитывала существенные признаки объектов и отражала объективные закономерности.

Допустим, необходимо отклассифицировать 10 объектов, каждый из которых характеризуется 10 величинами. Этими объектами могут быть астероиды, звезды, химические соединения, отпечатки пальцев, археологические находки, студенты вуза и т.д. Для того чтобы решить эту задачу, необходимо учесть 100 величин. А если объектов не 10, а 100 или 1000? Сложность задачи резко возрастает. Кроме того, участие человека неизбежно приводит к тому, что результаты классификации отражают не только объективные закономерности, но субъективные особенности кон-

кретного эксперта. Иными словами, другой эксперт (или тот же самый эксперт через неделю) отсортирует объекты несколько иначе.

Автоматическая классификация, построенная на использовании ЭВМ, называется *кластеризацией* объектов. Ее преимущество состоит в объективности, возможности учета нескольких характеристик большого числа классифицируемых объектов. Метод кластеризации наиболее эффективен, когда неизвестно исходное естественное распределение объектов.

Кластеризация иерархического типа состоит в последовательном объединении (разделении) групп объектов, начиная с самых близких, сходных (далеких, отличных), затем все более отдаленных друг от друга (приближенных друг к другу). В качестве меры близости двух объектов, как правило, используется геометрическое расстояние между точками, соответствующими этим объектам, в пространстве, образованном характеристиками, по которым осуществляется классификация.

Рассмотрим множество из n объектов $O_1(x_1, y_1, z_1), O_2(x_2, y_2, z_2), \dots$, каждый из которых характеризуется некоторой совокупностью признаков X, Y, Z . В пространстве признаков X, Y, Z каждому объекту соответствует точка. В качестве меры близости двух объектов O_i и O_k обычно выбирают геометрическое расстояние между этими точками:

$$S_{i,k} = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2}.$$

На начальном этапе считают, что каждый объект образует кластер с весовым коэффициентом 1. Для каждой пары точек определяют меру близости $S_{i,k}$ и, сравнивая их друг с другом, находят наиболее близко расположенные объекты, которые объединяются в один кластер с весовым коэффициентом 2. Координаты нового кластера вычисляются как средние взвешенные координат объединенных кластеров. Затем процесс повторяется. При слиянии двух кластеров O_i и O_k с весовыми коэффициентами p_i и p_k получается кластер O_{ik} , имеющий следующие координаты и весовой коэффициент соответственно:

$$\begin{aligned} x_{i,k} &= (p_i x_i + p_k x_k) / (p_i + p_k), & y_{i,k} &= (p_i y_i + p_k y_k) / (p_i + p_k), \\ z_{i,k} &= (p_i z_i + p_k z_k) / (p_i + p_k), & p_{i,k} &= p_i + p_k. \end{aligned}$$

В результате l шагов, на каждом из которых два кластера объединяются в один, получается разбиение на $n - l$ кластеров. На рис. 6.7.1 показан результат кластеризации двумерных объектов.

8. Обучение ЭВМ на примерах. Возможны два пути обучения ЭВМ: 1. Сообщение алгоритма, действуя по которому ЭВМ выполнит требуемые вычисления и решит задачу. 2. Обучение на примерах с последующим распознаванием объектов (букв, звуков, изображений и т.д.): усвоив ограниченное число примеров, ЭВМ правильно распознает новые объекты, которые не предъявляются в процессе обучения.

Пусть имеется множество X из n объектов: $X = (x_1, x_2, \dots, x_n)$, принадлежащее к небольшому числу классов K_1, K_2, \dots, K_m . Автомат (например, ЭВМ) решает задачу распознавания, если он относит объект x_i к соответствующему классу K_j . Рассмотрим рецепторное поле, например, матрицу из r фотодиодов, на которую проецируется изображение объекта.

Если каждый фотодиод может находиться в 2 состояниях, то число всех конфигураций 2^r . Если на j -ом выходе автомата появляется логическая 1, значит объект отнесен к j -ому классу.

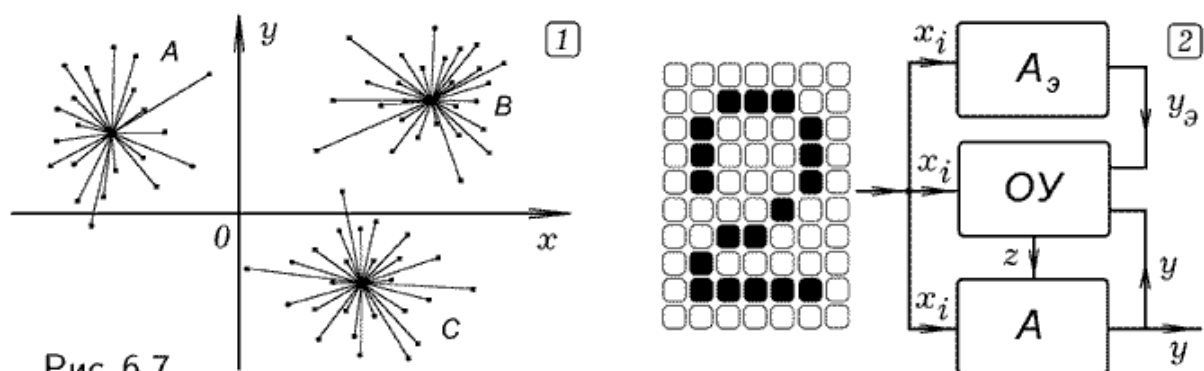


Рис. 6.7.

С целью обучения автомату A путем воздействия на рецепторное поле предъявляются k объектов, случайно выбранных из множества X (рис. 6.7.2). Эталонный автомат A_3 (машина, человек) указывает обучаемому автомату A , к какому классу относится каждый из l предъявленных объектов. Обучающее устройство ОУ сравнивает реакции y_3 эталонного автомата A_3 с реакциями y обучаемого автомата A и так изменяет его внутреннее состояние z , чтобы его реакции возможно чаще совпадали с реакциями A_3 . Затем автоматам предъявляется экзаменационная последовательность объектов $x_{k1}, x_{k2}, \dots, x_{ks}$. Если число ошибок автомата не превышает допустимого уровня, обучение закончено.

Проблема обучения распознаванию объектов может быть решена иначе. Каждую конфигурацию на рецепторном поле, можно изобразить точкой в r -мерном пространстве рецепторов, по осям которого $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_r$. В этом пространстве рецепторов можно построить поверхности $f_i(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_r) = 0$ ($i = 1, 2, \dots, n$), разделяющие его на области, в каждой из которых содержатся точки, изображающие объекты одного класса. При этом объекты, принадлежащие классу X_1 окажутся в области $f_1(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_r) > 0$.

Эффективность обучения зависит: 1) от того, какие именно характеристики подаются на вход A и как они кодируются; 2) от числа возможных состояний автомата; 3) от алгоритма работы обучающего устройства.

9. Обучение вероятностного автомата. Поведение *вероятностного автомата* A характеризуется вероятностями различных реакции (переходов в другое состояние) и действий, наблюдаемых на его выходе, в зависимости от стимула, подаваемого на его вход. Пусть известно конечное число стимулов x_i ($i = 1, \dots, n$) и возможные реакции y_j ($j = 1, \dots, m$), тогда поведение вероятностного автомата определяется матрицей вероятностей p_{ij} . Если множество (y_1, y_1, \dots, y_m) содержит полный набор всех реакций, то сумма соответствующих им вероятностей на каждый стимул (входной сигнал) x_i равна 1. То есть при любом i должно выполняться $p_{i1} + p_{i2} + p_{i3} + \dots + p_{im} = 1$.



ТЬЮРИНГ Алан Матисон (1912–1954 гг.)

После окончания Кембриджского университета занимался уточнением понятия алгоритма, предложил гипотетическую машину, позже названную машиной Тьюринга. Занимался теоретическими проблемами информатики, разрабатывал интерпретаторы для ЭВМ, внес значительный вклад в технологию программирования. Член королевского общества Великобритании.

Каждому отдельному действию автомата (x_i, y_j) ставится оценка α_{ij} , характеризующая эффективность поведения автомата при его реакции y_j на стимул x_i . Если автомат совершил правильное действие, то его "поощряют" высокой оценкой α_{ij} , в результате чего вероятность повторения этого действия увеличивается, а остальных — уменьшается. В случае ошибки автомат "наказывают", что приводит к уменьшению вероятности реакции y_j на стимул (входной сигнал) x_i .

10. Модульная сеть. Существует иной подход к проблеме обучения поведению. Представим себе автомат с электронной "нервной системой", состоящей из отдельных блоков, моделирующих нейроны. Каждый такой искусственный нейрон должен состоять из сумматора и порогового элемента (рис. 6.8.1). Нейрон возбуждается в случае, когда сумма весов синапсов, на которые поступают импульсы раздражения, превышает его порог срабатывания. Обучение автомата может быть осуществлено путем изменения весовых коэффициентов компьютерной модели нейросети.

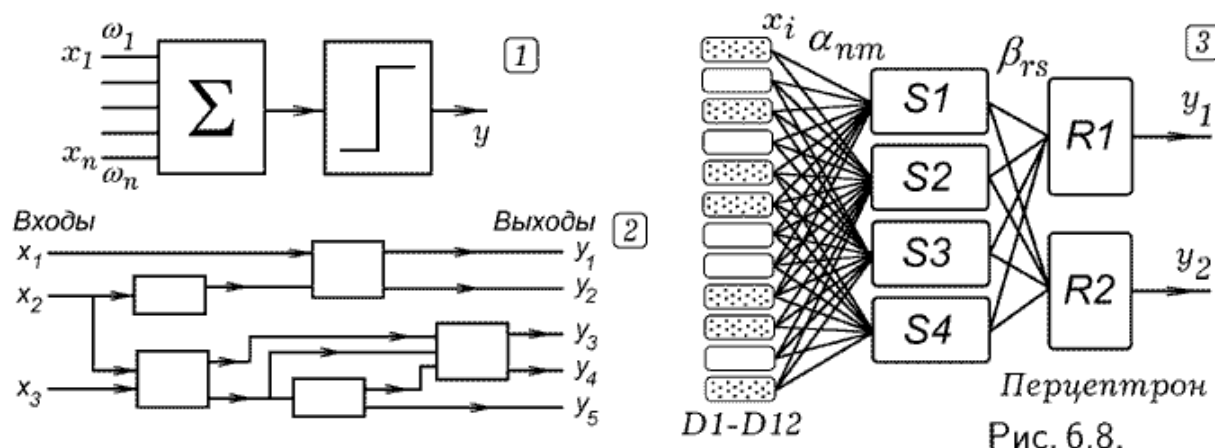


Рис. 6.8.

Формальный нейрон — автомат с n входами x_1, x_2, \dots, x_n и одним выходом y , который характеризуется порогом θ и весами $\omega_1, \dots, \omega_n$. Состояние его выхода в момент $t + 1$ зависит от входов в моменты t : на выходе $y = 1$ в момент $t + 1$ тогда, когда сумма всех весов возбужденных входов в момент t превышает порог срабатывания: $\omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n > \theta$. Если $\omega_i > 0$ означает, что вход возбуждающий, а если $\omega_i < 0$ — вход тормозящий.

Модульная сеть — множество модулей, соединенных так, что выход

одного после разветвления присоединяется к входам других модулей, причем каждый вход соединен не более чем с одним выходом (рис. 6.8.2). Все модули работают синхронно. Пусть модульная сеть состоит из g модулей, имеет n входных и m выходных линий. Каждый вход и выход могут быть возбуждены, поэтому имеется 2^n и 2^m комбинаций для входов и выходов соответственно. Вход и состояние сети в момент t определяют выход и состояние сети в момент $t + 1$. Модульная сеть моделирует нервную систему, состоящую из большого числа нейронов.

Можно доказать, что: 1. Любой конечный автомат может быть заменен модульной сетью. 2. Любая ЦЭВМ является конечным автоматом или модульной сетью.

Примером модульной сети, позволяющей распознавать образы, является перцептрон. Он содержит матрицу фотодатчиков D_i , суммирующие блоки S_i и решающие элементы R_j (рис. 6.8.3). Матрица фотоэлементов связана со всеми суммирующими элементами, а те соединены с решающими элементами. Пока перцептрон не обучен, веса всех связей элементов равны 1. В процессе обучения на фотоэлементы проецируются изображения различных объектов. Суммирующие элементы формируют на выходе сумму сигналов, поступивших на входы, с учетом веса каждой связи, изменяющегося в ходе обучения. Решающие элементы R_j выделяют наибольший или наименьший поступивший на них сигнал и выдают результат распознавания. Если перцептрон неверно распознал объект, то веса связей корректируются до получения правильного результата.

11. Эффективность системы счисления. Экономичность (эффективность) системы счисления характеризуется произведением числа различных символов P на количество разрядов N для выражения какого-то достаточно большого числа M . Чем меньше PN , тем выше эффективность системы счисления. Будем считать, что каждый разряд системы счисления с основанием P требует использования P элементов с одним устойчивым состоянием. Тогда PN равно общему количеству элементов, требующихся для отображения N -разрядного числа M в системе счисления с основанием P . С другой стороны, в N разрядах системы счисления с основанием P можно отобразить $M = P^N - 1$ различных целых чисел. Следовательно, $N = \log_P(M + 1)$. Тогда $PN = P \log_P(M + 1) = P \log_P P^N$. Показателем эффективности, позволяющим сравнить систему счисления с основанием P с двоичной, является коэффициент

$$K(P) = \frac{P \log_P P^N}{2 \log_2 P^N} = \frac{PN}{2N \log_2 P} = \frac{P}{2 \log_2 P}.$$

Учитывая, что основанием системы счисления P может быть любое рациональное число большее 1, исследуем функцию $K = K(P)$.

$$\frac{dK}{dP} = \frac{1}{2 \log_2 P} - \frac{P}{2} \frac{1}{P \ln 2 (\log_2 P)^2} = 0.$$

$$\frac{1}{\log_2 P} = \frac{1}{\ln 2 (\log_2 P)^2}, \quad \ln 2 \log_2 P = 1, \quad \ln P = 1, \quad P = e = 2,72...$$

Итак, самой экономичной системой счисления является система счисления с основанием $e = 2,72...$ То же самое видно из графика зависимости $K = K(P)$ (рис. 6.9.1).

С другой стороны, экономичность системы счисления определяется количеством чисел, которые можно представить в данной системе счисления в заданном количестве разрядов. Если основание системы P , то для отображения чисел в r разрядах требуется устройство с $n = rP$ устойчивыми состояниями, общее количество чисел $N = P^r$. Пусть количество состояний n остается равным 12. В таблице показано количество различных чисел, которые можно отобразить в системах счисления с основанием 12, 6, 4, 3, 2,7, 2,5, 2. Из нее также следует, что наиболее эффективной является система счисления с основанием $e \approx 2,72$. Высокая эффективность у троичной и двоичной систем счисления.

| | | | | | | | |
|------------------|----|------------|------------|------------|--------|------|------------|
| Основание P | 12 | 6 | 4 | 3 | 2,7 | 2,5 | 2 |
| Разрядность r | 1 | 2 | 3 | 4 | 4,4(4) | 4,8 | 6 |
| Кол-во чисел N | 12 | $6^2 = 36$ | $4^3 = 64$ | $3^4 = 81$ | 82,6 | 81,3 | $2^6 = 64$ |

12. Компьютерная модель ученика. Допустим, необходимо исследовать процесс обучения (формирования навыка). Заменяем учащегося вероятностным автоматом, выполняющим последовательность действий в зависимости от входной информации и своего внутреннего состояния. Его алгоритм удобно задать в виде стохастического графа — совокупности вершин, соединенных стрелками, которые соответствуют переходам от одной операции к другой (рис. 6.9.2). Вероятность перехода от i -ой операции к j -ой обозначим через p_{ij} . Все p_{ij} ($i, j = 1, 2, \dots, n$) образуют двумерную матрицу, называемую матрицей вероятностей.

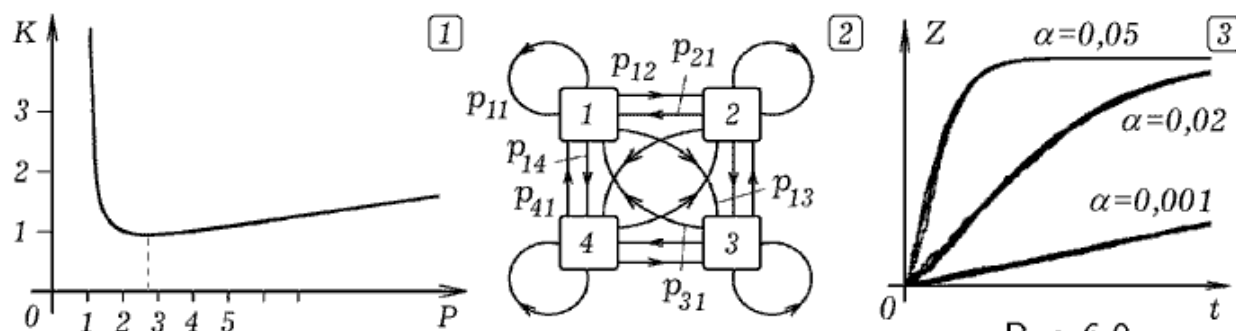


Рис. 6.9.

Если автомат необучен, то все элементы этой матрицы равны, то есть он выбирает каждую следующую операцию совершенно произвольно и после ее выполнения сравнивает свои действия с эталоном (учителем). Учитель подтверждает правильность выбора операции (то есть "поощряет"), или сообщает, что выбор сделан неверно ("наказывает"), подсказывая какую операцию следовало бы выбрать. Все это приводит к тому, что вероятности правильных переходов увеличиваются стремясь к 1, а вероятности неправильных — уменьшаются, приближаясь к 0. В конце обучения автомат практически без ошибок выполняет требуемую последовательность $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow \dots$

Можно создать компьютерную модель ученика, выполняющего последовательность из двух или четырех действий. Программа содержит цикл, в котором выбор каждой операции осуществляется с помощью генератора случайных чисел, их общее количество 500–2000. В случае правильного выбора операции или ошибки происходит соответствующий пересчет матрицы вероятностей так, что вероятность правильных действий увеличивается, а вероятность неправильных — уменьшается. Графики, получающиеся при различных коэффициентах научения α , представлены на рис. 6.9.3. При малом α уровень знаний растет пропорционально времени (количеству выполненных операций), а при большом α достигает насыщения и остается неизменным.

13. Представления о теории управления. *Кибернетика*, по словам Н. Винера, — наука об общих закономерностях процессов управления и связи в сложных системах (живых организмах и машинах). К центральным вопросам кибернетики относятся проблемы автоматизации процессов, самоуправления в сложных системах, создания роботов и машин-автоматов, способных самообучаться и саморазмножаться, автоматизация творческой деятельности.

Одним из важных понятий теории управления является понятие системы. Система — совокупность взаимосвязанных элементов, свойства которой не равны сумме свойств элементов по отдельности. Система имеет структуру — множество внутренних устойчивых связей между элементами, определяющее основные свойства данной системы. Общие свойства систем: целостность (несводимость свойств системы к сумме свойств отдельных элементов), структурность, взаимозависимость системы и среды, множественность описания, иерархичность.

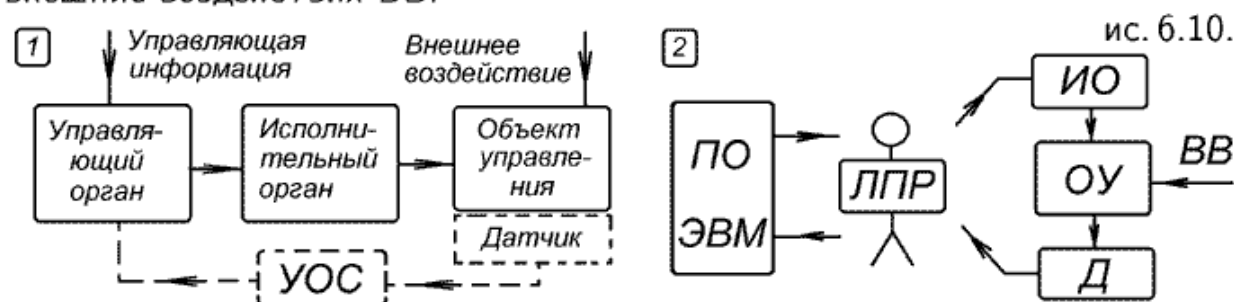
Информационная система обычно состоит из процессора, памяти, устройства ввода (рецептора), устройства вывода (исполнительный элемент), пользователя. *Кибернетическая система* (система управления) включает в себя управляющую систему (управляющий орган УО, исполнительный орган ИО) и объект управления ОУ. *Управление* — процесс целенаправленного воздействия на объект управления, приводящий к преобразованию информации, энергии или материи. *Контроль* — это измерение физической величины, характеризующей состояние объекта, и фиксация результатов в удобной форме. Системы автоматики подразделяются на автоматические, то есть функционирующие без участия человека, и автоматизированные, управляемые человеком. Различают разомкнутые и замкнутые системы автоматического управления (САУ) (без эксперта), и автоматизированные системы управления (АСУ) (с экспертом).

В *разомкнутой системе управления* (рис. 6.10.1) на основе входной управляющей информации управляющий орган УО посылает сигналы исполнительному органу ИО, который оказывает воздействие на объект управления ОУ. При этом различают: 1) системы автоматического управления: устройство управления — исполнительное устройство — объект; 2) системы автоматического контроля: объект — измерительное устройство — регистрирующее устройство. Автоматы с разомкнутой системой не

способны к адаптации, у них нет возможности самонастраиваться. Это обусловлено тем, что управляющий орган не получает информации от объекта управления. При воздействии помех и непредвиденных факторов они становятся неработоспособными

Также существуют *следающие системы*, осуществляющие слежение за уровнем сигнала, который поступает с датчика. В зависимости от его величины следящая система оказывает такое воздействие на исполнительный механизм, что сигнал с датчика удерживается внутри заданного интервала. Например, устройство, управляющее процессом фокусировки при колебаниях фото- и видеокамеры. *Пороговые автоматы* работают так: при достижении сигналом некоторого уровня автомат начинает выполнять определенную последовательность действий. Примером является терморегулятор, который при понижении температуры до заданного значения на определенное время включает нагреватель и подает сигнал.

Замкнутые САУ (рис. 6.10.1) имеют обратную связь, которая состоит из датчика Д и устройства обработки сигнала УОС. ОУ воздействует на датчик, тот посылает сигнал через УОС на управляющий орган УО. Все это происходит в соответствии с принципом регулирования: устройство управления осуществляет непрерывное или периодическое отслеживание заданной совокупности параметров объекта и воздействует на него с учетом их рассогласования. На объект управления ОУ также влияют внешние воздействия ВВ.



Автоматизированная система управления (АСУ) — человеко-машинная система, включающая технические, программные средства и эксперта (лицо, принимающее решение — ЛПР), обеспечивающего управление социальным или технологическим процессом (рис. 6.10.2). Эксперт (ЛПР), основываясь на сигналах от датчика Д о состоянии объекта управления ОУ и информации, идущей от ЭВМ, принимает решение и управляет исполнительным органом ИО, который воздействует на объект управления ОУ. Изменения состояния объекта контролируются датчиком Д.

6.2. ГРАФЫ, ДЕРЕВЬЯ И ЖАДНЫЕ АЛГОРИТМЫ

1. Основы теории графов. Многие задачи, связанные с изучением сложных систем, приводят к построению и анализу графов. *Граф* $G = (V, E)$ — это совокупность *вершин* $V = (v_1, v_2, \dots, v_n)$, которые связаны между собой *ребрами* $E = (e_1, e_2, \dots, e_m)$ (рис. 6.11.1). Если граф *ориентированный*, то на ребрах ставятся стрелки, указывающие направление (рис. 6.12.2). Каждое ребро соединяет две вершины, либо образует *петлю*, возвращаясь в ту же вершину из которой вышла. Если два или несколько ребер соединяют две одни и те же вершины, то такие ребра называются *параллельными*. *Связным* называется граф, двигаясь по ребрам которого можно из любой вершины попасть в любую другую вершину.

Связный граф без циклов называется *деревом*. На рис. 6.11.1 показан *лес* — совокупность деревьев (несвязный граф без циклов). *Ориентированное дерево* — ориентированный ациклический граф, у которого: 1) имеется один узел (корень), в который не входит ни одно ребро; 2) в каждую вершину входит 1 ребро; 3) из корня можно достичь каждую вершину по одному пути. *Бинарным* или *двоичным* деревом называется упорядоченное дерево, из каждой вершины которого выходит не более двух ребер, одно — правое, другое — левое (рис. 6.11.2). Если левым вершинам поставить в соответствие 0, а правым — 1, то каждому пути из корня будет отвечать двоичное слово.

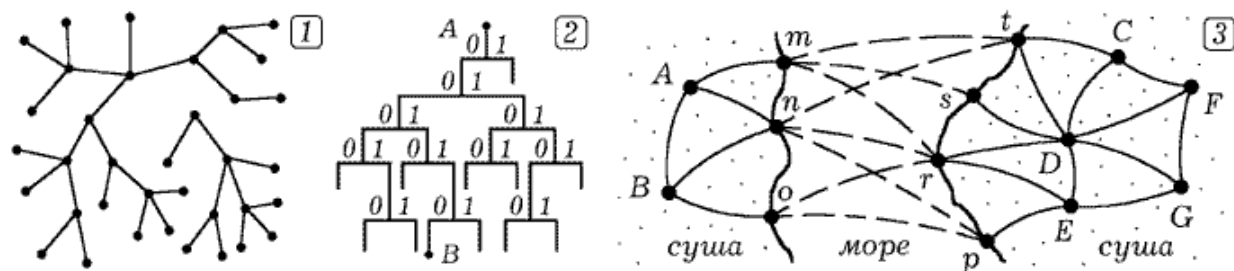


Рис. 6.11.

К графам сводятся целый ряд задач, например, задача о выборе оптимальных путей транспортировки товаров, хранящихся в пунктах А и В, которые необходимо переправить в пункты F и G, при известных затратах на перевозку 1 кг груза по суше и морю.

Если вершинам графа и (или) ребрам поставлены в соответствие некоторые коэффициенты w_1, w_2, \dots, w_m (рис. 6.12.1), называемые *весами*, то граф называется *взвешенным*. Весом для ребер могут быть пропускная способность дороги, канала связи, сопротивление ветви электрической цепи, стоимость перевозки груза и т.д. Если ребро e_k соединяет вершины v_i и v_j , то говорят, что ребро e_k *инцидентно* вершинам v_i и v_j . Сами вершины при этом называются *смежными*. Чтобы проанализировать граф на ЭВМ его необходимо разметить, то есть пронумеровать вершины, и задать с помощью следующих алгебраических матриц:

1. Матрица A смежности графа $G = (V, E)$ — это квадратная матрица размером $n \times n$, элементы которой определяются так:

$$a_{i,j} = \begin{cases} 1, & \text{если вершины } v_i, v_j \text{ смежные;} \\ 0, & \text{если вершины } v_i, v_j \text{ несмежные.} \end{cases}$$

Сумма элементов матрицы смежности по строкам или столбцам равна числу входящих и исходящих ребер.

2. Матрица B инцидентности графа $G = (V, E)$ — это матрица размером $n \times m$, элементы которой определяются так:

$$b_{i,j} = \begin{cases} 1, & \text{если вершина } v_i, \text{ инцидентно ребру } e_j; \\ 0, & \text{если вершина } v_i, \text{ неинцидентна ребру } e_j. \end{cases}$$

Для ориентированного графа матрица инцидентности содержит элементы, определяемые так:

$$b_{i,j} = \begin{cases} 1, & \text{если из вершины } v_i, \text{ выходит ребро } e_j; \\ 0, & \text{если вершина } v_i, \text{ неинцидентна ребру } e_j; \\ -1, & \text{если в вершину } v_i, \text{ входит ребро } e_j. \end{cases}$$

3. Матрица C весов ребер графа $G(V, E)$ — это прямоугольная матрица $n \times m$, каждый элемент $c_{i,j}$ которой равен весу $w_{i,j}$ ребра, соединяющего вершину v_i с v_j .

Матрицы смежности A , матрица инцидентности B и матрица весов C для неориентированного графа на рис. 6.12.1 представлены ниже:

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 12 & 35 & 48 \\ 12 & 0 & 23 & 0 \\ 35 & 23 & 0 & 18 \\ 48 & 0 & 18 & 0 \end{pmatrix}$$

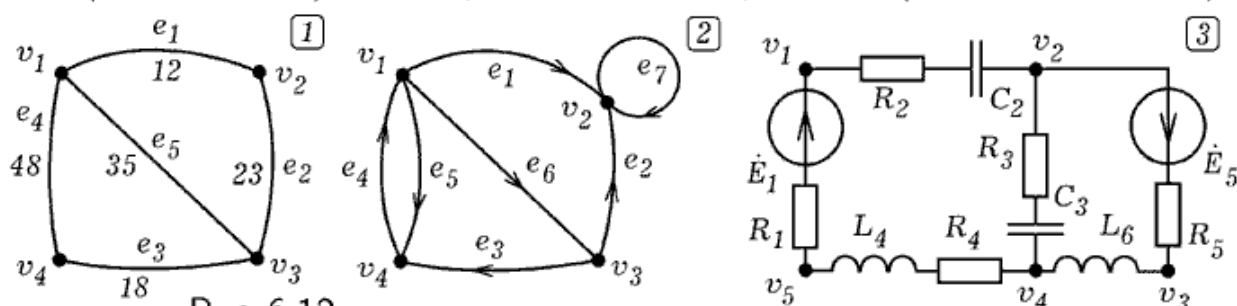


Рис. 6.12.

Рассмотрим электрическую цепь, изображенную на рис. 6.12.3. Она может быть представлена в виде графа, вершинами которого будут узлы цепи, а ребрами — ее ветви. Для того, чтобы ввести параметры цепи в ЭВМ следует записать матрицу весов ребер, которая в данном случае будет матрицей сопротивлений или импедансов ветвей:

$$\dot{Z} = \begin{pmatrix} 0 & R_2 + \dot{X}_{C2} & 0 & 0 & R_1 \\ R_2 + \dot{X}_{C2} & 0 & R_5 & R_3 + \dot{X}_{C3} & 0 \\ 0 & R_5 & 0 & \dot{X}_{L6} & 0 \\ 0 & R_3 + \dot{X}_{C3} & \dot{X}_{L6} & 0 & R_4 + \dot{X}_{L4} \\ R_1 & 0 & 0 & R_4 + \dot{X}_{L4} & 0 \end{pmatrix}$$

Аналогичным образом может быть задана матрица, характеризующая распределение источников ЭДС.

2. Представления о теории игр. Рассмотрим две системы S_1 и S_2 . Возможна конфликтная ситуация, когда определенные факторы, влияющие на систему S_1 с управляющим устройством Q_1 , зависят от действий управляющего устройства Q_2 системы S_2 , причем "интересы" систем S_1 и S_2 противоположны, то есть улучшение функционирования S_1 приводит к ухудшению функционирования S_2 и наоборот (рис. 6.13.1). Теория игр занимается разработкой стратегии, при которой управляемая система S_1 находилась бы в более благоприятном положении даже при самых неблагоприятных действиях противостоящей ей системы S_2 (противника).

Игра — упрощенная формализованная модель конфликтной ситуации. Она ведется по определенным правилам, то есть права и обязанности игроков, что считать выигрышем. Если два игрока, — *парная игра*, если много — *множественная*. Участники множественной игры могут образовывать постоянные или временные коалиции. Действия участников — *ходы*. Случайные ходы осуществляются механизмом случайного выбора, личные — игроком. Выигрыш игрока в результате одного хода называется *ценой игры*. *Игра с нулевой суммой* — игра, в которой алгебраическая сумма выигрышей всех сторон равна 0, то есть выигрыш одной стороны возможен за счет проигрыша другой. *Игра с полной информацией* — игра, в которой каждый игрок при каждом личном ходе знает всю предисторию ее развития, то есть личные и случайные ходы. Например, шахматы, шашки, крестики-нолики и т.д.

Рассмотрим парные игры с нулевой суммой. Ограничимся рассмотрением выигрыша 1 игрока, который он стремится максимизировать, а противник минимизировать. Оба противника знают возможные варианты своих и чужих действий и размер выигрыша при любом парном их сочетании. Такие игры представляют в виде *платежной матрицы игры*, каждый элемент которой a_{ij} указывает размер выигрыша игрока А, если он выбирает действие A_i , а его противник — действие B_j . Матрица игры содержит m строк и n столбцов по числу действий у игрока А и В соответственно (рис. 6.13.2).

3. Игра $m \times n$. Решить игру $m \times n$, значит найти такую стратегию, при которой средний выигрыш за большое число игр максимален. *Стратегия минимакса*: следует выбирать действие, при котором получается максимально возможный выигрыш в случае наименее благоприятного действия противника. Оптимальное действие A_k игрока А определяется из матрицы игры путем отыскания элемента $a_{kp} = \max_i \min_j a_{ij}$, который максимален по строкам и минимален по столбцам.

Основная теорема: любая конечная игра двух лиц с нулевой суммой имеет не менее одно решение (две оптимальных стратегий с ценой v).

Рассмотрим игру $G(4 \times 5)$ в матричной форме. Пусть у нас (игрок А) будет 4 стратегии, у противника (игрок В) — 5 (рис. 6.13.3). Нельзя выбирать стратегию A_3 с максимальным выигрышем 9, — игрок В в этом

случае выберет стратегию B_3 и выигрыш будет равен 1. Исходя из принципа осторожности, надо выбрать стратегию, при которой наш минимальный выигрыш максимален. Это *принцип минимакса*: поступай так, чтобы при наихудшем для тебя поведении противника получить максимальный выигрыш. Это стратегия A_4 , гарантированный выигрыш (нижняя цена игры) равен 3. Исходя из таких же соображений, противник выберет B_3 , так как при этом его максимальный проигрыш минимален и равен 6.

Минимаксные стратегии неустойчивы по отношению к информации о поведении другой стороны; они не обладают свойством равновесия. Если бы мы знали, что противник придерживается стратегии B_3 , то выбрали бы стратегию A_1 .

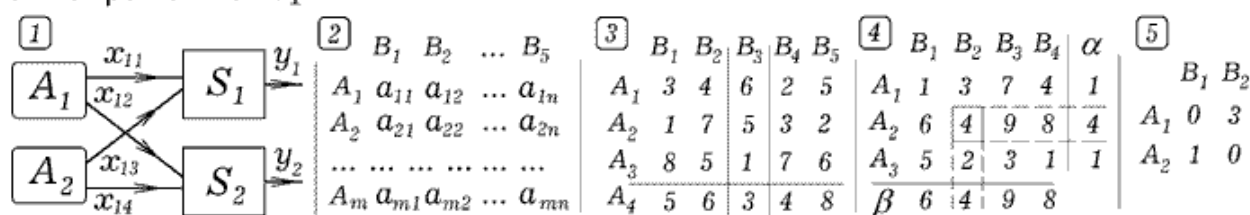


Рис. 6.13.

Кроме стратегии минимакса существуют стратегия минимаксного риска Сэвиджа (выбирается ход, при котором риск в наихудших условиях минимален), стратегия пессимизма–оптимизма Гурвица (задается коэффициент оптимизма, который определяет величину риска) и т.д.

Рассмотрим матрицу игры (рис. 6.13.4) и найдем максимальные элементы α по строкам, и минимальные по столбцам β . Видно, что нижняя и верхняя цены игры $\max \alpha$ и $\min \beta$ равны 4. Это *седловая точка* матрицы игры (минимум по одной координате, максимум по другой). Чтобы выиграть, игроки должны придерживаться минимаксной стратегии, которая является *оптимальной чистой стратегией*.

Теорема. Каждая игра с полной информацией имеет седловую точку.

Если матрица игры не имеет седловой точки, то не существует оптимальных чистых стратегий для каждого игрока, игра имеют более сложное решение. Рассмотрим игру 2 X 2 без седловой точки (рис. 6.13.5): игрок А атакует одно из двух оборонительных сооружений, игрок В может защищать только одно из двух сооружений. Одно сооружение в 3 раза важнее другого. Платежная матрица игры не имеет седловой точки.

Если игрок А придерживается какой-то одной стратегии, то игрок В, поняв это, будет его выигрывать. Стратегия, $S_A(p_1, p_2, \dots, p_n)$ состоящая в случайном применении с определенными вероятностями нескольких чистых стратегий, называется *смешанной стратегией*. При соответствующем подборе вероятностей p_j смешанная стратегия становится выигрышной или оптимальной.

Теорема: Если игра не имеет седловой точки, то игрок, играющий по определенной стратегии вероятнее всего проиграет игроку, который меняет стратегию случайным образом.

Если один игрок — ЭВМ, то она перебирает всевозможные варианты, каждый раз вычисляя оценочную функцию $y = F_0(x_1, x_2, \dots)$, где y — ценность данной позиции с точки зрения ее близости к цели, аргументы

x_i — факторы, учитываемые при оценке позиции. Для просчета игры в шахматы на 4 хода нужно перебрать около 10^{12} вариантов. Случайность выбора достигается с помощью генератора случайных чисел.

4. Некоторые задачи, решаемые на ЭВМ. *Исследование операций* — область математики, исследующая методы поиска наиболее оптимальных решений путем использования количественных методов. Это возможно в тех случаях, когда решение и учитываемые факторы могут быть записаны в числовой форме или форме предпочтений. Под *операцией* понимают последовательность действий, направленная на достижение конкретной цели. На развитие ситуации влияют неконтролируемые факторы, которые известны, но не зависят от лица, принимающего решение (ЛПР), и контролируемые факторы x_1, x_2, \dots, x_n , зависящих от ЛПР. Критерий оптимальности выражается в виде целевой функции $f(x_1, x_2, \dots, x_n)$, которую необходимо максимизировать.

Существуют также многокритериальные задачи, для которых необходимо вводить несколько целевых функций $f_j(x_1, x_2, \dots, x_n)$, $j = 1, 2, \dots, m$. Они решаются путем сведения к однокритериальной задаче путем свертки, в результате которой создается целевая функция $F = F(f_1, f_2, \dots, f_m)$. Часто рассматривают линейную комбинацию $F = \alpha_1 f_1 + \alpha_2 f_2 + \dots + \alpha_m f_m$, где α_i — весовые коэффициенты.

Основная задача линейного программирования может быть сформулирована так: найти неотрицательные (иногда целочисленные) значения переменных x_i , $i = 1, 2, \dots, n$, удовлетворяющие ограничениям:

$$a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jn}x_n \leq b_j, \quad j = 1, 2, \dots, m,$$

при которых целевая функция $f(x_1, x_2, \dots, x_n)$ достигает максимума. Ниже рассмотрены несколько типичных задач линейного программирования.

Задача о производстве. Необходимо произвести продукцию n видов в количествах x_i , $i = 1, 2, \dots, n$. Это требует m видов ресурсов, расход которых на производство каждого вида продукции составляет a_{ij} , а запасы равны b_j , $j = 1, 2, \dots, m$. Доход от продажи i -ого вида продукции равен d_i . Необходимо организовать производство так, чтобы доход был максимальным.

Задача сводится к нахождению значений x_i , $i = 1, 2, \dots, n$, для которых целевая функция

$$F(x_1, x_2, \dots, x_n) = d_1x_1 + d_2x_2 + \dots + d_nx_n$$

была максимальной и при этом выполнялись бы ограничения на расход ресурсов:

$$a_{1j}x_1 + a_{2j}x_2 + \dots + a_{nj}x_n \leq b_j, \quad j = 1, 2, \dots, m.$$

Транспортная задача. Имеются n пунктов отправления A_i , в которых сосредоточены грузы одного типа, и m пунктов назначения B_j . Известно сколько грузов находится в пунктах отправления (a_1, a_2, \dots, a_n) и сколько и какие грузы необходимо перевести в пункты назначения (b_1, b_2, \dots, b_m) , причем сумма заявок равна сумме запасов. Стоимость перевозок каждого груза от пункта A_i в пункт B_j задается матрицей c_{ij} . Необходимо так

составить план перевозок, чтобы общие затраты были бы минимальными.

Пусть x_{ij} — количество единиц грузов, отправляемых из i -ого пункта отправления A_i в j -ый пункт назначения B_j . Если считать, что стоимость перевозки грузов пропорциональна их количеству, то суммарная стоимость всех перевозок равна

$$F = c_{11}x_{11} + c_{12}x_{12} + \dots + c_{1m}x_{1m} + \dots + c_{n1}x_{n1} + \dots + c_{nm}x_{nm}.$$

Это и есть целевая функция, которую необходимо минимизировать. С другой стороны план перевозок x_{ij} , $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$ должен удовлетворять следующим ограничениям. Так как суммарное количество грузов, отправляемых от A_i во все пункты назначения равно запасу a_i в A_i , то:

$$x_{i1} + x_{i2} + \dots + x_{im} = a_i, \quad i = 1, 2, \dots, n.$$

Суммарное количество грузов, доставляемых в B_j из всех пунктов отправления, должно быть равно заявке b_j :

$$x_{1j} + x_{2j} + \dots + x_{nj} = b_j, \quad j = 1, 2, \dots, m.$$

Задача о рюкзаке. Имеются n товаров T_1, T_2, \dots, T_n . Вес i -ого товара T_i равен a_i и имеет стоимость c_i . Требуется загрузить рюкзак (грузовик, самолет, корабль) так, чтобы суммарная стоимость всех загруженных товаров была максимальна, а их масса не превосходила бы допустимого предела M .

Пусть x_1, x_2, \dots, x_n — число загруженных товаров T_1, T_2, \dots, T_n . Суммарная стоимость всех товаров

$$F(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

является целевой функцией. Задача сводится к нахождению таких целых значений $x_i \geq 0$, что

$$a_1x_1 + a_2x_2 + \dots + a_nx_n \leq M,$$

при которых функция F максимальна.

Задача о пищевом рационе. Дневной рацион некоторого животного включает в себя n компонентов K_1, K_2, \dots, K_n , взятых в количестве x_1, x_2, \dots, x_n соответственно. Известно, что стоимость 1 кг компонента K_i равна c_i . Рацион должен содержать не менее b_1 белков, b_2 жиров, b_3 углеводов, b_4 витаминов. В 1 кг компонента K_i содержатся a_{i1} белков, a_{i2} жиров, a_{i3} углеводов, a_{i4} витаминов. Необходимо определить количество каждого i -ого компонента x_i так, чтобы стоимость рациона была минимальной.

Целевой функцией является стоимость всего рациона

$$F(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n.$$

Решение задачи сводится к нахождению вектора (x_1, x_2, \dots, x_n) , элементы которого удовлетворяют ограничениям:

$$a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + \dots + a_{in}x_n \geq b_i, \quad x_i \geq 0, \quad i = 1, 2, 3, 4.$$

При этом целевая функция F должна быть минимальной.



ЛЯПУНОВ Алексей Андреевич (1911–1973 гг.)

Руководил первыми в СССР исследованиями в области кибернетики, разрабатывал общие и математические основы этой науки. Научные интересы: вычислительные машины, теория алгоритмов, программирование, машинный перевод, методологические аспекты развития науки. Создал учебные курсы программирования, разработал операторный метод, занимался вопросами компьютерной лингвистики.

Другой достаточно обширный класс задач касается *систем массового обслуживания*, к которым относятся телефонные станции, билетные кассы, компьютерные сети и другие системы, предназначенные для обработки потока заявок, поступающих случайным образом по каналам обслуживания. При этом различают системы с отказами и системы с очередью.

Система с отказами: при поступлении заявки в систему в момент t , когда все каналы заняты, заявка получает отказ. Например, вы позвонили, а телефон абонента занят, вы получаете отказ и необходимо снова звонить. *Система с очередью*: заявка, поступившая в момент t , когда все каналы заняты, встает в очередь и ждет, пока не освободится один из каналов. Различают системы с неограниченной очередью и с ограниченной очередью, при этом возможно обслуживание с приоритетом. Эффективность обслуживания характеризуется средней длиной очереди, средним временем ожидания, средним доходом.

5. Жадный алгоритм. Рассмотрим алгоритм, состоящий в принятии на каждом шаге локально оптимальных решений, в предположении, что конечное решение также будет оптимальным. Он называется *жадным*. Например, имеется большой ковер, которым необходимо покрыть пол в комнате. Использование жадного алгоритма требует на каждом шаге покрывать большую часть пола, а остальное отрезать. Алгоритм кодирования Хаффмана также является жадным.

Жадный алгоритм является эффективным, так как увеличение числа исходных данных приводит к пропорциональному росту объема вычислений. Однако он не всегда приводит к оптимальному решению. Например, рассмотрим задачу *коммивояжера*: торговец должен обойти все города по одному разу так, чтобы общая длина пути была минимальной, если расположение городов и дорог известны. Применение жадного алгоритма требует, чтобы сначала коммивояжер пошел в ближайший город, потом в другой ближайший город и т.д., что не является в общем случае оптимальным решением. На последних этапах коммивояжеру придется расплачиваться за "жадность" использованного им алгоритма.

К оптимизационной задаче принцип жадного выбора применим лишь тогда, когда последовательность локально оптимальных (жадных) выборов дает оптимальное решение в глобальном смысле. Если мы хотим показать, что применение к данной задаче жадного алгоритма приведет к

оптимальному решению, то необходимо доказать, что: 1) жадный выбор на первом этапе не закрывает путь к оптимальному решению, то есть для любого достаточно оптимального решения есть более приемлемое решение, согласующееся со сделанным жадным выбором; 2) подзадача, к которой свелась исходная проблема после осуществления жадного выбора аналогична исходной.

Рассмотрим задачу о рюкзаке: на складе имеются различные товары известной массы и стоимости. Необходимо так загрузить рюкзак (машину, корабль, самолет) известной грузоподъемности, чтобы за один рейс перевести товар как можно большей стоимости. Жадный алгоритм, состоит в следующем: 1) упорядочить товары по удельной стоимости, то есть стоимости в пересчете на 1 кг; 2) загружать в рюкзак товары с наиболее высокой удельной стоимостью. Его использование не всегда приводит к самому оптимальному решению.

Вот еще один пример. Пусть в кассе имеются монеты достоинством $a_1 = 1 < a_2 < \dots < a_n$. Как выдать сумму S , используя наименьшее число монет? Жадный алгоритм: 1) находят наибольшее число k_n монет достоинством a_n так, что $ka_n < S$; 2) вычисляется остаток $S_1 = S - ka_n$, он должен не превышать a_n ; 3) находят наибольшее число k_{n-1} монет a_{n-1} так, что $k_{n-1}a_{n-1} < S_1$ и т.д. Решение не всегда бывает оптимальным: сумма 12 копеек монетами в 1, 5 и 7 копеек выдается так: 1 монета — 7 коп., 5 монет — 1 коп. Оптимальный вариант: 2 монеты — 5 коп., 2 монеты — 1 коп.

Альтернативой жадному алгоритму является алгоритм *динамического программирования*, заключающийся в переборе всех разумных или возможных вариантов решения задачи. Например, прокладывается дорога между двумя пунктами. Как провести дорогу, чтобы затраты были минимальными, если местность содержит болота, леса и т.д.? Построим прямоугольную сетку, разбив местность на большое число узлов и определим стоимость участка дороги, соединяющего каждую пару соседних узлов. Затем, перебирая все возможные последовательности узлов, соединяющие населенные пункты, определяют как должна строиться дорога для того, чтобы затраты были бы минимальными. Аналогично решается задача с выбором оптимальных путей перевозки товаров (рис. 6.11.3).

Если жадный алгоритм берет "самый жирный кусок" на каждом шаге, то алгоритм динамического программирования предполагает перебор всевозможных ситуаций и вариантов и выбор из них наиболее оптимального. С другой стороны алгоритм динамического программирования не является эффективным: при увеличении числа n городов в задаче о коммивояжере число возможных вариантов, а значит и объем вычислений, возрастает не пропорционально n , а по экспоненте. Поэтому в задачах, для которых жадный алгоритм дает почти всегда оптимальное решение, используют именно его.

6. P- и NP-задачи. Рассмотрим множество однотипных алгоритмически разрешимых задач. Если машина Тьюринга решает задачу за полиномиальное время $t(n) = O(p(n))$, где $p(n)$ — некоторый полином от

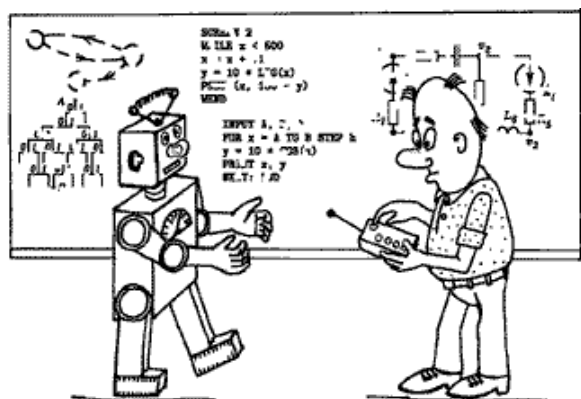
общего числа входных данных, то это задача P -типа. К этому классу относится подавляющее большинство задач, заключающихся в вычислении той или иной функции. Например, задача о расчете распределения температуры на поверхности пластины, — время ее решения пропорционально количеству узлов сетки, используемой для аппроксимации $T = T(x, y)$.

Наряду с задачами P -типа существует огромный класс математических проблем, решаемых не за полиномиальное, а за экспоненциальное время $t(n) = O(n^{\log_2 n})$. Это так называемые NP -задачи. Перечислим некоторые из них: 1) расшифровка сообщения путем подбора ключа; 2) нахождение простых делителей достаточно большого числа; 3) нахождение целочисленных значений переменных x_1, x_2, \dots, x_n , превращающих уравнение $f(x_1, x_2, \dots, x_n) = a$ в истинное высказывание; 4) ответ на вопрос: существуют ли в заданном множестве целых чисел такие, что их сумма равна 0? и т.д. NP -полной называется задача, к которой за полиномиальное время сводится любая NP -задача данного типа.

Рассмотрим NP -задачу о нахождении целых значений x_1, x_2, x_3 , обращающих уравнение $4x_1^3x_2 + x_2^2x_3 + 3x_1x_3^2 + 2x_1x_2^2 = 4272$ в истинное высказывание. Решить ее не просто, но нам известен ответ или сертификат (удостоверение) задачи: $x_1 = 5, x_2 = 7, x_3 = 3$. Подставляя известный сертификат в исходное равенство, мы уже решаем P -задачу и убеждаемся в том, что данная NP -задача имеет решение и им является приведенная выше тройка чисел. Возможно лишь подтвердить предположение о существовании решения, но нельзя его опровергнуть, даже если исходное равенство при подстановке конкретного сертификата не выполняется.

Другой пример, — имеется множество $\{-2, 3, 4, -6, 5, -1, \dots\}$ из n элементов. Необходимо ответить на вопрос: найдутся ли в нем такие k элементов множества, что их сумма равна 0. Чтобы решить эту NP -задачу необходимо перебрать все возможные суммы из k элементов. Как и в предыдущем случае, увеличение количества исходных данных приводит к экспоненциальному росту числа шагов, а значит и времени решения задачи. Знание ответа (сертификата) позволяет быстро удостовериться в существовании решения.

Проблема равенства классов P - и NP -задач ($P=?NP$) не решена до сих пор. Ее можно сформулировать иначе: если положительный ответ на данный вопрос быстро проверяется за полиномиальное время, то возможно ли быстро найти ответ на этот вопрос? Что проще: проверить задачу или ее решить?



ГЛАВА 7 ПРИКЛАДНАЯ ИНФОРМАТИКА

Безусловно, каждый уважающий себя начинающий компьютерщик должен разбираться в сумматорах и дешифраторах, знать формулу Шеннона и кое-что понимать в теории алгоритмов, уметь свободно обсуждать проблемы искусственного интеллекта и объяснять сущность частотной модуляции... Однако этого недостаточно! Еще необходимо ориентироваться в современных информационных технологиях, уметь устанавливать операционную систему и другое программное обеспечение, разбираться в текстовых, графических и видеоредакторах, электронных таблицах и базах данных. Наконец, нужно программировать на алгоритмических языках, решать математические задачи на компьютере. Пол Аллен и Билл Гейтс тоже начинали с Бейсика ...

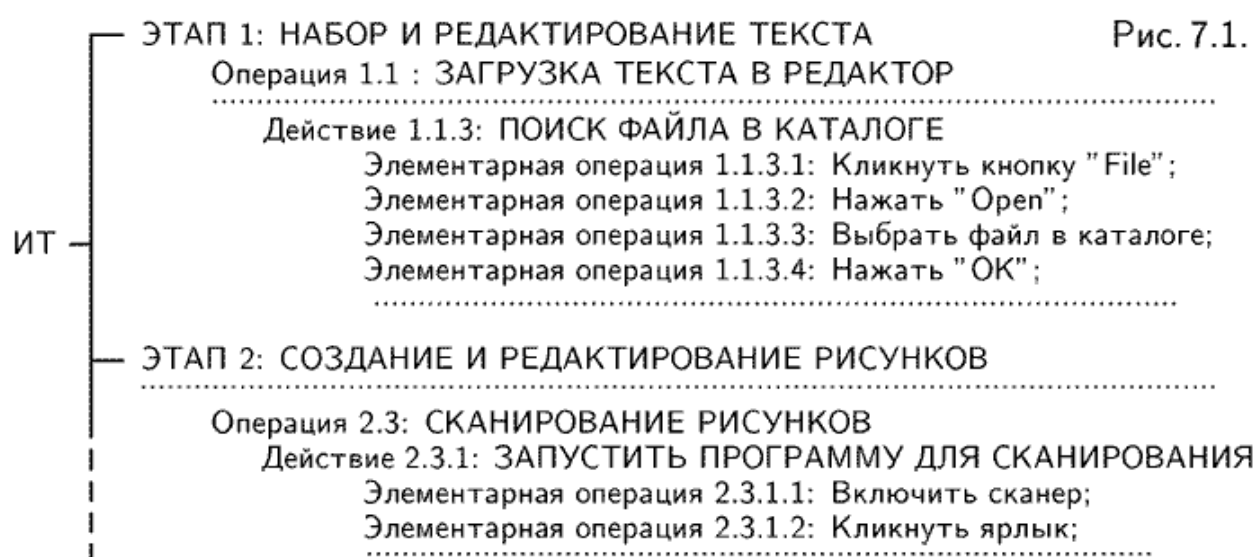
7.1. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

1. Основные понятия. Представим себе менеджера, к которому ежедневно поступают отчеты от подчиненных о состоянии бизнеса. На основе этих отчетов он заполняет сводную таблицу, строит диаграммы, графики и пишет свой отчет, создавая тем самым новый *информационный продукт*. Процесс создания нового информационного продукта называется *информационной технологией*. Ее результатом может быть не только устный, письменный или электронный отчет, фотография или видеоролик, но и телефонограмма, радиogramма, телевизионная передача и т.д.

Информационная коммуникационная технология (ИКТ) — система действий, связанная со сбором, хранением, обработкой и передачей информации с помощью ЭВМ, в результате которых получается новый информационный продукт либо осуществляется передача сообщений на большие расстояния. ИКТ является результатом функционирования *информационной системы*, в общем случае состоящей из нескольких экспертов (пользователей) и электронных устройств (компьютеров, сотовых телефонов, телевизионных передатчиков и приемников, видеокамер и т.д.). Особенности ИКТ являются: 1) интерактивный (диалоговый) режим работы с компьютером (смартфоном, телевизором и т.д.); 2) интегрированность (взаимосвязь) с другим ПО; 3) гибкость изменения данных и задач.

ИКТ имеет иерархическую структуру (рис. 7.1) и подразделяется на этапы, реализующие длительные технологические процессы. Этапы делятся на операции, которые позволяют, например, создавать конкретный объект в данной программной среде. Операции состоят из действий, изменяющих содержание экрана. Действия состоят из элементарных операций по управлению мышью и клавиатурой.

Перечислим примеры использования ИКТ: редактирование текстов и рисунков, обработка аудио и видеоинформации, машинный перевод, поиск информации с помощью информационно-поисковой системы, числовое программное управление работой станка, создание чертежей с помощью системы автоматизированного проектирования (САПР), информационная поддержка принятия решения с помощью экспертной системы (ЭС) или системы управления базой данных (СУБД), обучение с помощью ЭВМ, телеграф, радио- и телефонная связь, телевидение, передача информации через локальные и глобальные компьютерные сети.



2. Виды программного обеспечения. Все программное обеспечение (ПО) делится на следующие классы: 1. *Системные программы:* средства контроля и диагностики, ОС, оболочки, драйверы, утилиты, антивирусы; 2. *Прикладные программы:* офис, текстовые и графические редакторы, электронные таблицы, базы данных, математические пакеты, системы автоматизированного проектирования; 3. *Инструментальные системы программирования* для создания новых программ; 4. *Специальные программы*, созданные пользователем.

Приведем примеры прикладного ПО: текстовый редактор (процессор) Word; графический редактор Paint, Image; программы для создания анимации и видеофильмов 3D Studio, Adobe Premiere; электронные таблицы (табличный процессор) Excel; издательская система PageMaker; система управления базами данных Access; программы для подготовки презентаций PowerPoint; программы экономического назначения "1С: Бухгалтерия"; системы автоматизированного проектирования САПР "Компас", AutoCad; математические пакеты, программы для статистического ана-

лиза данных и моделирования; компьютерные игры, обучающие программы, электронные словари, энциклопедии.

3. ИТ обработки данных и поддержки принятия решения. Допустим, эксперт руководит каким-то сложным производственным процессом. Для того, чтобы принять то или иное решение и нажать на соответствующую кнопку ему необходима информация. Для ее получения используется *ИТ обработки данных и поддержки принятия решения*.

ИТ обработки данных состоит из следующих этапов: 1. *Сбор данных*. Датчики преобразуют изменения исследуемого объекта в электрические (или иные) сигналы, которые периодически измеряются и автоматически либо вручную заносятся в ЭВМ. 2. *Обработка данных*. С целью установления изучаемых закономерностей первичные данные классифицируются и группируются, сортируются и упорядочиваются, подвергаются математической обработке. 3. *Хранение данных* осуществляется на всевозможных внешних ЗУ. 4. *Создание отчетов* предполагает написание и распечатку на принтере бумажного документа или подготовку презентации. Отчет может содержать сводные таблицы, графики, диаграммы, презентации и т.д., полученные в результате обработки первичных данных и представляющие их в более удобном и компактном виде.

ИТ поддержки принятия решения требует качественно иного взаимодействия человека и компьютера. В процессе выработки решения участвуют: 1) ЭВМ с установленным на ней ПО, включающей экспертную систему (ЭС), программы, моделирующие управляемый технологический (или иной) процесс; 2) эксперт (ЛПР), задающий исходные данные, оценивающий результат вычислений ЭВМ и принимающий решение.

ЭС состоит из следующих компонентов: интерфейс пользователя, модуль базы знаний (БЗ), модуль логического вывода, модуль создания системы. *Интерфейс пользователя* обеспечивает ввод информации и команд в ЭС и получение выходной информации. *База знаний БЗ* содержит факты, описывающие предметную область, их логическую взаимосвязь и систему правил. Каждое правило состоит из двух частей: условия, и действия, которое следует произвести при выполнении условия. *Модуль логического вывода* — часть ЭС, производящая обработку знаний. В нем последовательно рассматриваются правила, проверяются их условия и осуществляются соответствующие действия, приводящие к решению проблемы. *Модуль создания системы* или *оболочка ЭС* — готовая программная среда, приспособленная к решению определенной проблемы путем создания соответствующей базы знаний и задания набора правил. Кроме того, ЭС может включать в себя компьютерную модель управляемого процесса (например, движения космической станции), позволяющую прогнозировать его реакцию на действия эксперта.

4. Текстовый и графический редакторы. *Текстовый редактор* (текстовый процессор) — прикладной пакет программ, позволяющий работать с текстовой информацией (Word, Блокнот и т.д.). Основные назначения текстовых редакторов — обеспечение ввода текста в компьютер (в

ОЗУ), их редактирование, печать на бумагу. Режимы работы: набор текста, считывание и сохранение в ВЗУ, форматирование, задание шрифтов, редактирование, проверка орфографии, печать текста на бумаге. Используемые аппаратные средства: ОЗУ, клавиатура, дисплей, ВЗУ, принтер.

При запуске программа текстового редактора помещается в ОЗУ. Текст набирается на клавиатуре и тоже попадает в ОЗУ. Для контроля за правильностью ввода используется дисплей. Перед выключением компьютера текст сохраняется в НЖМД, откуда снова может быть загружен в ОЗУ для очередного редактирования и печати с помощью принтера. Данными является вводимый текст. Рабочее поле экрана и интерфейс (средства управления текстовым редактором) образуют рабочую среду. Курсор показывает место ввода очередного символа. Для хранения копируемых фрагментов используется буфер обмена. Имеется словарь, развитая система меню и подсказок.

Основные функции *графического редактора* — создание изображений, их редактирование, сохранение в ВЗУ, получение копий на бумаге, пленке и т.п. Программа PaintBrush работает с растровой графикой, позволяя удалять и добавлять отдельные пиксели. Совокупность пикселей образует графическую сетку. Информация о цвете каждого пикселя хранится в специальной области ОЗУ — видеопамяти. Пакет CorelDraw преобразует загружаемый рисунок в векторный формат, что позволяет его растягивать, поворачивать, преобразовывать с помощью специальных фильтров и инструментов. Режимы работы графических редакторов: выбор инструмента, цвета, рисование изображения, его редактирование, работа с ВЗУ, мышью, клавиатурой и принтером. Рабочее поле экрана и интерфейс (средства управления программой) образуют среду. Помимо основной видеопамяти есть буфер обмена. Имеется система меню и подсказок.

5. Электронные таблицы и базы данных. *Электронные (динамические таблицы) или табличные процессоры* — программы, предназначенные для работы с большими массивами числовой информации. В электронных таблицах информация находится в клетках таблицы в текстовом и числовом формате. Информационная структура не статическая, а динамическая: при изменении исходных данных происходит автоматический пересчет остальных полей таблицы. Режимы работы: формирование таблицы, редактирование, вычисление по формулам, считывание и сохранение в НЖМД, построение графиков и диаграмм, статистическая обработка элементов таблицы, упорядочивание по признаку, работа как с базой данных, построение графиков и диаграмм, печать на бумагу.

Рассмотрим программу Excel из пакета Microsoft Office. Рабочее поле представляет собой экранное окно, в котором видна часть таблицы (максимальное число строк — более 10000, число столбцов — более 200). Имеются системы подсказок, меню, строка ввода, строка формул, строка состояния с информацией о текущей ячейке. При расчетах по формулам можно использовать встроенные функции. Программа хранит в табличной форме не только исходные данные, но и математические соотношения между ними.

База данных (БД) — специальным образом организованная и хранящаяся во внешней памяти компьютера информация (данные) об объектах. Система управления базами данных (СУБД) — система программ, позволяющих создавать БД, обновлять хранимую в ней информацию, обеспечивая удобный доступ к ней с целью просмотра и поиска. ВЗУ, предназначенные для хранения БД, должны иметь высокую информационную емкость и малое время доступа к хранимой информации. Режимы работы с СУБД: создание, редактирование, поиск, просмотр, копирование, распечатка, сортировка данных по заданному признаку. СУБД обеспечивает возможность внесения и чтения информации, работу с большим объемом данных, быстроту поиска данных, непротиворечивость данных, защиту от разрушения и несанкционированного доступа, систему подсказок.

Запись — группа связанных между собой элементов данных. Различают три типа организации данных и связей между ними: иерархический (древовидной), сетевой, реляционный. В иерархической БД один элемент считается главным, остальные — подчиненными. Сетевая БД дает возможность устанавливать дополнительно к вертикальным иерархическим связям горизонтальные связи. В реляционной БД под записью понимается строка таблицы. Элементы записи образуют столбцы этой таблицы (поля). Все элементы в столбце имеют одинаковый тип (числовой, символьный), каждый столбец — неповторяющееся имя. Преимущество таких БД — наглядность и понятность организации данных, скорость поиска нужной информации. БД и СУБД образуют *информационно-поисковую систему*, называемую *банком данных*.

6. Распределенная и централизованная обработка данных. Представим себе дорогую суперЭВМ, занимающую большое помещение, потребляющую много энергии и имеющую высокую производительность. При работе с ней используется принцип *централизованной обработки данных*: к центральной ЭВМ, подключают несколько терминалов (клавиатура, монитор), за которыми работают программисты. При этом снижается надежность, затрудняется развитие системы, падает быстродействие при работе в многопользовательском режиме. Выход из строя центральной ЭВМ приводит к прекращению работы.

Появление малых ЭВМ, микроЭВМ и ПЭВМ позволило реализовать принцип *распределенной обработки данных* и перейти от использования центральной ЭВМ к применению нескольких равноправных ЭВМ, объединенных в вычислительную сеть. **Вычислительная сеть (ВС)** — это совокупность ЭВМ, соединенных друг с другом и позволяющих осуществлять обмен информацией. Различают локальные, отраслевые, региональные и глобальные сети. **Абоненты сети** — устройства, выдающие или потребляющие информацию в сети (отдельные ЭВМ, терминалы, промышленные роботы, станки с числовым программным управлением). **Средства передачи** — физическая передающая среда (линии связи или пространство, в котором распространяются электрические сигналы) и специальная аппаратура (модем, сетевая плата и т.д.), обеспечивающая передачу

сообщений. *Физическая передающая среда* ВС: витая пара, коаксиальный кабель, оптоволоконный кабель. Информацию также передают с помощью радиоволн.

ВС представляет собой информационную систему, состоящую из источников информации (передающих ЭВМ), канала связи, кода (соглашения, позволяющего закодировать и раскодировать сигнал), потребителей информации (принимающих ЭВМ). Глобальная сеть строится на основе мощных компьютеров, называемых серверами, которые подключаются друг к другу по телефонным линиям и посредством спутниковой связи. К этим серверам подключены серверы и сети второго порядка (региональные), третьего порядка (отраслевые), четвертого порядка (локальные), а к ним — отдельные ПЭВМ.

Сервер — это компьютер, подключенный к сети и используемый для хранения данных, управления базами данных, удаленной обработки информации. *Рабочая станция* — это ПЭВМ, подключенная к сети, через который пользователь получает доступ к ее информационным ресурсам. *Концентратор* — устройство, коммутирующее несколько каналов связи на один путем частотного разделения. *Мультиплексор и демультиплексор* — устройства сопряжения ЭВМ с несколькими каналами связи. *Протокол* — набор правил, определяющий взаимодействие двух одноименных уровней модели взаимодействия открытых систем в различных абонентских ЭВМ. Передаваемая информация разбивается на отдельные *кадры*, которые последовательно передаются по сети. В качестве примера рассмотрим формат кадра протокола HDLC (High-level Data Link Control — высокоуровневый протокол управления каналом передачи данных). Как видно из рис. 7.2, кадр содержит поле заголовка, поле адреса получателя, поле адреса отправителя, поле команды управления, информационное поле данных, контрольное поле, конец кадра. *Байт-ориентированный протокол* обеспечивает передачу сообщения по информационному каналу в виде последовательности байтов. Кроме информационных байтов в канал передаются управляющие и служебные байты. *Бит-ориентированный протокол* обеспечивает передачу информации в виде потока бит, не разделенных на байты.

| | | | | | | |
|------------------------|-----------------------------------|------------------------------------|--------------------------------------|---------------------------------------|-------------------------------|------------------------|
| поле заго- ловка | поле адреса полу- чателя | поле адреса отпра- вителя | поле ко- манды управ- ления | информа- ционное поле данных | конт- роль- ное поле | поле конца кадра |
|------------------------|-----------------------------------|------------------------------------|--------------------------------------|---------------------------------------|-------------------------------|------------------------|

Рис. 7.2.

Протоколы описывают процессы в сетях на семи уровнях: 1) физический уровень (механическое и электрическое соединение устройств); 2) канальный уровень (соединение и разъединение физически связанных систем); 3) сетевой уровень (управление трафиком, маршрутизация сообщений, выбор оптимального канала связи); 4) транспортный уровень (передача информации независимо от ее способа); 5) сеансовый уровень (организация сеанса связи между физически связанными системами); 6) представительный уровень (представление информации в требуемом виде); 7)

прикладной уровень (управление прикладными программами).

Топология ЛВС — это усредненная геометрическая схема соединений узлов сети. Различают следующие виды топологии (рис. 7.3.1–5): 1) шинная; 2) радиальная (звездообразная); 3) кольцевая; 4) сетевая; 5) древовидная.

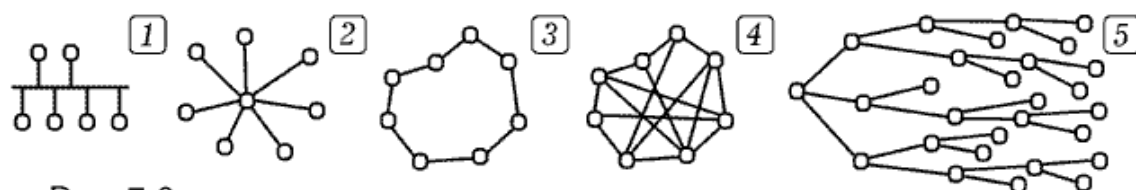


Рис. 7.3.

7. Локальные и глобальные ВС. *Локальные компьютерные (вычислительные) сети ЛВС* или Интранет (Intranet) представляют собой коммуникационные системы, охватывающие 10–100 ПЭВМ и ограниченные зданием или предприятием. Существенная особенность ЛВС — использование всеми ПЭВМ (рабочими станциями), включенными в сеть, аппаратных и информационных ресурсов других устройств сети. Составные части ЛВС: кабель, сетевые платы, сервер сети, центральное ЗУ, рабочие станции. В телекоммуникационных ВС обмен информацией между компьютерами и другими абонентами сети осуществляется по телефонным линиям, оптоволоконному кабелю и через радиоволны (спутниковая связь).

С целью идентификации в ЛВС каждая ПЭВМ имеет цифровой *IP*-адрес (Internet Protocol — межсетевой протокол для работы в Интернет), например, 10.100.0.12. В Интернет используется доменный способ адресации, когда все пространство адресов абонентов сети по региональному признаку разделено на области, называемые *доменами*.

Электронная почта (e-mail — electronic mail) выполняет функции обычной почты, то есть обеспечивает передачу сообщений из одного пункта в другой. Каждый пользователь имеет ПЭВМ, которая через модем подключена к телефонной линии. С помощью специальной программы он выходит в Интернет, подключается к узловому компьютеру сети (региональному узлу электронной почты) и пересылает на него сообщение в виде файла. Это сообщение попадает в электронный почтовый ящик — папку на том же или другом узловом компьютере, в соответствии с указанным электронным адресом. Узловые региональные компьютеры сети работают круглосуточно и связаны между собой выделенными линиями с высокой пропускной способностью. Адресат периодически проверяет свой почтовый ящик и обнаруживает там посланные ему сообщения. Эта система электронного обмена сообщениями позволяет организовывать телеконференции, скачивать архивы файлов, передавать мультимедиа-письма, содержащие текстовую, графическую, звуковую и видеoinформацию в одном сообщении. Формат адреса электронной почты имеет вид: *имя пользователя@адрес хост-компьютера: alexander@mail.ru*. Буквы перед символом "@" ("собака") задают имя абонента, а после него — имя узлового компьютера сети, на которой установлена почтовая система.

Интернет позволяет получать доступ к различным каталогам, базам данных, пользоваться электронной доской объявлений, проводить компьютерные конференции, общаться в реальном масштабе времени, то есть читать информацию по мере ее ввода другим пользователем, организовать телефонные переговоры (IP-телефония) и видеоконференции, осуществлять Web-мониторинг. *Всемирная Паутина* (World Wide Web — WWW) — это информационная служба Интернет, в которой используются гипермедиадокументы и возможность клиентов взаимодействовать с другими сетевыми приложениями и ресурсами. Файл в сети находят с помощью URL (Universal Resource Locator), — специального адреса, в котором указаны протокол, домен Web-сервера, содержащего файл, и путь файла на этом компьютере: *http://portal.ggpi.org/news.php*.

Гипермедиадокумент — текстовый файл, содержащий в себе связи с другими текстовыми, графическими, видео- или звуковыми файлами. Внутри гипертекстового документа некоторые фрагменты текста выделены. При их активизации можно перейти на другую часть этого же файла или запустить другой файл на этом или другом ПЭВМ. Все серверы WWW используют специальный язык HTML (Hyper Text Markup Language — язык разметки гипертекста), с помощью которого текстовый файл преобразуется в файл *.html. Для того, чтобы обеспечить обмен данными в ВС между различными ПЭВМ независимо от установленных на них ОС используется протокол FTP (File Transfer Protocol — протокол для передачи файлов).

8. Мультимедиа-технологии. *Мультимедиа* — компьютерная технология, обрабатывающая и сочетающая в себе текстовую, звуковую и видеоинформацию. При этом используются аппаратные и программные средства, позволяющие человеку обмениваться информацией с компьютером. В мультимедийных системах используются следующие аппаратные устройства: 1. Устройства ввода: клавиатура, мышь, трекбол, джойстик, сканер, устройства ввода аудиосигнала, цифровые фото- и видеокамеры, TV-устройства ввода. 2. Устройства вывода: принтеры, плоттеры, электронные экраны, системы аудиовывода, системы видеовывода. 3. Устройства ввода-вывода: рабочие станции, модемы, сенсорные дисплеи, аудио-видеомагнитофоны. 4. Преобразователи информации: аналого-цифровой преобразователь, цифро-аналоговый преобразователь, системы распознавания, конверторы, системы сжатия и восстановления, TV-преобразователи.

В последнее время развиваются *устройства речевого ввода-вывода* — акустические системы, позволяющие вводить информацию и управлять компьютером речью. Программное обеспечение может распознавать отдельные звуки и слова, являющиеся командами, узнавать голос человека по его частоте и тембру. Все это используется для идентификации пользователя, ввода текстов, а также для контроля правильности произношения звуков и слов в обучающих программах.

Мультимедиа-технологии используются в бизнес-приложениях для создания презентаций, телеконференций, для ввода команд и текста с голо-

са, работы с компьютерной графикой, производства видеофильмов, обработки звуковых файлов, в учебном процессе, для создания обучающих, развивающих программ, компьютерных энциклопедий и гипермедиа- и телемедиа-книг. При этом достигается эффект *виртуальной реальности*: в сознании пользователя возникает некоторая модель реального мира, содержащая несуществующие (воображаемые) объекты, с которыми он взаимодействует.

9. Защита информации. Информационная безопасность предполагает обеспечение целостности данных, конфиденциальности информации, доступности информации для авторизованных пользователей. Существуют следующие виды *средств защиты информации*: аппаратные, программные, организационные и законодательные. Перечислим *способы защиты информации*: препятствие или ограничение доступа к компьютеру, управление доступом к информационным ресурсам, маскировка информации путем ее шифрования, регламентация работы с компьютером, содержащим важную информацию. Управление доступом включает в себя идентификацию пользователей и ресурсов системы, проверку полномочий, разрешение работы в пределах установленного графика, регистрацию обращений к защищаемым ресурсам, реагирование при попытке несанкционированного считывания или изменения информации.

Угрозы информационным системам: 1) *угроза раскрытия* — информация станет известна людям, которые не должны ее знать; 2) *угроза целостности* — умышленное изменение документа или удаление данных; 3) *угроза отказа в обслуживании* — блокировка доступа к информационному ресурсу системы. Виды *компьютерных преступлений*: 1) несанкционированный доступ к информации; 2) ввод в ПО "логических бомб", срабатывающих при выполнении определенных условий; 3) разработка и распространение компьютерных вирусов; 4) небрежность в разработке и эксплуатации программно-вычислительных комплексов, приведшая к тяжким последствиям; 5) подделка компьютерной информации; 6) хищение компьютерной информации.

Политика безопасности — система мер управленческого, организационного и программного характера, направленная на защиту хранимой информации и самой информационной системы. При этом применяют шифрование, пароль, электронно-цифровую подпись (ЭЦП), подтверждающую подлинность и целостность документа. Использование ЭЦП позволяет предотвратить отказ одного из авторов договора, изменение и подделку принятого документа.

Компьютерный вирус — это программа, способная приписывать себя к другим программам ("заражать" их) и осуществлять нежелательные действия, связанные с порчей аппаратного и программного обеспечения ЭВМ. Существуют *вирусы*: 1) *Резидентными* называются вирусы, копирующие себя в ОЗУ, перехватывающие коды прерываний, что позволяет им вмешиваться в работу процессора и всей ЭВМ. В течение некоторого времени вирус себя не проявляет, а затем активизируется и копирует сам себя из ОЗУ в НЖМД.

2) *Нерезидентными* называются вирусы, не остающиеся в памяти ОЗУ после завершения работы зараженной программы. Они действуют только во время работы зараженной программы: сканируют каталоги, находят незараженный файл, пригодный для внедрения. К этому файлу дописывается тело вируса, он становится зараженным.

"Логические бомбы" — программы, активизирующиеся при наступлении некоторого события: заданной даты, определенного режима работы ЭВМ и т.д. *Загрузочный вирус* заражает загрузочный сектор НЖМД или НГМД и загружается в ОЗУ при загрузке ОС. Вирусы активизируются не сразу, а через некоторое время, действуют скрытно, чтобы усложнить поиск зараженной программы. Через некоторое время после размножения вирусов запускаются программы, печатающие ненужные сообщения, стирающие или портящие файлы, замедляющие работу процессора. *Троянские программы* — это вирусы, замаскированные под игровые или иные программы. *"Черви"* — программы, запускающиеся при загрузке ЭВМ, лавинообразное размножение которых приводит к блокированию системы.

Для выявления вирусов и зараженных файлов используются специальные *антивирусные* программы, позволяющие выявлять, "лечить" и удалять вредоносные программы. Чтобы защитить ПЭВМ и локальную сеть от попыток несанкционированного доступа из Интернет, отфильтровать *спам* (электронные письма рекламного характера), зараженные вирусом файлы, контролировать попытки программ самостоятельно выйти в Интернет, используют специальные программы — *межсетевые экраны* (*брандмауэры*).

7.2. ПРОГРАММИРУЕМ НА ЯЗЫКЕ QBASIC

1. Основные конструкции языка QBasic. Простейшие конструкции языка программирования Basic: константы, переменные, функции и выражения. Различают *целые константы* (345, -23, 0), *вещественные константы* (123.765, -0.89), *строковые константы* ("r12sd", "имя"). Запись вещественной константы с порядком состоит из мантиссы и порядка, которые разделены буквой "E": $0.213E-09$, что соответствует $0,213 \cdot 10^{-9}$.

Простые переменные обозначаются идентификатором (именем из одного или нескольких символов), хранятся в одной ячейке памяти и все время имеют одно определенное значение. *Переменные с индексом* — это элементы массива, которые обозначаются именем массива и индексом: $A(I)$ или $B(I, J)$. *Массивом* называется набор однотипных переменных, упорядоченный по возрастанию индексов. Оператор $DIMA(5, 6)$ задает массив A , состоящий из 5 строк и 6 столбцов.

Оператор присваивания используется для изменения значения переменной и записывается так: "идентификатор переменной = выражение" ($X = 5$, $Y = 2 * A / X$, $A\$ = \text{"компьютер"}$).

Стандартные функции языка Basic: $ASC(A\$)$, $CHR\$(B)$ — возвращает код ASCII первого символа строки $A\$$ и наоборот; $HEX\$(B)$, $OCT\$(B)$ — возвращают шестнадцатичный и восьмеричный код числа; $LCASE\$(A\$)$, $UCASE\$(A\$)$ — меняют верхний регистр строки на нижний и наоборот; $LEFT\$(A\$, B)$, $RIGHT\$(A\$, B)$, $MID\$(A\$, B, C)$ — возвращает часть строки; $LEN\$(A\$)$ — возвращает длину строки; $STR\$(B)$, $VAL(A\$)$ — преобразует число в строку и наоборот; $SIGN(X)$, $SQR(X)$, $ABS(X)$, $ATN(X)$, $EXP(X)$, $COS(X)$, $LOG(X)$, $SIN(X)$, $TAN(X)$ — возвращает значение математических функций; $TIME\$$ — возвращает текущее время ПЭВМ по системным часам; RND — возвращает случайное число от 0 до 1.

Перечислим виды используемых выражений: арифметические ($X * X$, $SQR(2 - TAN(2 * X - 5))$), строковые ($A\$ = B\$ + C\$$) и выражения типа отношения ($A <> B$, $C > D$). Для выполнения логических операций используются операторы AND , OR , NOT , EQV . Представленные ниже программы написаны в среде MS-DOS QBasic 1.0.

2. Линейные, разветвляющиеся и циклические алгоритмы. *Линейными* называются алгоритмы, в которых операторы выполняются последовательно друг за другом, как они записаны в программе. Рассмотрим программы для вычисления плотности прямоугольного параллелепипеда (Пр-1) и нахождения значения логических функций $x = (A) \text{ and } (C)$, $y = (A) \text{ or } (B)$, $z = (\text{not}(A)) \text{ and } (B \text{ or } C)$ при заданных A, B, C (Пр-2). В некоторых версиях QBasic "ложь" обозначается как 0, а "истина" как -1, поэтому программу Пр-2 следует изменить.

| | | | |
|--|---|---|---|
| <pre> INPUT "ВВЕДИТЕ ", a, b, c INPUT "ВВЕДИТЕ МАССУ ", m v = a * b * c : rho = m / v PRINT "ОБЪЕМ ", v PRINT "ПЛОТНОСТЬ ", rho </pre> | <div style="border: 1px solid black; padding: 2px; display: inline-block;">Пр-1</div> | <pre> A = FALSE: B = TRUE: C = FALSE x = A AND C: y = A OR B z = (NOT (A)) AND (B OR C) PRINT A, B, C, x, y, z </pre> | <div style="border: 1px solid black; padding: 2px; display: inline-block;">Пр-2</div> |
|--|---|---|---|

Разветвляющимися называются алгоритмы, исполнение которых может происходить по той или иной ветви в зависимости от промежуточных результатов вычислений. Для их программирования используется оператор условного перехода: $IF \{условие\} THEN \{оператор\ 1\} ELSE \{оператор\ 2\}$. Оператор безусловного перехода имеет формат: $GOTO \{номер\ строки\}$. В качестве примера рассмотрим вычисление функции $y = \log(x)$ (Пр-3). Если введенное значение x неположительно, программа должна сообщить об этом. Программа Пр-4 находит значение логической функции, оперируя с символьными переменными.

| | | | |
|--|---|---|---|
| <pre> INPUT "ВВЕДИТЕ X ", X IF X > 0 THEN Y = LOG(X) ELSE GOTO 1 PRINT Y: END 1 PRINT "X ДОЛЖЕН БЫТЬ >0": END </pre> | <div style="border: 1px solid black; padding: 2px; display: inline-block;">Пр-3</div> | <pre> A = FALSE: B = TRUE: C = FALSE z = (NOT (A)) AND (B OR C) IF z THEN A\$ = "И" ELSE A\$ = "Л" PRINT A\$ </pre> | <div style="border: 1px solid black; padding: 2px; display: inline-block;">Пр-4</div> |
|--|---|---|---|



АТАНАСОВ Джон Винсент (1903 гг.)

Руководитель работ по созданию первой вычислительной машины полностью на электронных лампах (1937–1942). Собранная ЭВМ отличалась разделением блоков, обработкой и хранением информации. ОЗУ состояло из конденсаторов и предусматривало их автоматическую подзарядку. Использовалась двоичная система счисления. Для ввода информации использовались перфокарты.

Часто требуется многократно вычислять значения функции по одним и тем же формулам для различных значений аргумента, либо повторять иную последовательность действий при значениях параметра, изменяющегося с некоторым шагом. В этом случае используют *циклы* — многократно повторяющиеся участки вычислительного процесса, в которых выполняются одни и те же операторы для различных входных величин. Различают циклы с *заданным* и с *неизвестным числом повторений*. Целочисленная переменная, изменяющаяся в цикле называется *параметром цикла*. Чтобы организовать цикл необходимо: 1) задать параметр цикла i ; 2) изменять параметр цикла на величину шага h ; 3) проверять условие повторения цикла; 4) выходить из цикла при его окончании.

```
SCREEN 2
WHILE x < 500
  x = x + .1
  y = 10 * LOG(x)
  PSET (x, 150 - y)
WEND
```

Пр-5

```
INPUT A, B, h
x = A
10 x = x + h
PRINT x, x ^ 2 / SIN(x)
IF x < B THEN GOTO 10
```

Пр-6

```
INPUT A, B, h
FOR x = A TO B STEP h
  y = 10 * COS(x)
  PRINT x, y
NEXT: END
```

Пр-7

Возможны три вида циклов: 1) цикл с предусловием "пока" (*WHILE* {условие} {Тело цикла} *WEND*): ПОКА (условие Q) ВЫПОЛНЯТЬ (Тело цикла); 2) цикл с постусловием "до" (используется оператор *DO* : {Тело цикла} *LOOP UNTIL* {условие}): ПОВТОРЯТЬ (Тело цикла) ДО (условие Q); 3) счетный цикл или цикл с параметром (используется оператор *FOR* {начальное значение параметра} *TO* {конечное значение} *STEP* {шаг} {Тело цикла}): ДЛЯ i от 1 ДО n ШАГ 2 ПОВТОРЯТЬ (Тело цикла).

```
DEF FNN (x) = x ^ 2
INPUT "ВВЕДИТЕ x И h ", x, h
PR = (FNN(x + h) - FNN(x)) / h
PRINT "ПРОИЗВОДНАЯ", PR
```

Пр-8

```
INPUT a: GOSUB 10: INPUT a
GOSUB 10: INPUT a: GOSUB 10: END
10 FOR x = 1 TO 10 '=== подпрограмма
  PRINT a * x ^ 2; : NEXT: PRINT : RETURN
```

Пр-9

3. Функции пользователя. Подпрограммы. Для сокращения программы пользователь может задавать новые функции и организовывать подпрограммы. Определить новые функции можно с помощью оператора *DEF* : *DEF FNN* (A,B) = A·SIN(B). Оператор *FNN*(3,2) возвраща-

ет значение $3 \sin(2)$ (см. Пр-8). Для вызова подпрограммы используется оператор *GOSUB* {номер первой строки подпрограммы}. Подпрограмма заканчивается оператором *RETURN* (см. Пр-9).

4. Решение уравнения методом половинного деления. Решение уравнения разделяют на этапы: 1) локализация корня, то есть приблизительное установление числового интервала, в котором он находится; 2) уточнение значения корня путем уменьшения содержащего его интервала до требуемого значения.

```
CLS : a = 0: b = 1: e = .000001
10 x = a: y1 = EXP(-x) - x * x
x = b: y2 = EXP(-x) - x * x
c = (b + a) / 2
x = c: y3 = EXP(-x) - x * x
PRINT a, b, c
IF (y1 * y3) > 0 THEN a = c ELSE b = c
IF (b - a) > e THEN GOTO 10
PRINT "Корень лежит в интервале", a, b
PRINT "0=", EXP(-a) - a * a
```

Пр-10

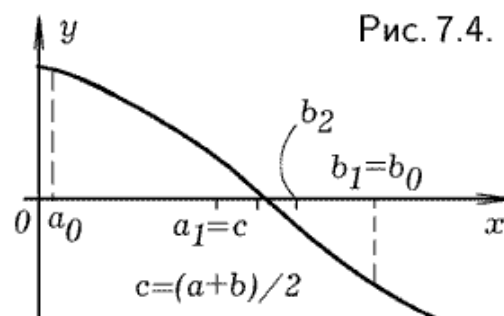


Рис. 7.4.

Самым простым методом нахождения решения уравнения является метод табуляции. Исходное уравнение представляют в виде $f(x) = 0$. Например, уравнение $x^2 = e^{-x}$ можно записать как $e^{-x} - x^2 = 0$. Затем устанавливают границы интервала $[a_0, b_0]$, внутри которого лежит один корень (рис. 7.4). Это можно сделать графическим методом: построить график функции $y = f(x)$ и приблизительно найти интервал, внутри которого он пересекает ось Ox . При этом функция $f(x)$ на концах интервала будет иметь разные знаки, в чем можно убедиться, проверив условие $f(a_0) \cdot f(b_0) < 0$. После этого табулируют функцию $f(x)$ с некоторым шагом и находят интервал, внутри которого она обращается в ноль. При необходимости эту процедуру повторяют снова с меньшим шагом.

Метод половинного деления состоит в том, что в цикле осуществляется деление отрезка $[a_0, b_0]$ пополам точкой $c = (a_0 + b_0)/2$. Если $f(c) = 0$, то значение $x = c$ и есть корень уравнения. В противном случае из двух отрезков $[a_0, c]$ и $[c, b_0]$ выбирается тот, на концах которого функция $f(x)$ имеет противоположные знаки. Если $f(c) \cdot f(b_0) < 0$, то $a_1 = c$, $b_1 = b_0$. Затем осуществляется новое приближение (итерация): деление отрезка $[a_1, b_1]$ повторяется. Так продолжается до тех пор, пока величина отрезка $[a_i, b_i]$ не окажется меньше заданного ε (программа Пр-10).

5. Нахождение суммы и произведения. Для вычисления суммы членов бесконечного ряда $S = \sum_{i=1}^{\infty} 1/i^2 = 1 + 1/4 + 1/8 \dots$ с заданной точностью $\varepsilon < 1$ используется цикл, в котором вычисляется i -ый член ряда $f_i = 1/i^2$ и накапливается сумма путем присвоения переменной S значения $S + f_i$. Перед циклом необходимо S присвоить 0. В цикле должно осуществляться сравнение f_i с ε . В случае, когда f_i меньше ε , происходит выход из цикла. Например, программа Пр-11 для нахождения суммы ряда с точностью $\varepsilon = 0,005$.

```
CLS : i = 1: f = 1
WHILE f > .005
f = 1 / i ^ 2: i = i + 1
s = s + f: PRINT s; f
WEND: PRINT "Сумма"; s
```

Пр-11

```
CLS : fact = 1: N = 25
FOR i = 1 TO N
fact = fact * i: PRINT i; "!="; fact
NEXT
```

Пр-12

Для нахождения произведения $P = \prod_{i=1}^n f_i = f_1 \cdot f_2 \cdot \dots \cdot f_n$, используется похожий алгоритм, отличающийся тем, что для накопления произведения применяется формула $P = P \cdot f_i$. Перед циклом необходимо P присвоить 1. Выше представлена программа Пр-12 для нахождения факториала $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$.

Интеграл представляет собой сумму бесконечно большого числа бесконечно малых величин. Интеграл функции пропорционален площади криволинейной трапеции, ограниченной графиком $y = f(x)$, осью абсцисс и пределами интегрирования. Для ее нахождения криволинейную трапецию разбивают на узкие полоски шириной $dx = h$ и высотой $y_i = f(x_i)$ (рис. 7.5). После этого вычисляется элементарная площадь каждой полоски $dS_i = y_i h$ и находится их сумма (программа ПР-13).

```
a = 1: B = 3: h = .0001
FOR x = a TO B STEP h
y = x ^ 3 - 2 * x ^ 2 + 3
s = s + y * h
NEXT: PRINT s
```

Пр-13

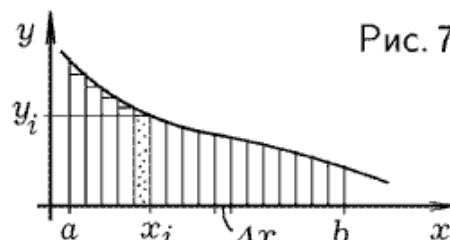


Рис. 7.5.

6. Массивы. Ввод и вывод массивов. Часто необходимо одинаковым образом обработать большую совокупность данных. В этом случае используют *массив* — набор однотипных переменных с одинаковым именем, которые различаются индексом. В начале программы массив следует объявить, для чего используется оператор $DIMA(N, M)$. Чтобы обратиться к элементу $a_{i,j}$ после идентификатора массива указывают индексы элемента: $A(I, J)$.

Возможны два способа ввода-вывода массивов: 1) путем перечисления данных и переменных с индексами с помощью операторов $READ, INPUT, PRINT$ (Пр-14); 2) ввод-вывод массивов в диалоговом режиме с помощью цикла с параметром (Пр-15, Пр-16).

```
DIM X(5): DATA 3, -4.2, 7, -1, 6
READ X(1), X(2), X(3), X(4), X(5)
PRINT X(1); X(2); X(3); X(4); X(5)
```

Пр-14

```
N = 4: M = 3: DIM a(N, M)
FOR i = 1 TO N: FOR j = 1 TO M
PRINT "Введите a("; i; j; ") "
INPUT a(i, j): NEXT: NEXT
FOR i = 1 TO N: FOR j = 1 TO M
PRINT 2 * a(i, j) - 5;
NEXT: PRINT : NEXT
```

Пр-16

```
DIM X(10,7): PRINT "Введите массив"
FOR I=1 TO 10: FOR J=1 TO 7: PRINT I, J
INPUT X(I,J): NEXT J : NEXT I: END
```

Пр-15

7. Нахождение наибольшего элемента массива. Для нахождения наибольшего элемента массива переменной *max* присваивают значение

первого элемента y_1 . Затем с помощью цикла с параметром перебирают все остальные элементы, сравнивая их с max . Если $y_i > max$, то переменной max присваивают y_i . По завершении перебора переменная max равна максимальному значению массива.

Для нахождения максимального значения функции $y = f(x)$ на интервале $[a, b]$, создают массив y_1, y_2, \dots, y_n , который соответствует значениям аргумента $a, a + h, a + 2h, \dots, a + (i + 1)h, \dots, b$. Перебирая элементы массива, находят максимальное значение y_i и соответствующее значение аргумента x_i (Пр-17). Минимальное значение ищется аналогично.

```
DIM y(300): A = 0.2: h = .01
FOR x = A TO 3.2 STEP h
i = i + 1: y(i) = SIN(x)
PRINT y(i): NEXT: max = y(1)
FOR i = 2 TO 300
IF y(i) > max THEN max = y(i): k=i
NEXT: PRINT "МАКСИМУМ ", max
PRINT "АРГУМЕНТ ", A + (k + 1) * h
```

Пр-17

8. Упорядочивание массивов. Пусть имеется файл, содержащий записи a_1, a_2, \dots, a_n , каждой из которых соответствует некоторый *ключ* — число K_i , где $i = 1, 2, \dots, n$. Осуществить сортировку — значит найти такую перестановку записей, чтобы их ключи следовали в порядке неубывания (невозрастания). При упорядочивании числового массива запись является ключом.

```
CLS : N = 10: DIM x(N)
DATA 8,-1,4,0,3,2,-4,5,1,6
FOR i = 1 TO N: READ x(i)
PRINT x(i); : NEXT i
FOR j = 1 TO N: min = x(j): k = j
FOR i = j TO N
IF x(i) < min THEN min = x(i): k = i
NEXT i: SWAP x(j), x(k): NEXT j:
PRINT : PRINT "Результат"
FOR i = 1 TO N: PRINT x(i); : NEXT i
```

Пр-18

```
CLS : N = 9: DIM x(N)
DATA 2,3,1,-4,5,-9,8,0,-3,-4
FOR i = 1 TO N
READ x(i): PRINT x(i)
NEXT i: PRINT
FOR i = 1 TO N: FOR j = 1 TO N
IF x(i) < x(j) THEN SWAP x(i), x(j)
NEXT: NEXT
FOR i = 1 TO N: PRINT x(i): NEXT i
```

Пр-19

Линейная сортировка: перебирая a_1, a_2, \dots, a_n , находят минимальный a_k и его номер k . После этого первый и k -ый элементы меняются местами. Перебирая a_2, \dots, a_n , элементы, снова находят минимальный и меняют его местами с a_2 . Решают аналогичную задачу для a_3, \dots, a_n , и т.д., пока не упорядочат все элементы (Пр-18).

```
SCREEN 2
LINE (10, 20)-(540, 120), 4
LINE (110, 120)-(340, 50), 5, B
LINE (30, 20)-(60, 50), 5, BF
CIRCLE (230, 135), 153
PSET (110, 5):PAINT (200, 100)
```

Пр-20

```
SCREEN 1
LINE (0, 240)-(640, 240)
FOR t = 0 TO 100 STEP .01
y = 10 * 2.7 ^ (-.05 * t) * SIN(t)
PSET (10 + 10 * t, 120 - 5 * y)
NEXT
```

Пр-21

Пузырьковая сортировка: осуществляют несколько проходов, в каждом из которых сравниваются соседние элементы массива (ключи) и при необходимости меняются местами. При этом "легкие" элементы "всплывают вверх", а тяжелые — "опускаются вниз" (программа Пр-19).

9. Работа с графикой и звуком. Для создания рисунков, построения графиков необходимо запустить графический режим. Для этого используется оператор *SCREEN n*, где *n* — номер режима (текстовый или графический с различным разрешением). Оператор *LINE* строит линию, а при наличии символов *B* и *BF* — незакрашенный или закрашенный прямоугольник. Оператор *CIRCLE* строит окружности и эллипсы. Оператор *PSET* рисует точку с заданными координатами. Оператор *PAINT* закрашивает соответствующую область экрана (Пр-20, Пр-21).

Для рисования ломанных линий может быть использован оператор *DRAW*, за которым следуют символы, показывающие направление движения и количество шагов. Например:

```
SCREEN 2: DRAW "D15 L100 F40 U10 E60"
```

Оператор *SOUND* позволяет генерировать звуковой сигнал известной частоты и длительности. Оператор *PLAY* позволяет воспроизводить простые мелодии (например, программа Пр-22).

```
SOUND 500, 3: SOUND 1500, 6: SOUND 2500, 9
Music$ = "CDEFGABBAGFEDC": PLAY Music$
```

Пр-22

10. Строковые переменные. Символьные (строковые) выражения: строковые константы, переменные и функции. Идентификатор строковых переменных содержит символ "\$". Примером работы со строковыми выражениями является программа Пр-23.

```
A$ = "ИНФОР": B$ = "МАТИКА": PRINT A$ + B$, LEN(A$)
X = ASC("A"): y$ = CHR$(100): PRINT X, y$, LEFT$(B$, 5)
A = 171: N$ = "23": PRINT VAL(N$), HEX$(A)
```

Пр-23

Оператор *INKEY\$* считывает символ при нажатии на клавишу клавиатуры. Например, программа Пр-24 после запуска ждет нажатия на клавишу "A", вращаясь в пустом цикле, после чего начинает печатать целые числа до тех пор, пока не нажмут клавишу "B".

```
DO: LOOP UNTIL INKEY$ = "A"
DO: i = i + 1: PRINT i;: LOOP UNTIL INKEY$ = "B"
```

Пр-24

Массив строковых значений объявляют оператором *DIMA\$(N)*, после чего можно обращаться к каждому элементу массива как *A\$(I)*.

| | |
|--|--|
| <pre>OPEN "a.txt" FOR OUTPUT AS #1 OPEN "b.txt" FOR OUTPUT AS #2 FOR x = 1 TO 10 PRINT #1, x; "^3 = "; x * x * x PRINT #2, x; "* 6 = "; x * 6 NEXT: CLOSE #1: CLOSE #2</pre> Пр-25 | <pre>OPEN "a.txt" FOR INPUT AS #1 OPEN "b.txt" FOR INPUT AS #2 FOR i = 1 TO 10 LINE INPUT #1, A\$ LINE INPUT #2, B\$ PRINT A\$, B\$ NEXT: CLOSE #1: CLOSE #2</pre> Пр-26 |
|--|--|

11. Запись в файл. Чтение из файла. Для создания текстового файла на диске или открытия уже существующего файла используется оператор *OPEN* "имя файла" *FOR OUTPUT AS*... Запись текста в файл на диске осуществляется оператором *PRINT*, а чтение из файла — оператором *LINE INPUT* или *INPUT*. Заккрытие файла осуществляется оператором *CLOSE*. Примеры — в программах Пр-25 и Пр-26.

7.3. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ. ПЕРСПЕКТИВЫ ИТ

1. Представление об искусственном интеллекте. Объект изучения искусственного интеллекта (ИИ) — метапроцедуры, позволяющие человеку или компьютеру решать творческие задачи, которые не имеют жесткого алгоритма. Перечислим основные проблемы ИИ:

1. Доказательство теорем, логический вывод следствий из исходных посылок, получение новых истинных логических выражений, которые не совпадают с исходными. Например, необходимо упростить сложное алгебраическое выражение, найти производную первого, второго или более высокого порядка от сложной функции, получить неопределенный интеграл (первообразную) и т.д. Во всех этих случаях компьютер, пользуясь соответствующими логическими правилами, решает задачу. Примером является пакет MathCAD, позволяющий упрощать математические выражения, брать производные и неопределенные интегралы от сложных функций (рис. 7.6).

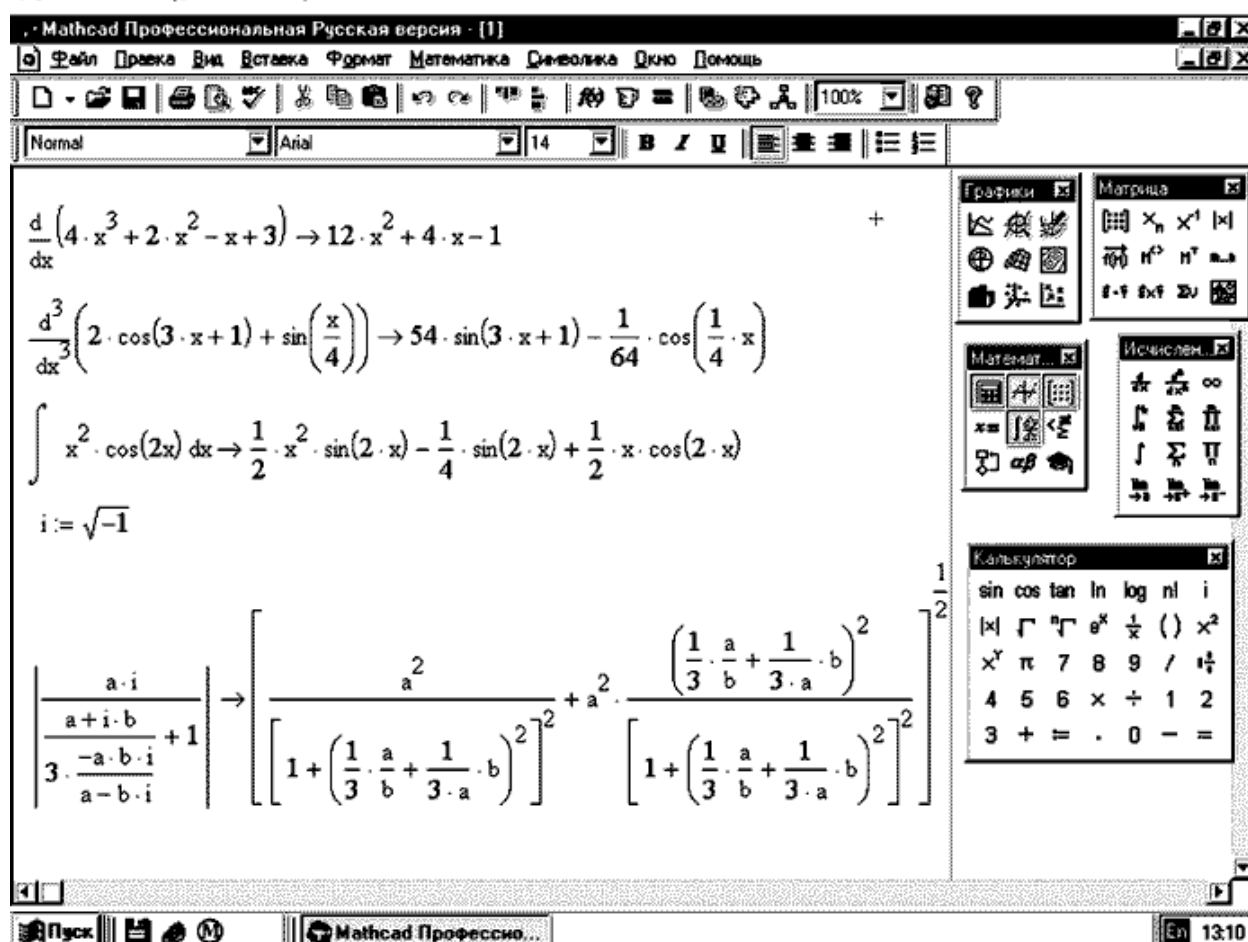


Рис. 7.6.

2. Распознавание образов. "Обучаемому" компьютеру предъявляют множество объектов первого класса $O_1^1, O_2^1, \dots, O_n^1$, второго класса $O_1^2, O_2^2, \dots, O_n^2$, m -ого класса $O_1^m, O_2^m, \dots, O_n^m$, каждый раз сообщая к какому классу относятся эти объекты. Сверяя реакцию "обучаемого" компью-

тера с правильным ответом, его "поощряют" или "наказывают". После окончания обучения компьютеру предъявляют объект O , не совпадающий ни с одним из рассмотренных ранее объектов, для того, чтобы он отклассифицировал его, то есть отнес к какому-то классу.

3. Создание экспертных систем. Врач вводит в ЭВМ результаты медицинского обследования, наблюдаемые симптомы, возраст, пол и другую информацию о больном. Компьютер обращается к базе знаний и ставит диагноз, перечисляет возможные болезни а также рекомендует те или иные медицинские препараты и методы лечения.

4. Робототехника. Важным направлением является создание роботов — многоцелевых кибернетических систем, способных целенаправленно взаимодействовать с изменяющейся окружающей средой. Робот воспринимает от окружающей среды те же сигналы, что и человек, и при этом выполняет сложные движения и механические операции лучше человека. Существуют роботы-андроиды, внешне похожие на человека.

Робот состоит из блока управления, блока восприятия и блока исполнительного механизма. Блок восприятия содержит механические, тактильные, оптические, температурные и иные датчики, воспринимающие информацию об окружающей среде. Блок исполнительного механизма включает в себя средства перемещения робота (тележка с двигателями) и средства манипулирования объектом (рука-манипулятор). Оператор обменивается информацией с блоком управления, получая, к примеру, видеоинформацию о месте, где находится робот, и отдавая команды, управляющие его движением. Блок управления включает ЭВМ, которая взаимодействует с оператором, обрабатывает сигналы, поступающие от блока восприятия, и сигналы обратной связи от блока исполнительного механизма. Роботы используются в промышленности для комплексной автоматизации производства, в космонавтике для пилотирования космических аппаратов, в глубоководных исследованиях и т.д.

5. Коллективное поведение автоматов. Изучается коллектив из n конечных вероятностных автоматов. Каждый автомат способен выполнять набор действий, с помощью которых он воздействует на среду и окружающие автоматы. Из одного состояния в другое автоматы переходят в дискретные моменты времени. Состояние системы на данном такте определяется состоянием каждого автомата и набором выбранных действий. Индивидуальная цель функционирования i -ого автомата состоит в максимизации некоторой целевой функции f_i , например, доходов. Глобальная цель заключается в максимизации функции $F(f_1, f_2, \dots, f_n)$.

6. Использование естественного языка. В последнее время развиваются системы распознавания отдельных слов, паролей, системы, позволяющие получить ответ на вопрос, заданный ЭВМ в устной форме, а также системы автоматического перевода. В будущем компьютер сможет воспринимать речь человека и отвечать на вопросы, поддерживая разговор на заданную тему.

7. Инженерия знаний. Это научное направление о способах представления знаний в вычислительных системах и методах работы с ними. В

системах ИИ с этой целью используют фреймы (кадры), каждый из которых имеет имя и состоит из частей — слотов:

Имя фрейма = (слот 1)(слот 2)(слот 3)... (слот k).

Рассмотрим следующий фрейм и пример его заполнения:

Ученый N = (фамилия)(имя)(наука)(страна)(открытие),

У-12 = (Менделеев)(Дмитрий)(химия)(Россия)(Периодический закон).

Увеличение общей производительности вычислительных систем, повышение емкости запоминающих устройств приводит к более широкому использованию баз знаний в системах автоматизированного управления.

8. Машинные игры. Способность ЭВМ быстро перебирать различные варианты, анализировать ситуации и просчитывать ходы позволяет использовать ее в качестве одного из игроков в интеллектуальных играх. Современные ЭВМ создают виртуальную реальность, моделирующую ту или иную ситуацию, что делает возможным разработку различных игр: стратегий, симуляторов, квестов и т.д.

2. Современное состояние информационных технологий. В основе современных ЭВМ — большие и сверхбольшие интегральные схемы (БИС и СБИС). Так, процессор Pentium состоит из 3,5 млн. транзисторов, образующих пятислойную структуру. Минимальный размер элементов составляет 0,25–0,13 мкм.

Согласно закону Мура, каждые полтора года плотность транзисторов в процессорах удваивается. Это приводит к повышению тактовой частоты и скорости обработки информации. Скоро будет достигнут предел, так как полупроводниковые процессоры принципиально не могут работать на частотах выше 30–40 ГГц: на высоких частотах им мешает квантовый шум (случайные колебания электронов).

Электронная промышленность продолжает развиваться в направлении уменьшения размеров транзисторов и увеличения плотности их размещения на кристалле. Это приводит к увеличению производительности процессора, росту тактовой частоты, снижению напряжения питания. В 2003 г. освоен 90-нм технологический процесс, в 2005 г. осуществлен переход на 65-нм технологию, в 2007 г. планируется внедрение 45-нм технологии. Уменьшение размеров транзистора в БИС связано с решением следующих проблем: 1) снижение энергопотребления; 2) рост влияния задержки распространения сигнала на быстродействие транзистора вследствие уменьшения длительности тактового импульса; 3) уменьшение быстродействия ОЗУ по мере роста ее объема. С целью повышения производительности ЭВМ осуществлены переход на многопроцессорные системы с параллельной обработкой данных, интеграция нескольких процессоров в одном многоядерном кристалле. На рис. 7.7.1 показана динамика роста числа элементов в микросхеме процессора. Уменьшение размера транзистора по мере развития технологии представлено на рис. 7.7.2.

Современные ЭВМ во многом обогнали человека. Создан *электронный нос* — специальное устройство, различающее запахи. Оно состоит из *кантилеверов* — микроскопических упругих пластин, которые в 100 раз тоньше человеческого волоса. Они прикреплены к пластине, которая коле-

блется. Каждый кантилевер покрыт специальным полимером. Молекулы летучего вещества оседают именно на тех пластинах, которые покрыты определенными полимерами. Это приводит к деформации кантилеверов и изменению частоты их колебаний, что контролируется с помощью лазеров и оптического детектора. В зависимости от состава летучего вещества изменяются колебания кантилеверов, что позволяет различать запахи.

В настоящее время получили широкое распространение информационные системы глобального позиционирования, использующие GPS-технологии (Global Positioning System). Пользователь с помощью персонального информационного устройства ПИУ (карманного или портативного ПК) может с достаточно высокой точностью определять свое географическое положение. GPS-технология основана на применении спутниковых систем, которые посылают специальные сигналы, улавливаемые ПИУ. На экране монитора изображается карта местности и показывается расположение пользователя.

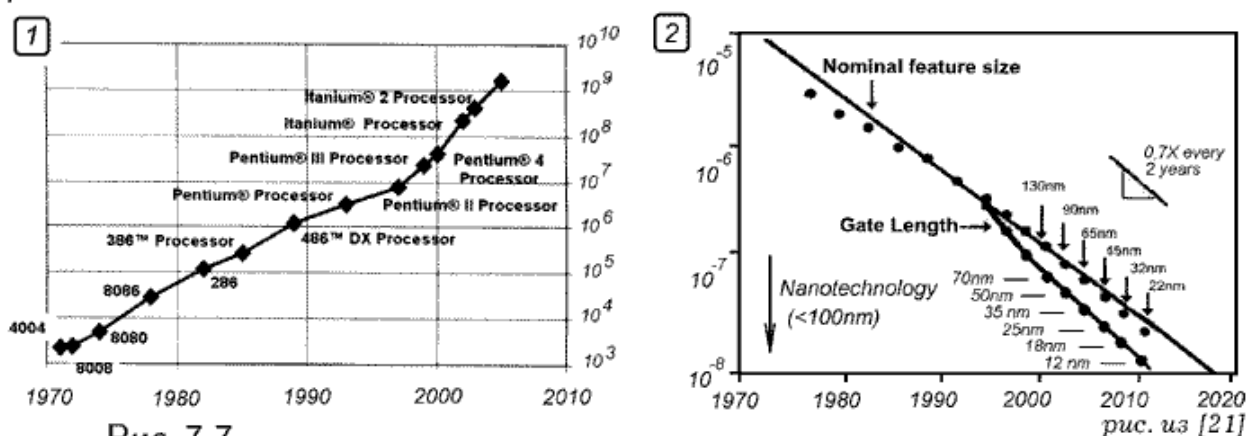


Рис. 7.7.

3. Ближайшее будущее компьютерной техники. В настоящее время происходит совершенствование информационно-коммуникационных технологий. Вот некоторые перспективные направления развития ЭВМ:

1. **Молекулярные компьютеры.** В 1974 г. IBM получило ротаксан — вещество, молекула которого обладает свойствами диода. Из нее можно сделать аналог транзистора, а из двух — аналог триггера. Переключения молекулы ротаксана из одного состояния в другое осуществляется с помощью света или слабого электрического поля. При этом тактовая частота процессора возрастает до $1 \text{ ТГц} = 10^{12} \text{ Гц}$. По прогнозам первый молекулярный компьютер появится к 2015 г.

2. **Биокомпьютеры.** Примером биокомпьютера является мозг человека. Применение в вычислительной технике биологических материалов делает возможным построение белковой памяти, создание биокомпьютера на ДНК. Такая ЭВМ будет иметь малые размеры, высокое быстродействие, потреблять мало энергии. Она может быть частью живого организма.

3. **Нейрокомпьютеры.** Это вычислительная система, созданная на базе нейронных систем живого мира. Примером искусственной нейронной сети является перцептрон. Нейрокомпьютерам присущи параллельность обработки информации, способность к обучению, к распознаванию образов,

установлению ассоциативных связей, высокая надежность.

4. *Оптические компьютеры.* Логические операции могут быть реализованы с помощью оптических элементов, что позволяет упростить работу оптических повторителей и усилителей оптоволоконных линий дальней связи. При этом используется явление *оптической бистабильности*: за счет нелинейности оптической среды возможны два стационарных состояния прошедшей световой волны, отличающихся интенсивностью и поляризацией. Оптоволокно имеет предел пропускной способности в 5–10 Гбит/с на один световой луч определенной длины волны. По каждому волокну может пропускаться до 8 лучей с разными длинами волн. Применяемые ЭВМ перейдут на оптическую основу, это позволит сохранять сигнал в световой форме и существенно повысить быстродействие.

5. *Квантовые компьютеры.* Квантовые ЭВМ состоят из совокупности микрочастиц (атомов), способных переходить из одного энергетического состояния в другое. Это осуществляется за счет вынужденных переходов атомов под действием световых волн (фотонов) с частотой $\nu = (E_2 - E_1)/h$. Спонтанные переходы должны быть исключены. При этом могут быть реализованы все логические операции: И, ИЛИ, НЕ. Единицей информации является *кубит* (qubit, Quantum Bit). Двум значениям кубита 0 и 1 могут соответствовать основное и возбужденное состояния атома, различная ориентация спина атомного ядра, направление тока в сверхпроводящем кольце и т.д. Условия работы квантового компьютера: 1) известно точное число частиц; 2) имеется способ приведения системы в определенное начальное состояние; 3) система изолирована от внешней среды; 4) возможно изменение состояния системы требуемым образом.

6. Создание вычислительных систем с искусственным интеллектом (ИИ). Под ИИ понимают способности компьютерных систем к таким действиям, которые назывались бы интеллектуальными, если бы исходили от человека: распознавание образов, эвристическое мышление.

В будущем электронно-вычислительная техника объединится не только со средствами связи, различными технологическими процессами, но и с биологическими организмами. Станет возможным создание искусственных имплантантов, человеко-машинных киборгов, разумных роботов-адроидов. К 2050 году ЭВМ достигнет мощности человеческого мозга.

Развитие компьютерной техники, нанотехнологии и биотехнологий неизбежно приведет к созданию сверхчеловеческого интеллекта, который качественным образом превосходит человеческий мозг по скорости поиска информации, скорости выполнения арифметических и логических операций, реализации различных эвристических алгоритмов. Этот момент в истории земной цивилизации получил название технологической сингулярности. Преимущества сверхчеловеческого интеллекта заключаются в следующем: 1) высокая скорость обработки и передачи информации; 2) практически бесконечное "время жизни", отсутствие необходимости каждый 50–70 лет обучаться "с нуля"; 3) возможность использования сигналов для управления различными электронными приборами и выхода в Интернет.

ПРИЛОЖЕНИЕ

1. Компьютерная модель машины Поста. Ниже представлена программа, моделирующая работу машины Поста. При запуске она должна считывать начальное состояние и программу из текстового файла progr.txt.

```

uses crt, dos; const N=35; zadergka=100; { Pascal }
var aaa,z: string; komand : array [1..N] of integer;
aa,a : array [1..N] of string; xxx, yyy : array [1..N] of integer;
error, s, NN, Koordinata, zapis, smeshen, P, x,y, i, ii : integer;
FF, f : text; Label K1, K2, Stop;
Procedure Pechat;
begin P:=P+1; writeln('НАЧАЛЬНОЕ СОСТОЯНИЕ: ');
sound(240); Delay(200*zadergka); Nosound;
For i:=1 to N do write(aa[i], ' '); writeln;
write('==',P,'=='); For i:=1 to N-2 do write('=='); writeln;
For i:=1 to N do write(a[i], ' '); writeln; delay(200*zadergka);
For i:=1 to Koordinata-1 do write('--'); write('M'); writeln;
writeln('K ',ii-2,' ',komand[ii-2],' ',xxx[ii-2],' ',yyy[ii-2]);
writeln('K ',ii-1,' ',komand[ii-1],' ',xxx[ii-1],' ',yyy[ii-1]);
writeln('K ',ii,' ',komand[ii],' ',xxx[ii],' ',yyy[ii]);
writeln('K ',ii+1,' ',komand[ii+1],' ',xxx[ii+1],' ',yyy[ii+1]);
writeln('K ',ii+2,' ',komand[ii+2],' ',xxx[ii+2],' ',yyy[ii+2]);
delay(200*zadergka); WRITELN; WRITELN; end;
Procedure Smestit(x:integer);
var i: integer;
begin for i:=1 to abs(x) do
  begin if x>0 then Koordinata:=Koordinata+1;
    if x<0 then Koordinata:=Koordinata-1; clrscr;
    writeln(' СМЕШАЮ ГОЛОВКУ '); writeln; writeln; Pechat;
    sound(500); Delay(200*zadergka); sound(1500);
    Delay(200*zadergka); Nosound; delay(200*zadergka); end; end;
Procedure Schitat;
Begin clrscr; writeln(' СЧИТЫВАЮ ИНФОРМАЦИЮ ');
writeln; writeln; Pechat;
z:=a[Koordinata]; Writeln('СЧИТАЛА a[' , Koordinata, ']=',z);
sound(50); Delay(200*zadergka); sound(150);
Delay(200*zadergka); Nosound; Delay(200*zadergka); end;
Procedure Postavit_Metku;
begin clrscr; writeln('СТАВЛЮ МЕТКУ'); writeln; writeln;
if a[Koordinata]='V' then begin Pechat; writeln;writeln;
writeln('ОШИБКА! МЕТКА УЖЕ СТОИТ. КОМАНДА ',ii);
error:=1; end else begin a[Koordinata]:='V'; Pechat; end;
sound(30); Delay(200*zadergka); sound(50); Delay(200*zadergka);
Nosound; delay(200*zadergka); end;
Procedure Steret;
Begin clrscr; writeln('СТИРАЮ МЕТКУ '); writeln; writeln;
if a[Koordinata]='-' then begin Pechat; writeln; writeln;
writeln('ОШИБКА! МЕТКИ УЖЕ НЕТ. КОМАНДА ',ii);
error:=1; end else begin a[Koordinata]:='-'; Pechat; end;
sound(120); delay(200*zadergka); nosound;
delay(200*zadergka); end;

```



```

Procedure Ustanovit(x:integer);
Begin clrscr; writeln('УСТАНОВЛИВАЮ ГОЛОВКУ');
writeln; writeln; Koordinata:=x; Pechat;
sound(150); delay(200*zadergka); nosound; delay(200*zadergka);
end;
BEGIN clrscr;
  for i:=1 to N do aa[i]:='-'; assign(FF,'progr.txt');
  reset(FF); readln(FF, Koordinata); readln(FF,aaa);
  for i:=1 to N do begin a[i]:=COPY(aaa,i,1); aa[i]:=a[i]; end;
  writeln('НАЧАЛЬНОЕ СОСТОЯНИЕ'); writeln; writeln; Pechat;
  sound(600); Delay(200*zadergka); sound(1600);
  delay(200*zadergka); nosound; delay(200*zadergka);
  delay(200*zadergka); writeln(' '); writeln(' '); i:=0;
  Repeat i:=i+1; readln(FF,komand[i],xxx[i],yyy[i]);
  until Eoln(FF); ii:=1; error:=0; k2:
  if Keypressed then goto k1; if komand[ii]=1010 then goto K1;
  if komand[ii]=1000 then
    begin Smestit(-1); ii:=xxx[ii]; goto k2; end;
  if komand[ii]=1001 then
    begin Smestit(1); ii:=xxx[ii]; goto k2; end;
  if komand[ii]=1011 then begin Steret;
    if error=1 then goto k1; ii:=xxx[ii]; goto k2; end;
  if komand[ii]=1111 then begin Postavit_Metku;
    if error=1 then goto k1; ii:=xxx[ii]; goto k2; end;
  if komand[ii]=1101 then begin Schitat;
    if z='- ' then ii:=xxx[ii] else ii:=yyy[ii]; goto k2; end;
  writeln('ОШИБКА В СТРОКЕ ', ii); Close(FF);
K1: writeln; writeln; writeln(' КОНЕЦ ИСПОЛНЕНИЯ ПРОГРАММЫ. ');
Repeat until Keypressed;
END.

```

Эту программу следует скомпилировать в ехе-файл и поместить в один каталог с файлом progr.txt, содержащим программу для МП. Рассмотрим две задачи с решениями (файлами progr.txt). Перед запуском пояснения следует удалить.

Задача 1. Напишите программу, складывающую два целых числа.

| 5 | | | | координата каретки |
|----------|-------|---|--|--------------------|
| VVVV-VVV | ----- | | | лента |
| 1111 | 2 | 0 | поставить метку | |
| 1001 | 3 | 0 | сместить вправо | |
| 1101 | 4 | 2 | если пусто -- команда 4, если метка -- команда 2 | |
| 1000 | 5 | 0 | сместить влево | |
| 1011 | 6 | 0 | удалить метку | |
| 1010 | 0 | 0 | остановить МП. | |

Задача 2. Напишите программу, вычитающую два целых числа.

| 7 | | | | координата каретки |
|-------------|-------|---|--|--------------------|
| VVVVVV-VVVV | ----- | | | лента |
| 1000 | 2 | 0 | сместить влево | |
| 1101 | 1 | 3 | если пусто -- команда 1, если метка -- команда 3 | |

3. Задано сообщение из 50 символов, алфавит содержит 5 букв. Напишите программу, которая определяет вероятности каждого символа и по формуле Шеннона вычисляет среднюю информацию, приходящуюся на один символ, и общую информацию в сообщении.

4. С помощью предыдущей программы убедитесь в том, что информативность сообщения максимальна тогда, когда все символы используются с равными вероятностями. Если в сообщении повторяется один символ, то информативность сообщения равна 0.

5. Сгенерируйте случайное сообщение из 50 символов, содержащее букву А с вероятностью 30 %, В — с вероятностью 35 %, С — с вероятностью 23 % и D — с вероятностью 12 % .

6. В сообщении буквы А, В, С, D встречаются с вероятностями $p_A = 0,45$, $p_B = 0,22$, $p_C = 0,14$, $p_D = 0,19$. Используются два кода: 1) А - 00, В - 01, С - 10, D - 11; 2) А - 0, В - 10, С - 110, D - 111. Определите избыточность этих кодов.

7. Напишите программу, кодирующую двоичным кодом сообщение из 20 букв на алфавите из 8 букв. На каждую букву приходится по 3 двоичных разряда: 000, 001, 010, ... , 111.

```

uses crt;
var f,s,a,b:string; i,k:integer; x:real;
BEGIN clrscr; s:='aaabbcbbddeeebbcbadeaabbdddeeffgh'; writeln(s); b:='';
  for i:=1 to length(s) do begin a:=s[i];
    if a='a' then b:=b+'000'; if a='b' then b:=b+'001';
    if a='c' then b:=b+'010'; if a='d' then b:=b+'011';
    if a='e' then b:=b+'100'; if a='f' then b:=b+'101';
    if a='g' then b:=b+'110'; if a='h' then b:=b+'111'; end;
  write(b); writeln; randomize; s:='';
  for i:=1 to length(b) do begin f:=copy(b,3*i-2,3);
    if f='000' then s:=s+'a'; if f='001' then s:=s+'b';
    if f='010' then s:=s+'c'; if f='011' then s:=s+'d';
    if f='100' then s:=s+'e'; if f='101' then s:=s+'f';
    if f='110' then s:=s+'g'; if f='111' then s:=s+'h'; end;
  write(s); readln;
END.

```

8. Создайте программу, декодирующую сообщение, закодированное предыдущим способом.

9. Имеется сообщение из 30 букв на алфавите из 8 букв. Напишите программу, которая кодирует каждую букву тремя битами, после чего случайно с вероятностью $p = 0,1$ вносит ошибки (инвертирует биты), и декодирует сообщение. Результат каждого действия должен выводиться на экран.

10. Имеется сообщение 01101 ... 01. Напишите программу, кодирующую его помехоустойчивым кодом, в котором каждый бит утраивается, затем вносит ошибки с заданной вероятностью p и декодирует сообщение.

```
uses crt; var aa1,aa,y,z,zz,x1,x : string; i : byte; s,k,p : real;
BEGIN clrscr;
    writeln('Исходное сообщение: ');
    aa1:='10101010101010101010101010101010101010101010101010';
    writeln(aa1); writeln('Задайте вероятность ошибки: ');
    readln(p);
    for i:=1 to length(aa1) do begin x:=copy(aa1,i,1);
        if x='0' then y:='000'; if x='1' then y:='111'; z:=z+y; end;
    writeln('Закодированное сообщение: '); writeln(z);
    randomize; zz:='';
    for i:=1 to length(z) do begin s:=random(1000)/1000;
        if (s<p)and(z[i]='0') then zz:=zz+'1';
        if (s<p)and(z[i]='1') then zz:=zz+'0';
```

```

        if s>=p then zz:=zz+z[i]; end;
writeln('Сообщение с ошибкой: '); writeln(zz); aa:=''; i:=1;
while i<length(zz) do begin x:=copy(zz,i,3);
    if x='000' then y:='0'; if x='001' then y:='0';
    if x='010' then y:='0'; if x='011' then y:='1';
    if x='100' then y:='0'; if x='101' then y:='1';
    if x='110' then y:='1'; if x='111' then y:='1';
    aa:=aa+y; i:=i+3; end;
writeln('Декодированное сообщение: '); writeln(aa);
for i:=1 to length(aa) do begin x1:=copy(aa,i,1);
x:=copy(aa,i,1); if x1<>x then k:=k+1; end; k:=k/length(aa);
writeln('Относительное количество ошибок: ',k); readln;
END.

```

11. Используя предыдущую программу, исследуйте зависимость относительного числа ошибок k декодирования от вероятности p , с которой инвертируются биты при передаче сообщения по линии связи. Величину k определите, сравнивая исходное и конечное сообщения.

12. Имеется сообщение 01101 ... 01. Напишите программу, которая разбивает его на кадры по 7 бит и добавляет восьмой бит четности.

13. Дополните предыдущую программу так, чтобы она вносила бы ошибки в сообщение, получающееся после добавления битов четности, а затем выявила бы их. Промежуточные результаты выведите на экран.

```

uses crt; {Вставляет бит четности, вносит ошибки, и находит ошибку}
var s,s1,a,a1,b,b1,p,f : string;
i,i1,j,j1,k,k1,q,q1,n,x,x1 : integer; w : real;
BEGIN clrscr; n:=49; s1:='';
s:='0010100101110110001010101001010110010010';
writeln('Исходное сообщение');
writeln(s); k:=round(length(s)/7);
writeln('Сообщение с битом четности');
for i:=1 to k do begin a:=copy(s,7*i-6,7); q:=round(length(a)/7);
for j:=1 to q do begin x:=0; b:=copy(a,j-1,1);
if b='1' then x:=x+1; end;
if (x mod 2)=0 then a:=a+'0' else a:=a+'1';
write(a); s1:=s1+a; f:=s1; end; writeln; randomize;
for i:=1 to length(f) do begin w:=random(100)/100;
if (w<0.1) and (s1[i]='1') then s1[i]:='0';
if (w<0.1) and (s1[i]='0') then s1[i]:='1'; end;
writeln('Сообщение с ошибкой'); writeln(s1); readln;
END.

```

14. Создайте программу, шифрующую сообщение из 32 букв русского языка путем их перемешивания и добавления новых символов.

15. Напишите программу, которая дешифрует зашифрованное в предыдущей задаче сообщение.

16. Создайте программу, которая архивирует строку 111011001...1 по принципу "значение бита — число повторов": 13011202..., а затем восстанавливает ее.

17. Напряжение на входе АЦП изменяется по закону $u = Ae^{-bt}$. Число уровней напряжения равно 512. Сохраните в массиве $u[i]$ результат

оцифровки. Методом покоординатного спуска восстановите значения A и b . Постройте график исходного сигнала, точки после оцифровки, график восстановленного сигнала. Программа – ниже.

```
uses crt, graph; var i: integer; A, b, nev, nevyazka : real;
u:array[1..100]of integer;
BEGIN clrscr; for i:=1 to 100 do u[i]:=round(512*exp(-0.05*i));
  for i:=1 to 100 do write(u[i], ' '); A:=100; b:=0.001;
  Repeat nevyazka:=0;
  for i:=1 to 100 do nevyazka:=nevyazka+abs(u[i]-A*exp(-b*i));
  A:=A+0.5; nev:=0;
  for i:=1 to 100 do nev:=nev+abs(u[i]-A*exp(-b*i));
  if nev>nevyazka then A:=A-0.5; nevyazka:=0;
  for i:=1 to 100 do nevyazka:=nevyazka+abs(u[i]-A*exp(-b*i));
  b:=b+0.00001; nev:=0;
  for i:=1 to 100 do nev:=nev+abs(u[i]-A*exp(-b*i));
  if nev>nevyazka then b:=b-0.00001;
  writeln(A, ' ', b, ' ', nevyazka);
  until nevyazka<0.1;
  Repeat until KeyPressed;
END.
```

18. Убедитесь в справедливости теоремы Котельникова, для этого оцифруйте входной гармонический сигнал частотой f и восстановите его, соединяя точки отрезками. Частота отсчетов должна быть больше $2f$ и меньше $2f$.

19. Создайте однослойную нейросеть с 5 входными нейронами и 3 выходными, которая распознавала бы объекты 11000, 00011, 01110.

20. Создайте двуслойную нейросеть с 8 входными нейронами, 5 промежуточными и 3 выходными нейронами, которая распознавала бы образы объектов 11110000, 00001111, 00111100. Как распознает нейросеть другие объекты?

21. Археологи А, Б и В нашли монету. Каждый высказал по 2 предположения: А: монета греческая, 5 век; Б: монета испанская, 3 век; В: монета не греческая, 4 век. Каждый из археологов прав только в одном из двух предположений. Где и когда была выпущена монета?

Решение: Обозначим простые высказывания: G — монета греческая, I — монета испанская, P — 5 век, C — четвертый век, T — третий век. Высказывания: $G\bar{P} + \bar{G}P = 1$, $I\bar{T} + \bar{I}T = 1$, $\bar{G}\bar{C} + GC = 1$. Монета не может быть изготовлена в двух государствах и двух веках: $GI = 0$, $PC = 0$, $PT = 0$, $CT = 0$. Необходимо написать программу, в которой перебираются все варианты и автоматически находится приемлемый.

```
uses crt;
var G,P,I,T,C: boolean; gg,pp,tt,cc,ii,k,l,m,n,o: integer;
BEGIN clrscr; k:=0;
  for gg:=0 to 1 do begin if gg=0 then g:=false else g:=true;
  for pp:=0 to 1 do begin if pp=0 then p:=false else p:=true;
  for tt:=0 to 1 do begin if tt=0 then t:=false else t:=true;
  for cc:=0 to 1 do begin if cc=0 then c:=false else c:=true;
  for ii:=0 to 1 do begin if ii=0 then i:=false else i:=true;
```

```

if ((g)and(not(p))or(not(g)and(p))=true) then k:=k+1;
if ((i)and(not(t))or(not(i)and(t))=true) then k:=k+1;
if (not(g)and(not(c))or((g)and(c))=true) then k:=k+1;
if(g)and(i)=false then k:=k+1;
if(p)and(c)=false then k:=k+1;
if(p)and(t)=false then k:=k+1;
if(c)and(t)=false then k:=k+1;
writeln(g,' ',p,' ',t,' ',c,' ',i,' ',k,' '); k:=0;
end; end; end; end; end; writeln(k); readkey;
END.

```

22. Одним из трех подозреваемых А, В, С совершено преступление. На допросе они дали следующие показания: А: 1) это сделал не я; 2) это сделал С; 3) меня там не было; В: 1) я этого не делал; 2) это сделал А; 3) с С я не знаком; С: 1) это сделал не я; 2) с А я знаком; 3) это сделал не В. Выяснилось, что каждый сделал 2 верных утверждения и одно ложное. Кто совершил преступление?

Решение. Обозначим простые высказывания через A_1, A_2, \dots, C_3 . Получаем систему логических уравнений:

$$\begin{aligned} \bar{A}_1 A_2 A_3 + A_1 \bar{A}_2 A_3 + A_1 A_2 \bar{A}_3 &= 1, \quad \bar{B}_1 B_2 B_3 + B_1 \bar{B}_2 B_3 + B_1 B_2 \bar{B}_3 = 1 \\ \bar{C}_1 C_2 C_3 + C_1 \bar{C}_2 C_3 + C_1 C_2 \bar{C}_3 &= 1, \quad A_1 \bar{B}_2 = 1, \quad B_1 \bar{C}_2 = 1, \quad A_2 \bar{C}_1 = 1. \end{aligned}$$

Задача решается методом перебора всех возможных вариантов.

23. Напишите программу, которая работает так: компьютер случайно загадывает число от 1 до 256. Вы пытаетесь угадать, а компьютер отвечает "больше" или "меньше". Предложите оптимальную стратегию отгадывания.

24. Промоделируйте игру "Жизнь". Клетка оживает при наличии 3 живых соседей. Если живых соседей 4 и больше, она умирает от перенаселенности. Если живых соседей меньше 2, она умирает от одиночества.

```

uses crt; const N=26;
type z1=record x,y,xy,x1,y1 : real end;
massiv = array[-1..N+1,-1..N+1] of integer;
var z,y,x,x1: massiv; s,i,j,k,l,m : integer;
procedure Print; var i,j : integer;
begin clrscr; for i:=1 to N do begin for j:=1 to N do begin
if x[i,j]=1 then Write(' * '); if x[i,j]=0 then Write(' ');
end; writeln; end; end;
procedure Oboznach; var i,j: integer;
begin for i:=1 to N do begin
for j:=1 to N do x1[i,j]:=x[i,j]; end; end;
procedure Raschet; var i,j : integer;
begin For i:=1 to N do For j:=1 to N do begin
S:=x1[i-1,j-1]+x1[i-1,j]+x1[i-1,j+1]+x1[i,j-1]+
x1[i,j+1]+x1[i+1,j-1]+x1[i+1,j]+x1[i+1,j+1];
if s=3 then x[i,j]:=1; if (s<2)or(s>3) then x[i,j]:=0; end; end;
BEGIN For i:=-1 to N+1 do For j:=-1 to N+1 do x[i,j]:=0;
{x[14,18]:=1; x[14,17]:=1; x[14,16]:=1; x[14,15]:=1;}
{x[14,17]:=1; x[14,16]:=1; x[14,15]:=1;}
{x[14,17]:=1; x[14,16]:=1; x[14,15]:=1; x[13,15]:=1; x[12,16]:=1;}

```

```
{x[15,13]:=1; x[15,14]:=1; x[15,15]:=1; x[16,14]:=1;}
{x[14,17]:=1; x[14,16]:=1; x[14,15]:=1;
x[15,16]:=1; x[15,15]:=1; x[15,14]:=1;}
x[14,17]:=1; x[14,16]:=1; x[14,15]:=1; x[15,16]:=1; x[15,15]:=1;
x[15,14]:=1; x[14,10]:=1; x[14,11]:=1; x[14,12]:=1; x[15,11]:=1;
x[15,15]:=1; x[15,13]:=1;
Print; Repeat delay(6000); Oboznach; Raschet; Print;
until keypressed;
END.
```

25. Промоделируйте работу автомата, имеющего 4 внутренних состояния. Нарисуйте диаграмму Мура.

```
uses crt; var i : integer; x,s,a,b,c,y : string;
q : integer; label 1;
BEGIN clrscr; s:='abacaaacccbbb'; writeln('S=',S); q:=1;
for i:=1 to length(s) do begin x:=copy(S,i,1);
if (x='a') and (q=1) then begin q:=1; y:='F'; goto 1;end;
if (x='c') and (q=1) then begin q:=2; y:='G'; goto 1;end;
if (x='b') and (q=1) then begin q:=4; y:='E'; goto 1;end;
if (x='b') and (q=2) then begin q:=2; y:='G'; goto 1;end;
if (x='a') and (q=2) then begin q:=3; y:='F'; goto 1;end;
if (x='c') and (q=2) then begin q:=4; y:='F'; goto 1;end;
if (x='a') and (q=3) then begin q:=4; y:='F'; goto 1;end;
if (x='c') and (q=3) then begin q:=3; y:='E'; goto 1;end;
if (x='b') and (q=3) then begin q:=2; y:='F'; goto 1;end;
if (x='a') and (q=4) then begin q:=4; y:='E'; goto 1;end;
if (x='b') and (q=4) then begin q:=3; y:='F'; goto 1;end;
if (x='c') and (q=4) then begin q:=2; y:='G'; goto 1;end;
1: writeln(q,' ',y); end; readkey;
END.
```

26. Промоделируйте работу исполнителя, перемещающегося по горизонтальной поверхности в соответствии с заданной программой.

```
uses graph, crt; var k4,k1,k,k2,k3,i,Gd,Gm,p: integer; a,x: string;
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
a:='drdrdruuululululululululldddd'; k:=0; moveto(320,240);
for i:=1 to length(a) do begin x:=copy(a,i,1);
if x='r' then begin k1:=k+30; linerel(k1,0) end;
if x='u' then begin k2:=k+30; linerel(0,k2) end;
if x='l' then begin k3:=k-30; linerel(k3,0) end;
if x='d' then begin k4:=k-30; linerel(0,k4) end;
delay(5000);sound(400); delay(1500);nosound;
end; Readkey; CloseGraph;
END.
```

27. Напишите обучающую программу, проверяющую умение решать примеры по арифметике.

```
uses crt; var s,i,j,a,b,c,n : integer;
BEGIN clrscr; randomize; s:=0;
for j:=1 to 5 do begin a:=random(9); b:=random(9);
c:=a*b; writeln; writeln('Сколько будет ',a,'*',b );
writeln('ответ'); readln(n);
```



```

if n=c then begin write('ВЕРНО') ;s:=s+1;
if n<>c then begin write('НЕТ'); end; end;
write('Число правильных ответов=',s); end;
if s=5 then begin write ('ваша оценка 5'); end;
if s=4 then begin write ('ваша оценка 4'); end;
if s=3 then begin write ('ваша оценка 3');end;
if s=2 then begin write ('ваша оценка 2');end;
if (s=1) or (s=0) then begin write('попробуйте заново'); end;
readkey;
END.

```

30. Создайте программу, переводящую десятичное число в шестнадцатичную систему счисления.

```

uses crt; var a,b,c,d : integer; label 10;
BEGIN clrscr; write('введите число в 10 система счисления ');
read(a); c:=20; repeat
c:=c-1; b:=a; a:=a div 16; d:=b-16*a; gotoxy(c,5);
if d=10 then begin write('a'); goto 10; end;
if d=11 then begin write('b'); goto 10; end;
if d=12 then begin write('c'); goto 10; end;
if d=13 then begin write('d'); goto 10; end;
if d=14 then begin write('e'); goto 10; end;
if d=15 then begin write('f'); goto 10; end; write(d);
10: until a=0; readkey;
END.

```

3. Моделирование физических явлений на ПЭВМ. Компьютер может быть использован для моделирования физических систем. Рассмотрим задачу: Имеется прямоугольная пластина из диэлектрика с диэлектрической проницаемостью $\varepsilon = \varepsilon(x, y)$. Задана плотность распределения заряда $\rho = \rho(x, y)$ и граничные условия $\varphi = \varphi(x, y)$. Необходимо рассчитать распределение потенциала $\varphi = \varphi(x, y)$ во всех точках пластины.

Потенциал поля удовлетворяет уравнению Пуассона, которое для неоднородной двумерной среды записывается так:

$$\frac{\partial}{\partial x} \left(\varepsilon(x, y) \frac{\partial \varphi}{\partial x} \right) + \frac{\partial}{\partial y} \left(\varepsilon(x, y) \frac{\partial \varphi}{\partial y} \right) = \rho(x, y).$$

Итак, мы получили задачу Дирихле: необходимо решить дифференциальное уравнение в частных производных в замкнутой области Ω при известном распределении искомой функции φ на границе области. Основная идея использования метода сеток для решения ДУЧП в области Ω состоит в следующем: 1) от непрерывной области Ω переходят к сеточной области $\Omega_{\Delta x, \Delta y}$; 2) для узлов сетки $\Omega_{\Delta x, \Delta y}$ составляют конечно-разностные уравнения; 3) исходя из начальных и граничных условий задаются начальные значения искомой функции $\varphi = \varphi(x, y)$ и ее значения в граничных узлах сетки $\Omega_{\Delta x, \Delta y}$; 4) решают получившуюся систему конечно-разностных уравнений, определяя значения функции в узлах сетки.

Переход от решаемого ДУЧП к конечно-разностному уравнению осуществляется путем замены производных на их конечно-разностные аппроксимации:

$$\frac{\varepsilon_{i-1,j} - \varepsilon_{i+1,j}}{2h} \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2h} + \varepsilon_{i,j} \frac{\varphi_{i-1,j} - 2\varphi_{i,j} + \varphi_{i+1,j}}{h^2} +$$

$$+ \frac{\varepsilon_{i,j-1} - \varepsilon_{i,j+1}}{2h} \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2h} + \varepsilon_{i,j} \frac{\varphi_{i,j-1} - 2\varphi_{i,j} + \varphi_{i,j+1}}{h^2} = -\rho_{i,j}.$$

$$\varphi_{i,j} = (\varphi_{i+1,j} + \varphi_{i-1,j} + \varphi_{i,j+1} + \varphi_{i,j-1} + \rho_{i,j}h^2/\varepsilon_{i,j})/4 + [(\varepsilon_{i-1,j} - \varepsilon_{i+1,j}) \cdot (\varphi_{i+1,j} - \varphi_{i-1,j}) + (\varepsilon_{i,j-1} - \varepsilon_{i,j+1}) \cdot (\varphi_{i,j+1} - \varphi_{i,j-1})]/(16\varepsilon_{i,j}).$$

Для решения рассматриваемой стационарной задачи используется релаксационный метод последовательных приближений: во всех внутренних узлах сетки задаются произвольные исходные значения искомой функции $\varphi = \varphi(x, y)$; значения функции во внешних узлах должны соответствовать граничным условиям. Осуществляется первая итерация, в ходе которой перебираются все внутренние узлы сетки и исходя из начальных значений определяют новые уточненные значения функции $\varphi(x_i, y_i)$. Затем осуществляется вторая, третья, ... и т.д. итерации, причем результаты i -ой итерации используются в качестве исходных для $i + 1$ -ой итерации. В результате получающиеся значения $\varphi(x_i, y_i)$ приближаются к истинному распределению потенциала $\varphi(x, y)$.

Используемый алгоритм состоит в последовательном переборе узлов сетки слева направо и справа налево, сверху вниз и снизу вверх, в ходе которого вычисляются значения потенциала $\varphi_{i,j}$. Перед каждым проходом следует учесть граничные условия задачи, допустим приравнять потенциалы на краю двумерной области к некоторой величине или функции координат. Если внутри области имеются точки, потенциалы которых остаются неизменными и равными φ_0 , например, точки проводника, то необходимо их приравнять к φ_0 .

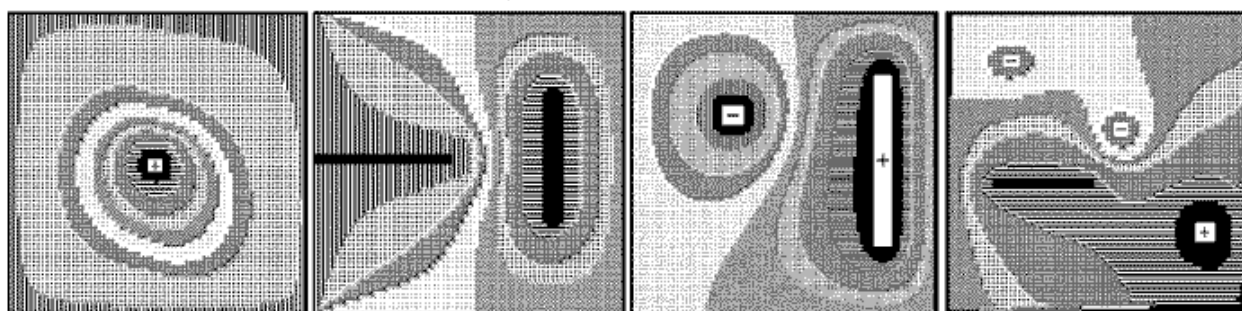


Рис. 8.1.

Программа содержит: а) процедуру Sreda, в которой задается диэлектрическая проницаемость среды $\varepsilon_{i,j}$ в узлах сетки; б) процедуру Gran, в которой учитывается распределение потенциала вдоль границы области; в) процедуру Raschet, в которой задается распределение заряда и вычисляется потенциал в различных узлах i, j ; г) процедуру Draw, выводящей

результат вычислений на экран; д) основную часть программы, в которой осуществляется вызов перечисленных процедур в требуемом порядке.

На рис. 8.1 показаны распределения потенциала двумерных полей: а) поля точечного заряда в неоднородной среде; б) поля, созданного острием и пластиной, имеющих разные потенциалы; в) поля небольшого тела и пластины, имеющих заряды разных знаков; г) поля нескольких заряженных тел вблизи, которых расположен заземленный проводник. Во каждом случае заданы свои граничные условия. Результаты расчетов потенциала сохраняются в двумерных массивах и могут быть использованы для построения силовых линий методом покоординатного спуска.

```

uses crt, graph;                                     {Pascal}
const N=70; M=70;
var i, j, ii, jj, k, q, DriverVar, ModeVar, ErrorCode :integer;
fi, e: array[1..N, 1..M] of real;
procedure Sreda; {---Диэлектрическая проницаемость среды---}
begin For i:=1 to N do For j:=1 to M do
begin e[i,j]:=1+0.015*j*i; end; end;
procedure Gran; {---- Граничные условия ----}
begin for i:=1 to N do begin fi[i,1]:=-350; fi[i,M]:=350;
fi[i,2]:=-350; fi[i,M-1]:=350; fi[2,i]:=-350+10*i;
fi[N-1,i]:=-350+10*i; fi[1,i]:=-350+10*i; fi[N,i]:=-350+10*i;
end; for j:=15 to 25 do fi[55,j]:=0; end;
procedure Raschet; {---- Расчет потенциала ----}
begin q:=0;
if ((i>20)and(i<30)and(j>30)and(j<35))then q:=500;
if ((i>40)and(i<55)and(j>40)and(j<55))then q:=-300;
fi[i,j]:=(fi[i+1,j]+fi[i-1,j]+fi[i,j+1]+fi[i,j-1]+q/e[i,j])/4+
((e[i-1,j]-e[i+1,j])*(fi[i+1,j]-fi[i-1,j])+
(e[i,j-1]-e[i,j+1])*(fi[i,j+1]-fi[i,j-1]))/(16*e[i,j])); end;
procedure Draw; {---- Вывод на экран ----}
begin setcolor(round(fi[i,j]/30));
rectangle(i*4+50,j*4,i*4+54,j*4+4);
rectangle(i*4+51,j*4+1,i*4+53,j*4+3); end;
BEGIN {---- Основная программа ----}
DriverVar:=Detect; InitGraph(DriverVar,ModeVar,'c:\bp\bgi');
ErrorCode:=GraphResult; if ErrorCode <> grOK then Halt(1);
Sreda; Repeat inc(k);
For i:=2 to N-1 do For j:=2 to M-1 do Raschet;
Gran; For j:=2 to M-1 do For i:=2 to N-1 do Raschet;
Gran; For i:=2 to N-1 do For jj:=2 to M-1 do
begin j:=M+1-jj; Raschet; end;
Gran; For j:=2 to M-1 do For ii:=2 to N-1 do
begin i:=N+1-ii; Raschet; end;
Gran; If k/10=round(k/10) then For i:=2 to N-1 do
For j:=2 to M-1 do Draw;
until KeyPressed; CloseGraph;
END.

```

ЛИТЕРАТУРА

1. Акатов, Р. В. Компьютерные измерения: Аналого–цифровой преобразователь [Текст] / Р. В. Акатов. — Учебная физика. — 1999. — N 3. — 48–64 с.
2. Аветисян, Р. Д., Аветисян Д. О. Теоретические основы информатики [Текст] / Р. Д. Аветисян, Д. О. Аветисян. — М.: РГНУ, 1997 — 167 с.
3. Акулов, О. А. Информатика: базовый курс: Учебник для студентов вузов, бакалавров, магистров, обучающихся по направлениям 552800, 65460 "Информатика и вычислительная техника" [Текст] / О. А. Акулов, Н. В. Медведев. — М.: Омега–Л, 2004. — 552 с.
4. Вентцель, Е. С. Исследование операций: задачи, принципы, методология [Текст] / Е. С. Вентцель. — М.: Наука. Гл. ред. физ.–мат. лит., 1988. — 208 с.
5. Гершензон, Е. М. Радиотехника: Учеб. пособие для студентов физ.–мат. фак. пед. ин–тов. [Текст] / Е. М. Гершензон, Г. Д. Полянина, Н. В. Соина. — М.: Просвещение, 1986. — 319 с.
6. Глушков, В. М. Введение в кибернетику [Текст] / В. М. Глушков. — Киев: Изд–во Академии наук Украинской ССР, 1964. — 324 с.
7. Гуц, А. К. Математическая логика и теория алгоритмов: Учебное пособие [Текст] / А. К. Гуц. — Омск: Изд–во Наследие. Диалог – Сибирь, 2004. — 108 с.
8. Душин, В. К. Теоретические основы информационных процессов и систем: Учебник [Текст] / В. К. Душин. — Издательско–торговая корпорация "Дашков и К^о", 2003. — 348 с.
9. Калиш Г. Г. Основы вычислительной техники: Учебн. пособ. для средн. проф. учебных заведений [Текст] / Г. Г. Калиш. — М.: Высш. шк., 2000. — 271 с.
10. Лидовский, В. В. Теория информации: Учебное пособие [Текст] / В. В. Лидовский. — М.: Компания Спутник+, 2004. — 111 с.
11. Майоров, С. А. Введение в микроЭВМ [Текст] / С. А. Майоров, В. В. Кириллов, А. А. Приблуда. — Л.: Машиностроение. Ленингр. отд–ние, 1988. — 304 с.
12. Майер, Р. В. Информационные технологии и физическое образование [Текст] / Р. В. Майер. — Глазов: ГГПИ, 2006. — 64 с.
13. Максимов, Н. В. Архитектура ЭВМ и вычислительных систем: Учебник [Текст] / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. — М.: ФОРУМ: ИНФРА–М, 2006. — 512 с.
14. Матаев, Г. Г. Компьютерная лаборатория в вузе и школе. Учебное пособие / Г. Г. Матаев. — М.: Горячая линия – Телеком, 2004. — 440 с.
15. Микаэлян, А. Л. Оптические методы в информатике: Запись, обработка и передача информации [Текст] / А. Л. Микаэлян. — М.: Наука, 1990. — 232 с.
16. Мнееян, М. Г. Физические принципы работы ЭВМ: Кн. для внеклас. чтения учащихся 8–10 кл. сред. шк. [Текст] / М. Г. Мнееян. — М.: Просвещение, 1987. — 192 с.
17. Могилев, А. В. Информатика: Учебн. пособие для студ. пед. вузов. [Текст] / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. — М.: Издательский центр "Академия", 2003. — 816 с.

18. Муттер, В.М. Основы помехоустойчивой телепередачи информации. [Текст] / В. М. Муттер. — Л.: Энергоиздат. Ленингр. отд-ние, 1990. — 288 с.
19. Нейман, Дж. Теория самовоспроизводящихся автоматов [Текст] / Дж. Нейман. — М.: Мир, 1971. — 382 с.
20. Новиков, Ю.Н. Компьютеры, сети, Интернет. Энциклопедия [Текст] / Ю.Н. Новиков, Д. Ю. Новиков, А. С. Черепанов, В. И. Чуркин. — СПб.: Питер, 2002. — 928 с.
21. Пахомов, С. Нанотехнологии на службе Intel [Текст] / С. Пахомов. / Компьютер Пресс. — 2005. — N 4. — С. 152–159.
22. Петцольд, Ч. Код [Текст] / Ч. Петцольд. — М.: Издательско-торговый дом "Русская редакция", 2001. — 512 с.
23. Попов, В.Б. Turbo Pascal для школьников: Учеб. пособие [Текст] / В. Б. Попов. — М.: Финансы и статистика, 2001. — 528 с.
24. Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации [Текст] / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко. — М.: Финансы и статистика, 2002. — 512 с.
25. Рыжиков, Ю. В. Информатика: лекции и практикум [Текст] / Ю. В. Рыжиков. — СПб.: КОРОНАпринт, 2000. — 256 с.
26. Сворень, Р. Электроника шаг за шагом: практическая энциклопедия юного радиолюбителя [Текст] / Р. Сворень. — М.: Дет. лит., 1986. — 431 с.
27. Соловьев, Г.Н. Схемотехника ЭВМ: Учебник для вузов [Текст] / Г. Н. Соловьев, Б. И. Кальнин, Ю. А. Попов и др. — М.: Высшая школа, 1985. — 391 с.
28. Стариченко, Б. Е. Теоретические основы информатики: Учебное пособие для вузов [Текст] / Б. Е. Стариченко. — М.: Горячая линия – Телеком, 2003. — 312 с.
29. Стратанович, Р.Л. Теория информации [Текст] / Р. Л. Стратанович. — М.: Сов. радио, 1975. — 424 с.
30. Трофимова, И. П. Системы обработки и хранения информации: Учеб. для вузов [Текст] / И. П. Трофимова. — М.: Высш. шк., 1989.— 191 с.
31. Успенский, В. А. Теория алгоритмов: основные открытия и приложения [Текст] / В. А. Успенский, А. Л. Семенов. — М.: Наука. Гл. ред. физ.-мат. лит., 1987. — 288 с.
32. Чисар, И. Теория информации: Теоремы кодирования для дискретных систем без памяти [Текст] / И. Чисар, Я. Кернер. — М.: Мир, 1985. — 395 с.
33. Шахинпур, М. Курс робототехники [Текст] / М. Шахинпур. — М.: Мир, 1990. — 527 с.
34. Электроника: Энциклопедический словарь [Текст] / Гл. ред. В. Г. Колесников. — М.: Сов. энциклопедия, 1991. — 688 с.
35. Энциклопедия кибернетики. В 2-х томах. [Текст] // Отв. ред. В. М. Глушков. — Киев: Гл. ред. Украинской Советской энциклопедии, 1974. — Т. 1. — 607 с. Т. 2. — 620 с.

СОДЕРЖАНИЕ

| | |
|--|------------|
| ВСТУПЛЕНИЕ | 3 |
| Глава 1. | |
| ОТ ТРАНЗИСТОРА ДО ТЕЛЕВИЗОРА | 6 |
| 1.1. Элементная база ЭВМ и других электронных устройств (6). | |
| 1.2. Устройства, вырабатывающие и преобразующие аналоговые сигналы (12). | |
| 1.3. Передача информации на большие расстояния (25). | |
| Глава 2. | |
| ИЗУЧАЕМ ЦИФРОВУЮ ЭЛЕКТРОНИКУ | 38 |
| 2.1. Основы цифровой электроники (38). | |
| 2.2. Как работают цифровые приборы (49). | |
| 2.3. Что такое цифровая связь (57). | |
| Глава 3. | |
| КАК УСТРОЕНА ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНАЯ МАШИНА | 63 |
| 3.1. Устройство и принцип действия ЭВМ (63). | |
| 3.2. Внешние устройства цифровых ЭВМ (81). | |
| Глава 4. | |
| ДЕЛАЕМ ЭКСПЕРИМЕНТЫ С КОМПЬЮТЕРОМ | 92 |
| 4.1. Ввод цифрового сигнала в ПЭВМ (92). | |
| 4.2. Ввод аналогового сигнала в ПЭВМ (101). | |
| 4.3. Вывод сигналов из ПЭВМ (108). | |
| 4.4. Замкнутые САУ на базе ПЭВМ (112). | |
| Глава 5. | |
| ОСНОВЫ ТЕОРИИ КОДИРОВАНИЯ И ПЕРЕДАЧИ ИНФОРМАЦИИ | 117 |
| 5.1. Данные, информация, кодирование (117). | |
| 5.2. Элементы теории кодирования (129). | |
| Глава 6. | |
| АЛГОРИТМЫ, ПЕРЦЕПТРОНЫ И ПРОЧИЕ ИГРЫ | 144 |
| 6.1. Об алгоритмах и автоматах (144). | |
| 6.2. Графы, деревья и жадные алгоритмы (160). | |
| Глава 7. | |
| ПРИКЛАДНАЯ ИНФОРМАТИКА | 169 |
| 7.1. Информационные технологии (169). | |
| 7.2. Программируем на языке QBASIC (178). | |
| 7.3. Искусственный интеллект. Перспективы ИТ (185). | |
| ПРИЛОЖЕНИЕ | 190 |
| ЛИТЕРАТУРА | 201 |
| | 203 |

Учебное издание

Майер Роберт Валерьевич

**Как стать компьютерным гением или
книга о информационных системах и технологиях**

Отпечатано с оригинал–макета автора
в авторской редакции.

Издательская лицензия ИД N 06035 от 12.10.2001.

Подписано в печать 01.08.08. Напечатано на ризографе.
Формат 60 x 90 1/16. Усл. печ.л. 14,1. Учетн.–изд.л. 14,8.

Тираж 200 экз. Заказ N

ГОУ ВПО "Глазовский государственный педагогический институт
им. В.Г. Короленко".

427621, Удмуртия, г. Глазов, ул. Первомайская, 25.
