

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ	6
ВВЕДЕНИЕ	9
1.ИНФОРМАЦИЯ И ИНФОРМАТИКА	11
1.1.Понятие информатики.....	11
1.2.Место информатики в ряду других фундаментальных наук.....	17
1.3.Социальные, правовые и этические аспекты информатики.....	21
1.4.Информация и ее измерение.....	31
1.4.1.Измерение информации в быту (информация как новизна)	32
1.4.2.Измерение информации в технике	32
1.4.3.Измерение информации в теории информации (информация как снятая неопределенность).....	34
1.5.Преобразование сообщения в сигналы.....	38
1.5.1.Кодирование сообщений	38
1.5.2.Равномерные простые коды	40
1.5.3.Неравномерные коды	42
1.5.4.Первичные коды	45
1.6. Сигналы, спектры сигналов, модуляция и кодирование	47
1.6.1.Понятие сигнала	47
1.6.2.Шумы и помехи	51
1.6.3.Размерность сигналов	54
1.6.4.Спектральное и математическое представление сигналов.....	56
1.6.5.Классификация сигналов.....	59
1.6.6.Типы сигналов	60
1.7.Помехоустойчивое кодирование и фильтрация	63
1.7.1.Статистическое (эффективное) кодирование.....	67
1.7.2.Помехоустойчивое кодирование.	68

1.7.3.Принцип обнаружения и исправления ошибок.	69
1.8.Фазы информационного цикла и их модели.....	70
1.8.1.Сбор информации	70
1.8.2.Передача информации	73
1.8.3.Обработка информации	75
1.9.Информационный ресурс и его составляющие	82
1.9.1.Определение информационной технологии	82
1.9.2.Свойства информационных технологий.....	84
1.9.3.Особенности информационных технологий.....	86
1.10.Виды обработки данных	89
1.10.1.Процедуры обработки данных в зависимости от видов представления данных	89
1.10.2.Технология обработки изображений	91
1.10.3.Видеотехнология	92
1.10.4.Обработка текстов.....	93
1.10.5.Обработка таблиц и технология гипертекста	93
1.10.6.Технология обработки речи	94
1.10.7. Технология обработки и преобразования сигналов 94	
1.10.8.Нейрокомпьютерные технологии.....	95
1.11.Алгоритм и его свойства.....	95
1.11.1.Описание алгоритмов на естественном языке	96
1.11.2.Описание алгоритмов с помощью блок-схем	99
1.12.Функциональная и структурная организация компьютера.....	101
1.12.1.Принцип двоичного кодирования	102
1.12.2.Принцип программного управления.....	103
1.12.3.Принцип однородности памяти.....	103
1.13.Принцип адресности.....	104
1.14.Технические характеристики персонального компьютера.....	106
1.15.Сетевые технологии обработки данных.....	108
1.15.1.Компьютерные информационные сети	108
1.15.2.Локальные вычислительные сети.....	112
1.15.3.Способы коммутации и передачи данных.....	119

1.16.Устройства хранения информации	124
1.16.1.Устройства обработки информации и управляющие устройства	125
1.16.2.Устройства ввода-вывода	126
1.17.Файловые структуры	131
1.17.1.Система FAT	132
1.17.2.Система FAT32	134
1.17.3.Система NTFS	136
1.18.Интерфейсы ввода-вывода.....	138
1.18.1.Последовательные порты	139
1.18.2.Параллельные порты.....	140
1.18.3.Интерфейсы USB и FireWire	144
1.19.Операционные системы и их классификация.....	150
1.20.Операционная система Windows.....	152
1.20.1.Общая характеристика Windows	152
1.20.2.Запуск Windows	153
1.20.3.Концепция окна	153
1.20.4.Рабочий стол и главное меню Windows.....	154
1.20.5.Создание значков программ в Windows	158
1.20.6.Создание новой папки.....	160
1.20.7.Работа с объектами.....	160
1.20.8.Проводник Windows.....	163
1.20.9.Операции с дисками. Сжатие информации на диске	166
1.20.10.Стандартные приложения Windows.....	167
1.20.11.Сетевые и коммуникационные возможности Windows	172
1.20.12.Окончание работы Windows	174
1.21.Защита информации	175
1.22.Текстовый редактор документов Microsoft Word ...	179
1.23.Электронные таблицы Microsoft Excel	191
КОНТРОЛЬНЫЕ ВОПРОСЫ.....	207
2.ОБЩИЕ СВЕДЕНИЯ О ПРЕДСТАВЛЕНИИ ИНФОРМАЦИИ В ЭВМ	210

2.1. Числа конечной точности	212
2.2. Диапазоны представления чисел	215
2.3. Позиционные системы счисления	217
2.4. Однородные и неоднородные системы счисления	221
2.5. Свойства систем счисления	223
2.6. Выбор системы счисления	224
2.7. Преобразование чисел в системах счисления	227
2.8. Восьмеричная и шестнадцатеричная системы	232
2.9. Отрицательные двоичные числа	234
2.10. Двоично-кодированная десятичная система счисления (<i>D</i> -коды)	237
2.11. Формы представления чисел в ЭВМ	240
2.12. Машинный нуль и переполнение разрядной сетки	247
2.13. Точность представления чисел в ЭВМ	247
2.14. Формы представления двоичных и десятичных чисел в ЭВМ	249
КОНТРОЛЬНЫЕ ВОПРОСЫ	253
3. ДВОИЧНАЯ АРИФМЕТИКА	256
3.1. Правила двоичной арифметики	256
3.2. Арифметические операции в двоичной системе счисления	258
3.3. Выполнение операции сложения в <i>D</i> -кодах	261
3.3.1. Сложение двоичных чисел	261
3.3.2. Сложение в коде D_1 с использованием обратного кода	264
3.3.3. Сложение в коде D_4 с использованием кода ДК ..	266
3.3.4. Алгоритм сложения с избытком шесть	267
3.4. Выполнение операции умножения в двоичной системе счисления	268

3.4.1. Умножение младшими разрядами множителя со сдвигом СЧП вправо при неподвижном множимом	270
3.4.2. Умножение младшими разрядами множителя со сдвигом множимого влево при неподвижном СЧП	272
3.4.3. Умножение старшими разрядами множителя со сдвигом СЧП влево при неподвижном множимом	272
3.4.4. Умножение старшими разрядами множителя со сдвигом множимого вправо при неподвижном СЧП	273
3.5. Умножение чисел со знаком	274
3.5.1. Множимое произвольного знака	276
3.5.2. Множимое произвольного знака. Множитель отрицательный	276
3.5.3. Умножение целых чисел и правильных дробей ..	278
3.6. Методы ускорения операции умножения	280
3.6.1. Алгоритм Бута	281
3.6.2. Модифицированный алгоритм Бута.....	284
3.6.3. Алгоритм Лемана	285
3.6.4. Умножение чисел в D -кодах	287
3.6.5. Аппаратные методы ускорения умножения.....	292
3.7. Выполнение операции деления	295
3.7.1. Деление меньшего числа на большее с восстановлением остатков.....	300
3.7.2. Деление меньшего числа на большее без восстановления остатков	304
3.7.3. Деление чисел со знаком	305
3.7.4. Деление в избыточных системах счисления	306
3.7.5. Замена деления умножением на обратную величину	307
3.7.6. Ускорение операции деления.....	308
КОНТРОЛЬНЫЕ ВОПРОСЫ.....	312
ЗАКЛЮЧЕНИЕ	314
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	315

ПРЕДИСЛОВИЕ

В учебном пособии приводятся подробные пояснения, касающиеся термина и понятия «информатика», ее связи с другими фундаментальными дисциплинами и науками, показаны мировоззренческие, правовые и экономические аспекты информационных технологий, дано наиболее полное и точное определение понятия информации и способов ее измерения, дана связь между сообщениями и сигналами, приведены основные существующие информационные технологии, дано понятие и свойства алгоритма и его связи с принципом программного управления с конкретными практическими примерами, показана функциональная и структурная организация компьютера и сетевых технологий обработки данных. Большое внимание уделено понятию сигнала, его спектра, принципам модуляции и кодирования, его математическому представлению и классификации. Также в работе излагаются основные базовые теоретические вопросы, касающиеся организации ЭВМ как вычислительной машины, рассмотрены основные сведения о представлении информации в ЭВМ, показаны различные варианты ее представления, рассмотрены наиболее распространенные варианты систем счисления, применяемые в ЭВМ. Основная часть пособия посвящена организации арифметических вычислений в ЭВМ и различным способам сложения, вычитания, умножения и деления, используемым в современных компьютерах.

Пособие предназначено для студентов младших курсов, обучающихся по специальности 230101.65 и направления 230100.62, может быть полезно при изучении других дисциплин по специальности 230101: например, «Теория автоматов», «Схемотехника ЭВМ», «Организация ЭВМ, комплексов и систем», «Программирование», при выполнении курсового проектирования по дисциплине «Организация ЭВМ, комплексов и систем», а также может быть использовано студентами старших курсов, магистрантами, аспирантами и специалистами–проектировщиками вычислительной техники.

Несмотря на то, что в последнее время вышло достаточно много различных изданий по вопросам теории информации, спо-

способам и правилам ее измерения, определения количества и качества информации, информационным ресурсам и его составляющим, способам и алгоритмам выполнения арифметических операций, применяемым в ЭВМ, информация в них слишком разрознена и иногда не совсем очевидна. В предлагаемом учебном пособии собраны сведения из различных изданий, касающихся теоретических аспектов, связанных с понятием «информатика» и «информационные технологии», показаны способы и принципы выполнения арифметических операций, а также рассмотрены интегрированные прикладные программы, наиболее необходимые при работе на персональных ЭВМ. Для изучения предлагаются такие прикладные программы семейства Microsoft: операционная система Microsoft Windows, пакет Microsoft Office и наиболее популярные программы этого программного пакета Word и Excel. Приведены общие сведения о представлении информации в ЭВМ. Рассмотрены различные системы счисления, преобразование чисел, формы представления чисел в ЭВМ и др.

При подготовке пособия авторы старались отразить по возможности полный спектр вариантов организации каждой из основных арифметических операций (сложения, вычитания, умножения и деления). Для каждого из вариантов арифметических операций приведены конкретные практические примеры, особенно необходимые при изучении данной дисциплины, так как многие из алгоритмов и способов выполнения операций достаточно сложно понять из теоретических сведений.

В конце каждой главы предлагаются контрольные вопросы и упражнения.

Авторы выражают искреннюю признательность рецензентам – кандидату технических наук, доценту Жмакину А.П. и кандидату технических наук Тюпину Д.В. за детальное изучение рукописи и ценные замечания, которые позволили улучшить содержание пособия. Также авторы благодарны сотрудникам редакционно-издательского отдела Курского государственного технического университета за нелегкий труд, связанный с редактированием рукописи.

Авторы приносят извинения читателям данного учебного пособия за допущенные ошибки и неточности, и просят направ-

лять свои отзывы и пожелания на кафедру вычислительной техники КурскГТУ.

ВВЕДЕНИЕ

Дисциплина цикла общих математических и естественно-научных дисциплин ЕН.Ф.2 «Информатика» является обязательной при подготовке специалиста по направлению 230100.62 Информатика и вычислительная техника и специальности 230101.65 Вычислительные машины, комплексы, системы и сети и занимает важное место в ряду других фундаментальных наук. Эта включает в себя изучение персонального компьютера и его возможностей, и является базовой для дисциплин «Теория автоматов», «Организация, ЭВМ и систем». Обязательным является использование компьютера как помощника при оформлении текстовой документации и использовании обработки чисел в электронных таблицах; ознакомление со способами задания чисел в ЭВМ; умение выполнять арифметические операции, используемые в вычислительной технике.

Учебное пособие состоит из трех глав.

Первая глава посвящена полному и подробному разъяснению термина и понятия «информатика», «информационные технологии», функциональная и структурная организация компьютера, вопросам измерения количества и качества информации, алгоритмизации и принципам программного управления, а также прикладным аспектам использования ЭВМ. В ней рассмотрены основные вопросы по применению и использованию операционной системы Windows, юридическим и этическим нюансам использования программного обеспечения и его защиты от вирусов, а также наиболее распространенным на сегодняшний день программам пакета Microsoft Office: Microsoft Word и Microsoft Excel. Эта глава особенно полезна студентам первого и второго курсов, начинающим общение с компьютером. В ней рассмотрены подготовка, редактирование и оформление текстовой документации, графиков, диаграмм и рисунков; обработка числовых данных в электронных таблицах.

Вторая глава рассматривает общие сведения о представлении информации в ЭВМ. В ней обсуждаются основные вопросы, касающиеся систем счисления и чисел, такие как представление информации в цифровых автоматах (ЦА); позиционные системы

счисления; методы перевода чисел; форматы представления чисел с плавающей запятой.

Третья глава учебного пособия обсуждает вопросы арифметики ЭВМ. В ней рассматриваются различные варианты и способы организации операций сложения, вычитания, умножения и деления, применяемых в компьютерах.

1. ИНФОРМАЦИЯ И ИНФОРМАТИКА

1.1. Понятие информатики

Информатика – молодая научная дисциплина, изучающая вопросы, связанные с поиском, сбором, хранением, преобразованием и использованием информации в самых различных сферах человеческой деятельности. Не нужно никого убеждать в том, что современный поток информации человечество может воспринять только с помощью компьютеров, которые осуществляют автоматическую обработку информации [6].

До настоящего времени толкование термин "информатика" (в том смысле как он используется в современной научной и методической литературе) еще не является установившимся и общепринятым. Обратимся к истории вопроса, восходящей ко времени появления электронных вычислительных машин.

После второй мировой войны возникла и начала бурно развиваться кибернетика как наука об общих закономерностях в управлении и связи в различных системах: искусственных, биологических, социальных. Рождение кибернетики принято связывать с опубликованием (1948г.) американским математиком Норбертом Винером ставшей знаменитой книги «Кибернетика или управление и связь в животном и машине». В этой работе были показаны пути создания общей теории управления и заложены основы методов рассмотрения проблем управления и связи для различных систем с единой точки зрения.

Совершенствуясь одновременно с развитием электронно-вычислительных машин, кибернетика со временем превращалась в более общую науку о преобразовании информации [17] (рис. 1.1).

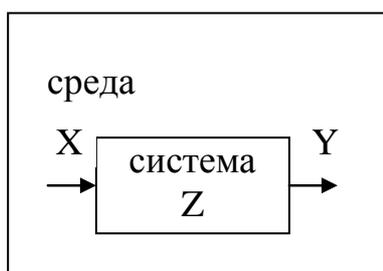


Рис. 1.1. Общая схема обмена информацией между системой и внешней средой

Под информацией (рис. 1.1) в кибернетике понимается любая совокупность сигналов, воздействий или сведений, которые

некоторая система воспринимает от окружающей среды (входная информация X), выдает в окружающую среду (выходная информация Y), а также хранит в себе (внутренняя, внутрисистемная информация Z).

В нашей стране развитие кибернетики переживало драматические периоды. Как писал академик А. И. Берг, "... в 1955-57 гг. и даже позже в нашей литературе были допущены грубые ошибки в оценке значения и возможностей кибернетики. Это нанесло серьезный ущерб развитию науки в нашей стране, привело к задержке в разработке многих теоретических положений и даже самих электронных машин". Достаточно сказать, что еще в философском словаре 1959 года издания кибернетики характеризовались как "буржуазная лженаука". Причиной этого явления послужили, с одной стороны, недооценка новой бурно развивающейся науки отдельными учеными "классического" направления, с другой - неумеренное пустословие тех, кто вместо активной разработки конкретных проблем кибернетики в различных областях спекулировал на полуфантастических прогнозах о безграничных возможностях кибернетики, дискредитируя тем самым эту науку [14].

Дело к тому же осложнялось тем, что развитие отечественной кибернетики в течение длительного периода сопровождалось серьезными трудностями в реализации крупных государственных проектов, например создания автоматизированных систем управления (АСУ), хотя работа в указанных направлениях привела к накоплению значительного опыта создания информационных систем и систем управления технико-экономическими объектами. Сложилась ситуация, в которой, с одной стороны, требовалось окончательно отмежеваться от шелухи и выделить из кибернетики научное и техническое ядро, а с другой – консолидировать силы для развития нового движения к давно уже стоящим глобальным целям.

Вскоре вслед за появлением термина «кибернетика» в мировой науке стало использоваться англоязычное "Computer Science", а чуть позже, на рубеже 60-х и 70-х годов, французы ввели получивший сейчас широкое распространение термин "Informatique". В русском языке наиболее раннее (примерно с

середины 60-х годов) употребление термина «информатика» связано с узко-конкретной областью изучения структуры и общих свойств научной информации, передаваемой посредством научной (да и не только научной) литературы. Эта информационно – аналитическая деятельность, совершенно необходимая и сегодня в библиотечном деле, книгоиздании и т.д., уже давно не отражает современного понимания информатики. Как отмечал академик А.П.Ершов, в современных условиях термин «информатика» вводится в русский язык в новом и куда более широком значении – как название фундаментальной естественной науки, изучающей процессы передачи и обработки информации. При таком толковании информатика оказывается более непосредственно связанной с философскими и общенаучными категориями, проясняется и ее место в кругу «традиционных» академических научных дисциплин.

Попытку определить, что же такое современная информатика, сделал в 1978 г. Международный конгресс по информатике: «Понятие информатики охватывает области, связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая машины, оборудование, математическое обеспечение, организационные аспекты, а также комплекс промышленного, коммерческого, административного и социального воздействия» [6,13].

Как следует из вышесказанного, информатика – не только «чистая наука». У нее, безусловно, есть научное ядро, но важной особенностью информатики является то, что она имеет широчайшие приложения, охватывающие почти все виды человеческой деятельности: производство, управление, науку, образование, проектные разработки, торговлю, финансовую сферу, медицину, криминалистику, охрану окружающей среды и др., а также быт, личную работу. Главное значение здесь имеет совершенствование социального управления на основе новых информационно технологий.

Как наука, информатика изучает общие закономерности, свойственные информационным процессам (в самом широком смысле этого понятия). Когда разрабатываются новые носители

информации, каналы связи, приемы кодирования, визуального отображения информации и многое другое, то конкретная природа этой информации почти не имеет значение. Для разработчика системы управления базами данных (СУБД) важны общие принципы организации и эффективность поиска данных, а не то, какие конкретно данные будут затем заложены в базу многочисленными пользователями. Эти общие закономерности есть предмет информатики как науки.

Объектом приложений информатики являются самые различные науки и области практической деятельности, для которых она стала непрерывным источником самых современных технологий, называемых часто «новые информационные технологии» (НИТ). Многообразные информационные технологии, функционирующие в разных видах человеческой деятельности (управлении производственным процессом, системы проектирования, финансовые операции, образование и т.п.), имея общие черты, в то же время существенно различаются между собой. Тем самым образуются и различные «предметные» информатики.

Выделим наиболее известные информационные технологии. Для обозначения ряда из них используются ставшие традиционными сокращения [15].

АСУ – автоматизированные системы управления. Комплекс технических и программных средств, которые во взаимодействии с человеком организуют управление объектами в производстве или общественной сфере. Например, в образовании используются системы АСУ-ВУЗ.

АСУТП – автоматизированные системы управления технологическими процессами. Например, такая система управляет работой станка с числовым программным управлением (СЧПУ), процессом запуска космического аппарата и т.д.

АСНИ – автоматизированная система научных исследований программно-аппаратный комплекс, в котором научные приборы сопряжены с компьютером, вводят в него данные измерений автоматически, а компьютер производит обработку этих данных и представление их в наиболее удобной для исследователя форме.

АОС – автоматизированная обучающая система. Это системы, помогающие учащимся осваивать новый материал, производящие контроль знаний, помогающие преподавателям готовить учебные материалы и т.д.

САПР – система автоматизированного проектирования – программно-аппаратный комплекс, который во взаимодействии с человеком (конструктором, инженером-проектировщиком, архитектором и т.д.) позволяет максимально эффективно проектировать механизмы, здания, узлы сложных агрегатов и др.

Не все такие технологии имеют общепринятые сокращенные названия. Это не значит, что они менее важны. Например, это диагностические системы в медицине, системы организации продажи билетов, системы ведения бухгалтерско-финансовой деятельности, системы обеспечения редакционно-издательской деятельности.

В связи с развитием информатики возникает вопрос о ее взаимосвязи и разграничении с кибернетикой. При этом, очевидно, требуется уточнение предмета кибернетики, более строгое его толкование. Информатика и кибернетика имеют много общего, основанного на концепции управления, однако они имеют и объективные различия. Один из подходов разграничения информатики и кибернетики – отнесение к области информатики исследований информационных технологий не в любых кибернетических системах (т.е. системах любой природы: биологических, технических и т.д.), а только в социальных системах. Кроме того, за кибернетикой сохраняются исследования общих законов движения информации в произвольных системах, в то время как информатика, опираясь на этот теоретический фундамент, изучает конкретные способы и приемы переработки, передачи, использования информации. Многим современным ученым такое разделение представляется искусственным, и они просто считают кибернетику одной из составных частей информатики.

Оставляя в стороне прикладные информационные технологии, опишем составные части ядра информатики. Каждая из этих частей может рассматриваться как относительно самостоятельная научная дисциплина; взаимоотношения между ними примерно такие же, как между алгеброй, геометрией и математиче-

ским анализом в классической математике – т.е. хоть это и самостоятельные дисциплины, но, несомненно, по духу и взаимопроникновению – части одной науки [6,13-15].

Теоретическая информатика – раздел информатики, пограничный с математикой. Он опирается на математическую логику и включает такие разделы как теория алгоритмов, системный анализ, теория информации и теория кодирования, исследование операций и другие. Этот раздел информатики использует математические методы для общего изучения процессов обработки информации.

Вычислительная техника – раздел, в котором разрабатываются общие принципы построения вычислительных систем. Речь идет не о технических деталях и электронных схемах (это лежит за пределами информатики как таковой), а о принципиальных решениях на уровне так называемой архитектуры вычислительных (компьютерных) систем, определяющей состав, на значение, функциональные возможности и принципы взаимодействия устройств. Примеры принципиальных, ставших классическими решений в этой области – Неймановская архитектура компьютеров первых поколений, шинная архитектура ЭВМ старших поколений, архитектура параллельной (многопроцессорной) обработки информации.

Программирование – деятельность, связанная с разработкой систем программного обеспечения. Отметим основные разделы современного программирования: создание системного программного обеспечения и создание прикладного программного обеспечения. Среди системного – разработка новых языков программирования и компиляторов к ним, разработка интерфейсных систем (пример – общеизвестная Windows). Среди прикладного программного обеспечения общего назначения самые популярные – системы обработки текстов, электронные процессоры, базы данных. В каждой области предметных приложений информатики существует множество специализированных прикладных программ более узкого назначения.

Информационные системы – решение вопросов об анализе потоков информации в различных сложных системах, их оптимизации, структурировании, принципах хранения и поиска. Ин-

формационно справочные системы, информационно – поисковые системы, гигантские современные глобальные системы хранения и поиска информации, включая широко известный Internet - все это выходит в последнее десятилетие XX века на передний план и привлекает внимание все большего круга пользователей. Без теоретического обоснования, поиска принципиальных решений в океане информации можно просто захлебнуться. Известным примером такого решения на глобальном уровне стала гипертекстовая поисковая система www; на значительно более низком уровне – справочная система, к услугам которой мы прибегаем позвонив по телефону 09.

Искусственный интеллект – область информатики, в которой решают сложнейшие проблемы, стоящие на пересечении с психологией, физиологией, лингвистикой, другими науками – как научить компьютер мыслить подобно человеку. Поскольку мы далеко не все знаем о том, как мыслит человек, эта деятельность, несмотря на полувековую историю, все еще имеет ряд принципиально нерешенных проблем. Перечислим основные направления разработок в данной сфере: моделирование рассуждений, компьютерная лингвистика, машинный перевод, создание экспертных систем, распознавание образов и другие. От успехов работ в области искусственного интеллекта зависит, в частности, решение такой важнейшей прикладной проблемы как создание интеллектуальных интерфейсных систем взаимодействия человека с компьютером, благодаря которым это взаимодействие будет походить на межчеловеческое и станет более эффективным, чем традиционное. Поэтому, по ряду признаков, кибернетика является составной частью информатики.

1.2. Место информатики в ряду других фундаментальных наук

Многообразие подходов к определению предмета и основных задач информатики как науки в настоящее время является вполне закономерным. В значительной степени оно обусловлено многообразием современных представлений об информации, которая является фундаментальным понятием современной науки, но до сих пор еще не имеет однозначного определения. Причина

этого заключается в том, что феномен информации по-разному проявляет себя в различных информационных средах, т.е. в тех конкретных условиях, в которых протекают информационные процессы, закономерности и методы реализации которых и изучает информатика как фундаментальная наука [20].

Поэтому в различных направлениях развития информатики (техническом, биологическом, социальном, физическом) анализируются лишь вполне определенные аспекты проявления феномена информации и информационных процессов, которые обусловлены тем или иным видом информационной среды.

Так, например, в технической информатике, анализируются различные методы генерации, кодирования, передачи, приема, преобразования и хранения сигналов, которые представляют собой лишь внешние отражения, т.е., некоторую форму представления информации. При этом ее внутреннее содержание, т.е. смысловое значение, а также ценность и значимость для потребителя полностью игнорируются. Такой подход к информации в технической информатике совсем не умаляет ее значения как научного направления. Наоборот, он дал возможность сформировать достаточно развитую теорию передачи и обработки сигналов, которая позволяет производить необходимые инженерные расчеты трактов передачи информации в технических системах связи и телекоммуникации, оценивать пропускную способность каналов связи, их помехоустойчивость и надежность и т.п. Одной из первых теоретических работ в этой области считается работа Клода Шеннона, которого принято считать основоположником общей теории информации. Хотя, строго говоря, научная дисциплина, которую он начал развивать, теорией информации в современном понимании, в общем-то, не является, а представляет собой лишь теорию передачи данных по каналам связи.

Гораздо ближе к современному пониманию термина «теория информации» находятся пионерские работы У. Эшби, который рассматривал информацию как меру разнообразия в природе и обществе, т.е. как степень организационной сложности технических, природных и социальных систем. Этот подход оказался исключительно плодотворным и, как будет показано далее, в последние годы находит все большее распространение при ис-

следовании проявлений феномена информации в физических, биологических и социальных системах.

Таким образом, именно фундаментальность понятия информации, ключевая роль информационных процессов в развитии живой и неживой природы, являются теми основными факторами, которые выдвигают информатику на уровень фундаментальных наук и ставят ее в один ряд с такими науками, как общая теория систем, синергетика, кибернетика, физика, химия, биология.

Междисциплинарный характер информатики.

Практически во всех современных энциклопедических словарях информатика определяется как комплексное междисциплинарное научное направление, которое является одной из наиболее перспективных «точек роста» современной науки. Она оказывает большое влияние на многие другие области научных исследований, передавая им свою научную методологию, главными достижения которой сегодня следует считать методологию информационного моделирования, а также информационный подход к анализу различных объектов, процессов и явлений в природе и обществе.

Именно поэтому, изучение информатики как фундаментальной науки в системе образования имеет исключительно большое значение для формирования современного научного мировоззрения. К сожалению, следует констатировать, что такой подход к изучению информатики, хотя и продекларирован в документах ЮНЕСКО [16], в трудах Российской академии наук, а также в проектах новых государственных образовательных стандартов России, практически еще очень медленно внедряется в систему образования. Причина здесь заключается не только в отсутствии хороших учебников по информатике для высшей и средней школы, но, главным образом, в консервативности мышления чиновников, работающих в сфере образования России, которые и сегодня все еще продолжают считать информатику вспомогательной технической дисциплиной о компьютерных технологиях и телекоммуникационных системах.

Справедливости ради, следует отметить, что не лучшим образом обстоит дело с изучением фундаментальных основ ин-

форматики и в образовательных системах зарубежных стран, где сегодня также доминирует инструментально-технологический подход к изучению ее основных проблем.

Информатика и философия

С философией информатику сегодня связывает, в первую очередь, проблема осмысления сущности феномена информации – этого удивительного по своей многогранности и распространённости проявления физической реальности окружающего нас материального мира, а также мира идеального, являющегося его отражением. Понятие информации является настолько сложным и многоаспектным, что до сих пор в науке не найдено его достаточно общего определения. И это несмотря на то, что, начиная со второй половины XX-го века, понятие информации стало общенаучной категорией и широко используется практически во всех современных научных дисциплинах. Правда, в каждой из них этому понятию придается свой собственной смысл, адекватный специфике данной научной дисциплины. Причина здесь заключается в том, что феномен информации представляет собой многоплановое явление, которое по-разному проявляет свои свойства в различных ситуациях. Именно поэтому представители различных научных дисциплин при проведении своих исследований, как правило, используют свои собственные определения понятия информации, характерные лишь для данной конкретной дисциплины. Сегодня таких определений насчитывается уже несколько десятков. Мало того, существуют философские работы [14], в которых утверждается, что дать достаточно общее определение содержания термина «информация» вообще невозможно, так как это понятие является принципиально неопределяемым. Поэтому утверждается, что нам и впредь придется довольствоваться теми частными определениями этого термина, которые уже сложились в различных областях научного знания. С этой точкой зрения трудно согласиться. Тем более, что понятие информации уже давно стало общенаучной категорией, которая сегодня широко используется в самых различных областях научного знания и практической деятельности людей (рис.1.2).

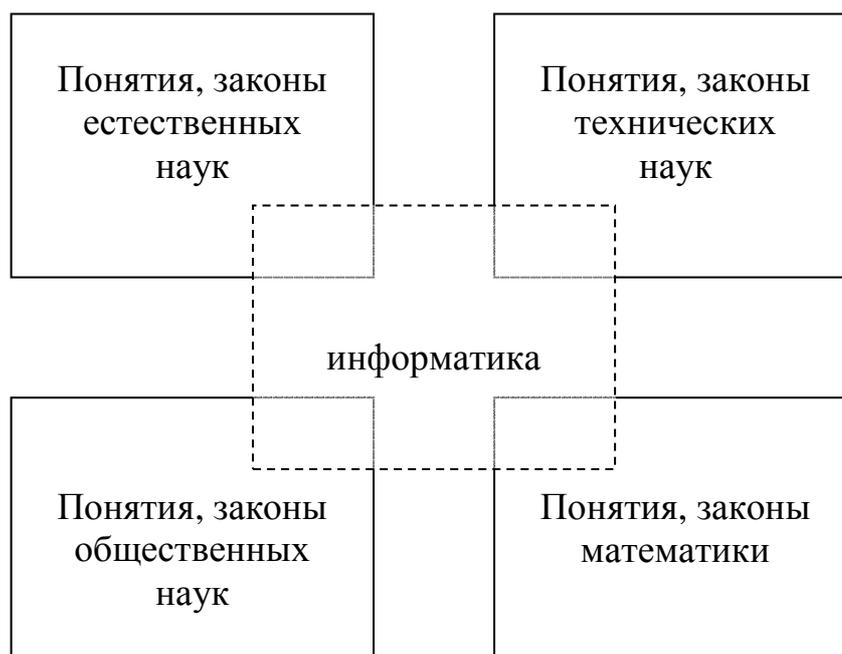


Рис. 1.2. Место информатики в системе наук

Таким образом, информатика является комплексной, междисциплинарной отраслью научного знания.

1.3. Социальные, правовые и этические аспекты информатики

Социальные аспекты

Термин «социальные аспекты» применительно к большей части наук, тем более фундаментальных, звучит странно. «Социальные аспекты математики» - вряд ли эта фраза имеет смысл. Однако, информатика - не только наука. Вспомним цитированное выше определение - в нем есть слова «... комплекс промышленного, коммерческого, административного и социального воздействий».

Информатизация общества – процесс повсеместного проникновения информационных технологий, как упомянутых выше, так и нет. Некоторые социологи и политологи полагают, что мир (точнее та его часть, которую составляют развитые страны) стоит на пороге информационного общества; к нему достаточно быстро приближается и Россия. В.А. Извозчиков предлагает следующее определение:

"Будем понимать под термином «информационное» («компьютеризированное») общество то, во все сферы жизни и деятельности членов которого включены компьютер, телематика,

другие средства информатики в качестве орудий интеллектуального труда, открывающих широкий доступ к сокровищам библиотек, позволяющих с огромной скоростью проводить вычисления и перерабатывать любую информацию, моделировать реальные и прогнозируемые события, процессы, явления, управлять производством, автоматизировать обучение и т.д.». Под "телематикой" понимаются службы обработки информации на расстоянии (кроме традиционных телефона и телеграфа).

За последние полвека информатизация была одной из причин значительного перехода людей из сферы прямого материального производства в информационную сферу. Промышленные рабочие и крестьяне, составлявшие в середине XX века более 2/3 населения, сегодня в странах указанной группы составляют менее 1/3. Все больше тех, кого называют «белые воротнички» – людей, не создающих материальные ценности непосредственно, а занятых обработкой информации (в самом широком смысле); это и учителя, и банковские служащие, и программисты, и многие другие категории работников. Появились новые, пограничные специальности. Можно ли назвать рабочим программиста, разрабатывающего программы для станков с числовым программным управлением? По ряду параметров можно, но труд в данном случае все же не физический, а интеллектуальный.

Информатизация меняет и облик традиционных отраслей промышленности и сельского хозяйства. Промышленные роботы, управляемые ЭВМ, станки с ЧПУ стали обычным оборудованием. Новейшие технологии в сельскохозяйственном производстве не только увеличивают производительность труда, но и облегчают его, вовлекают более образованных людей.

Казалось бы, что компьютеризация и информационные технологии несут в мир сплошную благодать, но социальная сфера столь сложна, что последствия любого, даже гораздо менее глобального процесса, редко бывают однозначными.

Рассмотрим, например, такое социальное последствие информатизации как быстрый рост производительности труда, его интенсификацию, гораздо более быструю чем раньше смену условий труда – в некоторых отраслях экономики по несколько раз

за время активной деятельности человека. Указанные выше факторы, с одной стороны, улучшают условия жизни многих людей, повышают степень материального и интеллектуального комфорта, стимулируют рост числа высокообразованных людей, а с другой – являются источником повышенной социальной напряженности. Например, появление на производстве промышленных роботов ведет к полному изменению технологии, которая перестает быть ориентированной на человека, и тем самым меняет номенклатуру профессий. На протяжении персонального периода трудовой деятельности значительная часть людей вынуждена менять либо специальность, либо место работы. Рост миграции населения характерен для большинства развитых стран. Государство, частные фирмы поддерживают постоянно действующую систему повышения квалификации и переподготовки, но не все люди справляются с сопутствующим стрессом без значительных потерь.

Другой достаточно опасный для демократического общества процесс также порожден прогрессом информатики. Все большее количество данных о каждом гражданине сосредоточивается в разных – государственных и негосударственных банках данных. Это и данные о профессиональной карьере (базы данных отделов кадров), здоровье (базы данных учреждений здравоохранения), имущественных возможностях (базы данных страховых компаний), перемещении по миру и т.д. – не говоря уже о тех, которые копят специальные службы. В каждом конкретном случае создание банка может быть оправдано, но в совокупности возникает система невиданной раньше ни в одном тоталитарном обществе прозрачности личности, чреватой возможным вмешательством государства или злоумышленников в частную жизнь.

Таким образом, жизнь в «информационном обществе» легче, по-видимому, не становится, а вот то, что она значительно меняется, несомненно.

Трудно, живя в самом разгаре описанных выше процессов, взвесить, чего в них больше – положительного или отрицательного, да и четких критериев для этого не существует. Тяжелая физическая работа в не слишком комфортабельных условиях, но с уверенностью, что она будет постоянным источником существ-

вованья тебя и твоей семьи, с одной стороны, либо интеллектуальный труд в комфортабельном офисе, но при высокой степени неопределенности в надежности положения [20].

Правовые аспекты

Деятельность программистов и других специалистов, работающих в сфере информатики, все чаще выступает в качестве объекта правового регулирования, и некоторые их действия при этом могут быть квалифицированы как правонарушения (преступления).

Правовое сознание в целом, а в области информатики особенно, в нашем обществе стоит невысоко. Все знают, что чаще всего нельзя копировать чужие программы – а в некоторых случаях такая возможность существует. Или как ответить на вопросы: можно ли не копируя купленную программу дать ей пользоваться другому лицу.

– студент создал в ходе выполнения дипломной работы хорошую программу – чья она юридически. Может ли он ее продавать лично или это может делать вуз;

– можно ли скопировать купленную программу для себя самого, чтобы иметь резервную копию;

– можно ли декомпилировать программу чтобы разобраться в ее деталях или исправить ошибки;

– в чем состоит разница между авторским и имущественным правом.

Вопросов, подобных этим, возникает множество. Есть, конечно, такие, ответы на которые очевидны: нельзя создавать вирусы, нельзя хулиганить в сетях, нельзя в некоммерческих телеконференциях запускать коммерческую информацию, нельзя вскрывать и искажать защищенную информацию в чужих базах данных и т.д., т.е. совершать поступки, которые могут быть объектом уголовного преследования. Но на многие вопросы ответы отнюдь не очевидны, а иногда казуистически запутаны, причем не только в нашей стране. Остановимся на правовом регулировании в области информатики в России более подробно.

Необходимо отметить, что регулирование в сфере, связанной с защитой информации, программированием и т.д., является для российского законодательства принципиально новым, еще

слабо разработанным направлением. К 1992 году был принят Закон Российской Федерации «О ПРАВОВОЙ ОХРАНЕ ПРОГРАММ ДЛЯ ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И БАЗ ДАННЫХ» [16], содержащий обширный план приведения российского законодательства в сфере информатики в соответствие с мировой практикой. Действие этого Закона распространяется на отношения, связанные с созданием и использованием программ для ЭВМ и баз данных. Также предусматривалось внести изменения и дополнения в Гражданский кодекс РФ, в Уголовный кодекс РФ, другие законодательные акты, связанные с вопросами правовой охраны программ для электронных вычислительных машин и баз данных, привести решения Правительства Российской Федерации в соответствие с Законом, обеспечить пересмотр и отмену государственными ведомствами и другими организациями Российской Федерации их нормативных актов, противоречащих указанному Закону, обеспечить принятие нормативных актов в соответствии с указанным Законом и т.д.

Главное содержание данного Закона - юридическое определение понятий, связанных с авторством и распространением компьютерных программ и баз данных, таких как Авторство, Адаптация, База данных, Воспроизведение, Декомпилирование, Использование, Модификация и т.д., а также установление прав, возникающих при создании программ и баз данных - авторских, имущественных, на передачу, защиту, регистрацию, неприкосновенность и т.д.

Авторское право распространяется на любые программы для ЭВМ и базы данных (как выпущенные, так и не выпущенные в свет), представленные в объективной форме, независимо от их материального носителя, назначения и достоинства. Авторское право распространяется на программы для ЭВМ и базы данных, являющиеся результатом творческой деятельности автора. Творческий характер деятельности автора предполагается до тех пор, пока не доказано обратное. Предоставляемая настоящим Законом правовая охрана распространяется на все виды программ для ЭВМ (в том числе на операционные системы и программные комплексы), которые могут быть выражены на любом языке и в

любой форме, и на базы данных, представляющие собой результат творческого труда по подбору и организации данных.

Предоставляемая правовая охрана не распространяется на идеи и принципы, лежащие на основе программы для ЭВМ и базы данных или какого-либо их элемента, в том числе идеи и принципы организации интерфейса и алгоритма, а также языки программирования.

Практически это, например, значит, что имея перед глазами редактор Word, Вы имеете право создать свою программу, которая будет работать точно также как Word (вопрос о смысле такой деятельности - другое дело).

Авторское право на программы для ЭВМ и базы данных возникает в силу их создания. Для признания и осуществления авторского права на программы для ЭВМ и базы данных не требуется опубликования, регистрации или соблюдения иных формальностей. Авторское право на базу данных признается при условии соблюдения авторского права на каждое из произведений, включенных в базу данных.

Автором программы для ЭВМ и базы данных признается физическое лицо, в результате творческой деятельности которого они созданы.

Если программа для ЭВМ и база данных созданы совместной творческой деятельностью двух и более физических лиц, то, независимо от того, состоит ли программа для ЭВМ или база данных из частей, каждая из которых имеет самостоятельное значение, или является неделимой, каждое из этих лиц признается автором таких программ для ЭВМ и баз данных.

Автору программы для ЭВМ или базы данных или иному правообладателю принадлежит исключительное право осуществлять и (или) разрешать осуществление следующих действий:

- выпуск программы для ЭВМ и базы данных;
- воспроизведение программы для ЭВМ и базы данных (полное или частичное) в любой форме, любыми способами;
- распространение программы для ЭВМ и баз данных;
- модификацию программы для ЭВМ и базы данных, в том числе перевод программы для ЭВМ и базы данных с одного языка на другой;

– иное использование программы для ЭВМ и базы данных.

Однако, имущественные права на программы для ЭВМ и базы данных, созданные в порядке выполнения служебных обязанностей или по заданию работодателя, принадлежат работодателю, если в договоре между ним и автором не предусмотрено иное. Таким образом, имущественно право на программу, созданную в ходе дипломного проектирования, принадлежит не автору, а вузу – по крайней мере, пока не между ними не будет заключено специальное соглашение. Имущественные права на программу для ЭВМ и базу данных могут быть переданы полностью или частично другим физическим или юридическим лицам по договору. Договор заключается в письменной форме и должен устанавливать следующие существенные условия: объем и способы использования программы для ЭВМ или базы данных, порядок выплаты и размер вознаграждения, срок действия договора.

Лицо, правомерно владеющее экземпляром программы для ЭВМ или базы данных, вправе без получения дополнительного разрешения правообладателя осуществлять любые действия, связанные с функционированием программы для ЭВМ или базы данных в соответствии с ее назначением, в том числе запись и хранение в памяти ЭВМ, а также исправление явных ошибок. Запись и хранение в памяти ЭВМ допускаются в отношении одной ЭВМ или одного пользователя в сети, если иное не предусмотрено договором с правообладателем. Также допускается без согласия правообладателя и без выплаты ему дополнительного вознаграждения осуществлять следующие действия:

- адаптацию программы для ЭВМ или базы данных;
- изготавливать или поручать изготовление копии программы для ЭВМ или базы данных при условии, что эта копия предназначена только для архивных целей и при необходимости (в случае, когда оригинал программы для ЭВМ или базы данных утерян, уничтожен или стал непригодным для использования) для замены правомерно приобретенного экземпляра.

Лицо, правомерно владеющее экземпляром программы для ЭВМ, вправе без согласия правообладателя и без выплаты дополнительного вознаграждения выполнять декомпилирование

программы для ЭВМ с тем, чтобы изучить кодирование и структуру этой программы при следующих условиях [13-15]:

– информация, необходимая для взаимодействия независимо разработанной данным лицом программы для ЭВМ с другими программами, недоступна из других источников;

– информация, полученная в результате этого декомпилирования, может использоваться лишь для организации взаимодействия независимо разработанной данным лицом программы для ЭВМ с другими программами, а не для составления новой программы для ЭВМ по своему виду существенно схожей с декомпилируемой программой для ЭВМ.

Свободная перепродажа экземпляра программы для ЭВМ и базы данных допускается без согласия правообладателя и без выплаты ему дополнительного вознаграждения после первой продажи или другой передачи права собственности на этот экземпляр.

Выпуск под своим именем чужой программы для ЭВМ или базы данных либо незаконное воспроизведение или распространение таких произведений влечет за собой уголовную ответственность.

В настоящее время уголовное законодательство Российской Федерации не в полной мере учитывает все возможные компьютерные преступления, кроме уже упоминавшихся нарушений авторского права. Вообще же, в законодательной практике многих стран отмечены различные виды компьютерных преступлений и разработаны методы борьбы с ними. Компьютерные преступления условно можно разделить на две большие категории:

1) преступления, связанные с вмешательством в работу компьютеров;

2) преступления, использующие компьютеры как необходимые технические средства.

Можно выделить следующие виды компьютерной преступности 1-го вида:

– несанкционированный доступ в компьютерные сети и системы, банки данных с целью шпионажа или диверсии (военного, промышленного, экономического), с целью так называемого компьютерного хищения или из хулиганских побуждений;

– ввод в программное обеспечение так называемых «логических бомб», срабатывающих при определенных условиях. Логические бомбы, угрожающие уничтожением данных, могут использоваться для шантажа владельцев информационных систем, либо выполнять новые, не планировавшиеся владельцем программы функции, но одновременно сохранять работоспособность системы.

Для современного состояния правового регулирования сферы, связанной с информатикой, в России в настоящее время наиболее актуальными являются вопросы, связанные с нарушением авторских прав. Большая часть программного обеспечения, используемого отдельными программистами и пользователями и целыми организациями, приобретена в результате незаконного копирования, т.е. хищения. Назрела потребность принятия шагов против этой порочной практики, поскольку она мешает прежде всего развитию самой информатики.

Задача каждого пользователя компьютеров и компьютерной информации, программиста - знать и соблюдать нормы законов в сфере информатики.

Этические аспекты

Далеко не все правила, регламентирующие деятельность в сфере информатики, можно свести в правовым нормам. Очень многое определяется соблюдением (зачастую неписанных) правил поведения тех, кто причастен к миру компьютеров. Впрочем, в этом отношении информатика ничуть не отличается от любой другой сферы жизни человека и общества.

Ограничимся здесь кратчайшими рассуждениями об этических нормах в информатике.

Как и в любой другой большой и разветвленной сфере человеческой деятельности, в информатике к настоящему времени сложились:

- определенные морально-этические нормы поведения и деятельности;
- особый жаргонный язык.

Морально-этические нормы в среде специалистов от информатики отличает от этики повседневной жизни несколько большая открытость, альтруизм, обязывающие оказать помощь в

случае, если в ней есть потребность. Большинство нынешних специалистов информатиков сформировались и приобрели свои знания и квалификацию при помощи бескорыстных консультаций и содействия других специалистов, так что они под влиянием чувства морального долга готовы оказать бескорыстную помощь, дать совет или консультацию, предоставить компьютер для выполнения каких-либо манипуляций с дискетами и т.д.

Ярким примером особой психологической атмосферы в среде информатиков является расширяющееся международное движение программистов, предоставляющих созданные ими программные средства для свободного распространения.

Это – положительные аспекты, но есть и масса отрицательных. Начнем с разговора о языке. Сленг российских информатиков построен в основном на искаженных под русское произношение англоязычных терминах и аббревиатурах, введенных иностранными фирмами – разработчиками компьютеров и программного обеспечения в технической документации. Одновременно формируется и словарный запас сленговых слов, заимствованных из русского языка на основе аналогий и ассоциаций по сходству и смежности (например: архивированный – "утопанный", компьютер – "железо" или "тачка" и т.д.).

С тем, что многие специальные термины пришли к нам из США, приходится мириться. Никто сегодня уже не перейдет от термина «принтер» к более-менее аналогичному «автоматическое печатающее устройство» (которым пользовались не так уж давно). Приживаемости подобных слов в отечественной литературе способствует, в частности, их относительная краткость. Однако трудно понять, зачем в телеконференции учителя иногда именуют себя «тичерами» – от этого они лучше не становятся. Подобных примеров можно привести массу. Итак, одно из этических правил – не искажай родной язык.

Не следует декларировать отсутствие законопослушности в сфере профессиональной деятельности (равно как и ни в какой другой). Бравата «воровал программы и буду воровать» не может быть оправдана недостаточностью финансирования. Задумайтесь: ведь подавляющее большинство тех, кто занимается не-

законным копированием программного обеспечения, никогда не позволит себе украсть компьютер.

Особую остроту этические проблемы приобретают при работе в глобальных телекоммуникационных сетях. Вскрыть защиту чужой базы данных – уголовное преступление. А позволять себе нецензурные выражения или прозрачные их эвфемизмы? Коммерческую рекламу в некоммерческой телеконференции? Независимо от того, предусмотрено за это законом возмездие или нет, порядочный человек этого делать не станет.

Подведя итог можно сделать заключение: этика – система норм нравственного поведения человека. Человек говорит себе: я не прочту содержимое дискеты, которую мой сосед забыл на рабочем месте, не потому, что это грозит мне наказанием, а потому, что это безнравственный поступок; я не скопирую программу, когда ее хозяин отвернулся, не потому, что он может подать на меня в суд, а потому, что стану неприятен сам себе, и т.д. Всякий раз, собираясь совершить сомнительный поступок в сфере профессиональной деятельности, человек должен задуматься, соответствует ли он этическим нормам, сложившимся в профессиональном сообществе.

1.4. Информация и ее измерение

Информационные процессы (сбор, обработка и передача информации) всегда играли важную роль в науке, технике и жизни общества. В ходе эволюции человечества просматривается устойчивая тенденция к автоматизации этих процессов, хотя их внутреннее содержание по существу осталось неизменным.

Информация не существует сама по себе, она проявляется в информационных процессах. Человек живет в мире информации и на протяжении всей жизни участвует во всевозможных информационных процессах. Основными информационными процессами являются: поиск, сбор, хранение, передача, обработка, использование и защита информации [6].

Действия, выполняемые с информацией, называются информационными процессами. Процессы, связанные с получением, хранением, обработкой и передачей информации, называются информационными.

Информационный процесс – совокупность последовательных действий (операций), производимых над информацией (в виде данных, сведений, фактов, идей, гипотез, теорий и пр.), для получения какого-либо результата (достижения цели) [14].

Информация проявляется именно в информационных процессах, которые всегда протекают в каких-либо системах (социальных, социотехнических, биологических и пр.). Информационные процессы, осуществляемые по определенным информационным технологиям, составляет основу информационной деятельности человека. Компьютер является универсальным устройством для автоматизированного выполнения информационных процессов.

1.4.1. Измерение информации в быту (информация как новизна)

Предположим, получено какое-то сообщение. В этом сообщении содержится некоторое количество информации. Как оценить, сколько информации вы получили или другими словами, как измерить информацию. К примеру, можно ли сказать, что чем больше статья, тем больше информации она содержит.

Разные люди, получившие одно и тоже сообщение, по-разному оценивают количество информации, содержащееся в нем. Это происходит оттого, что знания людей об этих событиях, явлениях до получения сообщения были различными. Поэтому те, кто знал об этом мало, сочтут, что получили много информации, те же, кто знал больше, чем написано в статье, скажут, что информации не получили вовсе. Количество информации в сообщении зависит от того, насколько важно сообщение для получателя. При таком подходе непонятно, по каким критериям можно ввести единицу измерения информации. С точки зрения информации как новизны можно оценить количество информации, содержащейся в научном открытии, музыкальном стиле, новой теории общественного развития.

1.4.2. Измерение информации в технике

В технике информацией считается любая хранящаяся, обрабатываемая или передаваемая последовательность знаков, символов. В технике под количеством информации понимают

количество кодируемых, передаваемых или хранимых символов [13].

Определить понятие «количество информации» сложно. В решении этой проблемы существуют два основных подхода. Исторически они возникли почти одновременно. В конце 40-х годов XX века Клод Шеннон развил вероятностный подход к измерению количества информации, работы по созданию ЭВМ привели к «объемному» подходу. В технике используют простой способ определения количества информации, названный объемным, основанный на подсчете числа символов в сообщении, не связан с его длиной и не учитывает содержания.

Пример: собака – 6 символов, dog – 3 символа.

Человеку привычно работать с символами, а компьютеру - с кодами. Каждый символ кодируется двоичным кодом, длиной в 8 знаков (восьмибитный код).

Прежде чем измерить информацию в битах, мы определяем количество символов в этом сообщении. Каждый символ не случайно кодируется 8-битным кодом. Для удобства введена более «крупная» единица информации в технике – байт, с помощью которой легче подсчитать количество информации в техническом сообщении - оно совпадает с количеством символов в нем.

В вычислительной технике: бит (binary digit) – двоичный знак двоичного алфавита {0,1}, минимальная единица измерения информации.

Байт (byte) – единица количества информации в системе СИ. Байт - восьмиразрядный двоичный код, с помощью которого можно представить один символ.

Единицы измерения информации в вычислительной технике представлены в таблице 1.1

Таблица 1.1. Единицы измерения информации

Бит	Элементарная единица информации
Байт (б)	8 бит
Килобайт (Кбайт)	2^{10} байт = 1024 байт
Мегабайт (Мбайт)	2^{10} Кбайт = 2^{20} байт
Гигабайт (Гбайт)	2^{10} Мбайт = 2^{30} байт
Терабайт (Тбайт)	1024 Гбайт = 2^{40} байт

Петабайт (Пбайт)	1024 Тбайт = 2^{50} байт
Эксабайт (Эбайт)	1024 Пбайт = 2^{60} байт

Информационный объем сообщения (информационная емкость сообщения) - количество информации в сообщении, измеренное в битах, байтах, производных единицах (Кб, Мб и т.д.) [14].

Длина сообщения зависит от числа различных символов, употребляемых для записи сообщения. Например, слово "мир" в русском алфавите записывается тремя знаками, в английском - пятью (peace), а в КОИ-8 - двадцатью четырьмя битами (111011011110100111110010).

1.4.3. Измерение информации в теории информации (информация как снятая неопределенность)

В теории информации количеством информации называют числовую характеристику сигнала, не зависящую от его формы и содержания и характеризующую неопределенность, которая исчезает после получения сообщения в виде данного сигнала - в этом случае количество информации зависит от вероятности получения сообщения о том или ином событии.

Для абсолютно достоверного события (событие обязательно произойдет, поэтому его вероятность равна 1) количество вероятности в сообщении о нем равно 0. Чем вероятнее событие, тем больше информации о нем несет сообщение.

Лишь при равновероятных ответах ответ "да" или "нет" несет 1 бит информации. Оценка количества информации основывается на законах теории информации. Сообщение имеет ценность, несет информацию, когда мы узнаем и понимаем смысл данного сообщения.

Какое количество информации содержится, к примеру, в тексте романа "Война и мир", в фресках Рафаэля или в генетическом коде человека? Ответа на эти вопросы наука не даёт и, по всей вероятности, даст не скоро.

Важнейшим результатом теории информации является вывод: в определенных, весьма широких условиях можно пренебречь качественными особенностями информации, выразить её количество числом, а также сравнить количество информации,

содержащейся в различных группах данных.

Теория информации как самостоятельная научная дисциплина была основана Клодом Шенноном в конце 40-х годов XX века. Предложенная им теория основывалась на фундаментальном понятии количественной меры неопределенности – энтропии и связанного с нею понятия количества информации.

Сигнал – это материальный носитель информации (предмет, явление, процесс) в пространстве и во времени. Любой сигнал неразрывно связан с определенной системой, которая является системой связи или системой передачи информации и состоит из следующих модулей: источник, передатчик, канал связи, приемник и адресат. Источник информации задает некоторое множество сообщений. Генерация определенного сообщения заключается в выборе его из множества всех возможных. Сообщения бывают дискретными и непрерывными. Светофор или передача сообщения с помощью азбуки Морзе – примеры дискретного сигнала.

Особым видом сигналов являются знаки, которые в отличие от сигналов естественного происхождения создаются самоорганизующимися системами и предназначаются для передачи и хранения информации. Есть знаки, входящие в четко организованную систему, и внесистемные знаки. Например: знаки дорожного движения, система цветов светофора, музыка, речь и языки, как естественные, так и искусственные. Внесистемные знаки – это или остатки некогда существовавших знаковых систем, или знаки, созданные временно, обычно в небольших коллективах людей. Например, языки жестов и поз.

В теории информации и кодировании принят энтропийный подход к измерению информации, который основан на том, что факт получения информации всегда связан с уменьшением разнообразия или неопределенности (энтропии) системы. Неопределенность может быть интерпретирована в смысле того, насколько мало известно наблюдателю о данной системе, энтропия системы снизилась, так как для наблюдателя система стала более упорядоченной.

При энтропийном подходе под информацией понимается количественная величина исчезнувшей в ходе какого-либо про-

цесса (испытания, измерения и т.д.) неопределенности. При этом в качестве меры неопределенности вводится энтропия [1].

Энтропия – мера внутренней неупорядоченности информационной системы.

Энтропия увеличивается при хаотическом распределении информационных ресурсов и уменьшается при их упорядочении.

На основе понятий энтропии и количества информации в теории информации введены важные характеристики сигналов и информационных систем:

- скорость создания информации;
- скорость передачи информации;
- избыточность;
- пропускная способность каналов связи.

Одним из самых замечательных результатов теории информации является доказательство, что при любых помехах и шумах можно обеспечить передачу информации без потерь.

Первая теорема Шеннона гласит, что при скорости создания информации меньшей пропускной способности канала можно передавать информацию со сколь угодно малой вероятностью ошибок, несмотря на шумы.

Шеннон сформулировал энтропию как меру хаоса в противовес количеству информации как меры упорядоченности структур.

Рассмотрим некоторую сложную систему и проследим ее эволюцию. Пусть эта система представляет собой находящийся в сосуде газ, состоящий из огромного числа беспорядочно движущихся молекул. Мы не знаем точного положения и скорости в каждый момент времени каждой частицы газа, но нам известны макропараметры: давление, объем, температура и состав газа. Фактически мы должны рассчитать число способов, которыми можно осуществить внутренние перестройки в системе, чтобы наблюдатель не заметил изменений макросостояния системы. При этом предполагается неотличимость атомов друг от друга.

Если в системе, состоящей из одного атома, произошло его энергетическое возбуждение, нам это может стать известно по значению температуры. При этом возможно только одно распределение возбуждения в системе равному единице. Энтропия свя-

зана с распределением следующим образом: $S = k \ln W$. В нашем случае $W = 1$, а значит, система обладает нулевой энтропией.

В системе из ста атомов, распределение возбуждения может быть осуществлено ста способами, т.е. $W = 100$, $\ln 100 = 4,61$. Энтропия системы выросла и стала хаотичной, поскольку мы не знаем, где находится в каждый момент возбужденный атом.

Принято считать, что любая система стремится к состоянию равновесия, т.е. растет энтропия системы. Однако второе начало термодинамики (закон сохранения энтропии и информации) требует компенсировать рост энтропии. Информация и является средством компенсации.

В настоящее время получили распространение подходы к определению понятия «количество информации», основанные на том, что информацию, содержащуюся в сообщении, можно не строго трактовать в смысле её новизны или, иначе, уменьшения неопределённости наших знаний об объекте.

Р. Хартли предложил в качестве меры неопределенности логарифм от числа возможностей, т.е. процесс получения информации рассматривает как выбор одного сообщения из конечного наперёд заданного множества из N равновероятных сообщений, а количество информации I , содержащееся в выбранном сообщении, определяет как двоичный логарифм N : $I = \log_2 N$ – формула Хартли.

Обычно количество информации представляется в виде: $\log_2 m$, где m – число возможных выборов. Тогда стандартной единицей количества информации будет выбор из двух возможностей. Такая единица получила наименование бит и представляется одним символом двоичного алфавита: 0 или 1.

Пример: нужно угадать одно число из набора чисел от единицы до ста. По формуле Хартли можно вычислить, какое количество информации для этого требуется: $I = \log_2 100 \approx 6,644$. Т. е. сообщение о верно угаданном числе содержит количество информации, приблизительно равное 6,644 единиц информации.

Другие примеры равновероятных сообщений: при бросании монеты: «выпала решка», «выпал орел»; на странице книги: «количество букв чётное», «количество букв нечётное».

В некоторых случаях, когда однозначно нельзя ответить на

вопросы распределения вероятности, для определения количества информации уже нельзя использовать формулу Хартли.

Пример: являются ли равновероятными сообщения "первой выйдет из дверей здания женщина" и "первым выйдет из дверей здания мужчина". Однозначно ответить на этот вопрос нельзя. Все зависит от того, о каком именно здании идет речь.

Для задач такого рода американский учёный Клод Шеннон предложил в 1948 г. другую формулу определения количества информации, учитывающую возможную неодинаковую вероятность сообщений в наборе [18].

Формула Шеннона:

$$I = -(p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_N \log_2 p_N), \quad (1.1)$$

где p_i - вероятность того, что именно i -е сообщение выделено в наборе из N сообщений.

Если вероятности p_1, \dots, p_N равны, то каждая из них равна $1/N$, и формула Шеннона превращается в формулу Хартли.

1.5. Преобразование сообщения в сигналы

1.5.1. Кодирование сообщений

Процесс передачи информации заключается в том, что сообщения преобразуются в сигналы и по системе связи передаются получателю. Получатель, зная закон соответствия между сообщениями и сигналами, может извлечь содержащуюся в сообщении информацию. Для верного декодирования каждому сигналу должно соответствовать одно определенное сообщение.

Преобразование сообщений в сигналы осуществляется с помощью кодирования и модуляции. Кодирование представляет собой отображение дискретных сообщений последовательностью символов позиционной системы счисления [17,18].

Последовательность символов, сопоставляемая одному элементарному сообщению (букве, знаку и т.д.) называется кодовой комбинацией. Систему правил преобразования элементарных сообщений в кодовые комбинации называют кодом. Основание используемой системы счисления называют основанием кода. Как правило, первичные коды задаются в виде таблиц.

При выборе основания системы счисления учитывают простоту, удобство и экономичность реализации цифрового представления информации в системе, ее преобразований и передачи по каналам связи. Наибольшее применение в технике передачи дискретной информации нашли коды с основанием 2, которые называются двоичными или бинарными. Символы двоичных кодов единица (1) и нуль (0) называются единичными элементами. Количество единичных элементов, образующих кодовую комбинацию, называется длиной кодовой комбинации.

Кодирование сообщений производится специальным устройством, которое называется кодером (кодирующим устройством) источника сообщения (датчика информации). В кодере кодовые комбинации представляются в виде определенных состояний накопительных элементов (триггеров, ферритов, механических рычагов, линеек и т.д.). Для передачи сообщения состояния накопительных элементов преобразуются в последовательность элементов дискретного электрического сигнала, как правило, в импульсы тока или напряжения. Каждый символ кодовой комбинации представляется единичным элементом цифрового сигнала. Процесс преобразования элементов кодовой комбинации в последовательность элементов сигнала называется модуляцией.

При представлении единичных элементов кодовых комбинаций токовыми и бестоковыми (или положительными и отрицательными) посылками, токовые (положительные) обозначаются "1", бестоковые (отрицательные) – "0".

В кодирующем устройстве производится первичное кодирование и первичная модуляция. Термин «первичное» подчеркивает то обстоятельство, что в процессе передачи по каналу связи сигналы, как правило, подвергаются дополнительному кодированию и модуляции.

Коды можно разделить на две большие группы: простые и корректирующие. Корректирующие коды (называют также помехоустойчивые) применяют для повышения верности информации. Простые коды (называют также: первичные, обыкновенные, безизбыточные) используются для первичного преобразования дискретных сообщений в сигналы и получают на выходе

кодера источника сообщения. Простые коды делят на равномерные и неравномерные.

Равномерными называются такие коды, в которых все кодовые комбинации имеют одинаковую длину, т.е. имеют одинаковое число единичных элементов.

Неравномерными называют такие коды, кодовые комбинации которых могут отличаться одна от другой числом единичных элементов.

Оценка простых кодов производится по скорости передачи, помехоустойчивости и сложности технической реализации.

1.5.2. Равномерные простые коды

Как следует из определения, простые равномерные коды состоят из комбинаций одинаковой длины. Естественно, возникает вопрос хорошо это или плохо. Для ответа на него рассмотрим пример.

Пусть имеется некоторое сообщение, состоящее из M элементов, представляющее собой некоторую последовательность m ($m \ll M$) знаков (например, книга имеет $M=100000$ элементов, представляющая собой некоторую последовательность из 32 букв, 10 цифр и 11 знаков препинания, т.е. из $m = 53$ знаков). Как известно, это сообщение несет некоторое количество информации I , равное:

$$I = \log_2 N,$$

где N – число возможных вариантов последовательностей из M элементов.

Поскольку последовательность из M элементов составлена знаками, каждый из которых (x_i) появляется в последовательности с различными вероятностями p_i , то, используя формулу Стерлинга, можно показать, что количество информации в этой последовательности будет [6]:

$$I = \log_2 N = M \sum_{i=1}^m p_i \log_2 \frac{1}{p_i}. \quad (1.2)$$

На один элемент сообщения будет приходиться в среднем количество информации:

$$H_l = \frac{I}{M} = \sum_{i=1}^m p_i \log_2 \frac{1}{p_i}. \quad (1.3)$$

Если каждый знак сообщения кодируется n -элементной кодовой комбинацией, состоящей из двоичных символов, то каждый из них будет содержать H_j количества информации:

$$H_j = \frac{H_l}{n} = \frac{1}{n} \sum_{i=1}^m p_i \log_2 \frac{1}{p_i}. \quad (1.4)$$

Очевидно, что код следует считать наилучшим с точки зрения скорости передачи тогда, когда H_j будет максимально возможным.

Из теории информации известно, что один двоичный элемент может содержать максимальное количество информации равно 1-му биту, т.е. всегда $H_j \leq 1$.

Следовательно, величина

$$R = 1 - H_j \quad (1.5)$$

может служить мерой, информационной недогрузки каждого двоичного элемента.

Если число знаков, из которых состоит сообщение, $m=2n$ и все знаки равновероятны $p_i=1/m$, то величина $R=0$ Действительно.

$$R = 1 - \frac{1}{n} \sum_{i=1}^m \frac{1}{m} \log_2 m = 1 - \frac{1}{n} \cdot \frac{m}{m} \log_2 2^n = 0. \quad (1.6)$$

Таким образом, максимальная скорость передачи равномерного простого кода будет тогда и только тогда, когда выполняются условия $p_i = \frac{1}{m}$ и $m = 2^n$, где n – целое число.

На практике, как правило, знаки сообщения неравновероятны, а также не выполняется условие $m = 2^n$, Поэтому равномерные коды имеют $R \neq 0$, т.е. скорость их практически всегда ниже максимально возможной. Однако, тот факт, что каждая кодовая комбинация в равномерных кодах имеет одинаковое количество двоичных элементов, позволяет получать простые правила кодирования и декодирования и, соответственно, простую техническую реализация кодирующих и декодирующих устройств.

Кроме того, за счет простых способов определения на приемной стороне начала и конца каждой кодовой комбинации, что является необходимым условием однозначного декодирования, помехоустойчивость равномерных кодов достаточно высокая. Важным фактором является также то, что простые равномерные коды легко преобразуются в корректирующие коды для повышения достоверности информации. Все это привело к тому, что равномерные коды получили широкое применение на практике.

Для расширения возможностей равномерных кодов используют следующие меры. Например, число русских букв, цифр и знаков препинания составляет 53, что требует применения 6-элементных кодовых комбинаций ($6 > \log_2 53$). Поэтому все множество знаков разбивается на два множества (регистра): буквенный и цифровой, что позволяет использовать 5-элементные кодовые комбинации. Для правильного декодирования вводятся специальные комбинации, указывающие о переходе с одного регистра на другой.

Современные отечественные телеграфные аппараты имеют три регистра: русский, латинский и цифровой.

Увеличение алфавита может быть достигнуто за счет того, что кодируются не только отдельные буквы (цифры), а и целые слова и даже отдельные фразы. Естественно – это вызывает необходимость увеличения числа регистров при использовании того же 5-элементного равномерного кода.

1.5.3. Неравномерные коды

Как отмечалось выше, неравномерными кодами называют такие коды, которые содержат разное число элементов [6].

Эти коды, как и равномерные коды, с точки зрения скорости передачи информации могут оцениваться величиной информационной нагрузки каждого двоичного символа:

$$R = 1 - \frac{1}{n} \sum_{i=1}^m p_i \log_2 \frac{1}{p_i}, \quad (1.7)$$

где $\bar{n} = \sum_{i=1}^m n_i + p_i$ – средняя длина кодовой комбинации; n_i

– длина комбинации, соответствующей i -му символу сообщения; p_i – вероятность появления i -го символа в сообщении.

Если более вероятным символам сообщения сопоставить более короткие кодовые комбинации и наоборот, то средняя длина кодовой комбинации будет меньше, т.е. скорость передачи информации таким кодом будет выше. Такие коды называют оптимальными.

При построении неравномерных кодов необходимо учитывать требование однозначного декодирования сообщения, первым этапом которого является правильное определение начала и конца каждой кодовой комбинации. Этого можно достичь, если между комбинациями ставить специальные разделительные группы или использовать неприводимые коды. Неприводимость кодов заключается в том, что в них из более длинной комбинации нельзя составить более короткие комбинации. В настоящее время разработан целый ряд неприводимых кодов.

Примером неприводимого кода может служить код, состоящий из следующих комбинация: 11, 10, 011, 001, 000, 00001, 000001.

Неприводимость этого кода заключается в том, что короткие кодовые комбинация не могут быть началом более длинных кодовых комбинаций и, следовательно, любая двоичная последовательность однозначно разбивается на указанные кодовые комбинации.

Например, последовательность

- 1011100000110000101111101000111 -

однозначно разбивается на комбинации

10, 11, 10, 00001, 10, 0001, 011, 11, 10, 10, 001, 11.

Необходимо отметить, что применение оптимальных неравномерных кодов не всегда будет обеспечивать большую скорость передачи по сравнению с равномерным кодом. Предпосылкой получения более высокой скорости передачи путем применения неравномерного кода может служить заметная неравновероятность символов сообщения и хорошая согласованность

выбранного неравномерного кода со статистической структурой сообщения.

Например, пусть имеется сообщение, состоящее из достаточно большого числа элементов. Каждый элемент представляет собой один из восьми различных символов. Вероятность появления i -го символа определяется из выражения

$$p_i = \frac{(9-i)^2}{204}. \quad (1.8)$$

Определить:

1) во сколько раз изменится скорость передачи информации при передаче указанного сообщения вышерассмотренным неприводимым кодом по сравнению с равномерным кодом;

2) может ли существовать другой код, с помощью которого можно получить более высокую скорость передачи информации при передаче данного сообщения.

Сопоставим комбинации неприводимого кода символам сообщения в соответствии с табл.1.2.

Таблица 1.2 Комбинации неприводимого кода

Комбинация	11	10	011	010	001	0001	00001	000001
i	1	2	3	4	5	6	7	8
$p_i \cdot 204$	64	49	36	25	16	9	4	1
n_i	2	2	3	3	3	4	5	6

Определим среднюю длину кодовой комбинации неприводимого кода:

$$\bar{n} = \sum_{i=1}^8 p_i n_i = \frac{1}{204} \left(1 \cdot 6 + 4 \cdot 5 + 9 \cdot 4 + 16 \cdot 3 + 25 \cdot 3 + 38 \cdot 3 + 49 \cdot 2 + 64 \cdot 2 \right) = 2.544 \text{ эл.}$$

При использовании равномерного кода число элементов в кодовой комбинации $n \gg \log_2 8 = 3$.

Следовательно, при применении данного неприводимого кода скорость увеличивается по сравнению с применением равномерного кода.

Далее определим количество информации, которую несет один элемент кодовой комбинации неприводимого кода.

$$H_9 = \frac{1}{m} \sum_{i=1}^8 p_i \log_2 \frac{1}{p_i} = 0.97 \text{ бит.} \quad (1.9)$$

Поскольку один двоичный элемент кодовой комбинации может нести 1 бит информации, то возможно, существует другой код или другая процедура кодирования, обеспечивающие большую скорость передачи.

В последнее время широкое применение получил неравномерный неприводимый код, построенный по следующему правилу.

Элементы сообщения записываются в виде последовательности натуральных десятичных чисел, записанных в двоичной форме, начиная с числа 2 (т.е.10):

10, 11, 100, 101, 110, 111, 10000,...

Полученные комбинации преобразуются в новые комбинации путем добавления в них нулей перед каждым нечетным элементом, начиная с третьего элемента.

Комбинации будут иметь вид

10, 11, 1000, 1001, 1100, 1101, 10000 и т.д.

Полученный таким образом код обладает следующими свойствами:

- каждая комбинация начинается единицей, а на нечетных позициях стоят нули;
- каждая комбинация имеет четное число элементов.

Эти свойства делают код неприводимым и обеспечивают простоту декодирования.

1.5.4. Первичные коды

Вопросами координации стандартных кодов в международном масштабе занимается Международный консультативный комитет по телеграфии и телефонии (МККТТ) и Международная организация стандартов (МОС) [15].

Для телеграфной связи в 1932 году был принят стандартный Международный телеграфный код МТК–2. Этот код представляет собой равномерный пятиэлементный код с двумя регистрами «буквы латинские» и «цифры». Для возможности пере-

дачи текстов на русском языке МТК–2 был дополнен третьим регистром «русские буквы». Этот код на регистрах «буквы латинские» и «цифры» совпадает со стандартом международной связи.

Код МТК-2 не полностью удовлетворяет ряду требований телеграфной связи. Ограниченность числа комбинаций пятизначного кода не позволяет передавать весьма важные служебные знаки, например: «Понял», «Ждите», «Квитанция», «Конец адреса» и др.

Поэтому ряд функциональных символов передают с помощью четырехкратного повторения комбинаций стандартного кода. Например, для обозначения начала текста передают В-Q'Cflf, конца телеграммы – +++++, конца сообщения – /VNNM и т.д. Однако многобуквенные сочетания можно использовать для передачи только служебных знаков или переключении аппаратуры, но их нельзя использовать для передачи графических печатных символов.

С появлением нового вида электросвязи – передачи данных возникла потребность введения новых служебных, арифметических, логических и других символов, отсутствующих в МТК–2. Это привело к необходимости создания нового стандартного кода, пригодного как для телеграфии, так и для передачи данных. Был разработан и утвержден в 1966 году МККТТ и МОС новый международный стандартный код МТК–5, который содержит не только строчные, но и прописные буквы, дополнительные знаки препинания, большое число символов управления устройствами связи и печати.

Код обеспечивает:

- обработку и передачу машинной символики в пределах машинного языка "КОБОЛ";
- простоту выделения при декодировании групп символов различного характера: служебных символов, цифр, букв и специальных знаков;
- простой алгоритм работы ЭВМ при обработке цифровой информации;
- упрощение процессов упорядочения информации по цифрам и буквам;

- передачу информации по каналам связи и телеуправления устройствами обработки данных;
- построение клавиатуры аппаратов с расположением клавиш, близким к расположению на клавиатуре пишущей машинки.

Международный код МТК–5 является равномерным семи-элементным однорегистровым кодом и содержит только буквы латинского языка. Поэтому для нашей страны был разработан стандартный код (ГОСТ 13052-67), который отличается от кода №5 тем, что он является двухрегистровым: первый регистр соответствует латинскому алфавиту, а второй – русскому.

1.6. Сигналы, спектры сигналов, модуляция и кодирование

1.6.1. Понятие сигнала

В XVIII веке в теорию математики вошло понятие функции, как определенной зависимости какой-либо величины y от другой величины – независимой переменной x , с математической записью такой зависимости в виде $y(x)$. Довольно скоро математика функций стала базовой основой теории всех естественных и технических наук. Особое значение функциональная математика приобрела в технике связи, где временные функции вида $s(t)$, $v(f)$ и т.п., используемые для передачи информации, стали называть сигналами [23-26].

В технических отраслях знаний термин "сигнал" (signal, от латинского signum – знак) очень часто используется в широком смысловом диапазоне, без соблюдения строгой терминологии. Под ним понимают и техническое средство для передачи, обращения и использования информации – электрический, магнитный, оптический сигнал; и физический процесс, представляющий собой материальное воплощение информационного сообщения – изменение какого-либо параметра носителя информации (напряжения, частоты, мощности электромагнитных колебаний, интенсивности светового потока и т.п.) во времени, в пространстве или в зависимости от изменения значений каких-либо других аргументов (независимых переменных); и смысловое содержание определенного физического состояния или процесса,

как, например, сигналы светофора, звуковые предупреждающие сигналы и т.п. Все эти понятия объединяет конечное назначение сигналов. Это определенные сведения, сообщения, информация о каких-либо процессах, состояниях или физических величинах объектов материального мира, выраженные в форме, удобной для передачи, обработки, хранения и использования этих сведений.

Термин “сигнал” очень часто отождествляют с понятиями “данные” (data) и “информация” (information). Действительно, эти понятия взаимосвязаны и не существуют одно без другого, но относятся к разным категориям.

Понятие информации имеет много определений, от наиболее широкого (информация есть формализованное отражение реального мира) до практического (сведения и данные, являющиеся объектом хранения, передачи, преобразования, восприятия и управления). В настоящее время мировая наука все больше склоняется к точке зрения, что информация, наряду с материей и энергией, принадлежит к фундаментальным философским категориям естествознания и относится к одному из свойств объективного мира, хотя и несколько специфичному. Что касается “данных” (от латинского datum – факт), то это совокупность фактов, результатов наблюдений, измерения каких-либо физических свойств объектов, явлений или процессах материального мира, представленных в формализованном виде, количественном или качественном. Это не информация, а только атрибут информации – сырье для получения информации путем соответствующей обработки и интерпретации (истолкования).

Наука и техника интернациональны, и используют, в основном, общепринятые термины, большинство из которых англоязычны. Термин "signal" в мировой практике является общепринятым для характеристики формы представления данных, при которой данные рассматриваются как результат некоторых измерений объекта исследований в виде последовательности значений скалярных величин (аналоговых, числовых, графических и пр.) в зависимости от изменения каких-либо переменных значений (времени, энергии, температуры, пространственных координат, и пр.). С учетом этого, в дальнейшем под термином

“сигнал” в узком смысле этого слова будем понимать каким-либо образом упорядоченное *отображение* изменения физического состояния какого-либо объекта – материального носителя сигнала. На это формализованное отображение переносятся данные о характере изменения в пространстве, во времени или по любой другой переменной определенных физических величин, физических свойств или физического состояния объекта исследований. А так как данные содержат информацию, как об основных целевых параметрах объекта исследований, так и о различных сопутствующих и мешающих факторах измерений, то в широком смысле этого слова можно считать, что сигнал является носителем общей измерительной информации. При этом материальная форма носителей сигналов (механическая, электрическая, магнитная, акустическая, оптическая и любая другая), равно как и форма отображения данных в каких-либо физических параметрах или процессах носителей, значения не имеет. Информативным параметром сигнала может являться любой параметр носителя сигнала, функционально и однозначно связанный со значениями информационных данных.

Наиболее распространенное представление сигналов – в электрической форме в виде зависимости напряжения от времени $U(t)$. Так, например, сигнал изменения напряженности магнитного поля по профилю аэросъемки – это и временная последовательность изменения электрического напряжения на выходе датчика аэромагнитометра, и запись этого напряжения на ленте регистратора, и последовательные значения цифровых отсчетов при обработке лент регистратора и вводе сигнала в ЭВМ.

С математической точки зрения сигнал представляет собой функцию, т.е. зависимость одной величины от другой, независимой переменной (рис. 1.3).



Рис. 1.3. Пример сигнала

По содержанию это информационная функция, несущая сообщение о физических свойствах, состоянии или поведении какой-либо физической системы, объекта или среды, а целью обработки сигналов можно считать извлечение определенных информационных сведений, которые отображены в этих сигналах (кратко - полезная или целевая информация) и преобразование этих сведений в форму, удобную для восприятия и дальнейшего использования.

Под "анализом" сигналов (analysis) имеется в виду не только их чисто математические преобразования, но и получение на основе этих преобразований выводов о специфических особенностях соответствующих процессов и объектов. Целями анализа сигналов обычно являются:

- Определение или оценка числовых параметров сигналов (энергия, средняя мощность, среднее квадратическое значение и пр.).

- Разложение сигналов на элементарные составляющие для сравнения свойств различных сигналов.

- Сравнение степени близости, "похожести", "родственности" различных сигналов, в том числе с определенными количественными оценками.

Математический аппарат анализа сигналов весьма обширен, и широко применяется на практике во всех без исключения областях науки и техники.

С понятием сигнала неразрывно связан термин *регистрации* сигналов, использование которого также широко и неоднозначно, как и самого термина сигнал. В наиболее общем смысле под этим термином можно понимать операцию выделения сигнала и его преобразования в форму, удобную для дальнейшего использования. Так, при получении информации о физических свойствах каких-либо объектов, под регистрацией сигнала понимают процесс измерения физических свойств объекта и перенос результатов измерения на материальный носитель сигнала или непосредственное энергетическое преобразование каких-либо свойств объекта в информационные параметры материального носителя сигнала (как правило – электрического). Но так же широко термин регистрации сигналов используют и для процессов выделения уже сформированных сигналов, несущих определенную информацию, из суммы других сигналов (радиосвязь, телеметрия и пр.), и для процессов фиксирования сигналов на носителях долговременной памяти, и для многих других процессов, связанных с обработкой сигналов.

Применительно к настоящему курсу под термином регистрации будем понимать *регистрацию данных* (data logging) которые проходят через конкретную систему или точку системы и определенным образом фиксируются на каком-либо материальном носителе или в памяти системы. Что касается процесса получения информации при помощи технических средств, обеспечивающих опытным путем нахождение соотношения измеряемой величины с принятой по определению образцовой единицей этой величины, и представление измеренного соотношения в какой-либо физической или числовой форме информационного сигнала, то для этого процесса будем применять, в основном, термин *детектирования*.

1.6.2. Шумы и помехи

При детектировании сигналов, несущих целевую для данного вида измерений информацию, в сумме с основным сигналом одновременно регистрируются и мешающие сигналы – шумы и помехи самой различной природы (рис. 1.4).

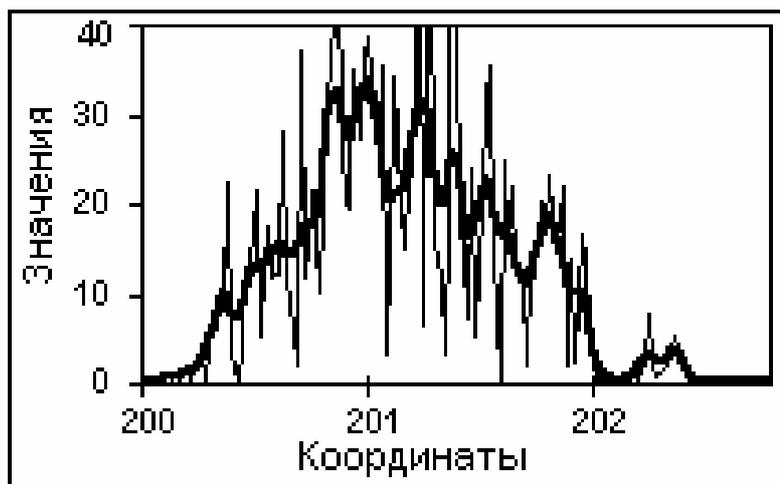


Рис. 1.4. Вид сигнала с помехами

К помехам относят также искажения полезных сигналов при влиянии различных дестабилизирующих факторов на процессы измерений, как, например, влияние грозových разрядов на электроразведочные методы измерений и т.п. Выделение полезных составляющих из общей суммы зарегистрированных сигналов или максимальное подавление шумов и помех в информационном сигнале при сохранении его полезных составляющих является одной из основных задач первичной обработки сигналов (результатов наблюдений).

Типы помех разделяют по источникам их возникновения, по энергетическому спектру, по характеру воздействия на сигнал, по вероятностным характеристикам и другим признакам.

Источники помех бывают внутренние и внешние.

Внутренние шумы могут быть присущи физической природе источников сигналов, как, например, тепловые шумы электронных потоков в электрических цепях или дробовые эффекты в электронных приборах, или возникают в измерительных устройствах и системах передачи и обработки сигналов от влияния различных дестабилизирующих факторов – температуры, повышенной влажности, нестабильности источников питания, влияния механических вибраций на гальванические соединения, и т.п.

Внешние источники шумов бывают искусственного и естественного происхождения. К искусственным источникам помех относятся промышленные помехи – двигатели, переключатели, генераторы сигналов различной формы и т.д. Естественными ис-

точниками помех являются молнии, флюктуации магнитных полей, всплески солнечной энергии, и т.д.

Электрические и магнитные поля различных источников помех вследствие наличия индуктивных, емкостных и резистивных связей создают на различных участках и цепях сигнальных систем паразитные разности потенциалов и токи, накладывающиеся на полезные сигналы.

Помехи подразделяются на флюктуационные, импульсные и периодические. Флюктуационные или шумовые помехи представляют хаотический и беспорядочный во времени процесс в виде нерегулярных случайных всплесков различной амплитуды. Как правило, флюктуационные помехи распределены по нормальному закону с нулевым средним и оказывают существенное влияние только на сигналы низкого уровня.

Импульсные помехи во многом похожи на шумовые помехи и проявляются как в виде отдельных импульсов, так и в виде последовательности импульсов, форма и параметры которых имеют случайный характер. Причинами импульсных помех являются резкие броски тока и напряжения в промышленных установках, транспортных средствах, а также природные электрические явления. Распределение импульсных помех, как правило, симметричное с произвольной плотностью распределения.

Периодические помехи вызываются периодическими низкочастотными или высокочастотными полями линий электропередач, силовых электроустановок и др. Если основная мощность помех сосредоточена на отдельных участках диапазона частот, например, на частоте напряжения промышленной сети или кратна этой частоте, то такие помехи называют сосредоточенными.

В зависимости от характера воздействия на сигнал помехи разделяют на аддитивные и мультипликативные. Аддитивные (налагающиеся) помехи суммируются с сигналом, не зависят от его значений и формы и не изменяют информативной составляющей самого сигнала. Мультипликативные или деформирующие помехи могут изменять форму информационной части сигнала, иметь зависимость от его значений и от определенных особенностей в сигнале и т.п. При известном характере мультипликативных помех возможна коррекция сигнала на их влияние.

Следует заметить, что деление сигналов на полезные и мешающие (шумовые) является достаточно условным. Источниками мешающих сигналов также могут быть определенные физические процессы, явления или объекты. При выяснении природы мешающих сигналов они могут переводиться в разряд информационных. Так, например, вариации диаметра скважин является мешающим фактором практически для всех ядерно-физических методов каротажа. Вместе с тем этот же фактор, при соответствующем методическом и аппаратном обеспечении, может дать возможность бесконтактного определения диаметра скважин в качестве дополнительного информационного параметра.

1.6.3. Размерность сигналов

Простейшими сигналами геофизической практики являются одномерные сигналы, как, например, сейсмические импульсы $s(t)$, измерения каких-либо параметров геофизических полей (электрических, магнитных, и пр.) по профилям на поверхности земли $s(x)$ или по стволу скважины $s(h)$, и т.п. Значения одномерных сигналов зависят только от одной независимой переменной, как, например, на рис. 1.3-1.5.

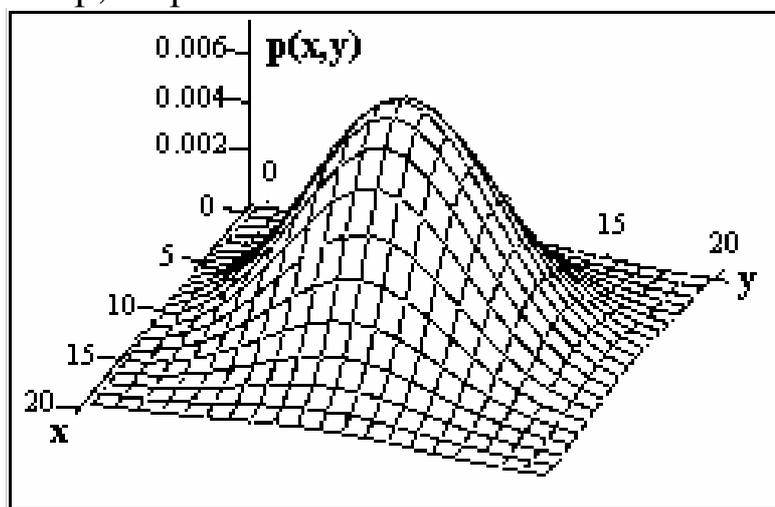


Рис. 1.5. Пример двумерного сигнала

В общем случае сигналы являются многомерными функциями пространственных, временных и прочих независимых переменных – сейсмическая волна вдоль линии профиля $s(x,t)$, аномалия гравитационного поля на поверхности наблюдений $s(x,y)$, пространственно - энергетическое распределение потока ионизирующих частиц или квантов от источника излучения

$s(x,y,z,E)$ и т.п. Все большее применение находят также многомерные сигналы, образованные некоторым множеством одномерных сигналов, как, например, комплексные каротажные измерения нескольких физических параметров горных пород по стволу скважины одновременно.

Многомерные сигналы могут иметь различное представление по своим аргументам. Так, полный акустический сигнал сейсмического профиля дискретен по пространству (точкам расположения приемников) и непрерывен по времени.

Многомерный сигнал может рассматриваться, как упорядоченная совокупность одномерных сигналов. С учетом этого при анализе и обработке сигналов многие принципы и практические методы обработки одномерных сигналов, математический аппарат которых развит достаточно глубоко, распространяются и на многомерные сигналы. Физическая природа сигналов для математического аппарата их обработки значения не имеет.

Вместе с тем обработка многомерных сигналов имеет свои особенности, и может существенно отличаться от одномерных сигналов в силу большего числа степеней свободы. Так, при дискретизации многомерных сигналов имеет значение не только частотный спектр сигналов, но и форма растра дискретизации. Пример не очень полезной особенности – многомерные полиномы сигнальных функций, в отличие от одномерных, не разлагаются на простые множители. Что касается порядка размерности многомерных сигналов, то ее увеличение выше двух практически не изменяет принципы и методы анализа данных, и сказывается, в основном, только на степени громоздкости формул и чисто техническом усложнении вычислений.

Учитывая эти факторы, при рассмотрении общей теории анализа, преобразований и обработки сигналов ограничимся, в основном, одно- и двумерными сигнальными функциями, а в качестве универсальных независимых переменных (аргументов функций) будем использовать, как правило, переменную " t " для одномерных сигналов и переменные " x,t " или " x,y " для двумерных сигналов, безотносительно к их физическому содержанию (пространство, время, энергия и пр.).

1.6.4. Спектральное и математическое представление сигналов

Кроме привычного динамического представления сигналов и функций в виде зависимости их значений от определенных аргументов (времени, линейной или пространственной координаты и т.п.) при анализе и обработке данных широко используется математическое описание сигналов по аргументам, обратным аргументам динамического представления. Так, например, для времени обратным аргументом является частота. Возможность такого описания определяется тем, что любой сколь угодно сложный по своей форме сигнал, не имеющий разрывов второго рода (бесконечных значений на интервале своего задания), можно представить в виде суммы более простых сигналов, и, в частности, в виде суммы простейших гармонических колебаний, что выполняется при помощи преобразования Фурье. Соответственно, математически разложение сигнала на гармонические составляющие описывается функциями значений амплитуд и начальных фаз колебаний по непрерывному или дискретному аргументу – частоте изменения функций на определенных интервалах аргументов их динамического представления. Совокупность амплитуд гармонических колебаний разложения называют *амплитудным спектром* сигнала, а совокупность начальных фаз – *фазовым спектром*. Оба спектра вместе образуют полный частотный спектр сигнала, который по точности математического представления тождественен динамической форме описания сигнала.

Линейные системы преобразования сигналов описываются дифференциальными уравнениями, причем для них верен принцип суперпозиции, согласно которому реакция систем на сложный сигнал, состоящий из суммы простых сигналов, равна сумме реакций от каждого составляющего сигнала в отдельности. Это позволяет при известной реакции системы на гармоническое колебание с определенной частотой определить реакцию системы на любой сложный сигнал, разложив его в ряд гармоник по частотному спектру сигнала. Широкое использование гармонических функций при анализе сигналов объясняется тем, что они являются достаточно простыми ортогональными функциями и

определены при всех значениях непрерывных переменных. Кроме того, они являются собственными функциями времени, сохраняющими свою форму при прохождении колебаний через любые линейные системы и системы обработки данных с постоянными параметрами (изменяются только амплитуда и фаза колебаний). Немаловажное значение имеет и то обстоятельство, что для гармонических функций и их комплексного анализа разработан мощный математический аппарат.

Примеры частотного представления сигналов приводятся ниже (рис. 1.1.5 – 1.1.12).

Кроме гармонического ряда Фурье применяются и другие виды разложения сигналов: по функциям Уолша, Бесселя, Хаара, полиномам Чебышева, Лаггера, Лежандра и др. Главное условие однозначности и математической идентичности отображения сигналов - ортогональность функций разложения. Но при качественном анализе сигналов могут применяться и неортогональные функции, выявляющие какие-либо характерные особенности сигналов, полезные для интерпретации физических данных.

Теория анализа и обработки физических данных базируется на математических моделях соответствующих физических полей и физических процессов, на основе которых создаются математические модели сигналов. Математические модели сигналов дают возможность обобщенно, абстрагируясь от физической природы, судить о свойствах сигналов, предсказывать изменения сигналов в изменяющихся условиях, заменять физическое моделирование процессов математическим. С помощью математических моделей имеется возможность описывать свойства сигналов, которые являются главными, определяющими в изучаемых процессах, и игнорировать большое число второстепенных признаков. Знание математических моделей сигналов дает возможность классифицировать их по различным признакам, характерным для того или иного типа моделей. Так, сигналы разделяются на неслучайные и случайные в зависимости от возможности точного предсказания их значений в любые моменты времени. Сигнал является неслучайным и называется *детерминированным*, если математическая модель позволяет осуществлять такое предсказание. Детерминированный сигнал задается, как правило,

математической функцией или вычислительным алгоритмом, а математическая модель сигнала может быть представлена в виде

$$s = F(t, z, \omega, \dots; A, B, C, \dots),$$

где s – информативный параметр сигнала; t, z, ω, \dots – независимые аргументы (время, пространственная координата, частота и др.); A, B, C, \dots – параметры сигналов.

Модель должна быть, по возможности, проще, минимизирована по количеству независимых аргументов и адекватна изучаемому процессу, что во многом предопределяет результаты измерений. Рассмотрим этот вопрос на примере геофизических данных.

Под геофизическим полем понимают собственное или индуцированное определенным внешним воздействием распределение какой-либо физической величины, создаваемое геологическим объектом или геологической структурой в пространстве, во времени или по любому другому аргументу (независимой переменной). В простейшем случае геофизический сигнал – это изменение какой-либо составляющей геофизического поля, т.е. сечение поля по одному из аргументов. В пределе все геофизическое поле в целом может рассматриваться как первичный многомерный сигнал в прямом физическом отображении, с которого путем измерений могут сниматься формализованные копии определенных составляющих (сечений) сигнала на материальные носители информации.

Геофизическим полям в определенных условиях их регистрации соответствуют определенные математические модели сигналов, т.е. их описание на каком-либо формальном языке. Математическое описание не может быть всеобъемлющим и идеально точным и, по существу, всегда отображает не реальные объекты, а их упрощенные (гомоморфные) модели. Модели могут задаваться таблицами, графиками, функциональными зависимостями, уравнениями состояний и переходов из одного состояния в другое и т.п. Формализованное описание может считаться математической моделью оригинала, если оно позволяет с определенной точностью прогнозировать состояние и поведение изучаемых объектов путем формальных процедур над их описанием.

Неотъемлемой частью любой математической модели сигнала является область определения сигнала, которая устанавливается интервалом задания независимой переменной. Примеры задания интервала для переменных:

$$a \leq x \leq b, \quad x \in [a, b].$$

$$a < y \leq b, \quad y \in (a, b].$$

$$a < z < b, \quad z \in (a, b).$$

Пространство значений независимой переменной обычно обозначается через индекс R . Так, например, $R := (-\infty, +\infty)$, $x \in R$.

Кроме задания области определения сигнала могут быть также заданы виды численных значений переменных (целые, рациональные, вещественные, комплексные).

1.6.5. Классификация сигналов

Классификация сигналов осуществляется на основании существенных признаков соответствующих математических моделей сигналов. Все сигналы разделяют на две крупных группы: детерминированные и случайные. Классификация сигналов внутри групп приведена на рис. 1.6.

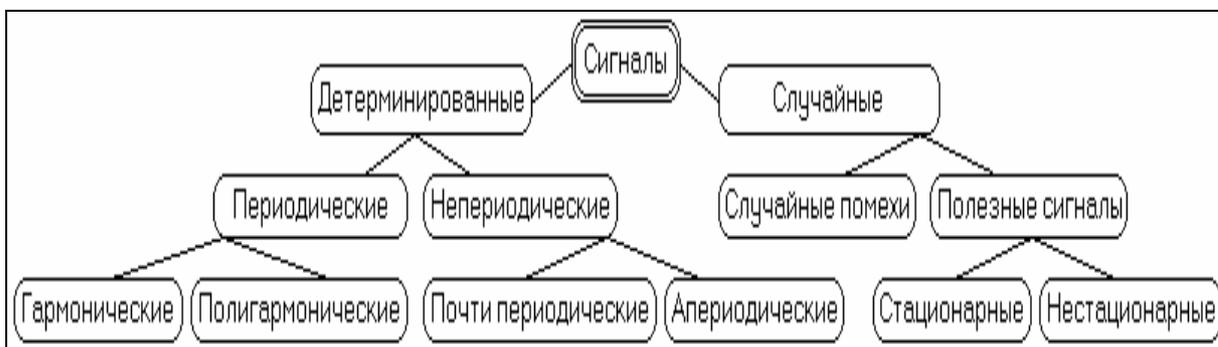


Рис. 1.6. Классификация сигналов

С математических позиций группы сигналов обычно называют множествами, в которые объединяют сигналы по какому-либо общему свойству. Принадлежность сигнала s к множеству L_P записывается в виде $L_P = \{s; P\}$, где P – определенное свойство данного множества сигналов.

1.6.6. Типы сигналов

Выделяют следующие основные типы сигналов, которым соответствуют определенные формы их математического описания: аналоговые, дискретные и цифровые.

Аналоговый сигнал (analog signal) является непрерывной функцией непрерывного аргумента, т.е. определен для любого значения аргументов. Источниками аналоговых сигналов, как правило, являются физические процессы и явления, непрерывные в динамике своего развития во времени, в пространстве или по любой другой независимой переменной, при этом регистрируемый сигнал подобен (“аналогичен”) порождающему его процессу. Пример математической записи сигнала: $y(t)=4.8\exp[-(t-4)^2/2.8]$. Пример графического отображения данного сигнала приведен на рис. 1.7, при этом как сама функция, так и ее аргументы, могут принимать любые значения в пределах некоторых интервалов $y_1 \leq y \leq y_2, t_1 \leq t \leq t_2$.

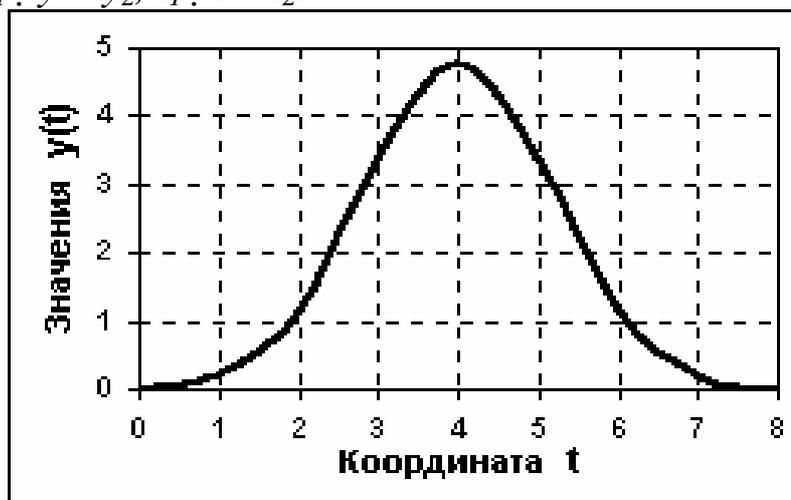


Рис. 1.7. Аналоговый сигнал

Если интервалы значений сигнала или его независимых переменных не ограничиваются, то по умолчанию они принимаются равными от $-\infty$ до $+\infty$. Множество возможных значений сигнала образует континуум — непрерывное пространство, в котором любая сигнальная точка может быть определена с точностью до бесконечности. Примеры сигналов, аналоговых по своей природе — изменение напряженности электрического, магнитного, электромагнитного поля во времени и в пространстве.

Дискретный сигнал (discrete signal) по своим значениям также является непрерывной функцией, но определенной только

по дискретным значениям аргумента. По множеству своих значений он является конечным (счетным) и описывается дискретной последовательностью отсчетов (samples) $y(n\Delta t)$, где $y_1 \leq y \leq y_2$, Δt – интервал между отсчетами (интервал или шаг дискретизации, sample time), $n=0, 1, 2, \dots, N$. Величина, обратная шагу дискретизации: $f=1/\Delta t$, называется частотой дискретизации (sampling frequency). Если дискретный сигнал получен дискретизацией (sampling) аналогового сигнала, то он представляет собой последовательность отсчетов, значения которых в точности равны значениям исходного сигнала по координатам $n\Delta t$.

Пример дискретизации аналогового сигнала, приведенного на рис. 1.7, представлен на рис. 1.8.

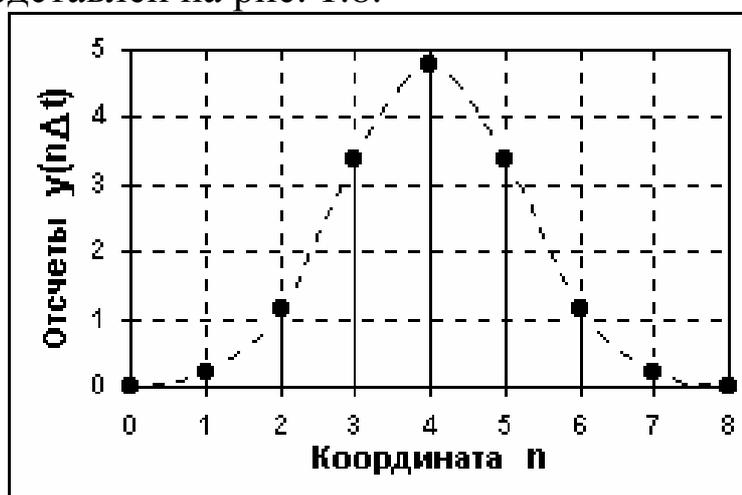


Рис. 1.8. Дискретный сигнал

При $\Delta t = const$ (равномерная дискретизация данных) дискретный сигнал можно описывать сокращенным обозначением $y(n)$. В технической литературе в обозначениях дискретизированных функций иногда оставляют прежние индексы аргументов аналоговых функций, заключая последние в квадратные скобки – $y[t]$. При неравномерной дискретизации сигнала обозначения дискретных последовательностей (в текстовых описаниях) обычно заключаются в фигурные скобки – $\{s(t_i)\}$, а значения отсчетов приводятся в виде таблиц с указанием значений координат t_i . Для числовых последовательностей (равномерных и неравномерных) применяется и следующее числовое описание: $s(t_i) = \{a_1, a_2, \dots, a_N\}$, $t = t_1, t_2, \dots, t_N$. Примеры дискретных геофизических сигналов – результаты вертикального электрического зон-

дирования (дискретная величина разноса токовых электродов), профили геохимического опробования, и т.п.

Цифровой сигнал (digital signal) квантован по своим значениям и дискретен по аргументу. Он описывается квантованной решетчатой функцией $y_n = Q_k[y(n\Delta t)]$, где Q_k – функция квантования с числом уровней квантования k , при этом интервалы квантования могут быть как с равномерным распределением, так и с неравномерным, например – логарифмическим. Задается цифровой сигнал, как правило, в виде дискретного ряда (discrete series) числовых данных – числового массива по последовательным значениям аргумента при $\Delta t = const$, но в общем случае сигнал может задаваться и в виде таблицы для произвольных значений аргумента.

По существу, цифровой сигнал по своим значениям (отсчетам) является формализованной разновидностью дискретного сигнала при округлении отсчетов последнего до определенного количества цифр, как это показано на рис 1.9.

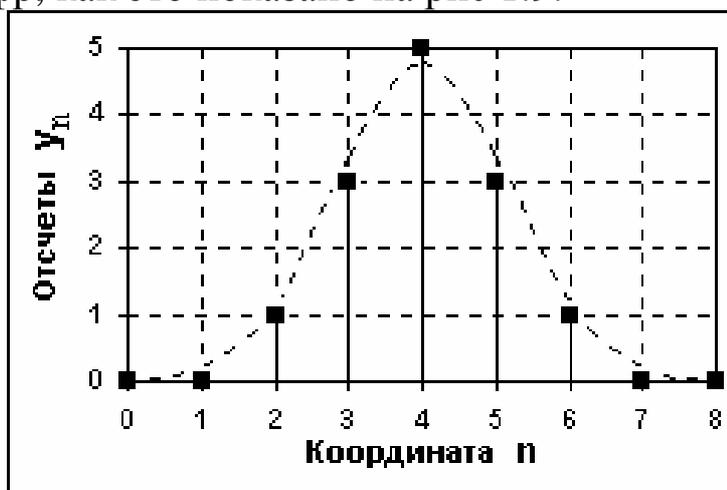


Рис. 1.9. Цифровой сигнал

Цифровой сигнал конечен по множеству своих значений. Процесс преобразования бесконечных по значениям аналоговых отсчетов в конечное число цифровых значений называется квантованием по уровню, а возникающие при квантовании ошибки округления отсчетов (отбрасываемые значения) – шумами (noise) или ошибками (error) квантования (quantization).

В системах цифровой обработки данных и в ЭВМ сигнал всегда представлен с точностью до определенного количества разрядов, а, следовательно, всегда является цифровым. С учетом

этих факторов при описании цифровых сигналов функция квантования обычно опускается (подразумевается равномерной по умолчанию), а для описания сигналов используются правила описания дискретных сигналов. Что касается формы обращения цифровых сигналов в системах хранения, передачи и обработки, то, как правило, они представляет собой комбинации коротких одно- или двуполярных импульсов одинаковой амплитуды, которыми в двоичном коде с определенным количеством числовых разрядов кодируются числовые последовательности сигналов (массивов данных).

1.7. Помехоустойчивое кодирование и фильтрация

Как правило способ передачи информации (кодирование и модуляция) задан и задача сводится к поиску оптимальной помехоустойчивости, которую обеспечивают различные способы приема.

Под помехоустойчивостью системы связи подразумевается способность системы восстанавливать сигналы с заданной достоверностью. Предельно допустимая помехоустойчивость называется потенциальной. Сравнение потенциальной и реальной помехоустойчивости позволяет дать оценку качества приема данного устройства и найти еще не использованные ресурсы.

Сведения о потенциальной помехоустойчивости приемника при различных способах передачи позволяют сравнить эти способы между собой и найти наиболее совершенные.

Следует помнить, что оптимальный приемник, является корреляционным, то есть сигнал на его выходе представляет собой функцию корреляции принимаемого и ожидаемого сигналов, благодаря чему обеспечивается максимально-возможное отношение сигнал/шум.

Задача оптимальной фильтрации непрерывного сигнала ставится так, чтобы обработав принятый сигнал, получить на выходе приемника сигнал, наименее отличающийся от переданного сигнала. Решение этой задачи основывается на трех основных предположениях:

- Сигнал $S(t)$ и помеха $w(t)$ представляют собой стационарные случайные процессы;
- Операция фильтрации предполагается линейной;

– Критерием оптимальности считается минимум средне-квадратичной ошибки.

Как правило, фильтры ставятся перед решающим устройством, задача которого – вынести решение в пользу того или иного сигнала. Нужно отметить важное обстоятельство, что при приеме дискретных сигналов нет необходимости заботиться о сохранении формы сигнала. Основная задача – обеспечить минимум ошибочных решений при приеме сигналов. Поэтому при синтезе фильтров для дискретных сигналов используется критерий максимума отношения сигнал/шум на выходе фильтра. Фильтры, удовлетворяющие данному критерию могут называться оптимальными фильтрами, или фильтрами, максимизирующими отношение сигнал/шум.

Непрерывный сигнал можно передавать мгновенными значениями этого сигнала (отсчетами), следующими с определенной частотой повторения. Последняя должна быть больше не менее, чем в 2 раза передаваемой частоты входного сигнала. Такое представление сигнала во времени называется дискретизацией. Этот принцип лежит в основе импульсно-кодовой модуляции (ИКМ).

Информация о мгновенном значении входного непрерывного сигнала может быть передана в сторону приемника непосредственно в форме отсчетов – амплитудно-модулированных импульсов, взятых в определенные временные моменты, причем длительность импульсов, как правило очень мала по сравнению с периодом их повторения. В интервалах между двумя соседними отсчетами одного сигнала последовательно во времени можно разместить отсчеты других передаваемых сигналов, а на приемной стороне эти отсчеты распределить между каналами.

В основе амплитудно-импульсной модуляции (АИМ) лежит передача сигналов в виде импульсов, промодулированных по амплитуде. Под влиянием помех, возникающих в тракте передачи, происходят случайные изменения формы и амплитуды передаваемых импульсов, что при восстановлении исходного непрерывного сигнала проявляется в виде дополнительного шума. Физически уменьшение этого шума возможно лишь за счет сниже-

ния уровня помех в тракте передачи, что практически приводит к уменьшению дальности связи.

Изменение амплитуды можно передавать в виде изменения длительности импульсов. Амплитуда широтно-модулированных импульсов (ШИМ) постоянно, при этом удается снизить влияние внешних помех при передаче импульсов, что дает возможность значительно увеличить дальность связи.

Передача информации путем изменения положения импульса постоянной амплитуды и длительности лежит в основе время-импульсной модуляции (ВИМ).

Описанные виды импульсной модуляции (АИМ, ШИМ, ВИМ) соотносятся как обычные (АМ, ЧМ, ФМ) и являются аналоговыми методами импульсной модуляции, общим недостатком которых являются жесткие требования к параметрам линии связи, т.к. помехи, которые накладываются на передаваемый модулированный импульс, изменяют его форму, что в приемнике отражается как дополнительный шум. Этот шум значительно увеличивается при передаче информации на большие расстояния, т.к. искажения импульсов отдельных участков складываются. Технические ограничения, накладываемые на приведенные выше способы импульсной модуляции вели к дальнейшему поиску способов, при которых для передачи информации можно было полностью перейти к чисто цифровой форме сигнала, передаваемого по тракту передачи. Результатом этого поиска явилась импульсно-кодовая модуляция (ИКМ).

При таком типе модуляции входной непрерывный сигнал $x=f(t)$ дискретизируется в соответствии с теоремой отсчетов, а амплитуда АИМ импульсов, отображающая мгновенное значение входного сигнала в момент дискретизации, преобразуется кодером в двоичные числа. Так как число символов n в двоичном числе, отражающем амплитуду импульса, ограничено, то ограничено и число цифр, позволяющих обозначить амплитуду соответствующего импульса. Поэтому кодер не может в большинстве случаев точно закодировать амплитуду импульсов, а производит “округление” до ближайшей нормированной амплитуды, которая может быть передана двоичным числом с ограниченным количеством разрядов. Отсюда следует, что кодер должен последова-

тельно переводить непрерывно изменяющиеся амплитуды АИМ импульсов в квантованные по уровню АИМ импульсы и кодировать, т.е. выразить их через дискретно-квантованные по уровню величины в двоичном коде. Группа двоичных символов, которая используется для передачи одной дискретно-квантованной амплитуды, называется кодовой группой (кодовое слово).

Дискретизация – первый шаг при преобразовании аналогового сигнала в цифровую форму. На входе декодера она появляется в виде АИМ импульсов, поступающих на выход через фильтр нижних частот.

Форма амплитудно-модулированных импульсов может быть различной и зависит от схемы дискретизатора и способов кодирования и декодирования. При передаче необходимо получать как можно более узкие импульсы отсчетов, чтобы в интервалах между ними разместить отсчеты сигналов остальных каналов система, а при приеме, наоборот, как можно более широкие импульсы отсчетов, так как мощность низкочастотного сигнала на входе приемника зависит от энергии импульсов отсчетов, восстановленных на выходе декодера. Сигнал на выходе АИМ ключа – самая простая форма дискретизированного сигнала, у которого вершины импульсов повторяют форму исходного непрерывного сигнала.

Передача аналоговых сигналов цифровыми методами сопровождается шумом квантования, возникающим из-за деления динамического диапазона кодека на конечное число дискретных величин (ступеней квантования).

Сравнение аналоговых импульсных видов модуляции (АИМ, ШИМ, ВИМ) с ИКМ позволяет сделать следующие выводы:

– информация о мгновенных параметрах входного непрерывного сигнала при аналоговых импульсных видах модуляции передается при непрерывном изменении аналоговых величин (амплитуды, длительности, временного положения) импульса. Длительность действия систем передачи с этими видами модуляции, как правило, ограничена искажениями, возникающими в процессе передачи, главной причиной которых является чувствительность передаваемого сигнала к внешним помехам;

– информация о мгновенных параметрах непрерывного сигнала в системах с ИКМ передается в виде двоичных чисел (кодовых групп), представленных последовательностью импульсов одинаковой формы и амплитуды. Так как искажения этих импульсов при условии безошибочной регенерации не влияют на качество передачи и их сравнительно легко регенерировать, то практически можно достичь независимости качества передачи входного непрерывного сигнала от дальности связи. Необходимо помнить, что при ограничении числа уровней квантования входного непрерывного сигнала появляется дополнительный шум. Кроме того, цифровые системы передачи по сравнению с аналоговыми занимают более широкую полосу частот, что объясняется заменой аналогового сигнала группой импульсов.

1.7.1. Статистическое (эффективное) кодирование

Для рационального использования пропускной способности канала необходимо применять соответствующие способы кодирования сообщений. Статическим или оптимальным называется кодирование, при котором пропускная способность канала связи без помех используется наилучшим образом. При оптимальном кодировании фактическая скорость передачи сообщений по каналу R приближается к пропускной способности C , что достигается путем согласования источника с каналом. Сообщения источника кодируются таким образом, чтобы они в наибольшей степени соответствовали ограничениям, которые накладываются на сигналы, передаваемые по каналу связи. Поэтому структура оптимального кода зависит как от статистических характеристик источника, так и от особенностей канала.

Кодирование с исправлением ошибок (помехоустойчивое кодирование), по существу, представляет собой метод обработки сигналов, предназначенный для увеличения надежности передачи по цифровым каналам. Хотя различные схемы кодирования очень непохожи друг на друга и основаны на различных математических теориях, всем им присущи два общих свойства. Одно из них – избыточность. Закодированные цифровые сообщения всегда содержат дополнительные, или избыточные символы. Эти символы используются для того, чтобы подчеркнуть индивидуаль-

ность каждого сообщения. Из приведенной выше информации можно сделать вывод, что помехоустойчивое кодирование, проигрывает по скорости передачи с оптимальным кодированием из-за избыточности кода, с другой стороны оптимальное кодирование применимо лишь в каналах, в которых влияние помех незначительно.

Всякая система связи строится для передачи сообщений от источников к потребителю. При этом каждое сообщение имеет свое содержание и определенную ценность для потребителя. Однако для канала связи существенным является лишь тот факт, что в передаваемом сообщении содержится какое-то количество информации.

Для того, чтобы иметь возможность сравнивать различные каналы связи, необходимо иметь некоторую количественную меру, позволяющую оценить содержащуюся в передаваемом сообщении информацию. Такая мера в виде количества передаваемой информации была введена К.Шенноном.

1.7.2. Помехоустойчивое кодирование

При передаче цифровых данных по каналу с шумом всегда существует вероятность того, что принятые данные будут содержать некоторый уровень частоты появления ошибок. Получатель, как правило, устанавливает некоторый уровень частоты появления ошибок, при превышении которого принятые данные использовать нельзя. Если частота ошибок в принимаемых данных превышает допустимый уровень, то можно использовать кодирование с исправлением ошибок., которое позволяет уменьшить частоту ошибок до приемлемой.

Кодирование с обнаружением и исправлением ошибок как правило связано с понятием избыточности кода, что приводит в конечном итоге к снижению скорости передачи информационного потока по тракту связи. Избыточность заключается в том, что цифровые сообщения содержат дополнительные символы, обеспечивающие индивидуальность каждого кодового слова. Вторым свойством, связанным с помехоустойчивым кодированием, является усреднение шума. Этот эффект заключается в том, что из-

быточные символы зависят от нескольких информационных символов.

При увеличении длины кодового блока (т.е. количества избыточных символов) доля ошибочных символов в блоке стремится к средней частоте ошибок в канале. Обработывая символы блоками, а не одного за другим можно добиться снижения общей частоты ошибок и при фиксированной вероятности ошибки блока долю ошибок, которые нужно исправлять.

Все известные в настоящее время коды могут быть разделены на две большие группы: блочные и непрерывные. Блочные коды характеризуются тем, что последовательность передаваемых символов разделена на блоки. Операции кодирования и декодирования в каждом блоке производится отдельно. Непрерывные коды характеризуются тем, что первичная последовательность символов, несущих информацию, непрерывно преобразуется по определенному закону в другую последовательность, содержащую избыточное число символов. При этом процессы кодирования и декодирования не требует деления кодовых символов на блоки.

Разновидностями как блочных, так и непрерывных кодов являются делимые (с возможностью выделения информационных и контрольных символов) и неделимые коды. Наиболее многочисленным классом делимых кодов составляют линейные коды. Их особенность состоит в том, что контрольные символы образуются как линейные комбинации информационных символов.

1.7.3. Принцип обнаружения и исправления ошибок

Корректирующие коды строятся так, чтобы количество комбинаций M превышало число сообщений M_0 источника. Однако в этом случае используется лишь M_0 комбинаций источника из общего числа для передачи информации. Такие комбинации называются разрешенными, а остальные – запрещенными $M - M_0$. Приемнику известны все разрешенные и запрещенные комбинации, поэтому, если при приеме некоторого разрешенного сообщения в результате ошибки это сообщение попадает в разряд запрещенных, то такая ошибка будет обнаружена, а при опреде-

ленных условиях исправлена. Следует заметить, что при ошибке, приводящей к появлению другого разрешенного сигнала, такая ошибка не обнаружима.

Расстоянием Хемминга d между двумя последовательностями называется число позиций, в которых две последовательности отличаются друг от друга. Наименьшее значение d для всех пар кодовых последовательностей называется кодовым расстоянием.

Ошибка обнаруживается всегда, если её кратность, т.е. число искаженных символов в кодовой комбинации: $g < d-1$. Если $g > d$, то некоторые ошибки также обнаруживаются. Однако полной гарантии обнаружения ошибок нет, т.к. ошибочная комбинация может совпадать с какой-либо разрешенной комбинацией. Минимальное кодовое расстояние, при котором обнаруживаются любые одиночные ошибки, $d=2$.

Исправление ошибок в процессе декодирования сводится к определению переданной комбинации по известной принятой. Расстояние между переданной разрешенной комбинацией и принятой запрещенной комбинацией d_0 равно кратности ошибок g .

1.8. Фазы информационного цикла и их модели

Совокупность больших сложных человеко-машинных информационных систем является важнейшей составляющей инфраструктуры общества, где информация выступает одним из главных ресурсов его жизнедеятельности. Поэтому является чрезвычайно важным понимание сложного технологического процесса сбора, передачи и обработки информации. Это три важнейших фазы информационного цикла [13-15].

1.8.1. Сбор информации

Система сбора информации представляет собой сложный программно-аппаратный комплекс. Как правило, современные системы сбора информации не только обеспечивают кодирование информации и ее ввод в ЭВМ, но и выполняют предварительную обработку этой информации. Сбор информации – это процесс получения информации из внешнего мира и приведение ее к виду, стандартному для данной информационной системы. Обмен информацией между воспринимающей информацией сис-

темой и окружающей средой осуществляется посредством сигналов. Сигнал – это средство перенесения информации в пространстве и времени. Переносчиками сигнала могут являться звук, свет, электрический ток, магнитное поле и др. Технические системы, также как и живые организмы, воспринимают информацию из внешней среды с помощью специальных приспособлений. Если у живых организмов это органы чувств (обоняния, осязания, слуха, зрения), то у технических систем это множество специальных устройств (клавиатура, мышь, сканер, различного рода световоспринимающие устройства). Вне зависимости от носителя информации процесс обработки сигнала может быть охарактеризован несколькими шагами. На первом шаге сигнал с помощью специального устройства преобразуется в эквивалентный ему электрический сигнал. На втором шаге вторичный сигнал оцифровывается специальным устройством – аналого-цифровым преобразователем (АЦП). АЦП ставит в соответствии со значением электрического сигнала некоторое число. Таким образом, устройство восприятия и АЦП, представляют собой некий цифровой измерительный прибор. Если этот прибор оснастить некоторым устройством для хранения измеренной величины – регистром, то на следующем шаге по команде от ЭВМ можно ввести это число в машину и подвергать затем любой необходимой обработке. Конечно, не все технические средства сбора информации работают по такой схеме. Так, клавиатура не имеет АЦП. Первичный сигнал непосредственно преобразуется в соответствующий нажатой клавише цифровой код. Но в любом случае поступающая в ЭВМ информация представлена в виде цифрового кода – двоичного числа. Современные системы сбора информации могут включать в себя большое количество всевозможных устройств ввода информации (от человека в ЭВМ, от ЭВМ к ЭВМ и т.п.). Такое разнообразие средств приводит к необходимости управления процессом сбора информации и к разработке соответствующего программного обеспечения. Современная развитая система сбора информации представляет собой совокупность технических средств, и программ, обеспечивающих ввод информации с отдельных устройств. Так как информация очень разнообразна по содержанию и виду обслуживаемой

ею человеческой деятельности (научная, управленческая, производственная, экономическая и др.), то каждый вид информации имеет свои особенные технологии обработки, смысловую ценность, формы представления и отображения на физическом носителе, требования к точности, достоверности, оперативности отражения фактов, явлений, процессов. Обработка информации с применением комплекса технических средств вызывает необходимость манипулировать с отдельными информационными элементами, обеспечивать их изучение и описание, идентификацию для удобства обработки, хранения и передачи. Информация, представленная в формализованном виде, получила название данные. Информация, являясь сложным по структуре образованием, размещается на физических носителях (бумажных или магнитных документах, в виде сигналов передаваемых по каналам связи) и может находиться в статичном или динамичном состояниях. Информационная технология решения задач включает следующие важнейшие процедуры, которые могут быть сгруппированы по функционально-временным стадиям: сбор и регистрация информации, передача ее к месту обработки, машинное кодирование данных, хранение и поиск, вычислительная обработка, тиражирование информации, использование информации, т.е. принятие решений и выработка управляющих воздействий. Обычно информация подвергается всем процедурам преобразования, но в ряде случаев некоторые процедуры могут отсутствовать. Последовательность их выполнения также бывает различной, но при этом некоторые процедуры могут повторяться. Состав процедур преобразования и особенности их выполнения во многом зависят от самого объекта, ведущего автоматизированную обработку информации. Рассмотрим особенности выполнения основных процедур преобразования информации.

Сбор и регистрация информации происходят по-разному в различных объектах. Так как информация это важнейшая движущая сила экономики, а экономика – движущая сила обучения, то рассмотрим процедуру в автоматизированных управленческих процессах промышленных предприятий, где производится сбор и регистрация первичной учетной информации, отражающей производственно-хозяйственную деятельность предприятия.

Особое значение при этом придается достоверности, полноте и своевременности первичной информации. Сбор и регистрация информации происходят при выполнении хозяйственных операций (прием готовой продукции, получение и отпуск материалов и т.п.). Сначала информацию собирают, затем ее фиксируют. Учетные данные могут возникать на рабочих местах в результате подсчета количества обработанных деталей, прошедших сборку узлов, изделий, выявление брака и т.д. Для сбора фактической информации производятся измерение, подсчет, взвешивание материальных объектов, получение временных и количественных характеристик работы отдельных исполнителей. Сбор информации, как правило, сопровождается ее регистрацией, т.е. фиксацией информации на материальном носителе (документе или машинном носителе). Запись в первичные документы в основном осуществляется вручную, поэтому процедуры сбора и регистрации остаются пока наиболее трудоемкими. В условиях автоматизации управления предприятием особое внимание придается использованию технических средств сбора и регистрации информации, совмещающих операции количественного измерения, регистрации, накоплению и передаче информации по каналам связи в ЭВМ с целью формирования первичного документа.

1.8.2. Передача информации

Передача информации осуществляется различными способами: с помощью почты, курьера, перевозки транспортными средствами, дистанционной передачи по каналам связи. При передаче почтой, курьером и транспортными средствами информация поступает очень медленно. Дистанционная передача информации по каналам связи отличается от предыдущих способов огромной скоростью. Для ее осуществления необходимы специальные технические средства. Некоторые технические средства сбора и регистрации, собирая автоматически информацию с датчиков, установленных на рабочих местах, передают ее в ЭВМ. Обмен данными позволяет взаимодействовать между собой территориально удаленным объектам. Доставка данных производится по заданному адресу с использованием сетей передачи данных. В современных условиях большое распространение по-

лучила распределенная обработка информации, при этом сети передачи данных превращаются в информационно-вычислительные сети (ИВС). ИВС представляют наиболее динамичную и эффективную отрасль автоматизированной технологии процессов ввода, передачи, обработки и выдачи информации. Важнейшим элементом ИВС является канал передачи данных (рис. 1.3).

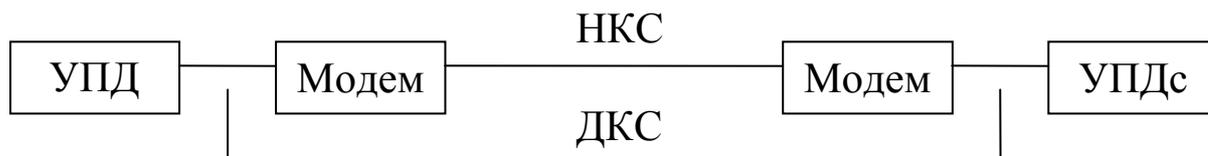


Рис. 1.3. Структурная схема канала передачи данных

На рис. 1.3. приняты следующие сокращения: УПД – устройство подготовки данных; НКС – непрерывный канал связи; ДКС – дискретный канал связи; УПДс – устройство повышения достоверности.

Непрерывный канал связи (НКС) совместно с функционирующими на его концах модемами образует дискретный канал связи (ДКС). В свою очередь, ДКС и устройства повышения достоверности (УПДс) образуют канал передачи данных. В НКС элементы данных передаются в виде физических сигналов в течение какого-то промежутка времени. Большинство НКС оказываются непригодными для передачи сигналов, отображающих данные, без предварительного их согласования. Для такого преобразования предусматривают специальные устройства – модемы. Модем представляет собой совокупность модулятора и демодулятора. С помощью модулятора информационный сигнал воздействует на некоторый параметр сигнала-переносчика, благодаря чему спектр сигнала смещается в область частот, для которых наблюдается наименьшее затухание в выбранном НКС. Обратную операцию, переход от модулированного сигнала к информационному сигналу, осуществляет демодулятор. Понятие ДКС позволяют, отвлекаясь от физической природы процессов, происходящих в НКС, представлять совокупность НКС и модемов, на вход которой подается последовательность кодовых символов – входное сообщение. Входным сообщением может

являться, как и двоичный код, так и другие различные кодировки (алфавиты). Для выходящего сообщения определение звучит точно также. В простейшем случае алфавиты на входе и выходе ДКС совпадают. УПДс может представлять собой специальную аппаратуру, предназначенную для повышения достоверности передачи данных, а может представлять собой специальную программу и ЭВМ, на которой она выполняется, может являться как элементом канала связи, так и элементом системы обработки информации. Повысить достоверность передачи информации можно с помощью кода Хэмминга, но это не единственный способ, их существует большое количество. Код Хэмминга позволяет не только распознавать правильность передачи информации, но и исправлять одиночные ошибки, избегая, таким образом, повторной передачи информации. Дистанционно информация может передаваться как первичная с мест ее возникновения, так и результативная в обратном направлении. Поступление информации по каналам связи в центр обработки в основном осуществляется двумя способами: на машинном носителе и непосредственно в ЭВМ при помощи специальных программных и аппаратных средств. Дистанционная передача постоянно развивается и совершенствуется. Особое значение этот способ передачи информации имеет в многоуровневых межотраслевых системах, где применение дистанционной передачи значительно ускоряет прохождение информации с одного уровня управления на другой и сокращает общее время обработки данных.

1.8.3. Обработка информации

В современных развитых информационных системах машинная обработка информации предполагает последовательно-параллельное во времени решение вычислительных задач. Это возможно при наличии определенной организации вычислительного процесса. Вычислительная задача, формируемая источником вычислительных задач (ИВЗ), по мере необходимости решения обращается с запросами в вычислительную систему. Организация вычислительного процесса предполагает определение последовательности решения задач и реализацию вычислений. Последовательность решения задается, исходя из их информаци-

онной взаимосвязи, когда результаты решения одной задачи используются как исходные данные для решения другой. Процесс решения определяется принятым вычислительным алгоритмом. Вычислительные алгоритмы должны объединяться в соответствии с требуемой технологической последовательностью решения задач в вычислительный граф системы обработки информации. Поэтому в вычислительной системе можно выделить систему диспетчирования (СД), которая определяет организацию вычислительного процесса, и ЭВМ (возможно и не одну), обеспечивающую обработку информации. Каждая вычислительная задача, поступающая в вычислительную систему, может быть рассмотрена как некоторая заявка на обслуживание. Последовательность вычислительных задач во времени создает поток заявок. В соответствии с требованиями на организацию вычислительного процесса возможно перераспределение поступающих задач на основе принятой схемы диспетчирования. Поэтому в структуре вычислительной системы должны быть предусмотрены соответствующие накопители и устройства диспетчирования, которые обеспечивают реализацию оптимальной организации вычислительного процесса.

На рисунке 1.4 представлена обобщенная структурная схема вычислительной системы. ИВЗ формирует входной поток заявок на их решение.

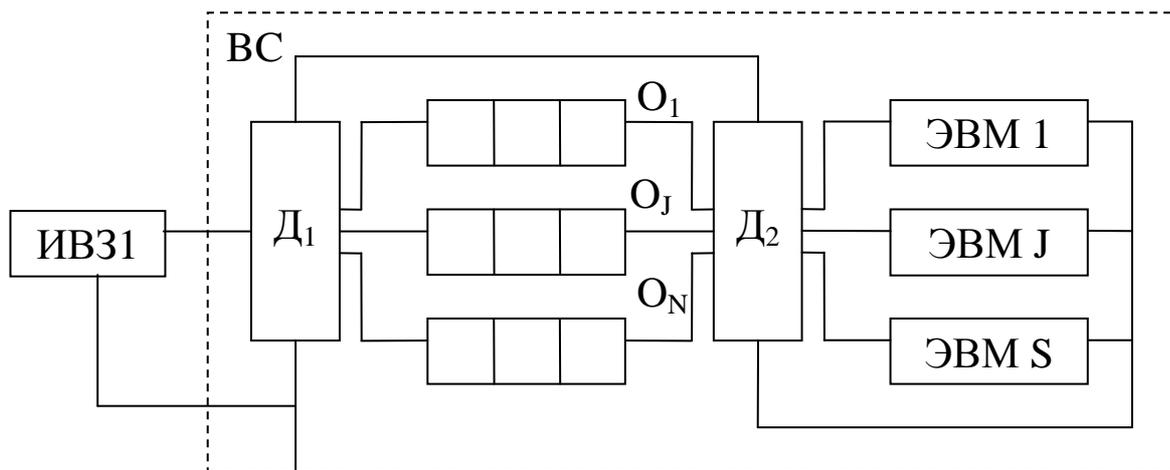


Рис. 1.4. Обобщенная структура вычислительной системы

На рис. 1.4 приняты следующие сокращения: ИВЗ – информационно-вычислительная заявка; Д – диспетчер; О – очередь заявок на обслуживание.

С помощью диспетчера D_1 реализуется обоснование поступившей заявки и постановка ее в очередь $O_1 \dots O_N$, которые реализуются на ячейках оперативной памяти. Заявки отображаются кодами и ожидают начала обслуживания в зависимости от информационной взаимосвязи между задачами. Диспетчер D_2 выбирает из очередей заявку на обслуживание, т.е. передает вычислительную задачу для обработки ЭВМ. Обслуживание обычно осуществляется в соответствии с принятым планом организации вычислительного процесса. Процесс выбора заявки из множества называется диспетчированием. Обычно выбирается заявка, имеющая преимущественное право на обслуживание. При этом иницируется соответствующая программа, реализующая вычислительный алгоритм решения задачи. При отсутствии заявок в очередях диспетчер D_2 переключает процессоры ЭВМ в состояние ожидания. В общем случае в вычислительной системе реализуется параллельное обслуживание за счет наличия нескольких ЭВМ. Можно считать, что процесс обслуживания осуществляется в два этапа. Сначала заявки ставятся в очередь с помощью диспетчера D_1 , а на следующем этапе они обслуживаются путем выбора заявок из очереди диспетчером D_2 . Диспетчеры D_1 и D_2 реализуются программным путем и представляют собой управляющие программы. Информационные процессы в автоматизированных системах организационного управления реализуются с помощью ЭВМ и других технических средств. Деятельность ИВЦ и ИВЦКП характеризовалась обработкой больших объемов информации, использованием нескольких средних и больших ЭВМ, квалификационным персоналом для обслуживания техники и разработки программного обеспечения. Централизованное применение вычислительных и других технических средств позволяло обеспечить их надежную работу, планомерную загрузку и квалификационное обслуживание. Централизованная обработка информации наряду с рядом положительных сторон имела ряд отрицательных черт, порожденных прежде всего отрывом конечного пользователя от технологического процесса обработки информации. Децентрализованные формы использования вычислительных ресурсов начали формироваться со второй половины 80-х годов, когда сфера экономики получи-

ла возможность перейти к массовому использованию персональных ЭВМ. Децентрализация предусматривает размещение ПЭВМ в местах возникновения и потребления информации, где создаются автономные пункты (АП) и автоматизированные рабочие места.

В начале работы в машины загружаются программы и различные информационные массивы (условно-постоянные, переменные, справочные), каждый из которых сначала, как правило, обрабатывается для получения каких-либо результатных показателей, а затем массивы объединяются для получения сводных показателей.

При обработке экономической информации на ЭВМ выполняются арифметические и логические операции. Арифметические операции обработки данных в ЭВМ включают все виды математических действий, обусловленных программой. Логические операции обеспечивают соответствующее упорядочение данных в массивах (первичных, промежуточных, постоянных, переменных), подлежащих дальнейшей арифметической обработке.

Технология электронной обработки информации – человеко-машинных процессов исполнения взаимосвязанных операций, протекающих в установленной последовательности с целью преобразования исходной (первичной) информации в результатную. В условиях электронной обработки данных преобладают операции, выполняемые автоматически на машинах и устройства, которые считывают данные, выполняют операции по заданной программе в автоматическом режиме при участии человека или сохраняя за пользователем функции контроля, анализа и регулирования.

Технология автоматизированной обработки информации строится на следующих принципах интеграции обработки данных и возможности работы пользователей в условиях эксплуатации автоматизированных систем централизованного хранения и коллективного использования данных (банков данных): распределение обработки данных на базе развитых систем передачи; рациональное сочетание централизованного и децентрализованного управления и организации вычислительных систем; моде-

лирование и формализованное описание данных, процедур их преобразования, функции и рабочих мест исполнителей; учет конкретных особенностей объекта, в котором реализуется машинная обработка информации.

Различают два основных типа организации технологических процессов: предметный и пооперационный.

Предметный тип организации технологии предполагает создание параллельно действующих технологических линий, специализирующихся на обработке информации и решении конкретных комплексов задач (учет труда и заработной платы, снабжение и сбыт, финансовые операции и т.п.) и организующих пооперационную обработку данных внутри линии.

Пооперационный (поточный) тип построения технологического процесса предусматривает последовательное преобразование обрабатываемой информации согласно технологии, представленной в виде непрерывной последовательности сменяющих друг друга операций, выполняемых в автоматическом режиме. Такой подход к построению технологии оказался приемлемым при организации работы абонентских пунктов и автоматизированных рабочих мест.

Основной этап информационного технологического процесса связан с решением функциональных задач на ЭВМ. Внутримашинная технология решения задач на ЭВМ, как правило, реализует многочисленные типовые процессы преобразования экономической информации. Решение каждой отдельной задачи или комплекса задач требует выполнения следующих операций: ввод программы машинного решения задачи и размещения ее в памяти ЭВМ; ввод исходных данных; логический и арифметический контроль введенной информации; исправление ошибочных данных; компоновка входных массивов и сортировка введенной информации; вычисления по заданному алгоритму; получение выходных массивов информации; редактирование выходных форм; вывод информации на экран и машинные носители; печать таблиц с выходными данными. Выбор того или иного варианта технологии определяется, прежде всего, как объемно-временными особенностями решаемых задач, периодичностью, срочностью, требованиями к скорости связи пользователя с

ЭВМ, так и режимных возможностей технических средств – в первую очередь ЭВМ.

Различают следующие режимы взаимодействия пользователя с ЭВМ: пакетный и интерактивный (запросный, диалоговый). Сами же ЭВМ могут функционировать в следующих режимах: одно- и многопрограммном, разделении времени, реального времени, телеобработки. При этом предусматривается цель удовлетворения потребности пользователей в максимально возможной автоматизации решения разнообразных задач.

Пакетный режим был наиболее распространен в практике централизованного решения экономических задач, когда большой удельный вес занимали задачи отчетности о производственно-хозяйственной деятельности экономических объектов разного уровня управления.

Организация вычислительного процесса при пакетном режиме строилась без доступа пользователя к ЭВМ. Его функции ограничивались подготовкой исходных данных по комплексу информационно взаимосвязанных задач и передачей их в центр обработки, где формировался пакет, включающий задание для ЭВМ на обработку, программы, исходные, нормативно-расценочные и справочные данные.

Пакет вводился в ЭВМ и реализовывался в автоматическом режиме без участия пользователя и оператора, что позволяло минимизировать время выполнения заданного набора задач. При этом работа ЭВМ могла проходить в однопрограммном или многопрограммном режиме, что предпочтительнее, так как обеспечивалась параллельная работа основных устройств машины.

Интерактивный режим предусматривает непосредственное взаимодействие пользователя с информационно-вычислительной системой, может носить характер запроса (как правило, регламентированного) или диалога с ЭВМ. Запросный режим необходим пользователям для взаимодействия с системой через значительное число абонентских терминальных устройств, в том числе удаленных на значительное расстояние от центра обработки. ЭВМ в подобных случаях реализует систему массового обслуживания, работает в режиме разделения времени, при котором несколько независимых абонентов (пользователей) с помощью

устройств ввода-вывода имеют в процессе решения своих задач непосредственный и практически одновременный доступ к ЭВМ. Этот режим позволяет дифференцированно в строго установленном порядке предоставлять каждому пользователю время для общения с ЭВМ, а после окончания сеанса отключать его.

Диалоговый режим открывает пользователю возможность непосредственно взаимодействовать с вычислительной системой в допустимом для него темпе работы, реализуя повторяющийся цикл выдачи задания, получения и анализа ответа. При этом ЭВМ сама может инициировать диалог, сообщая пользователю последовательность шагов (представление меню) для получения искомого результата.

Хранение и накопление информации вызвано многократным ее использованием, применением постоянной информации, необходимостью комплектации первичных данных до их обработки. Хранение информации осуществляется на машинных носителях в виде информационных массивов, где данные располагаются по установленному в процессе проектирования группировочному признаку. Поиск данных — это выборка нужных данных из хранимой информации, включая поиск информации, подлежащей корректировке или замене запроса на нужную информацию.

Развитие организационных форм вычислительной техники строится на сочетании централизованной и децентрализованной — смешанной — форм. Предпосылкой появления смешанной формы явилось создание сетей ЭВМ на основе различных средств связи. Сети ЭВМ предполагают объединение в систему с помощью каналов связи вычислительных средств, программных и информационных ресурсов (баз данных, баз знаний). Сетями могут охватываться различные формы использования ЭВМ, причем каждый абонент имеет возможность доступа не только к своим вычислительным ресурсам, но и к ресурсам всех остальных абонентов, что создает ряд преимуществ при эксплуатации вычислительной системы.

В последнее время организация применения компьютерной техники претерпевает значительные изменения, связанные с переходом к созданию интегрированных информационных систем.

Интегрированные информационные системы создаются с учетом того, что они должны осуществлять согласованное управление данными в пределах предприятия (организации), координировать работу отдельных подразделений, автоматизировать операции по обмену информацией, как в пределах отдельных групп пользователей, так и между несколькими организациями, отстоящими друг от друга на десятки и сотни километров. Основой для построения подобных систем служат локальные вычислительные сети (ЛВС). Характерной чертой ЛВС является предоставление возможности пользователям работать в универсальной информационной среде с функциями коллективного доступа к данным.

Все больше внимания уделяется развитию не только локальных, но и распределенных сетей, без которых немислимо решение современных задач информатизации.

1.9. Информационный ресурс и его составляющие

1.9.1. Определение информационной технологии

Чтобы определить понятие «информационная технология», обратимся к термину «технология», который имеет множество толкований.

В широком смысле под *технологией* понимают науку о законах производства материальных благ [14].

Следуя данному определению, в технологии выделяют три аспекта: идеологию, или принципы производства; орудия труда, т.е. станки, машины, агрегаты, и кадры, владеющие профессиональными навыками. Эти аспекты составляют соответственно, информационную, инструментальную и социальную, составляющую.

Информационная составляющая включает описание принципов и методов производства; инструментальная – орудия труда, с помощью которых реализуется производство; социальная – кадры и их организацию.

Для конкретного производства *технология* понимают в узком смысле как совокупность приемов и методов, определяющих последовательность действий для реализации производственного процесса.

Поэтому можно говорить о технологии, как совокупности методов обработки, изготовления, изменения состояния, свойств, формы сырья, материала или полуфабриката в процессе производства, например технология получения металлов, химическая технология, технология строительных работ и т.д.

Рассматривая в качестве материалов информацию, над которой производятся определенные действия, изменяющие ее характеристики, приходим к определению информационной технологии.

Информационная технология (ИТ) – совокупность методов и способов получения, обработки, представления информации, направленных на изменение ее состояния, свойств, формы, содержания и осуществляемых в интересах пользователей. При сопоставлении технологии материального производства и информационной технологии выделим их цели.

Цель технологии материального производства – выпуск продукции, удовлетворяющей потребности человека или системы.

Цель информационной технологии – производство информации для ее анализа человеком и принятия на его основе решения по выполнению какого-либо действия. Информационные технологии предназначены для снижения трудоемкости процессов использования информационных ресурсов. Результат применения информационных технологий обособляется в так называемых информационных продуктах.

Информационный продукт – документированная информация, подготовленная в соответствии с потребностями пользователей и представленная в форме товара. Информационными продуктами являются программные продукты, базы и банки данных и другая информация.

Информационные технологии обеспечивают переход от рутинных к промышленным методам и средствам работы с информацией в различных сферах человеческой деятельности, обеспечивая ее рациональное и эффективное использование.

С современных позиций информационные технологии реализуются с использованием средств компьютерной и оргтехни-

ки. Поэтому, в современном понимании под информационными технологиями подразумевается следующее.

Информационная технология – совокупность методов, производственных и программно-технологических средств, объединенных в технологическую цепочку, обеспечивающую сбор, хранение, обработку, вывод и распространение информации. Последнее определение отражает использование в информационных технологиях принципов современных автоматизированных систем. С учетом их использования, информационная технология может называться *автоматизированной информационной технологией* (АИТ).

Информационная технология – это системно-организованная последовательность операций, выполняемых над информацией с использованием средств и методов автоматизации. При этом под операциями понимаются элементарные действия над информацией, которые могут быть объединены в типовые технологические операции (действия) информационной технологии:

- сбор и регистрация информации;
- передача информации;
- ввод информации;
- обработка информации;
- вывод информации;
- хранение информации;
- накопление информации;
- поиск информации;
- анализ информации

1.9.2. Свойства информационных технологий

Информационная технология имеет свою цель, методы и средства реализации. Как было отмечено, *целью ИТ* является создание из информационного ресурса качественного информационного продукта, удовлетворяющего требованиям пользователя [14].

Методами ИТ являются методы обработки и передачи данных.

Средства (инструментарий) ИТ – это математические, программные, информационные, технические и другие средства. В итоге получаем развернутое определение для информационной технологии.

Под *информационной технологией* понимается целостная техническая система, обеспечивающая целенаправленное создание, передачу, хранение и отображение информационного продукта (данных, идей, знаний) с наименьшими затратами и в соответствии с закономерностями той социальной среды, где она развивается.

Поскольку существенную часть технических средств для реализации информационных технологий занимают средства компьютерной техники, то часто под *информационными технологиями*, особенно под новыми информационными технологиями, понимаются компьютерные информационные технологии.

Можно выделить три уровня рассмотрения информационных технологий:

- первый уровень – теоретический. Основная задача связана с созданием комплекса взаимосвязанных моделей информационных процессов;
- второй уровень – исследовательский. Основная задача направлена на разработку методов автоматизированного конструирования оптимальных конкретных информационных технологий;
- третий уровень – прикладной, связанный с инструментальными и предметными аспектами информационных технологий.

В соответствии с определением информационных технологий, отметим их характерные свойства:

- целью процесса в информационных технологиях является получение информации (информационного продукта);
- предметом процесса в информационных технологиях (предмет обработки) являются данные или знания;
- средства осуществления процесса в информационных технологиях представляются;
- различными вычислительными комплексами (программными, аппаратными, программно-аппаратными);

- процессы обработки данных в информационных технологиях разделяются на операции в соответствии с выбранной предметной областью;

- управляющие воздействия на процессы в информационных технологиях осуществляются лицами, принимающими решения;

- критериями оптимальности процесса в информационных технологиях служат своевременность доставки информации пользователям, ее надежность, достоверность, полнота.

- информационные технологии обеспечивают высокую степень расчленения всего процесса обработки данных на этапы, операции, действия;

- информационные технологии включают весь набор элементов для достижения поставленной цели;

- информационные технологии должны иметь регулярный характер.

Кроме того, информационные технологии различаются:

- составом и последовательностью операций;

- степенью их автоматизации (долей машинного и ручного труда);

- надежностью их выполнения.

Свойство *надежности* в информационных технологиях реализуется качеством выполнения основных операций и наличием разнообразного их контроля.

Организация информационных технологий определяется рядом факторов и критериев:

- объемы информации;

- срочность и точность ее обработки;

- структурные и предметные особенности объекта управления;

- соответствие временным регламентам взаимодействия процессов в предметной области и их элементов.

1.9.3. Особенности информационных технологий

В числе отличительных особенностей информационных технологий выделим следующие наиболее важные [14].

1. Информационные технологии позволяют активизировать и эффективно использовать информационные ресурсы общества, которые сегодня являются наиболее важным стратегическим фактором его развития. Под информационными ресурсами понимаются данные и документированная информация о жизнедеятельности общества, организованные в базы и банки данных, а также другие формы организации информации.

Эффективное использование информационных ресурсов (научных знаний, открытий, изобретений, технологий, передового опыта) позволяет получить существенную экономию других видов ресурсов – сырья, энергии, полезных ископаемых, материалов и оборудования, людских ресурсов, социального времени.

2. Информационные технологии позволяют оптимизировать и во многих случаях автоматизировать информационные процессы, которые в последние годы занимают все большее место в жизнедеятельности человеческого общества.

Общеизвестно, что развитие цивилизации происходит в направлении становления информационного общества, в котором объектами и результатами труда большинства занятого населения становятся уже не материальные ценности, а главным образом информация и научные знания.

3. Информационные процессы являются важными элементами других более сложных производственных или же социальных процессов. Очень часто информационные технологии выступают в качестве компонентов соответствующих *производственных* или *социальных технологий*. Характерными примерами могут служить системы автоматизированного проектирования промышленных изделий, гибкие автоматизированные и роботизированные производства, автоматизированные системы управления технологическими процессами и т.п.

4. Информационные технологии на современном этапе играют исключительно важную роль в обеспечении информационного взаимодействия между людьми, а также в системах подготовки и распространения массовой информации. Характерными примерами здесь могут служить электронная почта, факсимиль-

ная передача информации и другие виды телекоммуникационной связи.

5. Информационные технологии занимают сегодня центральное место в процессе интеллектуализации общества, развития его системы образования и культуры. Использование обучающих информационных технологий оказалось весьма эффективным методом для систем самообразования, продолженного обучения, а также для систем повышения квалификации и переподготовки кадров.

6. Информационные технологии играют в настоящее время ключевую роль также и в процессах получения и накопления новых знаний. В первую очередь здесь необходимо отметить *методы* информационного моделирования исследуемых наукой процессов и явлений, позволяющие ученому проводить своего рода «*вычислительный эксперимент*».

Вторым перспективным направлением представляют собой *методы искусственного интеллекта*, позволяющие находить решения плохо формализуемых задач, а также задач с неполной информацией и с нечеткими исходными данными.

Третье перспективное направление связано с использованием методов *когнитивной компьютерной графики*. При помощи этих методов, позволяющих образно представлять различные математические формулы и закономерности, уже удалось доказать несколько весьма сложных теорем в теории чисел. Кроме того, их использование открывает новые возможности для познания человеком самого себя, принципов функционирования своего сознания.

7. Принципиально важное для современного этапа развития общества значение развития информационных технологий заключается в том, что их использование может оказать существенное содействие в решении глобальных проблем человечества.

Методы *информационного моделирования глобальных процессов*, особенно в сочетании с методами космического информационного мониторинга, могут обеспечить уже сегодня возможность прогнозирования многих кризисных ситуаций в регионах повышенной социальной и политической напряженности, а также в районах экологического бедствия, в местах природных

катастроф и крупных технологических аварий, представляющих повышенную опасность для общества.

Из всех видов технологий информационная технология, применяемая в сфере экономики и управления, предъявляет самые высокие требования к “человеческому фактору”, оказывая принципиальное влияние на квалификацию работника, содержание его труда, физическую и умственную нагрузку, профессиональные перспективы и уровень социальных отношений.

Таким образом, свойства и особенности информационных технологий в конечном итоге имеют стратегическое значение для развития общества, их необходимо учитывать при проектировании автоматизированных информационных систем.

1.10. Виды обработки данных

1.10.1. Процедуры обработки данных в зависимости от видов представления данных

Обработка данных представляет собой последовательность операций, производимых над данными [17]. Процедуры обработки данных могут различаться в зависимости от форм и видов представления данных (рис. 1.5, рис. 1.6).

В *экономической деятельности* наиболее распространено цифровое и буквенное отображение информации в различных вариантах и сочетаниях: документы, тексты, таблицы, файлы, базы данных и др. В информационных технологиях, применяемых в экономической деятельности, как и в телевидении, кино-, мультимедийных технологиях широко используются также изображения, речь, звуки, сигналы и т.д.

В *управлении технологическими процессами и объектами дискретного и непрерывного действия* более всего актуальна обработка сигналов, сообщений для управления на низовом, производственном уровне.

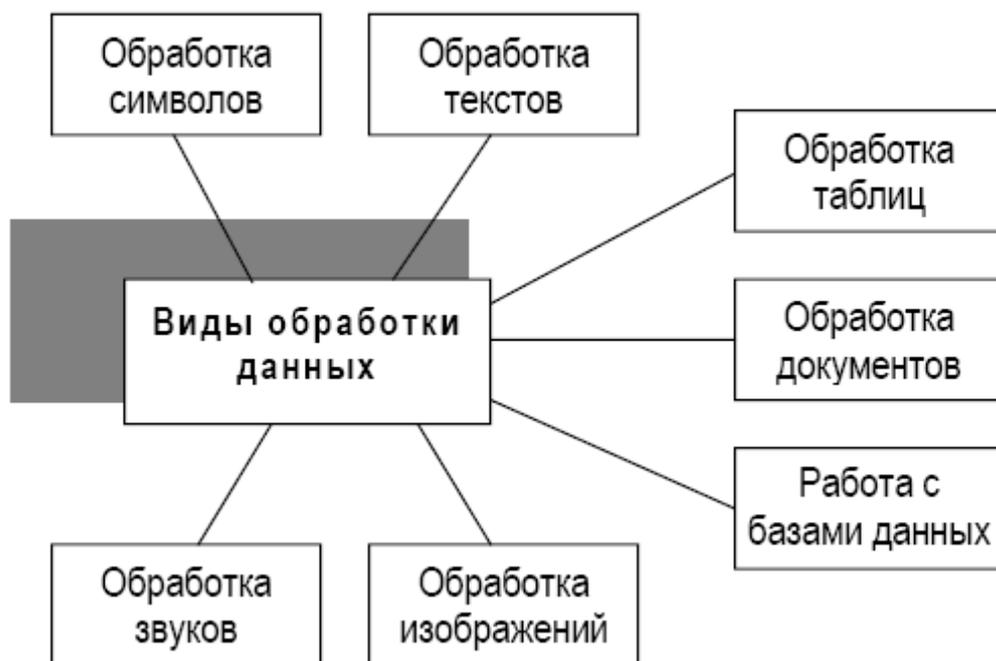


Рис. 1.5. Виды обработки данных

Для среднего и верхнего уровней управления предприятием информация обобщается, группируется, агрегируется, чтобы получить более полную и достоверную картину состояния всего производства при принятии управленческих решений.

Информационные технологии разделяются на классы (инструментарий) по типу обрабатываемой информации (рисунок) и могут объединяться в интегрированные технологии. Обработка цифровой, символьной, текстовой, табличной информации, в виде баз данных, сигналов, речи, звуков, документов, изображений имеет свои особенности и специфику, реализуется видами инструментария информационных технологий:

1. Текстовые процессоры: Microsoft Word, Word Perfect, Лексикон, Lotus и др.;
2. Электронные таблицы: Microsoft Excel, Lotus 1-2-3, Corel Quattro Pro, Sun Star Office Calc и др.;
3. Программы презентационной графики: Microsoft Power Point, Corel Presentations, Lotus Freelance Graphics, Sun Star Office Impress и др.;
4. WEB-редакторы: Microsoft Front Page, Netscape Composer, Macromedia Free Hand и др.;
5. Почтовые клиенты: Microsoft Outlook, Microsoft Outlook Express, Netscape Messenger, The Bat и др.;

6. Редакторы растровой графики: Adobe Photoshop, Corel Photo-Paint и др.;

7. Редакторы векторной графики: Corel Draw, Adobe Illustrator и др.;

8. Настольные издательские системы: Adobe Page Maker, Quark Xpress, Corel Ventura, Microsoft Publisher и др.;

9. Средства разработки: Borland Delphi, Microsoft Visual Basic, Borland C++ Builder, Microsoft Visual C++ и др.

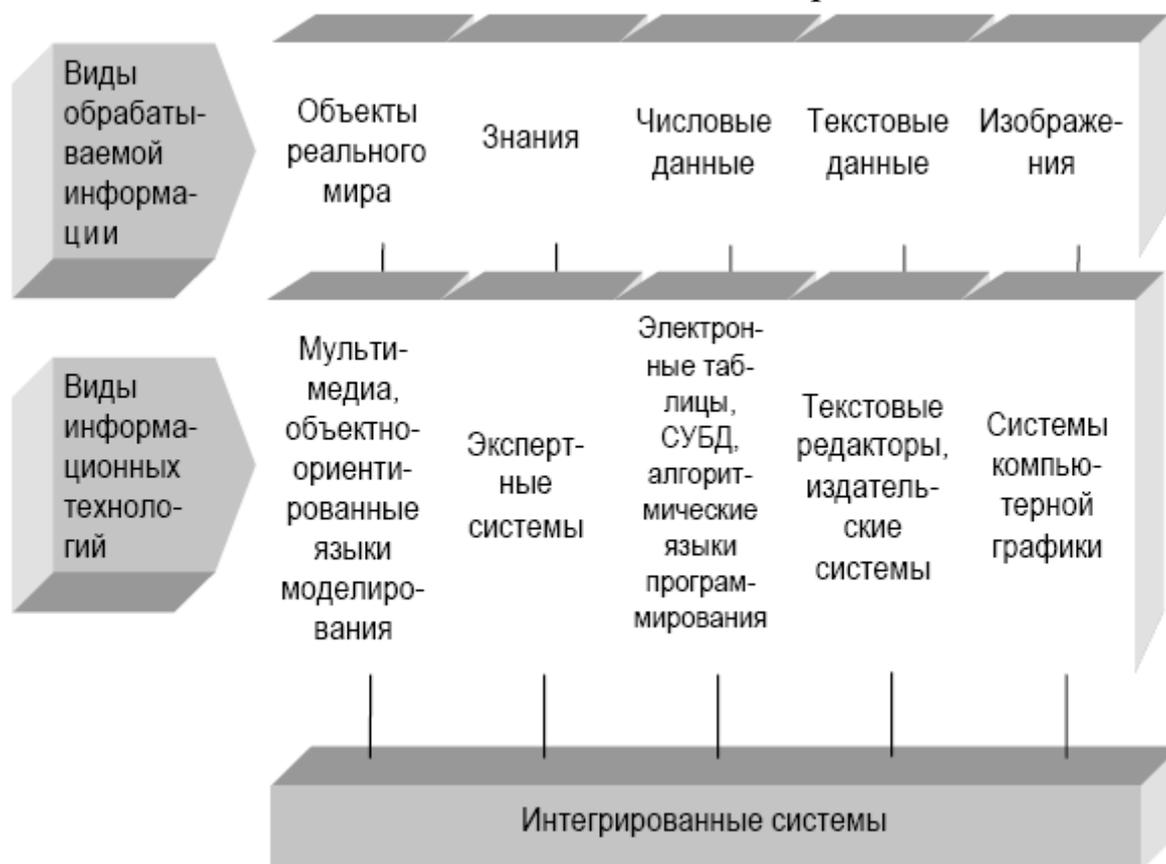


Рис. 1.6. Классификация ИТ в зависимости от типа обрабатываемой информации

Кратко охарактеризуем наиболее распространенные виды обработки данных, организуемых современными информационными технологиями.

1.10.2. Технология обработки изображений

В технологии обработки изображений используется анализ, преобразование и трактовка изображений. Сначала изображения вводятся через видео или другие устройства. В результате сканирования изображений формируется большой объем данных.

Данные с изображениями используют специальные графические форматы.

Введенное изображение подвергается различным видам обработки: распознаванию объектов и образов, устранению искажений, что требует высоких скоростей, большой памяти и специальных технологий. Обработка изображений используется в компьютерной рекламе, в издательском деле, в интегрированных технологиях мультимедиа.

Обработка изображений как направление связано с развитием электронной техники и технологий. При обработке изображений требуются высокие скорости, большие объемы памяти, специализированное техническое и программное оснащение.

Изображения относятся к разного рода объектам, выделению их контуров, перемещению, распознаванию и т.д. Объектами могут быть пользователи, клиенты, прикладные процессы, документы, предметы, явления, которые являются источниками или адресатами информации.

Кроме того, данные могут быть представлены в виде неподвижных или движущихся изображений. Например, использование изображений осуществляется при проведении видеоконференций, в видеосюжетах, в анимации, в создании музыкальных и видеообразов и др.

1.10.3. Видеотехнология

Видеотехнология строится на разработке и демонстрации движущихся изображений, что открыло широкие возможности в возникновении мультисреды. Видеотехнология применяется для создания видеосюжетов, фильмов, деловой графики и др. Для этой технологий необходимо сжатие изображений. Оно обеспечивает уменьшение файла в 160–200 раз, после этого данные записываются во внешнюю память.

Технология визуализации – процесс многооконного представлений данных в виде изображений (обратный сжатию). Визуализация обеспечивает преобразование любого типа данных в разноцветные движущие или неподвижные изображения. Каждый зрительный образ по объему данных соответствует тысячам страниц текста. Представление информации в виде видеосюже-

тов позволяет оживлять образы, наблюдать динамику процессов и явлений.

Визуализация широко используется в создании виртуальной реальности (нереальное, воображаемое, объемное представление, создаваемое звуком и изображениями). Технология виртуальной реальности используется в конструкторской, рекламной деятельности, в создании мультипликационных фильмов. Этот процесс именуется мультипликацией.

1.10.4. Обработка текстов

Технология обработки текстов является одним из средств так называемого *электронного офиса*. В этой технологии наиболее трудоемким является ввод текста; следующими этапами являются подготовка текста, его оформление и вывод. При работе с текстами пользователь должен иметь разнообразные функции (инструментарий), повышающие эффективность и производительность его деятельности.

Электронные тексты могут сопровождаться изображениями и звуком. Обработка текстов тесно связана с организацией гипертекста и электронной почтой. Технологии обработки текстов реализуются с помощью программ текстовых редакторов (процессоров) и издательских систем.

1.10.5. Обработка таблиц и технология гипертекста

Технологии обработки таблиц реализуются с помощью комплекса прикладных программ электронных таблиц в составе электронного офиса и дополняются рядом аналитических возможностей.

Работа с электронной таблицей позволяет вводить и обновлять данные, команды, формулы, определять взаимосвязь и взаимозависимость между клетками, данными в виде функций, аргументами которых являются записи в клетках. В ячейках таблицы могут размещаться записные книжки, календари, справочники, списки мероприятий. Обработка текстов и таблиц является главной составляющей, на которой строится обработка текстов.

Гипертекст формируется в результате представлений текста как ассоциативно связанных блоков информации. Ассоциативная связь – это соединение, сближение представлений, смеж-

ных, противоположных, аналогичных и т.д. Гипертекст значительно отличается от обычного текста. Обычные (линейные) тексты имеют последовательную структуру и предусматривают их чтение слева направо и сверху вниз.

Использование гипертекста позволяет фиксировать отдельные идеи, мысли, факты, а затем связывать их друг с другом, двигаясь в любых направлениях, определяемых ассоциативными связями. В тех случаях, когда к блокам текста добавляются большое число изображений и запись звука, гипертекст превращается в *гиперсреду*.

1.10.6. Технология обработки речи

Технология обработки речи является многоплановой проблемой, охватывающей широкий круг задач. В их перечень прежде всего входят распознавание и синтез речи.

Распознавание речи преобразует ее в текст, открывает возможность использования ее в качестве источника информации. Обратной распознаванию является задача синтеза речи, т.е. преобразования текста в речь. Так как речь, представленная дискретными сигналами, характеризуется большим объемом данных, то при ее записи в память или при передаче по сети осуществляется операция сжатия данных.

Обработка речи может использоваться в образовательной, медицинской сферах деятельности, а также для управления объектами при голосовом вводе.

1.10.7. Технология обработки и преобразования сигналов

Технология обработки и преобразования сигналов выполняется при решении многих информационных задач. Сигналы обрабатываются различными методами (аналоговыми и дискретными). Обработка сигналов используется в распознавании образов, телеобработке данных и опирается на методологию искусственного интеллекта.

Обработка сигналов, в первую очередь дискретных, используется в управлении производством для таких объектов, как станки, автоматические линии, для мониторинга (контроля и

слежения) выпуска изделий, например, в машиностроительных отраслях, медицине, радиолокации и т.д.

1.10.8. Нейрокомпьютерные технологии

Нейрокомпьютерные технологии используют взаимодействующие друг с другом специальные нейрокомпоненты на базе микропроцессоров. Такой подход основан на моделировании поведения нервных клеток (нейронов).

Нейротехнология применяется в создании искусственного интеллекта для решения сложных задач: распознавание образов, управление кредитными рисками, прогноз фондовых ситуаций, определение стоимости недвижимости с учетом качеству зданий, их состояния, окружающей обстановки и среды, автоматическое распознавание чеков и др.

Компонентами нейротехнологий являются нейронные компьютеры и процессоры, а также *нейронные сети*, как класс алгоритмов, обеспечивающих решение сложных задач.

Нейросети обладают способностью самообучения, имеют высокое быстродействие, так как обработка информации в них осуществляется многими компонентами, функционирующими параллельно.

1.11. Алгоритм и его свойства

Один из важнейших этапов решения задач на ЭВМ – составление алгоритма. В 1983 году отмечалось 1200-летие со дня рождения одного из величайших ученых Средней Азии и средневекового Востока Мухамада ибн Мусы аль-Хорезми. Он написал ряд трактатов по арифметике и алгебре, в том числе книгу "Арифметика индусскими цифрами" – о счете с помощью десяти цифр и правилах арифметических действий с числами.

Имя ученого аль-Хорезми превратилось в понятие *algorithmi*, первоначально обозначавшее десятичную систему исчисления и правила арифметических действий в этой системе. Отсюда и возник современный научный термин "алгоритм".

Каждый из нас ежедневно использует различные алгоритмы: инструкции, правила, рецепты и т.п. Обычно мы это делаем не задумываясь. Например, открывая дверь ключом, никто не размышляет над тем, в какой последовательности выполнять

действия. Однако чтобы научить кого-нибудь открывать дверь, придется четко указать и сами действия, и порядок их выполнения. То же потребует и при указании маршрута поездки.

С другой стороны алгоритм относится к основным понятиям математики, а поэтому не имеет определения. Часто это понятие формулируют так: «точное предписание о порядке выполнения действий, из заданного фиксированного множества, для решения всех задач, заданного класса».

Поэтому при составлении и записи алгоритма необходимо обеспечить, чтобы он обладал рядом свойств.

Однозначность алгоритма, под которой понимается единственность толкования исполнителем правила построения действий и порядок их выполнения. Чтобы алгоритм обладал этим свойством, он должен быть записан командами из системы команд исполнителя.

Конечность алгоритма – обязательность завершения каждого из действий, составляющих алгоритм, и завершенность выполнения алгоритма в целом.

Результативность алгоритма, предполагающая, что выполнение алгоритма должно завершиться получением определённых результатов.

Массовость, т. е. возможность применения данного алгоритма для решения целого класса задач, отвечающих общей постановке задачи. Для того чтобы алгоритм обладал свойством массовости, следует составлять алгоритм, используя обозначения величин и избегая конкретных значений.

Правильность алгоритма, под которой понимается способность алгоритма давать правильные результаты решения поставленных задач.

Эффективность – для решения задачи должны использоваться ограниченные ресурсы компьютера (процессорное время, объём оперативной памяти и т. д.).

1.11.1. Описание алгоритмов на естественном языке

Если речь идёт о составлении алгоритмов для процессора ЭВМ (электронно-вычислительной машины), исполнителем является процессор. Упрощённая модель процессора содержит

устройство считывания данных, стёк (специальную оперативную память небольшого объёма, предназначенную для временного хранения данных) и арифметическое устройство, которое может выполнять арифметические действия [9,10,21].

Предположим, что программа, составленная для такого процессора, содержит числовые данные и символы арифметических действий над этими данными. Вот пример такой программы, предназначенной для вычисления сумм двух чисел 2 и 3:

2, 3, +

Проследим выполнение этой программы. Первая операция - считывание в стек значения 2. Затем в стек считывается второе значение (3). Первое значение при этом сдвигается во вторую ячейку памяти. Третий шаг выполнения программы – вычисление суммы двух считанных значений (они называются *операндами*). Результат этой операции – значение 5 – записывается в первую ячейку стёка.

Был рассмотрен пример простейшей программы. Она является записью алгоритма решения некоторого класса задач – задач вычисления суммы двух чисел. Обозначим эти числа a и b . Тогда алгоритм можно записать следующим образом:

1. Считать число a .
2. Считать число b .
3. Выполнить суммирование $c := a + b$.
4. Вывести число c .

Это пример записи алгоритма *на естественном языке*, то есть на языке человеческого общения. Видно, что формулировка алгоритма не зависит от конкретных значений переменных a и b , поэтому его можно применять для решения достаточно большого числа сходных задач, вместе составляющих целый *класс* задач суммирования. Алгоритм описывает действия не над конкретными значениями, а над абстрактными объектами.

Основными объектами программирования являются *переменные*. Переменные в программе отличаются от переменных, используемых в записи математических формул. Несмотря на сходство терминов, правила использования переменных в программах для компьютера отличаются от правил работы с математическими переменными. Это различие необходимо уяснить.

В программировании переменную можно трактовать как одну или несколько ячеек оперативной памяти компьютера, которым присвоено определённое имя. Содержимое этих ячеек может меняться, но имя переменной остаётся неизменным. В математике значение переменной в рамках определённой задачи неизменно, но меняется в других задачах из данного класса. Именно поэтому конструкция

$$a := a + 1 \quad (1.10)$$

воспринимается программистом совершенно естественно, а уравнение

$$a = a + 1 \quad (1.11)$$

математик сочтёт неверным. В первом случае имеется в виду вычисление суммы содержимого ячейки a и числовой константы 1 и занесение полученного результата в ту же ячейку a . Вторым случаем равносильно неверному тождеству $0 = 1$.

Составим алгоритм решения следующей задачи. Пусть заданы два значения x и y . Необходимо сравнить эти значения и напечатать имя большей переменной. Для этой задачи достаточно сравнить оба значения и в зависимости от результата сравнения вывести на печать символ « x » и символ « y »:

1. Ввести значение x .
2. Ввести значение y .
3. Если $x < y$, то напечатать « y », иначе напечатать « x ».

В этом алгоритме используются *алгоритмические структуры* - *линейная последовательность операций* и *ветвление* (шаг 3, условный оператор). Последняя структура называется так потому, что после передачи в неё управления выполнение алгоритма может пойти по одному из двух возможных ветвлений. То, какая ветвь будет выбрана, зависит от выполнения условия. Линейная последовательность в данном примере состоит из *блоков ввода/вывода данных*.

Для записи алгоритмов использовался естественный язык. Иногда используют полуформальный язык с ограниченным словарём (часто на основе английского языка), промежуточный между естественным языком и языком программирования. Такой язык называется *псевдокодом*. Запись алгоритма на псевдокоде

называется *структурным планом*. Псевдокод удобен тем, что позволяет программисту сосредоточиться на формулировке алгоритма, не задумываясь над синтаксическими особенностями конкретного языка программирования.

1.11.2. Описание алгоритмов с помощью блок-схем

Для разработки структуры программы удобнее пользоваться записью алгоритма в виде *блок-схемы* (в англоязычной литературе используется термин *flow-chart*). Для изображения основных алгоритмических структур и блоков на блок-схемах используют специальные графические символы [9,10,21]. Они приведены на рисунке 1.7.

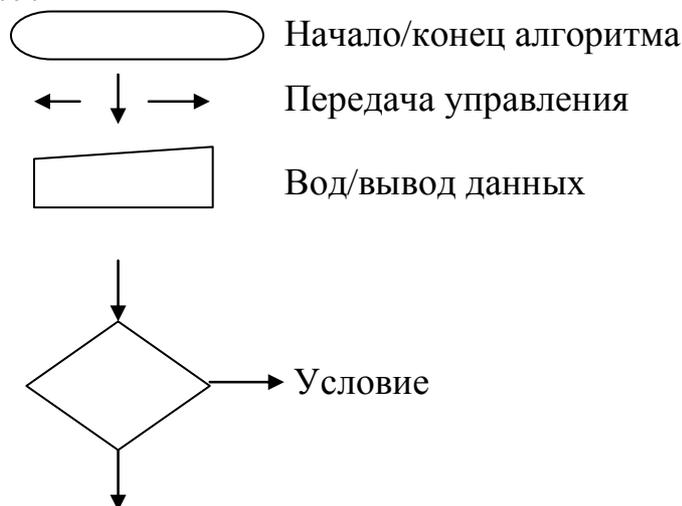


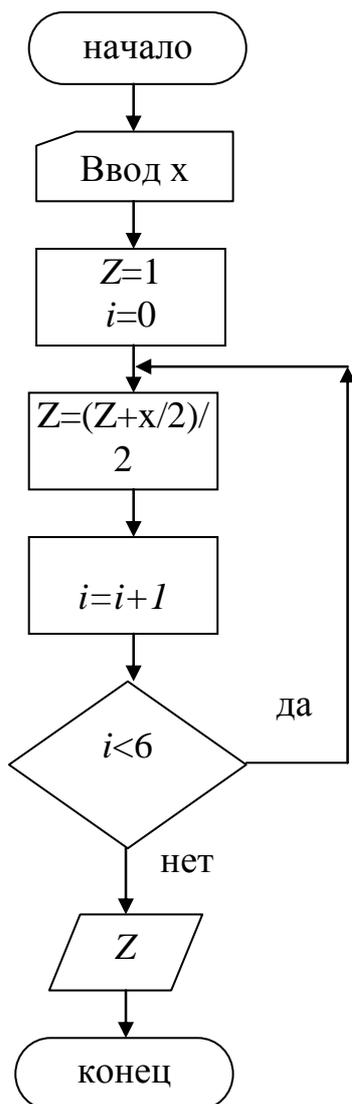
Рис. 1.7. Графический способ задания алгоритмов

Составим алгоритм вычисления квадратного корня из произвольного положительного вещественного числа x методом Герона и запишем его на естественном языке, а также в виде блок-схемы. Метод основан на многократном применении формулы:

$$z_{n+1} = \frac{1}{2} \left(z_n + \frac{x}{z_n} \right), \quad (1.12)$$

при $z_0 = 1$.

Числовая последовательность z_n в пределе при $n \rightarrow \infty$ сходится к искомому значению. Выполним только 5 итераций метода, считая, что при этом будет достигнута достаточно хорошая точность. Обычно десяти итераций более чем достаточно для достижения хорошей точности расчёта. Оба варианта записи алгоритма выглядят следующим образом (рис. 1.8):



1. Ввести x .
2. Присвоить $z = 1$.
3. Присвоить $i = 0$.
4. Присвоить $z = (z + x/z)/2$.
5. Присвоить $i = i + 1$.
6. Если $i < 6$, то перейти к шагу 4, иначе напечатать значение z .

Рис. 1.8. Варианты записи алгоритма: а) графический; б) на естественном языке

Следует заметить, что приведённый алгоритм предназначен для решения узкого класса задач – квадратных уравнений с «хорошими» коэффициентами. Если допустить, что коэффициенты могут принимать произвольные вещественные значения, есть опасность, что при определённых значениях коэффициента (например, $a = 0$) возникает аварийная ситуация (деление на ноль). Качественный алгоритм и качественная программа должны быть устойчивыми, то есть при любых входных параметрах завершение работы программы должно быть нормальным, хотя, возможно, и сопровождаться предупреждающим сообщением о некорректности входных данных.

1.12. Функциональная и структурная организация компьютера

Персональным компьютером (ПК, РС – от англ. Personal Computer) называют небольшую ЭВМ, ориентированную на широкий круг [2,9,10]. До появления персональных компьютеров инженеры, ученые, экономисты, представители других профессий общались с ЭВМ только с помощью посредников – инженеров-системотехников и программистов, поскольку работа на ЭВМ старых типов требовала специальной подготовки. С появлением персональных ЭВМ необходимость в таком посредничестве отпала, так как процесс общения с ЭВМ значительно упростился.

Впервые производство персональных компьютеров было поставлено на поток в 1975 году американской фирмой APPLE. В 1981 г. появились первые персональные компьютеры фирмы IBM. Они были более дешевыми и в них были использованы последние разработки сразу нескольких других фирм, в частности программное обеспечение фирмы MICROSOFT (произносится "Майкрософт"). В настоящее время большой популярностью пользуется продукция не только IBM, но и других фирм производящих IBM совместимые компьютеры, а также компьютеры фирмы APPLE (им присвоили имя "Мэкинтош").

В вычислительной технике архитектура определяет состав, назначение, логическую организацию и порядок взаимодействия всех аппаратных и программных средств, объединенных в единую вычислительную систему. Иными словами, архитектура описывает то, как ЭВМ представляется пользователю.

В современных персональных компьютерах используется принцип открытой архитектуры. Он состоит в том, что основные устройства, непосредственно участвующие в обработке информации (процессор, сопроцессор, оперативная память), соединяется с остальными (периферийными) устройствами единой магистралью – системной шиной (магистральным интерфейсом). При этом часть структурных элементов объединена (конструктивно) системной (материнской или основной – англ. motherboard или mainboard) платой.

В настоящее время распространено два способа построения вычислительных машин: *с непосредственными связями* и *на основе шины*.

Представителем первого способа служит фон-неймановская ВМ (рис. 1.9).

Сущность фон-неймановской концепции вычислительной машины сводится к четырем принципам:

- двоичного кодирования;
- программного управления;
- однородности памяти;
- адресности.

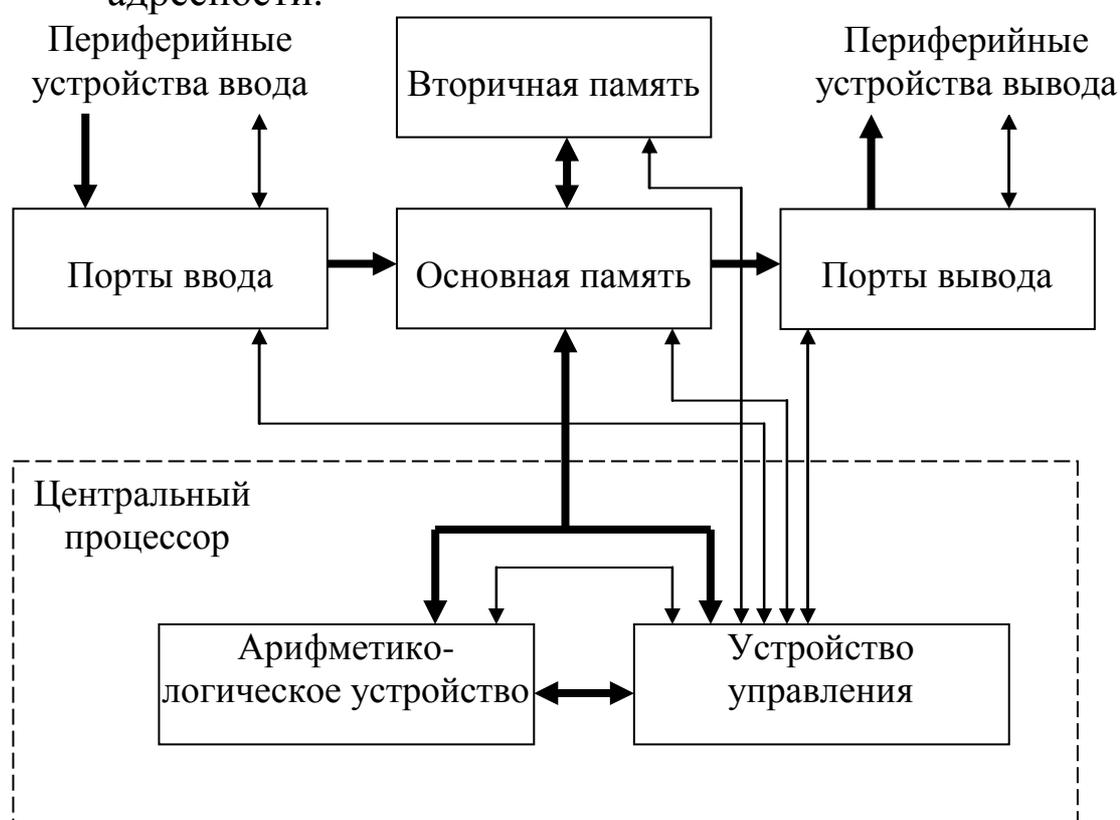


Рис. 1.9. Фон-неймановская организация компьютера

Большинство ЭВМ по своей структуре отвечают принципу программного управления и содержат: память, устройство управления, арифметико-логическое устройство и устройство ввода/вывода.

1.12.1. Принцип двоичного кодирования

Согласно этому принципу, вся информация, как данные, так и команды, кодируются двоичными цифрами 0 и 1. Каждый тип информации представляется двоичной последовательностью и

имеет свой *формат*. Последовательность битов в формате, имеет определенный смысл и называется *полем*. В числовой информации обычно выделяют *поле знака* и *поле значащих разрядов*. В формате команды можно выделить два поля (рис. 1.10): *поле кода операции* (КОП) и *поле адресов* (адресную часть — АЧ).

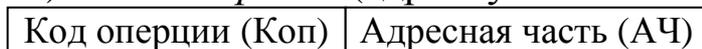


Рис. 1.10. Типовой формат команды

Код операции представляет собой указание, какая операция должна быть выполнена, и задается с помощью r -разрядной двоичной комбинации.

Вид адресной части и число составляющих ее адресов зависят от типа команды: в командах преобразования данных АЧ содержит адреса объектов обработки (*операндов*) и результата; в командах изменения порядка вычислений — адрес следующей команды программы; в командах ввода/вывода — номер устройства ввода/вывода. Адресная часть также представляется двоичной последовательностью, длину которой обозначим через p . Таким образом, команда в вычислительной машине имеет вид $(r+p)$ -разрядной двоичной комбинации.

1.12.2. Принцип программного управления

Все вычисления, предусмотренные алгоритмом решения задачи, должны быть представлены в виде *программы*, состоящей из последовательности управляющих слов — *команд*. Команды программы хранятся в последовательных ячейках памяти вычислительной машины и выполняются *в естественной последовательности*, то есть в порядке их положения в программе.

1.12.3. Принцип однородности памяти

Команды и данные хранятся в одной и той же памяти. Циклически изменяя адресную часть команды обеспечивается обращение к последовательным элементам массива данных. Такой прием носит название *модификации команд*. Другой подход — принцип однородности, когда команды одной программы могут быть получены как результат исполнения другой программы. Эта возможность лежит в основе *трансляции* — перевода текста программы с языка высокого уровня на язык конкретной ВМ.

1.13. Принцип адресности

Структурно основная память состоит из пронумерованных ячеек, причем процессору в произвольный момент доступна любая ячейка. Двоичные коды команд и данных разделяются на единицы информации, называемые *словами*, и хранятся в ячейках памяти, а для доступа к ним используются номера соответствующих ячеек – *адреса*.

Команды и данные должны располагаться в *основной памяти* (ОП), организованной так, что каждое двоичное слово хранится в отдельной ячейке, определяемой адресом, причем соседние ячейки памяти имеют следующие по порядку адреса. Доступ к любым ячейкам запоминающего устройства (ЗУ) основной памяти может производиться в произвольной последовательности. Такой вид памяти известен как *память с произвольным доступом*. При необходимости, в состав ОП включают *постоянные запоминающие устройства* (ПЗУ), также обеспечивающие произвольный доступ.

Устройство управления (УУ) – важнейшая часть ВМ, организующая автоматическое выполнение программ (путем реализации функций управления) и обеспечивающая функционирование ВМ как единой системы. Для пояснения функций УУ ВМ следует рассматривать как совокупность элементов, между которыми происходит пересылка информации, в ходе которой эта информация может подвергаться определенным видам обработки. Пересылка информации между любыми элементами ВМ инициируется своим *сигналом управления* (СУ), то есть управление вычислительным процессом сводится к выдаче нужного набора СУ в нужной временной последовательности. Основной функцией УУ является формирование управляющих сигналов, отвечающих за извлечение команд из памяти в порядке, определяемом программой, и последующее исполнение этих команд. Кроме того, УУ формирует СУ для синхронизации и координации внутренних и внешних устройств ВМ.

Еще одной частью ВМ является *арифметико-логическое устройство* (АЛУ). АЛУ обеспечивает арифметическую и логическую обработку двух входных переменных, в результате чего формируется выходная переменная. Функции АЛУ обычно сво-

дятся к простым арифметическим и логическим операциям, а также операциям сдвига. Помимо результата операции, АЛУ формирует ряд *признаков результата* (флагов), характеризующих полученный результат и события, произошедшие в процессе его получения (равенство нулю, знак, четность, перенос, переполнение и т. д.). Флаги могут анализироваться в УУ с целью принятия решения о дальнейшей последовательности выполнения команд программы.

УУ и АЛУ обычно рассматривают как единое устройство, известное как *центральный процессор* (ЦП) или просто *процессор*. Помимо УУ и АЛУ в процессор входит также *набор регистров общего назначения* (РОН).

Второй способ организации носит название *организации с общей шиной*. Здесь все устройства вычислительной машины подключены к магистральной шине, служащей единственным трактом для потоков команд, данных и управления (рис. 1.11).

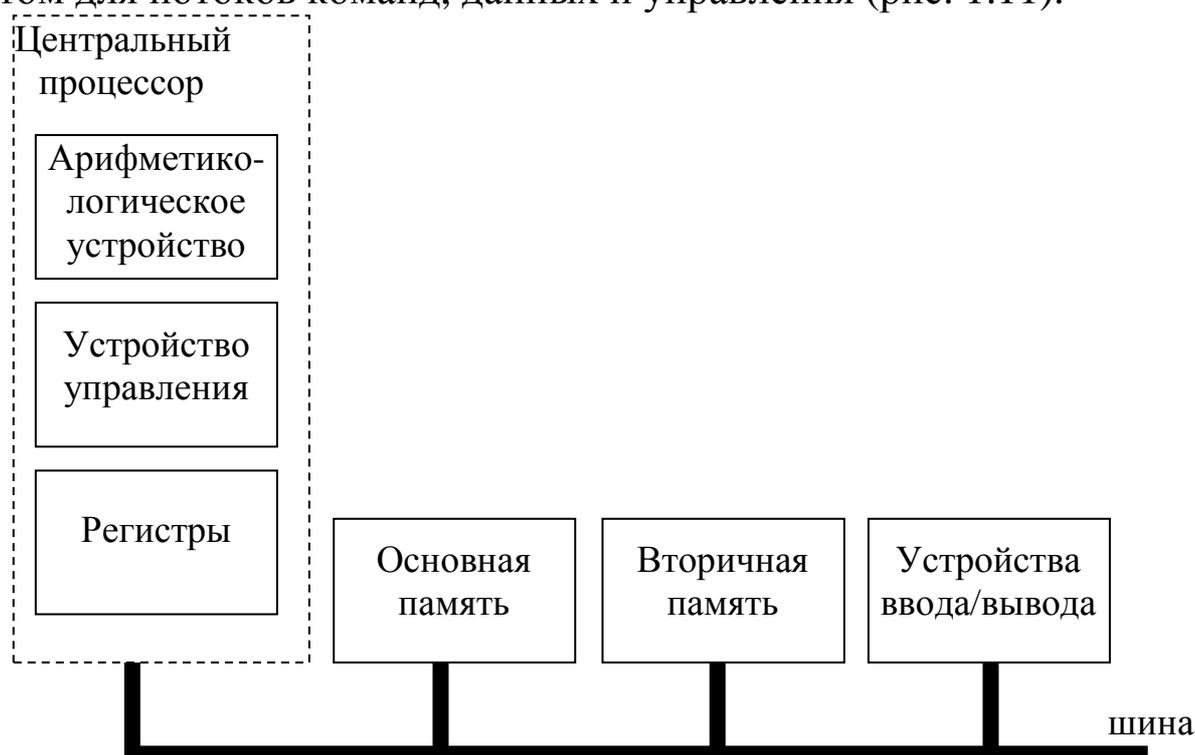


Рис. 1.11. Организация компьютера с общей шиной

Наличие общей шины упрощает реализацию ВМ, позволяет легко менять состав и конфигурацию машины. Благодаря этим свойствам шинная архитектура получила широкое распространение в мини и микроЭВМ. С шиной связан и основной недостаток архитектуры: в каждый момент передавать информацию по

шине может только одно устройство. Основную нагрузку на шину создают обмены между процессором и памятью, связанные с извлечением из памяти команд и данных и записью в память результатов вычислений. На операции ввода/вывода остается лишь часть пропускной способности шины.

1.14. Технические характеристики персонального компьютера

Для оценки возможностей вычислительной машины необходимо знать ее технические характеристики. Наиболее характерные из них: *быстродействие, производительность, тактовая частота*. Единицами измерения быстродействия служат [2,9]:

МИПС (MIPS – Mega Instruction Per Second) – миллион операций над числами с фиксированной запятой (точкой);

МФЛОПС (MFLOPS – Mega FLoating Operations Per Second) – миллион операций над числами с плавающей запятой (точкой);

КОПС (KOPS – Kilo Operations Per Second) для низкопроизводительных ЭВМ – тысяча неких усредненных операций над числами;

ГФЛОПС (GFLOPS – Giga FLoating Operations Per Second) – миллиард операций в секунду над числами с плавающей запятой (точкой).

Оценка производительности ЭВМ всегда приблизительная, ибо при этом ориентируются на некоторые усредненные или, наоборот, на конкретные виды операций. Реально при решении различных задач используются и различные наборы операций. Поэтому для характеристики производительности ПК вместо производительности обычно указывают тактовую частоту, более общую характеристику, определяющую быстродействие машины, так как каждая операция требует для выполнения вполне определенного количества тактов. Такт – это интервал времени, затрачиваемый на выполнение одной простейшей машинной операции. Следовательно, *тактовая частота* – это количество тактов в секунду.

Один такт в секунду равен одному *Герцу*. Современные компьютеры работают на тактовых частотах в несколько сотен МегаГерц, т.е. выполняют несколько десятков или сотен миллионов простейших машинных операций за одну секунду.

Зная тактовую частоту, можно достаточно точно определить время выполнения любой машинной операции.

Разрядность машины и кодовых шин интерфейса. Разрядность – это максимальное количество разрядов двоичного числа, над которым одновременно за один такт может выполняться машинная операция, в том числе и операция передачи информации; чем больше разрядность, тем, при прочих равных условиях, будет больше и производительность ПК.

Типы системного и локальных интерфейсов. Разные типы интерфейсов обеспечивают разные скорости передачи информации между узлами машины, позволяют подключать разное количество внешних устройств и различные их виды.

Тип процессора. Компьютер на базе процессора более современного типа будет при всех прочих равных условиях производительнее, чем машины на базе процессоров старых типов.

Емкость оперативной памяти. В оперативной памяти хранится обрабатываемая в данный момент информация. Ее объем должен быть достаточным для этого. Если это не так, соответствующие программы не смогут быть запущены на данной машине. Поэтому при описании программ всегда указывают, какой должен быть объем оперативной памяти, чтобы можно было запустить данную программу. В настоящее время объем оперативной памяти достигает нескольких сотен Мегабайт.

Виды и емкость КЭШ-памяти. КЭШ-память – это буферная, не доступная для пользователя быстродействующая память, автоматически используемая компьютером для ускорения операций с информацией, хранящейся в более медленно действующих запоминающих устройствах. Например, для ускорения операций с основной памятью организуется регистровая КЭШ-память внутри микропроцессора (КЭШ-память первого уровня) или вне микропроцессора на материнской плате (КЭШ-память второго уровня); для ускорения операций с дисковой памятью организуется КЭШ-память на ячейках электронной памяти.

Емкость накопителя на жестких магнитных дисках (винчестера). Емкость винчестера измеряется обычно в мегабайтах или гигабайтах (1Гбайт=1024 Мбайта). Современное программное обеспечение и решаемые пользователем задачи предъявляют все более высокие требования не только к объему оперативной памяти, но и объему винчестера. Если еще совсем недавно объем винчестера составлял 500 Мбайт – 4 Гбайт, то теперь никого не удивляют 80 – 360 Гбайт.

Характеристики периферийных устройств. Почти все предыдущие характеристики касались устройств, находящихся на материнской плате. К характеристикам периферийных устройств относятся емкость жесткого диска (приводилась выше), число и типы дисководов для дискет, тип дисплея и объем видеопамяти, тип и скорость печати принтера, быстродействие модема и т.д.

Несомненно, что при определенных условиях эксплуатации имеют значение такие технические характеристики как *габариты, масса и надежность*. Габариты и массу чаще учитывают, когда речь идет о персональных компьютерах в мобильном исполнении (ноутбуках). Повышенная надежность и защищенность важна, например, в случае использования компьютера вне дома и офиса (на улице, в цехах, в автомобиле и т.д.). Обычно стоимость таких компьютеров значительно выше по сравнению с обычными.

1.15. Сетевые технологии обработки данных

1.15.1. Компьютерные информационные сети

Компьютерная сеть представляет собой совокупность компьютеров, объединенных средствами передачи данных. Средства передачи данных в общем случае могут состоять из следующих элементов: связанных компьютеров, каналов связи (спутниковых, телефонных, цифровых, волоконнооптических, радио- и других), коммутирующей аппаратуры, ретрансляторов, различного рода преобразователей сигналов и других элементов и устройств [10,17,22].

Архитектура сети ЭВМ определяет принципы построения и функционирования аппаратного и программного обеспечения элементов сети.

Современные сети можно классифицировать по различным признакам: по удаленности компьютеров, топологии, назначению, перечню предоставляемых услуг, принципам управления (централизованные и децентрализованные), методам коммутации (без коммутации, телефонная коммутация, коммутация цепей, сообщений, пакетов и дейтаграмм и т. д.), видам среды передачи и т. д.

Вычислительные сети, состоящие из программно-совместимых ЭВМ, являются или *гомогенными*. Если ЭВМ, входящие в сеть, программно несовместимы, то такая сеть называется *неоднородной* или *гетерогенной*.

По типу организации передачи данных различают сети: с *коммутацией каналов*, с *коммутацией сообщений*, с *коммутацией пакетов*. Имеются сети, использующие смешанные системы передачи данных.

По способу управления вычислительные сети классифицируются на следующие:

- *сети с децентрализованным управлением;*
- *сети с централизованным управлением;*
- *сети со смешанным управлением.*

В первом случае каждая ЭВМ, входящая в состав сети, включает полный набор программных средств для координации выполняемых сетевых операций. Сети такого типа сложны и достаточно дороги. Так как операционные системы отдельных ЭВМ разрабатываются с ориентацией на коллективный доступ к общему полю памяти сети. При этом в каждый конкретный момент времени доступ к общему полю памяти предоставляется только для одной ЭВМ. А координация работы ЭВМ осуществляется под управлением единой операционной системы сети.

В условиях смешанных сетей под централизованным управлением ведется решение задач, обладающих высшим приоритетом и, как правило, связанных с обработкой больших объемов информации.

По структуре построения (топологии) сети подразделяются на классы:

- *одноузловые;*
- *многоузловые;*
- *одноканальные;*
- *многоканальные.*

Все известные компьютерные сети по организационному признаку и предоставляемому пользователю множеству возможностей для использования информационных ресурсов можно классифицировать следующим образом:

- локальные вычислительные сети;
- сеть Internet (Интернет);
- корпоративные сети Intranet (Интранет);
- сети электронных досок объявлений (сети BBS);
- компьютерные сети на основе FTN-технологий.

В рамках приведенной классификации существуют, создаются и развиваются сети, ориентированные на научную, учебную и учебно-научную проблематику.

Сети делятся на общественные, частные и коммерческие. Для физического уровня определены следующие классы общественных сетей:

- до 1000 км – средней длины;
- до 10 000 км – длинные;
- до 25 000 км – самые длинные наземные;
- до 80 000 км – магистральные через спутник;
- до 160 000 км – магистральные международные через два спутника.

В зависимости от удаленности компьютеров сети условно разделяют на *глобальные*, *региональные* и *локальные*.

Произвольная *глобальная сеть* (GAN-Global Area Network) объединяет абонентов, расположенных в различных странах, на различных континентах. Сеть может включать другие глобальные сети, локальные сети, а также отдельно подключаемые к ней компьютеры (удаленные компьютеры) или отдельно подключаемые устройства ввода-вывода.

Взаимодействие между абонентами в глобальной сети осуществляется на базе телефонных линий связи, радиосвязи и систем спутниковой связи. Глобальная вычислительная сеть позволяет решить проблему объединения мировых информационных ресурсов и организации доступа к этим. Глобальные сети бывают четырех основных видов: городские, региональные, национальные и транснациональные.

В качестве устройств ввода-вывода в сети могут использоваться, например, печатающие и копирующие устройства, кассовые и банковские аппараты, дисплеи (терминалы) и факсы, причем они могут быть удалены друг от друга на значительное расстояние.

Региональная вычислительная сеть (MAN-Metropolitan Area Network) связывает абонентов, расположенных на значительном расстоянии друг от друга. Она может включать абонентов внутри большого города, экономического региона, отдельной страны. Обычно расстояние между абонентами составляет десятки, сотни километров.

Локальные вычислительные сети (ЛВС), Local Area Network (LAN), объединяют абонентов, расположенных в пределах небольшой территории. В настоящее время не существует четких ограничений на территориальный разброс абонентов ЛВС. Компьютеры в ЛВС могут быть расположены на расстоянии до нескольких километров и обычно соединены при помощи скоростных линий связи со скоростью обмена от 1 до 10 и более Мбит/с.

ЛВС обычно развертываются в рамках некоторой организации (корпорации, учреждения). Поэтому их иногда называют *корпоративными системами* или *сетями*. Компьютеры при этом, как правило, находятся в пределах одного помещения, здания или соседних зданий. Объединение глобальных, региональных и локальных вычислительных сетей позволяет создавать многосетевые иерархии, обеспечивая доступ к мировым информационным ресурсам.

Таким образом, для того, чтобы создать компьютерную сеть, нужны компьютеры, линии связи, а также специальные устройства для подключения компьютеров к линиям связи. На-

конец, необходимо установить специальное программное обеспечение для управления совместной работы в сети.

1.15.2. Локальные вычислительные сети

С появлением микроЭВМ и персональных ЭВМ возникли локальные вычислительные сети. Они позволили поднять на качественно новую ступень управление производственным объектом, повысить эффективность использования ЭВМ, улучшить качество обрабатываемой информации, реализовать безбумажную технологию, создать новые технологии.

Локальная вычислительная сеть – компьютерная сеть для ограниченного круга пользователей, объединяющая компьютеры в одном помещении или в рамках одного предприятия.

Локальная вычислительная сеть (ЛВС) – это совокупность технических средств (компьютеров, кабелей, сетевых адаптеров и др.), работающих под управлением сетевой операционной системы и прикладного программного обеспечения.

Локальные сети получили широкое распространение, начиная с 80-х годов. Локальная компьютерная сеть позволяет легко обмениваться информацией внутри отдельной организации.

По назначению (характеру реализуемых функций) ЛВС их можно разделить на следующие категории:

- *вычислительные*, выполняющие преимущественно расчетные работы;
- *информационно-вычислительные*, кроме расчетных выполняющие работу по информационному обслуживанию пользователей;
- *информационные*, выполняющие в основном информационное обслуживание пользователей (создание и оформление документов, доставку пользователю директивной, текущей, справочной и другой нужной ему информации);
- *информационно-поисковые* – разновидность информационных, специализирующуюся на поиске информации в сетевых хранилищах по нужной пользователю тематике;
- *информационно-советующие*, обрабатывающие текущую организационную, техническую и технологическую информа-

цию и вырабатывающие результирующую информацию для поддержки принятия пользователем правильных решений;

- *информационно-управляющие*, обрабатывающие текущую техническую и технологическую информацию и вырабатывающие результирующую информацию, на базе которой автоматически вырабатываются воздействия на управляемую систему и т. д.

По количеству подключенных к сети компьютеров сети можно разделить на малые, объединяющие до 10-15 машин, средние – до 50 машин и большие – свыше 50 машин.

По территориальной расположенности ЛВС делятся на компактно размещенные (все компьютеры расположены в одном помещении) и распределенные (компьютеры сети размещены в разных помещениях).

По пропускной способности ЛВС делятся на три группы:

- ЛВС с малой пропускной способностью (скорости передачи данных в пределах до десятка мегабит в секунду), использующие чаще всего в качестве каналов связи тонкий коаксиальный кабель или витую пару;

- ЛВС со средней пропускной способностью (скорости передачи данных несколько десятков мегабит в секунду), использующие чаще всего в качестве каналов связи толстый коаксиальный кабель или экранированную витую пару;

- ЛВС с большой пропускной способностью (скорости передачи данных сотни и даже тысячи мегабит в секунду), использующие чаще всего в качестве каналов связи волоконно-оптические кабели.

Объединение компьютеров в ЛВС обеспечивает решение задач коллективной работы с информацией.

1. *Разделение файлов.* ЛВС позволяет многим пользователям одновременно работать с одним файлом, хранящимся на центральном файл-сервере. Например, на предприятии или фирме несколько сотрудников могут одновременно использовать одни и те же руководящие документы.

2. *Передача файлов.* ЛВС позволяет быстро и надежно копировать файлы любого размера с одной машины на другую.

3. *Доступ к информации и файлам.* ЛВС позволяет запускать прикладные программы с любой из рабочих станций, где бы она ни была расположена.

4. *Разделение прикладных программ и баз данных.* ЛВС позволяет двум пользователям использовать одну и ту же копию программы. При этом, конечно, они не могут одновременно редактировать один и тот же документ или запись в базе данных.

5. *Одновременный ввод данных в прикладные программы.* Сетевые прикладные программы позволяют нескольким пользователям одновременно вводить данные, необходимые для работы этих программ. Например, вести записи в базе данных так, что они не будут мешать друг другу. Однако только специальные сетевые версии программ позволяют одновременный ввод информации. Обычные компьютерные программы позволяют работать с набором файлов только одному пользователю.

6. *Разделение принтера или другого технического устройства.* ЛВС позволяет нескольким пользователям на различных рабочих станциях совместно использовать один или несколько дорогостоящих лазерных принтеров или других устройств.

7. *Электронная почта.* Пользователь может использовать ЛВС как почтовую службу и рассылать служебные записки, доклады, сообщения и т.п. другим пользователям. В отличие от телефона электронная почта передаст ваше сообщение даже в том случае, если в данный момент абонент (группа абонентов) отсутствует на своем рабочем месте, причем для этого ей не потребуются бумаги.

Топология вычислительной сети во многом определяется структурой сети связи, т.е. способом соединения абонентов друг с другом и ЭВМ [4]. По топологическим признакам ЛВС делятся на сети следующих типов: с общей шиной, кольцевые, иерархические, радиальные и многосвязные рис. 1.12.

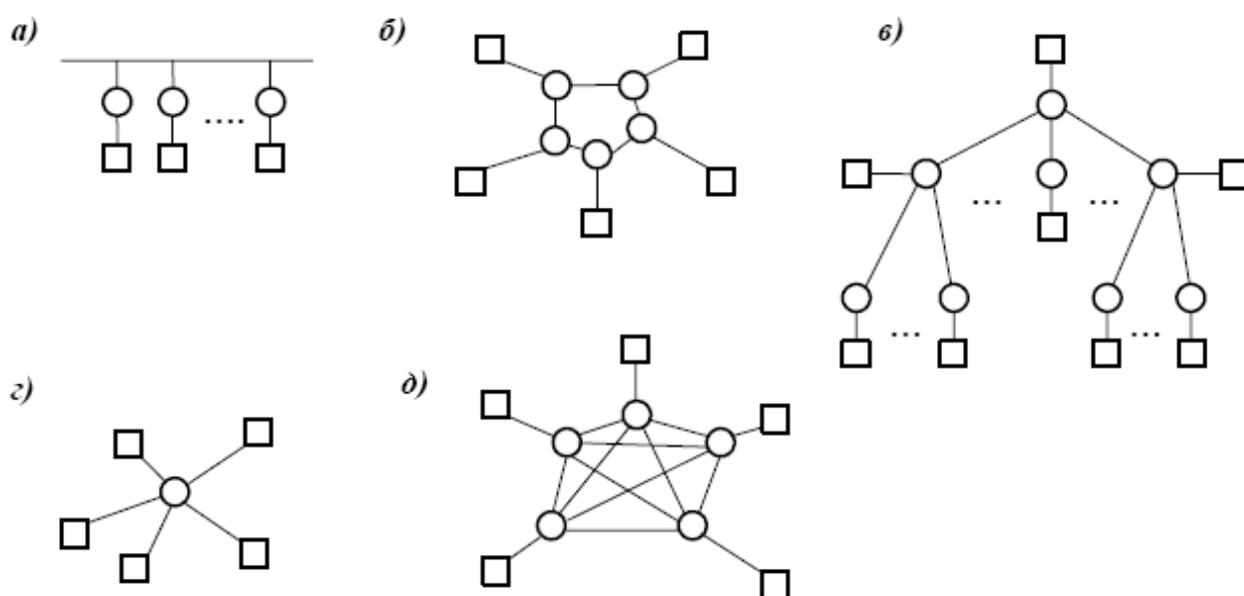


Рис. 1.12. Типы структур компьютерных сетей: а) общая шина; б) иерархическая структура; в) радиальная (звезда); д) многозвенная

На рис. 1.12 квадратом показана абонентская ЭВМ, а кружком – узел коммутации. Топология вычислительной сети в ЛВС с общей шиной (рис. 1.12а) характеризуется тем, что одна из машин служит в качестве системного обслуживающего устройства, обеспечивающего централизованный доступ к общим файлам и базам данных, печатающим устройствам и другим вычислительным ресурсам.

Сети данного типа приобрели большую популярность благодаря низкой стоимости, высокой гибкости и скорости передачи данных, легкости расширения сети (подключение новых абонентов к сети не сказывается на ее основных характеристиках). К недостаткам шинной топологии следует отнести необходимость использования довольно сложных протоколов и уязвимость в отношении физических повреждений кабеля.

Кольцевая топология (рис. 1.12б) в сети отличается тем, что информация по кольцу может передаваться только в одном направлении и все подключенные ПЭВМ могут участвовать в ее приеме и передаче. При этом абонент-получатель должен помечать полученную информацию специальным маркером, иначе могут появиться «заблудившиеся» данные, мешающие нормальной работе сети.

Как последовательная конфигурация кольцо особенно уязвимо в отношении отказов: выход из строя какого-либо сегмента кабеля приводит к прекращению обслуживания всех пользователей. Разработчики ЛВС приложили немало усилий, чтобы справиться с этой проблемой. Защита от повреждений или отказов обеспечивается либо замыканием кольца на обратный (дублирующий) путь, либо переключением на запасное кольцо. И в том, и в другом случае сохраняется общая кольцевая топология.

Иерархическая ЛВС (конфигурация типа «дерево») представляет собой более развитый вариант структуры ЛВС, построенной на основе общей шины (рис. 1.12в). Дерево образуется путем соединения нескольких шин с корневой системой, где размещаются самые важные компоненты ЛВС. Оно обладает необходимой гибкостью для того, чтобы охватить средствами ЛВС несколько этажей в здании или несколько зданий на одной территории, и реализуется, как правило, в сложных системах, насчитывающих десятки и даже сотни абонентов.

Радиальную (звездообразную) конфигурацию (рис. 1.12г) можно рассматривать как дальнейшее развитие структуры «дерево с корнем» с ответвлением к каждому подключенному устройству. В центре сети обычно размещается коммутирующее устройство, обеспечивающее жизнеспособность системы. ЛВС подобной конфигурации находят наиболее частое применение в автоматизированных учрежденческих системах управления, использующих центральную базу данных. Звездообразные ЛВС, как правило, менее надежны, чем сети с общей шиной или иерархические, но эта проблема решается дублированием аппаратуры центрального узла. К недостаткам можно также отнести значительное потребление кабеля (иногда в несколько раз превышающее расход в аналогичных по возможностям ЛВС с общей шиной или иерархических).

Наиболее сложной и дорогой является *многосвязная топология* (рис. 1.12д), в которой каждый узел связан со всеми другими узлами сети. Эта топология в ЛВС применяется очень редко, в основном там, где требуются исключительно высокие надежность сети и скорость передачи данных.

На практике чаще встречаются *гибридные* ЛВС, приспособленные к требованиям конкретного заказчика и сочетающие фрагменты шинной, звездообразной и других топологий.

Основными аппаратными компонентами ЛВС являются:

- рабочие станции;
- серверы;
- интерфейсные платы;
- кабели.

Рабочие станции (РС) – это, как правило, персональные ЭВМ, которые являются рабочими местами пользователей сети.

Требования, предъявляемые к составу рабочих станций, определяются характеристиками решаемых в сети задач, принципами организации вычислительного процесса, используемой операционной системой и некоторыми другими факторами.

Иногда в рабочей станции, непосредственно подключенной к сетевому кабелю, могут отсутствовать накопители на магнитных дисках. Такие рабочие станции называют *бездисковыми рабочими станциями*.

Основным *преимуществом* бездисковых РС является низкая стоимость, а также высокая защищенность от несанкционированного проникновения в систему пользователей и компьютерных вирусов. *Недостаток* бездисковой РС заключается в невозможности работать в автономном режиме (без подключения к серверу), а также иметь свои собственные архивы данных и программ.

Серверы в ЛВС выполняют функции распределения сетевых ресурсов. Обычно его функции возлагают на достаточно мощный ПК, мини-ЭВМ, большую ЭВМ или специальную ЭВМ-сервер. В одной сети может быть один или несколько серверов. Каждый из серверов может быть отдельным или совмещенным с РС. В последнем случае не все, а только часть ресурсов сервера оказывается общедоступной.

При наличии в ЛВС нескольких серверов каждый из них управляет работой подключенных к нему рабочих станций. Совокупность компьютеров сервера и относящихся к нему рабочих станций часто называют доменом. Иногда в одном домене находится несколько серверов. Обычно один из них является глав-

ным, а другие – выполняют роль резерва (на случай отказа главного сервера) или логического расширения основного сервера.

Существует два основных принципа управления в локальных сетях: централизация и децентрализация.

Согласно этим принципам локальные сети бывают:

- одноранговые сети;
- сети с выделенным сервером (файл-сервером).

Одноранговые сети не предусматривают выделение специальных компьютеров, организующих работу сети. Каждый пользователь, подключаясь к сети, выделяет в сеть какие-либо ресурсы (дисковое пространство, принтеры) и подключается к ресурсам, предоставленным в сеть другими пользователями. Такие сети просты в установке, наладке, они существенно дешевле сетей с выделенным сервером.

В свою очередь, *сети с выделенным сервером*, несмотря на сложность настройки и относительную дороговизну, позволяют осуществлять централизованное управление. В данном случае все компьютеры, кроме сервера, называются рабочими станциями.

Сервер – компьютер, выделенный для совместного использования участниками сети, поставляющий ресурсы и услуги.

Клиент – компьютер, использующий ресурсы и услуги сервера.

Каждый компьютер сети имеет уникальное сетевое имя. Каждому пользователю серверной сети необходимо согласовать с администратором сети свое сетевое имя и сетевой пароль.

Следует заметить, что в серверной сети на компьютеры с разными ролями устанавливают различные операционные системы. Так, на сервер устанавливают одну из серверных операционных систем. В качестве примера можно указать Windows NT Server. На компьютеры-клиенты можно устанавливать любую операционную систему, содержащую средства для выполнения роли клиента серверной сети, например, Windows 95/98.

Каждый компьютер сети имеет уникальное сетевое имя, позволяющее однозначно его идентифицировать. Для каждого пользователя серверной сети необходимо иметь свое сетевое имя

и сетевой пароль. Имена компьютеров, сетевые имена и пароли пользователей прописываются на сервере.

Для удобства управления локальной компьютерной сетью, несколько компьютеров, имеющих равные права доступа, объединяют в рабочие группы.

Совокупность приемов разделения и ограничения прав доступа участников компьютерной сети к ресурсам называется *политикой сети*.

Обеспечением работоспособности сети и ее администрированием занимается *системный администратор* – человек, управляющий организацией работы локальной сети.

Рабочая группа – группа компьютеров в локальной сети.

Политика сети – совокупность приемов разделения и ограничения прав доступа участников компьютерной сети к ресурсам.

Системный администратор – человек, управляющий организацией работы локальной сети.

1.15.3. Способы коммутации и передачи данных

Основная функция систем передачи данных в условиях функционирования вычислительных сетей заключается в организации быстрой и надежной передачи информации произвольным абонентам сети, а также в сокращении затрат на передачу данных.

Важнейшая характеристика сетей передачи данных - время доставки информации – зависит от структуры сети передачи данных, пропускной способности линий связи, а также от способа соединения каналов связи между взаимодействующими абонентами сети и способа передачи данных по этим каналам. В настоящее время различают системы передачи данных с постоянным включением каналов связи (некоммутируемые каналы связи) и коммутацией на время передачи информации по этим каналам.

При использовании *некоммутируемых каналов связи* средства приема-передачи абонентских пунктов и ЭВМ постоянно соединены между собой, т.е. находятся в режиме «*online*». В этом случае отсутствуют потери времени на коммутацию, обес-

печиваются высокая степень готовности системы к передаче информации, более высокая надежность каналов связи и, как следствие, достоверность передачи информации. Недостатками такого способа организации связи являются низкий коэффициент использования аппаратуры передачи данных и линий связи, высокие расходы на эксплуатацию сети. Рентабельность подобных сетей достигается только при условии достаточно полной загрузки этих каналов.

При коммутации абонентских пунктов и ЭВМ только на время передачи информации (т.е. нормальным режимом для которых является режим «*off-line*») принцип построения узла коммутации определяется способами организации прохождения информации в сетях передачи данных.

Существуют три основных способа подготовки и передачи информации в сетях, основанных на коммутации:

- каналов;
- сообщений;
- пакетов.

Коммутация каналов

Способ *коммутации каналов* заключается в установлении физического канала связи для передачи данных непосредственно между абонентами сети (Рис. 1.13). При использовании коммутируемых каналов тракт (путь) передачи данных образуется из самих каналов связи и устройств коммутации, расположенных в узлах связи.

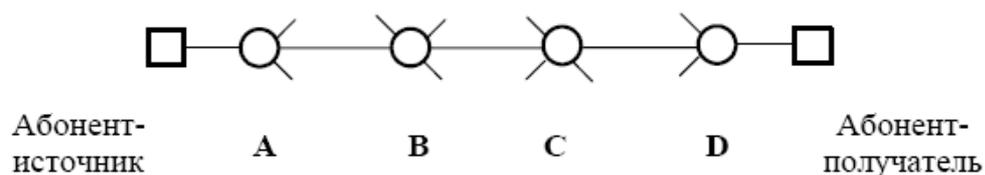


Рис. 1.13. Способ коммутации каналов

Установление соединения заключается в том, что абонент посылает в канал связи заданный набор символов, прохождение которых по сети через соответствующие узлы коммутации вызывает установку нужного соединения с вызываемым абонентом. Этот транзитный канал образуется в начале сеанса связи, остается фиксированным на период передачи всей информации и разрывается только после завершения передачи информации.

Такой способ соединения используется в основном в сетях, где требуется обеспечить непрерывность передачи сообщений (например, при использовании телефонных каналов связи и абонентского телеграфа). В этом случае связь абонентов возможна только при условии использования ими однотипной аппаратуры, одинаковых каналов связи, а также единых кодов.

К достоинствам данного способа организации соединения абонентов сети следует отнести:

- гибкость системы соединения в зависимости от изменения потребностей;
- высокую экономичность использования каналов, достигаемую за счет их эксплуатации только в течение времени установления связи и непосредственно передачи данных;
- невысокие расходы на эксплуатацию каналов связи (на порядок меньше, чем при эксплуатации некоммутируемых линий связи).

Способ коммутации каналов более оперативный, так как позволяет вести непрерывный двусторонний обмен информацией между двумя абонентами.

Недостатками коммутируемых каналов связи является необходимость использования специальных и коммутирующих устройств, которые снижают скорость передачи данных и достоверность передаваемой информации.

Использование специальных методов и средств, обеспечивающих повышение достоверности передачи информации в сети, влечет за собой снижение скорости передачи данных за счет увеличения объема передаваемой информации, вызванного необходимостью введения избыточных знаков, за счет потерь времени на кодирование информации в узле-передатчике и декодирование, логический контроль и другие преобразования – в узле-приемнике.

Наконец, сокращение потоков информации ниже пропускной способности аппаратной части и каналов связи ведет к недогрузке канала, а в период пиковой нагрузки может вызвать определенные потери вызовов.

Коммутация сообщений

При *коммутации сообщений* поступающая на узел связи информация передается в память узла связи, после чего анализируется адрес получателя.

В зависимости от занятости требуемого канала сообщение либо передается в память соседнего узла, либо становится в очередь для последующей передачи.

Таким образом, способ коммутации сообщений обеспечивает поэтапный характер передачи информации. В этом случае сообщения содержат адресный признак (заголовок), в соответствии с которым осуществляется автоматическая передача информации в сети от абонента-передатчика к абоненту-приемнику.

Все функции согласования работы отдельных участков сети связи, а также управление передачей сообщений и их соответствующую обработку выполняют центры (узлы) коммутации сообщений.

Основное функциональное назначение центра коммутации сообщений - обеспечить автоматическую передачу информации от абонента к абоненту в соответствии с адресным признаком сообщения и требованиями к качеству и надежности связи.

Метод коммутации сообщений обеспечивает независимость работы отдельных участков сети, что значительно повышает эффективность использования каналов связи при передаче одного и того же объема информации (которая в этом случае может достигать 80 - 90% от максимального значения).

В системе с коммутацией сообщений происходит сглаживание несогласованности в пропускной способности каналов и более эффективно реализуется передача многоадресных сообщений (так как не требуется одновременного освобождения всех каналов между узлом-передатчиком и узлом-приемником). Передача информации может производиться в любое время, так как прямая связь абонентов друг с другом необязательна.

К недостаткам метода следует отнести односторонний характер связи между абонентами сети. Для более полной загрузки каналов и их эффективного использования возможно совместное применение перечисленных методов коммутации, основой которого служат следующие условия:

- использование в одном и том же узле связи аппаратуры для коммутации каналов и для коммутации сообщений (того или иного способа коммутации в узле осуществляется в зависимости от загрузки каналов связи);

- организация сети с коммутацией каналов для узлов верхних уровней иерархии и

- коммутации сообщений для нижних уровней.

Коммутация пакетов

В последние годы появился еще один способ коммутации абонентов сети – так называемая *коммутация пакетов*. Этот способ сочетает в себе ряд преимуществ методов коммутации каналов и коммутации сообщений. При коммутации пакетов перед началом передачи сообщение разбивается на короткие пакеты фиксированной длины, которые затем передаются по сети.

В пункте назначения эти пакеты вновь объединяются в первоначальное сообщение, а так как их длительное хранение в запоминающем устройстве узла связи не предполагается, пакеты передаются от узла к узлу с минимальной задержкой во времени. В этом отношении указанный метод близок методу коммутации каналов.

При коммутации пакетов их фиксированная длина обеспечивает эффективность обработки пакетов, предотвращает блокировку линий связи и значительно уменьшает емкость требуемой промежуточной памяти узлов связи. Кроме того, сокращается время задержки при передаче информации, т.е. скорость передачи информации превышает аналогичную скорость при методе коммутации сообщений.

К недостаткам метода следует отнести односторонний характер связи между абонентами сети.

Различают два основных типа систем связи с коммутацией пакетов:

- в системах *первого типа* устройство коммутации анализирует адрес места назначения каждого принятого пакета и определяет канал, необходимый для передачи информации;

- в системах *второго типа* пакеты рассылаются по всем каналам и терминалам, каждый канал (терминал), в свою очередь, проанализировав адрес места назначения пакета и сравнив его с

собственным, осуществляет прием и дальнейшую передачу (обработку) пакета либо игнорирует его.

Первый тип систем коммутации пакетов характерен для глобальных сетей с огромным числом каналов связи и терминалов, второй тип применим для сравнительно замкнутых сетей с небольшим числом абонентов.

1.16. Устройства хранения информации

Различают устройства хранения информации, реализованные в виде электронных схем, и накопители информации, при помощи которых данные записываются на какой-либо носитель, например магнитный или оптический (ранее использовались даже бумажные носители – перфокарты и перфоленты).

Устройства, представляющие собой электронные схемы, отличаются небольшим временем доступа к данным, но не позволяют хранить большие объемы информации. К таким устройствам относятся постоянное запоминающее устройство (ПЗУ, ROM), оперативное запоминающее устройство (оперативная память, ОЗУ, RAM).

Накопители информации на гибких и жестких магнитных дисках (винчестеры, Zip, JAZ, ORB), оптических (CD, CD-RW, DVD) и магнитооптических дисках наоборот дают возможность хранить большие объемы информации, но время ее записи и считывания там велико. Поэтому эффективная работа на компьютере возможна только при совместном использовании накопителей информации и устройств хранения, реализованных в виде электронных схем.

Оперативная память предназначена для хранения исполняемых в данный момент программ и необходимых для этого данных. То есть, в ОЗУ хранится информация, с которой ведется работа в данный момент времени. Содержимое оперативной памяти пропадает при выключении питания.

ПЗУ предназначено для хранения неизменяемой информации. В компьютере постоянно должна храниться информация, которая нужна при каждом его включении. Например, в ПЗУ записываются команды, которые компьютер должен выполнить сразу после включения питания для начала работы (например,

BIOS). ПЗУ называют энергонезависимой памятью, так как содержимое ПЗУ при отключении питания сохраняется.

Принцип записи информации на магнитные ленты и диски аналогичен принципу записи звука в магнитофоне. В магнитооптических дисках информация также хранится на магнитном носителе, но чтение и запись осуществляются лучом лазера, что значительно повышает сохранность информации.

Для работы с дискетами предназначено устройство, называемое дисководом, а для работы с компакт-дисками – CD-ROM (Compact Disc Read Only Memory). В настоящее время можно встретить в продаже такие устройства для работы с компакт-дисками, поддерживающие одновременно функции чтения CD, CDR, CDRW, DVD и обеспечивающие запись CDR и CDRW. То есть Вам уже не требуется приобретать два или три отдельных устройства, так как все функции объединены в одном. Такие устройства разрабатывались в первую очередь для использования в ноутбуках (переносных портативных компьютерах), где габариты и вес и функциональность играют далеко не последнюю роль.

1.16.1. Устройства обработки информации и управляющие устройства

Основным устройством обработки информации в ЭВМ является арифметико-логическое устройство (АЛУ). АЛУ выполняет все логические операции (логическое умножение – операция "И", логическое сложение – операция "ИЛИ", логическое отрицание – операция "НЕ"), а также арифметические операции над данными, представленными в виде двоичных кодов (нулей и единиц).

Булеву алгебру, в которой используются только два числа – 0 и 1, положил в основу теории электрических и электронных переключательных схем-сумматоров, американский инженер Клод Шеннон в 1938 г. Именно это во многом определило появление ЭВМ, способных автоматически производить арифметические вычисления.

Так как выполнение самых сложных действий в ЭВМ сводится к большому числу простейших арифметических и логиче-

ских операций, производительность процессора при выполнении простейших операций определяет быстродействие ЭВМ.

В ЭВМ арифметико-логическое устройство объединено с управляющими устройствами в единую схему – процессор, представляющий собой микросхему с большим числом контактов. Для ускорения работы процессора при выполнении математических вычислений обычно используется специальное устройство сопроцессор, который в последних моделях конструктивно встроен в микросхему процессора.

Изобретателем микропроцессора как схемы, в которую собрана практически вся основная электроника компьютера, стала американская фирма INTEL, выпустившая в 1970 г. процессор 8008. С их появления и началась история ЭВМ четвертого поколения. По настоящее время фирма INTEL занимает ведущие позиции на мировом рынке в производстве и разработке новых типов процессоров.

1.16.2. Устройства ввода-вывода

Устройства ввода и вывода можно условно разделить на устройства, с помощью которых информация передается машине от человека, человеку от машины и от одной машины другой машине [2,10,17,22].

Наиболее распространенные устройства следующие:

– устройства ввода: клавиатура, сканер, устройства местонаказания (мышь, джойстик, графический планшет, световое перо);

– устройства вывода: дисплей, принтер, плоттер, звуковая карта;

– устройства, обеспечивающие связь с другими машинами: модем, сетевой адаптер, инфракрасный порт и т.д.

Кроме них имеются специальные устройства, обеспечивающие совместную работу ЭВМ с специальной аппаратурой, микрофонами, видеокамерами, видеоманитофонами, научными приборами и т.д.

Клавиатура – основное устройство ввода информации. Расположение латинских букв на ней соответствует расположению клавиш на латинской печатной машинке (клавиатура QWERTY –

по первым буквам в верхнем ряду), русских букв – русской печатной машинке.

Сканер – устройство для ввода графической информации в компьютер. Сканеры бывают ручные, настольные (планшетные), интегрированные в корпус принтера. Ручные сканеры (более дешевые, но обладающие более скромными возможностями) проводят над изображением (чаще всего их используют для считывания штрих-кодов), а в настольные лист бумаги вкладывают целиком. Кроме того, сканеры бывают цветные и черно-белые.

Устройства местоуказания предназначены для ввода координат в компьютер. Мышь – наиболее распространенный манипулятор, позволяющий перемещать указатель (курсор мыши) по экрану дисплея и указывать им на определенные объекты на экране (т.е. вводить в компьютер координаты выбранной точки на экране). Наиболее просты и дешевы механические мыши, в основании которых имеется шарик, вращающийся при перемещении мыши по ровной поверхности. Вращение шарика передается на датчики, вырабатывающие электрические сигналы, отслеживая тем самым движения кисти руки человека, что и приводит к соответствующим перемещениям курсора на экране. Более дорогой и сложной, но более точной и надежной является оптическая мышь, в которой в ее основании вместо шарика используется излучатель света и фотодатчики, считывающие отраженный свет от поверхности стола или коврика мышки.

Трекбол – это своеобразная "мышь вверх ногами". Он представляет собой шарик, как правило, встраиваемый в клавиатуру, который вращают пальцами. Трекбол обычно используют в переносных компьютерах – ноутбуках (англ. notebook – записная книжка).

Джойстик – манипулятор, выполняемый в виде рычажка (ручки) на массивном основании. Управляющие сигналы вырабатываются движениями ручки и нажатием кнопки (или кнопок) на ней. Джойстики, как правило, используют для работы с игровыми программами.

Графический планшет (дигитайзер или диджитайзер – англ. digitizer – оцифровыватель) – планшет, покрытый сеткой пьезоэлементов – элементов, вырабатывающих электрический ток при

механическом воздействии. На нем размещают лист бумаги с изображением и надавливанием на определенные точки на нем вводят их координаты в компьютер. Дигитайзеры, как правило, используются для ввода карт или планов в ЭВМ.

Световым пером также указываются координаты определенной точки, но непосредственно на экране дисплея. На его конце имеется фотоэлемент. Им при поднесении к экрану фиксируется момент попадания на него электронного луча, формирующего изображение (как известно, этот электронный луч несколько раз в секунду обегает все точки поверхности экрана). На основе этого вычисляются координаты точки, к которой поднесено световое перо в данный момент времени.

Дисплей (монитор) – основное устройство вывода информации. Дисплеи бывают основанными на электронно-лучевой трубке или панели на жидких кристаллах (LCD, от англ. Liquid Crystal Display). Кроме того различают цветные и монохромные (одноцветные) дисплеи.

Формирование изображения на экране обеспечивает как электронная начинка монитора, так и специальное устройство – видеоадаптер, который формирует, хранит и передает изображение на экран дисплея. Конструктивно видеоадаптер представляет собой плату, которая вставляется в корпус компьютера (в системный блок). Дисплей подключается непосредственно к ней. На этой плате находятся, в частности, схемы видеопамати, в которых запоминается изображение, выводимое на экран.

Современные дисплеи должны соответствовать очень строгим требованиям, установленным международными нормами – стандартами. Защита человека от разного рода излучений в дисплеях выполняется на более серьезном уровне. Специальные защитные фильтры, навешиваемые на экран, защищают его поверхность от бликов, позволяют несколько увеличить четкость изображения, обеспечивают дополнительную защиту от излучений. Защита от негативных полей обеспечивается лишь при подключении к заземлению как корпуса компьютера, так и защитного экрана. Следует помнить, что излучения имеют место не только со стороны экрана, где конструкторы предусматривают максимально возможную защиту для человека, но и с задней

стороны дисплея, где никакой защиты, как правило, не устраивается. Поэтому размещать компьютер в помещении следует так, чтобы с задней стороны дисплея люди в течение длительного времени не находились.

Дисплей может работать либо в текстовом (когда на экран могут быть выведены только стандартные ASCII – символы), либо в графическом режиме, когда изображения строятся из большого числа точек – пикселей (для работы с чертежами, графикой, текстом, видео и т.д.).

Качество изображения в графическом режиме зависит от разрешающей способности, т.е. количества пикселей по горизонтали и вертикали (640*480, 800*600, 1024*768 и т.д. Разрешающая способность не зависит от размера экрана дисплея.

Существуют различные видеорежимы. Они отличаются разрешающей способностью и палитрой – количеством выводимых цветов. С течением времени и развитием техники появляются новые графические режимы с большей разрешающей способностью и более богатой палитрой. От типа дисплея зависит способность поддерживать различные видеорежимы. Как правило, для дисплеев выполняется правило совместимости "сверху вниз". Это значит, что дисплей более современного типа (дисплеи типа SVGA – 800·600 и 1024·768) может работать как в режимах с высокой разрешающей способностью и большим количеством выводимых цветов, так и в режимах, разработанных для дисплеев старых типов, т.е. с меньшей разрешающей способностью и меньшим количеством цветов (дисплеи типа VGA 640·480). Принтер – устройство вывода информации на бумагу. Принтеры бывают матричные, струйные, лазерные. Иногда встречаются принтеры других типов – литерные, лепестковые, светодиодные и другие. Кроме того, по формату бумаги различают "широкие" и "узкие" принтеры.

В матричном принтере изображение выводится на бумагу с помощью специальной движущейся головки, в которой имеется несколько (9, 24 или 48) иголок, наносящих удары по листу бумаги через красящую ленту. Скорость работы матричных принтеров невысока (от 10 секунд на страницу при низком качестве, до нескольких минут – при высоком), кроме того, они издадут

неприятный звук при работе. К их преимуществам следует отнести низкую стоимость, высокую надежность. В струйных принтерах красящее вещество (тонер) выдувается на бумагу с помощью системы сопел. Эти принтеры обеспечивают более высокие скорость и качество печати, позволяют создавать цветные изображения. При этом по стоимости струйные принтеры незначительно отличаются от матричных. Эксплуатационные расходы (стоимость тонера и обслуживания) у них выше.

Наиболее высокую скорость печати (до 5 секунд на страницу) при наилучшем качестве обеспечивают лазерные принтеры. В них изображение переносится на бумагу со специального барабана, к участкам поверхности которого, наэлектризованного лучом лазера, притягиваются частицы красящего порошка. Лазерные принтеры являются достаточно дорогими, но наиболее экономически выгодными при печати большого количества документов в офисе и дома.

Плоттер (графопостроитель) – устройство для вывода чертежей на бумагу. Бывают струйные и механические плоттеры. Устройство струйных плоттеров аналогично устройству струйных принтеров. В механических плоттерах пишущий узел с перьями (шариковыми, керамическими или фитильными, как во флوماстерах) перемещается по направляющим относительно листа и1074 ватмана, или бумага, зажатая в прижимных устройствах, перемещается относительно пишущего узла.

В корпус компьютера обычно встраивается динамик, способный выдавать звуковые сигналы, например при ошибках загрузки компьютера. Для возможности прослушивания музыки в качественном исполнении, речи, звуковых эффектов необходимо оснастить компьютер звуковой приставкой – специальной платой (саунд-бластером, англ. sound blaster – "выдувающий" звук), вставляемой в системный блок (корпус) компьютера, и подключаемыми к ней колонками. В настоящее практически все производители встраивают звуковую карту непосредственно в материнскую плату (основную плату, на которой расположены процессор память и др.). Мощный компьютер, оснащенный этими и другими устройствами для создания звуковых эффектов называют мультимедийным (от англ. multimedia-"многие среды", т.е.

возможность одновременно использовать всевозможные способы представления информации – текстовой, графической, звуковой, видео и пр.).

Модем (МОдулятор – ДЕМодулятор) – устройство, преобразующее информацию к виду, в котором ее можно передавать по линиям связи (по телефонным линиям). Модемы бывают внутренние (вставляемые в корпус компьютера) и внешние (представляющие собой отдельные устройства, подключаемые к компьютеру и телефонной линии). Кроме того, различают телефонные модемы, позволяющие передавать только текстовые сообщения, и факс-модемы, позволяющие передавать и графические изображения.

Сетевой адаптер (сетевая плата) – устройство, обеспечивающее подключение компьютера к локальной компьютерной сети. Сетевой адаптер представляет собой вставляемую в корпус компьютера плату с разъемом для подключения линии связи компьютерной сети.

1.17. Файловые структуры

Жесткий диск- основное устройство хранения данных. Он может иметь различную структуру хранения файлов и каталогов, которая обеспечивает непосредственное расположение данных на диске. Файловая система чаще всего интегрирована в операционную, а некоторые операционные системы поддерживают несколько файловых систем [10,17,19].

Большинство существующих на сегодняшний день файловых систем построены на основе *таблицы размещения файлов (File Allocation Table - FAT)*, которая содержит дорожки данных в каждом кластере на диске. Существует несколько типов файловой системы FAT, например, FAT 16 и FAT 32. Они отличаются количеством цифр, используемых в таблице размещения файлов. Другими словами в файловой системе FAT 32 используется 32-разрядное число для хранения дорожки данных в каждом кластере, в FAT 16- 16-разрядное число. В настоящее время существуют следующие типы файловой системы FAT:

- FAT 16, используемая в разделах емкостью от 16 Мбайт до 2 Гбайт;

- FAT 32, используемая (необязательно) в разделах емкостью от 512 Мбайт до 2 Тбайт.

Файловая система FAT 16 изначально применяются в операционных системах DOS и Windows и поддерживаются практически всеми известными на сегодняшний день операционными системами. Большинство персональных компьютеров поставляется с жесткими дисками, на которых установлена одна из файловых систем FAT.

Файловая система FAT 32 поддерживается начиная с операционной системы Windows 95B и более поздних версий. А операционная система Windows 2000, которая также поддерживает файловую систему NTFS. Некоторые операционные системы имеют собственную файловую систему. Например, Windows NT, Windows 2000, Windows XP, Windows Vista поддерживают файловую систему NT File System (NTFS); операционная система OS/2 поставляется с собственной файловой системой High Performance File System (HPFS).

Для обеспечения пользовательским приложениям доступа к файлам независимо от типа используемого диска в операционной системе предусмотрено несколько структур. Эти структуры поддерживаются операционными системами Windows 9x, Windows NT и более старшие версии представлены ниже в порядке расположения на диске:

- загрузочные секторы главного и дополнительного разделов;
- загрузочный сектор логического диска;
- таблицы размещения файлов;
- корневой каталог;
- область данных;
- цилиндр для выполнения диагностических операций чтения/записи.

1.17.1. Система FAT

Впервые поддержка жестких дисков была реализована в DOS 2.0, выпущенной в 1983 году. В этой операционной системе впервые использовалась 16-разрядная файловая система FAT и поддерживалось деление диска на разделы, т.е. создание на дис-

ке логических томов. Выполнять разделение диска необходимо даже в том случае, если вы собираетесь использовать только один раздел. Разделы диска иногда называют логическими томами, поскольку операционная система присваивает каждому разделу отдельную букву.

В настоящее время практически все накопители на жестких дисках делятся на несколько разделов, с которыми работает установленная операционная система. Однако, разделив диск, можно установить несколько операционных систем по одной в каждый раздел, и эти системы будут нормально сосуществовать на одном диске. Такая мультизагрузка становится все более популярной в настоящее время.

Чтобы установить на один жесткий диск несколько операционных систем, его надо разбить на разделы. Например, вы можете с помощью программы PartitionMagic создать на диске один или несколько разделов для установки на них DOS, Windows 9x, Windows XP, либо Windows Vista, а оставшуюся часть диска предоставить для другой операционной системы. Каждый раздел в операционной системе будет иметь вид отдельного диска.

Информация о каждом разделе сохраняется в загрузочном секторе раздела (или логического диска) в начале каждого раздела. Существует также основная таблица списка разделов, помещенная в загрузочный сектор главного раздела.

Загрузочный сектор главного раздела (или *главная загрузочная запись* (*Master Boot Record – MBR*)) является первым сектором на жестком диске (цилиндр 0, головка 0, сектор 1) и состоит из двух элементов.

Таблица главного раздела. Содержит список разделов на диске и расположение загрузочных секторов соответствующих логических дисков. Эта таблица очень маленькая и может содержать максимум четыре записи. Таким образом, для получения большего количества разделов в операционной системе (например, DOS) можно создать один дополнительный раздел и поместить в него несколько логических дисков.

Главный загрузочный код. Небольшая программа, которая выполняется системной BIOS. Основная функция этого кода-

передача управления в раздел, который обозначен как активный (или загрузочный).

В файловой системе FAT количество разделов на всех жестких дисках в системе может достигать 24. Это означает, что в компьютере может быть установлено либо 24 отдельных накопителя, в каждом из которых имеется по одному разделу, либо один жесткий диск с 24 разделами, либо несколько накопителей с разным количеством разделов, но при условии, что общее количество разделов не больше 24. Если общее количество разделов превысит эту цифру, DOS просто проигнорирует их, хотя другие операционные системы могут работать и с большим количеством томов. Такое ограничение DOS связано с тем, что в латинском алфавите от С до Z всего 24 буквы.

В начале каждого раздела DOS содержится загрузочный сектор логического диска. При разбиении диска на разделы необходимо создать активный (или загрузочный) раздел. Программа, содержащаяся в самом первом секторе на жестком диске, определяет, какой раздел активен, и передает управление его загрузочному сектору. Вы также можете создать дополнительный раздел диска для Novell NetWare, NTFS (Windows NT), HPFS (OS/2), AIX (UNIX), XENIX или другой файловой системы, используя системный диск с соответствующей программой разделения диска.

Разделы, используемые этими операционными системами, недоступны при работе в DOS. Все дело в различиях между файловыми структурами. DOS использует структуру FAT, которая также поддерживается OS/2, Windows NT и некоторыми другими операционными системами. В то же время в OS/2 обычно вместо FAT применяется файловая система HPFS (High Performance File System), а Windows NT пользуется собственной файловой системой NTFS (NT File System) и т.д.

1.17.2. Система FAT32

Когда разрабатывалась файловая система FAT, жесткие диски размером 2 Гбайт можно было встретить разве что в научно-фантастических романах. В настоящее время практически все системы нижнего уровня оснащаются жестким диском не менее

160 Гбайт, а чаще всего 320 или 540 Гбайт. При использовании стандартной FAT вы можете создать раздел размером не более 2 Гбайт. Это ограничение приводит к тому, что пользователи больших жестких дисков испытывают неудобства в организации файлов: в их распоряжении как минимум три диска (диск размером 6 Гбайт можно разбить на три диска по 2 Гбайт).

Для устранения такого ограничения фирма Microsoft предложила новую файловую систему с расширенными возможностями, называемую FAT32. Эта файловая система работает как стандартная FAT, но имеет отличия в организации хранения файлов. Кроме того, FAT32 можно установить с помощью программы Fdisk, в отличие от VFAT, которая является частью Vmm.vxd. Файловая система FAT32 была впервые реализована в операционной системе Windows 95 OEM Service Release 2 (OSR2). Она встроена также и в Windows 98.

Основное преимущество FAT32 – это возможность использования 32-разрядных записей, вместо 16-разрядных, что приводит к увеличению числа кластеров в разделе до 268 435 456 (вместо 65 536, или 2^{16}). Это значение эквивалентно 2^{28} , а не 2^{32} , поскольку четыре бита из 32 зарезервированы для других целей.

При использовании FAT32 размер раздела может достигать 2 Гбайт (1 Гбайт равен 1 024 Мбайт). Новая файловая система может иметь 4 294 967 296 (2^{32}) кластеров размером 512 байт, а размер единичного файла может составлять 4 Гбайт.

Существует еще одно отличие FAT32 от ее предшественниц положение корневого каталога: он не занимает фиксированного места на диске, как в FAT16. Корневой каталог в FAT32 может располагаться в любом месте раздела и иметь любой размер. Устранение ограничений записей корневого каталога обеспечивает динамическое изменение размера раздела FAT32. Фирма Microsoft не реализовала это замечательное свойство в операционных системах Windows 9x, чем и воспользовались независимые разработчики, такие как фирма PowerQuest, создавшая программу PartitionMagic.

Основной недостаток файловой системы FAT32 – несовместимость с предыдущими версиями DOS и Windows 95. Вы не сможете загрузить предыдущую версию DOS или ориги-

нальную Windows 95 с диска с FAT32, кроме того, раздел с FAT32 будет недоступен этим системам при их загрузке с другого диска.

Поскольку раздел FAT32 имеет большее количество кластеров, чем раздел FAT16, размер кластера уменьшается. Использование меньшего кластера снижает потери дискового пространства. Например, раздел размером 2 Гбайт с 5000 файлов в FAT32 использует кластер размером 4 Кбайт, вместо 32 Кбайт в FAT16. Такое уменьшение размера кластера приводит к снижению потерь дискового пространства с 78 до 10 Мбайт.

1.17.3. Система NTFS

Файловая система NTFS применяется в операционной системе Windows NT. Несмотря на то, что Windows NT может использовать разделы с FAT, файловая система NTFS обеспечивает большие преимущества по сравнению с FAT: большие размеры файлов и разделов, дополнительные атрибуты файлов и расширенные средства безопасности. При разработке операционной системы Windows NT не существовало проблем обеспечения обратной совместимости, поэтому файловая система обладает особыми свойствами и поддерживается только Windows NT.

Все операционные системы Windows (кроме Windows NT) основаны на DOS, поэтому внутри системы существует часть кода DOS. В Windows NT DOS-программы работают в режиме эмуляции DOS. При загрузке другой операционной системы раздел с NTFS недоступен.

В файловой системе имя файла может содержать до 255 символов, включая пробелы, точки и другие символы, кроме * ? \ / ; < > |. Поскольку NTFS — 64-разрядная файловая система, размер файла и раздела может быть просто огромным — 2^{64} байт, или 17 179 869 184 Тбайт!

Несмотря на существующие различия в структуре раздела файловых систем FAT и NTFS, они имеют подобные элементы, например загрузочную область. Раздел NTFS состоит из главной таблицы файлов (masterfile table – MFT). MFT – это не то же самое, что FAT. Вместо использования таблицы со ссылками на кластеры, MFT содержит большее количество информации о

файлах и каталогах в разделе. В некоторых случаях MFT может даже содержать файлы и каталоги.

Первая запись в MFT называется дескриптором {descriptor} и содержит информацию о расположении самой MFT. Загрузочный сектор в разделе NTFS содержит ссылку на расположение записи дескриптора.

Вторая запись в MFT – это зеркальная копия дескриптора. Такое избыточное хранение данных обеспечивает большую устойчивость к ошибкам.

Третья запись— это запись файла журнала. Все операции (транзакции) в NTFS записываются в специальный файл журнала, что позволяет восстановить данные после сбоя. Остальная часть MFT состоит из записей для файлов и каталогов, которые хранятся в разделе. В файле NTFS хранятся атрибуты, определенные пользователем и системой. Атрибуты в разделе NTFS – это не простые флаги из раздела FAT. Вся информация о файле, т.е. атрибуты, в файловой системе NTFS сохраняется вместе с файлом и является частью самого файла. Каталоги в NTFS состоят в основном из индексов файлов в этом каталоге и не содержат такой информации о файле, как размер, дата, время и др.

Таким образом, MFT – это не просто список кластеров, это основная структура хранения данных в разделе. Если файл или каталог относительно небольшой (около 1 500 байт), его запись может храниться в MFT. Для больших массивов данных в MFT помещается указатель на файл или каталог, а сами данные располагаются в других кластерах в разделе. Эти кластеры называются экстендами (extents). Все записи в MFT, включая дескрипторы и файл журнала, могут использовать экстенды для хранения дополнительных атрибутов. Атрибуты файла, которые являются частью записи MFT, называются резидентными (resident), а атрибуты, расположенные в экстендах, – нерезидентными (nonresident).

Получить доступ к разделу NTFS из DOS и других операционных систем нельзя. Windows NT предназначена для использования в качестве сетевой операционной системы, поэтому доступ к файлам в разделе NTFS можно получить посредством сети.

Для этого в NTFS поддерживаются имена файлов, удовлетворяющие стандарту восемь–точка–три.

Основное преимущество файловой системы NTFS – это обеспечение безопасности файлов и каталогов. Атрибуты безопасности в NTFS называются разрешениями (permissions) и устанавливаются системным администратором посредством предоставления доступа к данным на уровне прав пользователей и групп пользователей.

Однако вы можете установить FAT-атрибуты файлов в NTFS с помощью стандартных инструментов операционной системы Windows NT, например программы Windows NT Explorer или команды DOS Attrib. При копировании файлов из раздела NTFS в FAT все атрибуты файла сохраняются, и пользователь с правами полного доступа не сможет удалить файл с FAT-атрибутом "только для чтения".

1.18. Интерфейсы ввода-вывода

Последовательные порты, параллельные порты, универсальная последовательная шина (Universal Serial Bus – USB), Fire Wire, SCSI и IDE – являются интерфейсами ввода-вывода. Основными средствами коммуникации, используемыми в PC, являются последовательные и параллельные порты. К последовательным портам чаще подключаются двунаправленные устройства, которые должны как передавать информацию в компьютер, так и принимать ее [9,10,19].

Параллельные порты, как правило, используются для подключения принтеров, сканеров и работают в однонаправленном режиме, хотя могут применяться и как двунаправленные. Некоторые фирмы выпускают программы, предназначенные для организации высокоскоростной передачи данных между компьютерами через последовательные или параллельные порты. Версии этих программ передачи файлов были включены еще в DOS версии 6.0. Существуют версии сетевых адаптеров, накопителей на магнитной ленте, дисководов для гибких дисков и CD-ROM, которые также подключаются к параллельным портам.

1.18.1. Последовательные порты

Асинхронный последовательный интерфейс – это основной тип интерфейса, с помощью которого осуществляется взаимодействие между компьютерами. Термин асинхронный означает, что при передаче данных не используются никакие синхронизирующие сигналы и отдельные символы могут передаваться с произвольными интервалами, как, например, при вводе данных с клавиатуры [9,10,19].

Каждому символу, передаваемому через последовательное соединение, должен предшествовать стандартный стартовый сигнал, а завершать его передачу должен стоповый сигнал. Стартовый сигнал - это нулевой бит, называемый стартовым битом. Его назначение - сообщить принимающему устройству о том, что следующие восемь битов представляют собой байт данных. После символа передаются один или два стоповых бита, сигнализирующие об окончании передачи символа. В принимающем устройстве символы распознаются по появлению стартовых и стоповых сигналов, а не по моменту их передачи. Асинхронный интерфейс ориентирован на передачу символов (байтов), а при передаче используется примерно 20% информации только для идентификации каждого символа.

Термин *последовательный* означает, что передача данных осуществляется по одиночному проводнику, а биты при этом передаются последовательно, один за другим. Такой тип связи характерен для телефонной сети, в которой каждое направление обслуживает один проводник. Многие компании выпускают дополнительные последовательные порты для компьютеров, обычно эти порты устанавливаются на многофункциональных платах или на плате с параллельным портом.

К последовательным портам можно подключить самые разнообразные устройства: модемы, плоттеры, принтеры, другие компьютеры, устройства считывания штрих-кода или схему управления устройствами. В основном во всех устройствах, для которых необходима двунаправленная связь с компьютером, используется ставший стандартом последовательный порт RS-232C (Reference Standard number 232 revision C – стандарт обмена но-

мер 232 версии C), который позволяет передавать данные между несовместимыми устройствами.

Основой любого последовательного порта является микросхема UART (Universal Asynchronous Receiver/Transmitter – универсальный асинхронный приемник/передатчик). С ее помощью осуществляется управление преобразованием данных из принятого от компьютера параллельного формата в последовательный и наоборот.

В настоящее время фирмы-производители предлагают несколько видов микросхем UART. В первых компьютерах PC и XT применялась микросхема UART 8250, которая до сих пор устанавливается на многих дешевых платах последовательных портов. В компьютерах PC/AT (и в других компьютерах на базе процессора 286 и последующих) используется микросхема UART 16450. Единственное различие между этими двумя микросхемами заключается в способности обеспечивать высокоскоростной обмен данными: микросхема 16450 лучше приспособлена для этих целей.

Микросхема UART 16550 была первой схемой последовательного порта, которая использовалась в компьютерах PS/2. Она могла работать так же, как и микросхемы 8250 и 16450, но содержала еще и буфер на 16 байт, позволяющий передавать данные с более высокой скоростью. Буфер использовался по принципу FIFO (First In/First Out, т.е. первым пришел первым ушел). К сожалению, эта схема имела существенные недостатки, связанные именно с работой буфера. Они были устранены в микросхеме UART 16550A. В настоящее время фирма National Semiconductor выпускает микросхему UART 16550D.

1.18.2. Параллельные порты

В параллельных портах для одновременной передачи байта информации используется восемь линий. Этот интерфейс отличается высоким быстродействием, часто применяется для подключения к компьютеру принтера, а также соединения компьютеров. (Ведь при этом существенно выше скорость передачи данных, чем при соединении через последовательные порты: 4, а не 1 бит за раз.) Существенным недостатком параллельного пор-

та является то, что соединительные провода не могут быть слишком длинными. При большой длине соединительного кабеля в него приходится вводить промежуточные усилители сигналов, так как в противном случае возникает большое количество помех [9,10,19].

Интерфейс передачи данных по параллельным портам был доработан в 1994 году, когда был принят стандарт IEEE 1284, в котором определены физические характеристики параллельных портов (режимы передачи данных и т.д.).

Хотя IEEE 1284 был выпущен для стандартизации форм сигналов, с помощью которых компьютер "общается" с подключаемыми устройствами, в частности с принтером, этот стандарт интересен и для производителей периферийных устройств, подключаемых к параллельным портам (дисководов, сетевых адаптеров и др.).

Поскольку IEEE 1284 предназначен только для аппаратного обеспечения и не определяет требований к программному обеспечению, работающему с параллельными портами, вскоре был разработан стандарт, определяющий требования к такому программному обеспечению и направленный на устранение различий между микросхемами параллельных портов разных производителей. В нем, в частности, описана спецификация для поддержки режима EPP через BIOS.

Стандартом IEEE 1284 предусмотрена более высокая пропускная способность соединения между компьютером и принтером или двумя компьютерами. Для реализации этой возможности стандартный кабель принтера не подходит. Стандартом IEEE 1284 для принтера предусмотрена витая пара.

В стандарте IEEE 1284 определен также новый разъем. Разъем типа А определен как штыревой DB25, разъем типа В как Centronics 36. Новый разъем типа С является разъемом высокой плотности. Такие разъемы (типа С) уже устанавливаются на принтерах Hewlett-Packard. Разъемы всех трех типов показаны на рис. 1.14.

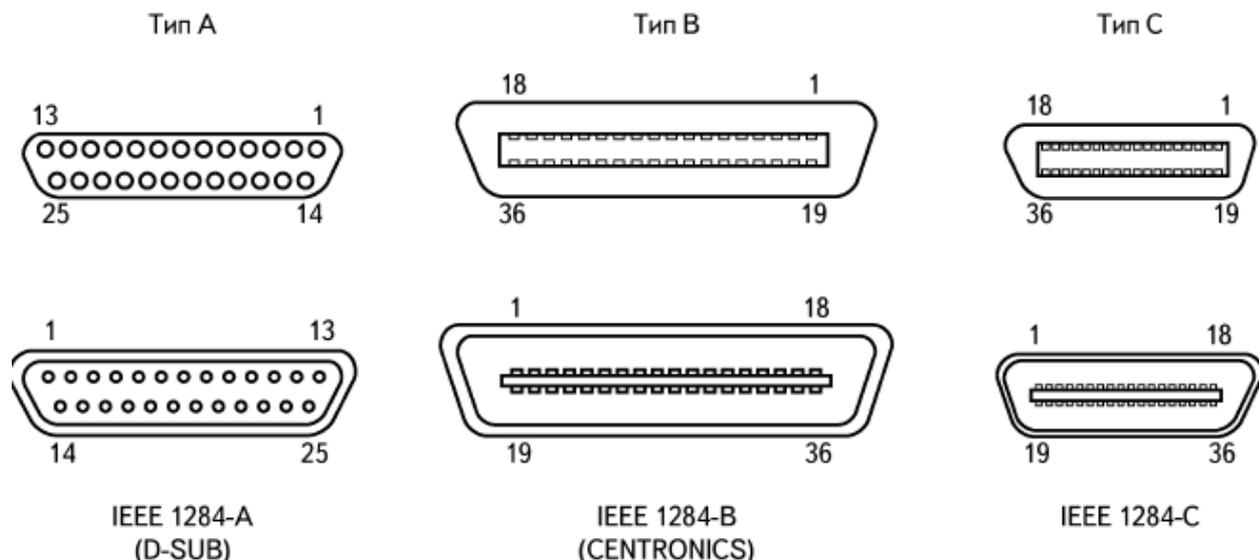


Рис. 1.14. Типы разъемов для параллельных портов

В первом компьютере IBM PC существовал только один параллельный порт, предназначенный для передачи информации от компьютера к какому-либо устройству, например к принтеру. Однонаправленность параллельного порта первого PC вполне соответствовала его основному назначению передаче данных на принтер. Однако во многих случаях желательно было иметь двунаправленный параллельный порт даже для принтера (чтобы можно было реализовать обратную связь, например для принтеров типа PostScript). С однонаправленным параллельным портом осуществить это было невозможно.

Такой тип параллельных портов не предназначался для использования в качестве ввода, однако с помощью специальных схем (в которых четыре сигнальные линии могла быть представлены как 4-разрядное соединение) и однонаправленного параллельного порта можно обеспечить 8-разрядный вывод и 4-разрядный ввод. В настоящее время этот тип портов используется довольно редко, так как в компьютерах, выпущенных после 1993 года как правило устанавливаются параллельные порты наподобие 8-разрядного, UPP и UCP.

Стандартный параллельный порт обеспечивает скорость передачи данных 50 Кбайт/с, но при использовании различных усовершенствований пропускную способность можно увеличить до 150 Кбайт/с.

Двунаправленные порты (8-разрядные)

Двунаправленный параллельный порт впервые появился в 1987 году в компьютерах PS/2. Даже сегодня в PC-совместимых компьютерах можно найти порты, которые обычно обозначаются как параллельные 'типа PS/2', "двунаправленные" и "расширенные" (extended) параллельные порты. С помощью такого порта появилась возможность организовать двусторонний обмен данными между компьютером и различными периферийными устройствами. Для этого используется несколько бывших до этого свободными контактов разъема параллельного порта, а направление передачи информации определяется специальным битом состояния.

Двунаправленные порты могут работать с 8-разрядным вводом и выводом, используя для этого восемь стандартных линий передачи данных, пропускная способность которых при подключении внешних устройств значительно выше, чем для 4-разрядных портов. Скорость передачи данных при работе через двунаправленный порт около 150 Кбайт/с.

Усовершенствованный параллельный порт (EPP)

EPP (Enhanced Parallel Port) – это новый тип параллельного порта, который иногда называют *быстродействующим параллельным портом*. Он разработан фирмами Intel, Xircom и Zenith Data Systems и выпущен в октябре 1991 года. Первыми устройствами, предлагающими возможности усовершенствованного параллельного порта, были портативные компьютеры фирмы Zenith Data Systems, адаптеры фирмы Xircom и микросхема Intel 82360 SL I/O.

EPP работает практически на всех скоростях, поддерживаемых шиной ISA, и предлагает десятикратное увеличение пропускной способности по сравнению с обычным параллельным портом. Этот тип портов разработан специально для таких подключаемых к параллельному порту устройств, как сетевые адаптеры, дисководы и накопители на магнитной ленте. Усовершенствованный параллельный порт соответствует требованиям нового стандарта IEEE 1284 для параллельных портов и передает данные со скоростью до 2 Мбайт/с.

Порт с расширенными возможностями (ECP)

Другой тип высокоскоростного параллельного порта, называемый *портом с расширенными возможностями* (Enhanced Capabilities Port - ECP), разработан фирмами Microsoft и Hewlett-Packard и выпущен в 1992 году. Подобно EPP, этот порт обладал повышенной производительностью и требовал для своей работы специальной логики устройств.

Порт с расширенными возможностями соответствует требованиям стандарта IEEE 1284. Однако, в отличие от порта EPP, он не является портом, специально разработанным для подключения устройств к PC-совместимым компьютерам. Основная цель разработки и выпуска этого типа параллельных портов - поддержка "недорогого" подключения высокоскоростных принтеров. Еще одним отличием ECP от EPP является то, что режим работы первого из них требует использования канала прямого доступа к памяти, который никак не определен в EPP (что зачастую приводит к конфликтам, связанным с устройствами, которые также используют прямой доступ к памяти). Большинство компьютеров могут работать как в режиме ECP, так и в EPP.

1.18.3. Интерфейсы USB и FireWire

В настоящее время для настольных и портативных компьютеров разработано два высокоскоростных устройства с последовательной шиной, получивших название *USB (Universal Serial Bus* – универсальная последовательная шина) и *IEEE 1394*, называемая *также i.Link* или *FireWire*. Возможности этих высокоскоростных коммуникационных портов намного выше стандартных параллельных и последовательных портов, которые установлены в большинстве современных компьютеров. Преимущество новых портов состоит в том, что их можно использовать как альтернативу SCSI для высокоскоростных соединений с периферийными устройствами, и в том, что к ним могут подсоединяться все типы внешних периферийных устройств (т.е. была предпринята попытка объединения устройств ввода-вывода) [9,10,19].

Последним направлением в развитии высокоскоростных периферийных шин является использование последовательной архитектуры. Для передачи информации в параллельной архитектуре, где биты передаются одновременно, необходимы ли-

нии, имеющие 8, 16 и более проводов. Может показаться, что за одно время через параллельный канал передается больше данных, чем через последовательный, однако на самом деле увеличить пропускную способность последовательного соединения намного легче, чем параллельного.

Параллельное соединение обладает рядом недостатков, одним из которых является фазовый сдвиг сигнала, из-за чего длина параллельных каналов, например SCSI, ограничена (не должна превышать 3 м). Проблема здесь заключается в том, что, хотя 8- и 16-разрядные данные одновременно пересылаются передатчиком, из-за задержек одни биты прибывают в приемник раньше других. Соответственно, чем длиннее кабель, тем больше время задержки между первым и последним прибывшими битами на приемном конце.

В последовательном соединении данные передаются друг за другом, поэтому один бит не сможет "обогнать" другой и скорость передачи может быть значительно увеличена.

Еще одним преимуществом последовательного способа передачи данных является возможность использования только одно- или двухпроводного кабеля, поэтому помехи, возникающие при передаче, очень малы, чего нельзя сказать о параллельном соединении.

Стоимость параллельных кабелей довольно высокая, поскольку провода, предназначенные для параллельной передачи, не только используются в большом количестве, но и специальным образом укладываются, дабы предотвратить возникновение помех, а это очень трудоемкий и дорогостоящий процесс. Кабели для последовательной передачи данных, напротив, очень дешевые, так как состоят из нескольких проводов и требования к их экранированию намного ниже, чем у параллельных.

Именно поэтому, а также в силу требований внешнего периферийного интерфейса Plug and Play и необходимости устранения физического нагромождения портов в портативных компьютерах были разработаны эти две высокоскоростные последовательные шины, используемые в компьютерах сегодня повсеместно.

Универсальная последовательная шина USB

В шине USB (Universal Serial Bus) реализована возможность подключения большого количества периферийных устройств к компьютеру. При подключении устройств к USB не нужно устанавливать платы в разъемы системной платы и реконфигурировать систему; кроме того, экономно используются такие важные системные ресурсы, как IRQ (запросы прерывания). При подключении периферийного оборудования к персональным компьютерам, оснащенным шиной USB, его настройка происходит автоматически, сразу после физического подключения, без перезагрузки или установки.

Основным инициатором разработки стандарта USB выступила компания Intel. Начиная с набора микросхем системной логики Triton II (82430HX), в котором стандарт USB был воплощен в микросхеме PIIX3 South Bridge, фирма Intel поддерживает этот стандарт во всех своих наборах микросхем системной лотки. Совместно с компанией Intel над созданием универсальной последовательной шины работало еще семь компаний, среди которых Compaq, Digital, IBM, Microsoft, NEC и Northern Telecom. Эти фирмы создали форум USB Implement Forum, целью которого является развитие, поддержка и распространение архитектуры USB.

Универсальная последовательная шина это интерфейс, работающий со скоростью 12 Мбит/с (1,5 Мбайт/с) и основанный на простом 4-проводном соединении. Эта шина поддерживает до 127 подключаемых устройств и использует топологию звезды, построенную на расширяющих концентраторах, которые могут входить в персональный компьютер, любое периферийное устройство USB и даже быть отдельными устройствами.

Для подключения нескольких USB-устройств одновременно необходимо использовать концентратор. С помощью концентратора к одному порту USB можно подключить клавиатуру, мышь, цифровую камеру, принтер, телефон и т.д. В компьютере устанавливается модуль, называемый *корневым концентратором*, - начальная точка для подключения всех остальных устройств. Практически все системные платы имеют два порта USB.

Максимальная длина кабеля между двумя работающими на предельной скорости (12 Мбит/с) устройствами или устройством

и концентратором - пять метров. В кабеле используется экранированная витая пара. Максимальная длина кабеля для низкоскоростных (1,5 Мбит/с) устройств при использовании нескрученной пары проводов три метра. Причем эти расстояния уменьшаются, если используется более толстый провод.

Шина USB удовлетворяет требованиям технологии Plug and Play фирмы Intel, в том числе требованию горячего подключения, при котором устройство может подсоединяться к компьютеру без выключения питания и перезагрузки системы. Нужно просто подключить устройство, после чего контроллер USB, установленный в компьютер, самостоятельно его обнаружит, а также добавит необходимые для работы ресурсы и драйверы. Компания Microsoft разработала специальные драйверы USB и включила их последние версии операционных систем. Поддержка универсальной последовательной шины необходима также и в BIOS; шина USB устанавливается в новых системах, имеющих встроенные порты USB. Существуют также платы USB, с помощью которых можно добавить возможности универсальной последовательной шины в уже существующие компьютеры. К шине USB можно подключить такие периферийные устройства, как модемы, телефоны, джойстики, клавиатуры и устройства управления указателем (мыши), различные типы видео и аудиоаппаратуры.

Интересной особенностью универсальной последовательной шины является возможность подвода мощности ко всем подключаемым устройствам через шину. Поддержка Plug and Play позволяет системе «опрашивать» подключаемое устройство о его энергетических потребностях и выдает предупреждение, если уровень мощности превосходит допустимый. Это наиболее эффективно для портативных компьютеров, емкость батарей которых ограничена.

Благодаря устройствам USB осуществляется самоопределение периферийного оборудования, что значительно упрощает его установку. Это означает, что не нужно устанавливать уникальные адреса для каждого периферийного устройства шина USB делает это автоматически. Причем при подключении или

отключении устройств USB не нужно выключать компьютер или перезагружать систему.

Одно из самых значительных достоинств интерфейса типа USB состоит в том, что для обслуживания всех устройств шины USB требуется только одно-единственное прерывание. Это означает, что можно присоединить 127 устройств, и все они будут использовать одно прерывание. В современных персональных компьютерах так часто не хватает свободных адресов прерываний, что это, пожалуй, самое ценное достоинство USB.

IEEE 1394 (Шина FireWire или i.Link)

FireWire – это высокоскоростная локальная последовательная шина, способная передавать данные со скоростью 100, 200 и 400 Мбит/с (12,5, 25 и 50 Мбайт/с), а при работе с некоторыми типами файлов до 1 Гбит/с. Стандарт на пишу IEEE 1394 (официальное название шины Fire-Wire) опубликован в конце 1995 года. Он разработан на основе шины FireWire, представленной фирмами Apple и Texas Instruments, и является частью стандарта Serial SCSI.

Эта шина использует простой 6-проводный кабель, состоящий из двух различных пар линий, предназначенных для передачи тактовых импульсов и информации, а также двух линий питания. Как и USB, IEEE 1394 полностью поддерживает технологию Plug and Play, включая возможность горячего подключения (установка и извлечение компонентов без отключения питания системы). Но структуре шина 1394 не так сложна, как параллельная шина SCSI, и устройства, подключаемые к ней, могут потреблять от нее ток до 1,5 А. По производительности шина IEEE 1394 превосходит Ultra-Wide SCSI, стоит гораздо меньше, а устройства к ней подсоединить намного проще. На рис. 1.15 показаны компоненты разъемов шины 1394.

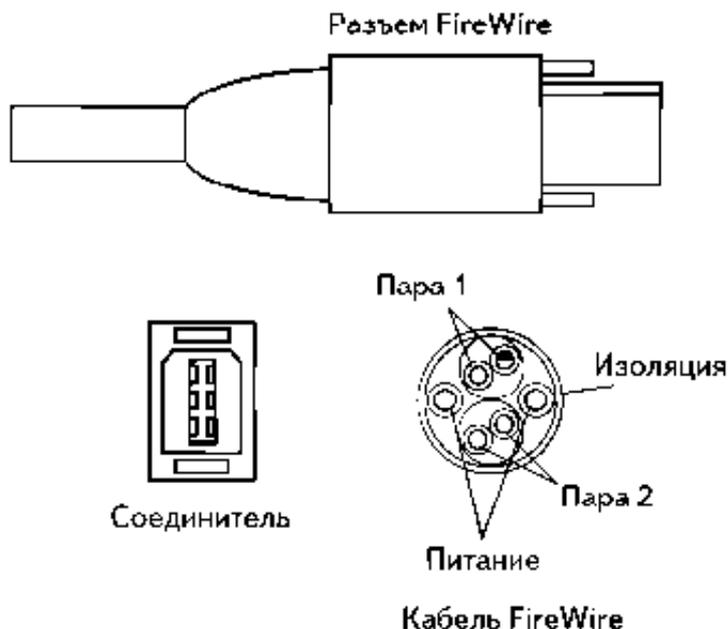


Рис. 1.15. Типы разъемов 1394.

Шина 1394 построена на разветвляющейся топологии и позволяет использовать до 63 узлов в цепочке и подсоединять при этом к каждому узлу до 16 устройств. Если этого недостаточно, то можно дополнительно подключить до 1 023 шинных переключателей, которые могут соединять более 64 000 узлов! Кроме того, шина 1394 может поддерживать устройства, построенные на одной шине, но работающие на разных скоростях передачи данных, как и SCSI.

Шина 1394 построена на разветвляющейся топологии и позволяет использовать до 63 узлов в цепочке и подсоединять при этом к каждому узлу до 16 устройств. Если этого недостаточно, то можно дополнительно подключить до 1 023 шинных переключателей, которые могут соединять более 64 000 узлов. Кроме того, шина 1394 может поддерживать устройства, построенные на одной шине, но работающие на разных скоростях передачи данных, как и SCSI.

Подключить к компьютеру через шину 1394 можно практически все устройства, имеющие возможность работать со SCSI. Сюда входят все виды дисковых накопителей, включая жесткие, оптические, CD-ROM и новые цифровые видеодиски DVD. К шине 1394 могут подключаться цифровые видеокамеры, устройства с записью на магнитную ленту и многие другие высокоскоростные периферийные устройства.

1.19. Операционные системы и их классификация

Одним из условий эффективного использования вычислительной техники является создание специализированных прикладных программ [1,2], среди которых особая роль отводится офисным прикладным программам (рис. 1.16).

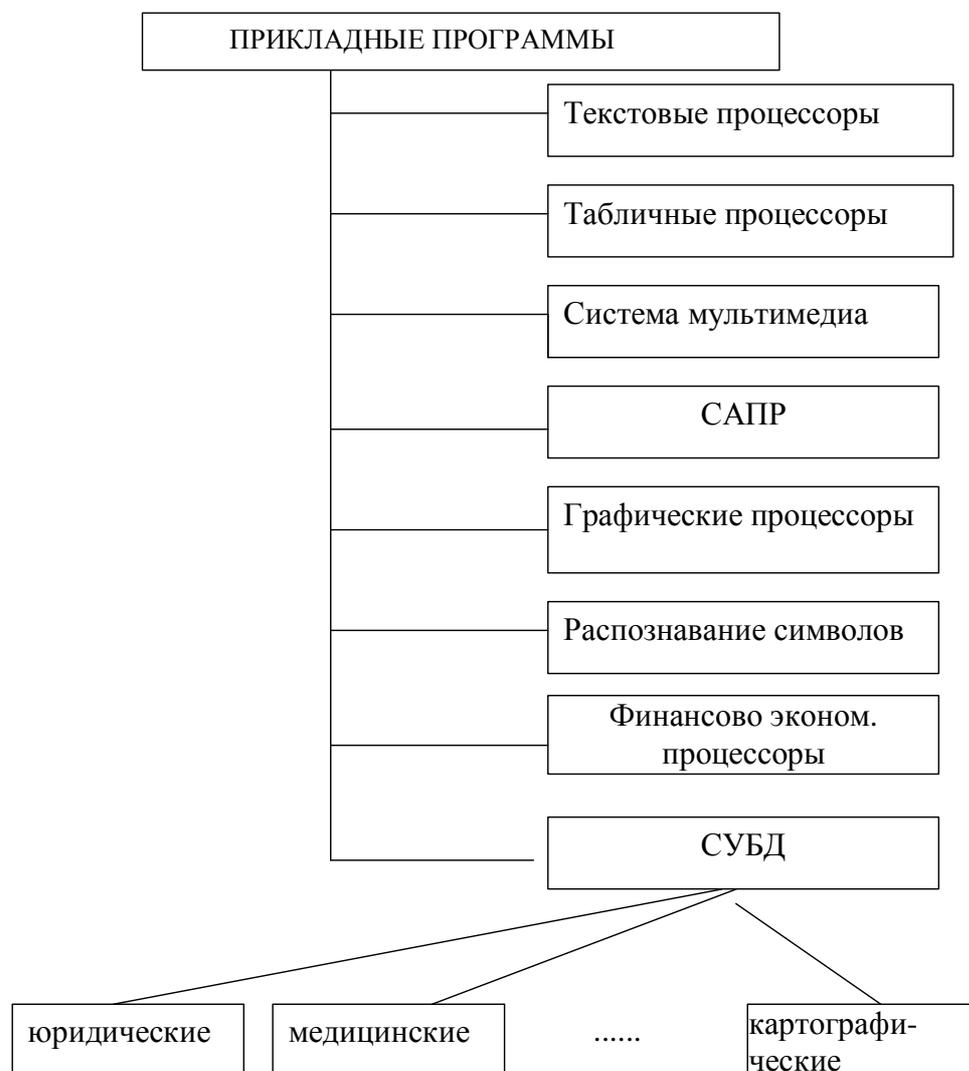


Рис. 1.16. Структура прикладных программ Windows

Кроме прикладных программ, ориентированных на конкретную предметную область деятельности, интенсивно проектируются интегрированные прикладные программы, соединяющие в себе различные функции отдельных предметных программ. Наиболее известным примером интегрированных прикладных программ является офисный пакет MS Office. Он содержит такие прикладные программы, как Microsoft Word, Mi-

Microsoft Excel, Microsoft Access, Microsoft PowerPoint, Microsoft Visio и т.д. Рассмотрим в данном учебном пособии две прикладные программы Microsoft Word и Microsoft Excel.

В настоящее время существует достаточно большое количество различных типов операционных систем (MS DOS, Windows (95, 98, Me, NT, 2000, XP и др., OS/2, MacOS, Linux и т.д.). В нашей стране наиболее используемой, но не самой наилучшей по ряду параметров является Windows. К числу ее "минусов" этой ОС можно отнести: однозадачность и слишком большую "открытость" и сложность интерфейса, а также невысокую надежность и большое количество недоработок (особенно Windows 98, Me).

ОС обычно классифицируют по следующим признакам:

- число пользователей, одновременно работающих с системой (однопользовательские и многопользовательские);
- число задач, которые могут решаться с их помощью в любой момент времени (однозадачные и многозадачные);
- базовый способ общения пользователя с ОС (диалог на языке команд, диалог на языке меню, диалог на языке графических представлений);
- число разрядов адресной шины (16, 32, 64...);
- минимально требуемые ресурсы, т.е. минимально необходимые объемы оперативной и дисковой памяти, класс микропроцессора.

В этой классификации MS DOS – однопользовательская, однозадачная, 16-разрядная ОС, общающаяся с пользователем на языке команд.

Windows – это многопользовательская, многозадачная, 32-разрядная ОС.

Еще совсем недавно ОС MS DOS пользовалась достаточно большой популярностью. Но в настоящее время большинство прикладных программ требуют для своей работы ОС Windows (Windows 98 и выше). Поэтому, рассмотрим работу с ОС Windows подробнее.

1.20. Операционная система Windows

1.20.1. Общая характеристика Windows

В настоящее время широко используются операционные системы (ОС) Windows 95, 98, Me, 2000, XP, которые представляет собой интегрированные, объектно-ориентированные операционные системы, основанные на 32-разрядной технологии со встроенной поддержкой работы в сети. В отличие от них, Windows 3.x являлась надстройкой над операционной системой MS DOS, т.е. это была программная оболочка, которую чаще называют операционной средой [1,2,17].

Излагаемый далее материал поясняет возможности и работу на примере наиболее распространенной ОС Windows 98. Основные приемы работы с Windows 2000, Windows XP будут аналогичными.

Windows 2000 и Windows XP обеспечивают более надежную и безопасную (с точки зрения защиты информации) работу, но требуют больших ресурсов компьютера.

ОС Windows предоставляет следующие возможности:

- предоставляет пользователю удобный и понятный интерфейс;
- работает на IBM PC большинства современных конфигураций. При этом сохранена и даже улучшена совместимость с существующими DOS и Windows-приложениями;
- содержит 32-разрядное ядро, обеспечивающее более быстрое и эффективное управление памятью и всеми процессами;
- имеет новую 32-разрядную файловую систему с открытой для дальнейшего развития архитектурой;
- имеет встроенные средства для работы со звуком, видео и компакт-дисками;
- предоставляет такие инструменты, как Мастера (Wizards), которые автоматизируют процесс выполнения операций путем задания достаточно простых вопросов пользователю;
- включает встроенные средства диагностики, оптимизации и исправления ошибок, которые помогают устранять конфликты между устройствами и повышают эффективность функционирования всей системы.

- выполнение всех функций DOS (Windows 2000 и Windows XP для обеспечения безопасности не поддерживают режим DOS);
- поддержку независимой мультизадачности для 32-разрядных приложений, т.е. обеспечивает одновременную работу нескольких приложений, распределяя кванты времени (временные интервалы) между активными приложениями и автоматически передавая управление другой задаче по окончании выделенного времени;
- максимально упрощает установку и настройку периферийных устройств за счет поддержки стандарта Plug and Play;
- обеспечивает работу в режиме удаленного доступа и синхронизации файлов настольного и портативного компьютеров;
- поддерживает работу ПК в неоднородных сетях.

1.20.2. Запуск Windows

Запуск Windows, как правило, осуществляется автоматически, сразу же после включения компьютера.

В процессе загрузки Windows выводит в качестве заставки свой логотип, и курсор мыши имеет вид песочных часов. Когда загрузка завершена, логотип сменяется рабочим окном, на нем очень быстро расставляются значки программ и курсор мыши принимает вид стрелки.

1.20.3. Концепция окна

Как понятно из названия, отличительной особенностью графической среды Windows являются окна. Для любой программы, запущенной в среде Windows открывается свое окно, в которое программа и выводит всю информацию.

Окна могут быть разными, разными не только по размеру, но и по назначению, но все они имеют некоторые общие атрибуты, определяемые средой Windows.

Основным атрибутом окна является его заголовок, который размещается вверху окна и занимает целую строку. В строке заголовка указывается имя программы. Если это программа-редактор и работает с каким-либо файлом, то при определенных обстоятельствах рядом указывается имя файла. Обычно имя файла указывается в заголовке собственного окна, но если оба

окна: окно программы и окно файла – "распахнуты" на весь экран, то их заголовки объединяются. Если одновременно запущены несколько программ, то каждая из них работает в своем окне, при этом заголовки активного окна резко отличаются по цвету от других заголовков. Обычно цвет активного заголовка – синий. Такой цвет предусмотрен в стандартной конфигурации Windows. В общем случае можно выбрать любую гамму цветов, или придумать ее самому, но в любом случае цвет заголовка будет отличаться.

С помощью заголовка можно перенести окно с одного места на экране в другое. Для этого подведите курсор мышки к строке заголовка, нажмите левую кнопку и, удерживая ее нажатой, перемещайте мышь по столу. Вслед за курсором мыши начнет перемещаться окно программы.

Такой прием получил название Drag & Drop – "схватил и тащи". Обычно, в таких случаях пользуются термином "перетащить".

На строке заголовка справа находятся три кнопки: свернуть, восстановить и закрыть. На кнопке со значением "Свернуть" изображена горизонтальная черта. Свернутое окно превращается в кнопку на панели задач, обычно внизу экрана, но иногда (очень редко) сбоку и даже вверху. Чтобы восстановить прежние размеры окна, достаточно щелкнуть мышкой (один раз) по кнопке в панели задач.

Третья кнопка имеет значение "Завершить выполнение программы". На ней изображен крестик. Однако, если вы щелкните мышкой в самом левом углу окна, послушно раскроется меню, где можно выбрать строку "Закрыть" (Close). Завершить выполнение любой программы можно, нажав комбинацию клавиш Alt+F4.

1.20.4. Рабочий стол и главное меню Windows

Ведущие производители в области разработки операционных систем (Apple, Microsoft, IBM) придерживаются следующей концепции организации пользовательского интерфейса.

Весь экран представляет собой *рабочий стол*, который создан таким образом, что выглядит как поверхность обычного сто-

ла. За ним пользователь может выполнять самые разнообразные работы: создавать документы сложной структуры (письма, отчеты, приказы, и др.), рисовать, раскладывать *папки* с документами, использовать встроенный калькулятор и пр. Реализация этой концепции преследует одну цель – сделать работу проще как для профессионалов, так и для новичков.

Современные системы, использующие концепцию "рабочего стола", являются объектно-ориентированными. Это означает, что большинство действий в таких программах осуществляется путем манипуляций над объектами системы. Каждому типу объекта в таких системах присваивается свой *значок* и некоторый набор свойств определяющий способы использования того или иного объекта. Эти свойства можно увидеть в контекстном меню, нажав правую клавишу мышки на интересующем объекте.

На рабочем столе Windows располагаются самые разнообразные объекты: программы, файлы данных, *папки* (folder) и т.д., которые представлены *значками* (*пиктограммами*, *иконками*) – графическими изображениями различного вида. Вид значка позволяет определить, с каким объектом имеет дело пользователь. Папки иногда называют аналогом каталога в MS DOS, но на самом деле это более широкое понятие. *Папка* – это хранилище, в котором могут содержаться фаты, устройства (диски, принтеры, компьютеры сети) и др. папки. Рабочий стол также является папкой, которая является материнской для всех других папок, и находится на высшем уровне иерархии.

Каждый значок на рабочем столе соответствует программе, документу (файлу данных) или папке.

Чтобы упорядочить положение значков, достаточно щелкнуть правой кнопкой мышки в поле рабочего стола и в открывшемся меню выбрать строку "Выстроить значки. Выбрав строку "Упорядочить значки" можно упорядочить их по имени, типу, размеру и дате создания.

Запуск программы осуществляется двойным щелчком мышки на значке этой программы. Если значок соответствует не программе, а документу, то сначала будет запущена программа, способная работать с этим документом, а потом – сам документ. Среди прочих есть значки, соответствующие папкам. Например,

на рабочем столе всегда находится значок, открывающий папку "Мой компьютер" и значок папки "Корзина".

Удаление документа из папки приводит к удалению файла на диске, чего не происходит, например, при удалении значка программы с рабочего стола.

В нижней части экрана Windows расположена панель задач. Если вы не видите панель задач, то поместите курсор мышки в то место экрана, где она по-вашему должна быть. Спустя секунду, панель "всплывет". Если нет, то попробуйте поискать панель справа, слева, наконец, вверху экрана. Если вы отыскали панель, то чтобы задать (или отменить) режим, когда панель автоматически исчезает с экрана, когда она не нужна, щелкните правой кнопкой мышки по панели (попытайтесь выбрать место подальше от разного рода кнопок и значков) и в открывшемся меню (контекстное меню) выберите строку Свойства. Необходимый режим задается галочкой в строке Автоматически убирать с экрана.

То же контекстное меню можно использовать для автоматического упорядочения окон на экране. Для этого следует выбрать строки Каскадом, Сверху вниз, Слева направо, определяющие порядок представления окон на экране.

Панель задач, как и любой объект, можно переместить к любой стороне экрана с помощью мышки в соответствии с принципом Drag & Drop. На панели задач в виде кнопок отображаются все запущенные программы. Каждой программе соответствует отдельная кнопка. В левой части кнопки виден значок запущенной программы и рядом – заголовок из окна программы. Если кнопок много, то их размер пропорционально уменьшается и заголовок может воспроизводиться не полностью. Кнопка соответствующая задаче, которая в настоящий момент является активной, выглядит "нажатой", т.е. более "утопленной", по сравнению с остальными.

Используя кнопки в панели задач удобно переключаться между программами – достаточно один раз щелкнуть мышкой по кнопке и окно соответствующей программы всплывет наверх экрана. Если у вас нет мышки (или она временно не работает), выбрать необходимую программу можно с помощью комбинации

клавиш Alt+Tab. Нажмите клавиши, отпустите Tab но не отпустите клавишу Alt. На экране появится список значков запущенных программ. Нажимая клавишу Tab (клавиша Alt по-прежнему остается нажатой), выберите интересующую вас программу, после чего Alt можно отпустить. Последний метод может быть особенно полезен, когда у вас запущены несколько DOS-программ и вы не можете из под них добраться до панели задач.

Контекстное меню, о котором мы говорили применительно к панели задач, может быть открыто для любого объекта. Для этого достаточно привести курсор мышки на объект и щелкнуть правой кнопкой (Для значков документов, программ и папок контекстное меню может быть вызвано также нажатием клавиш Alt+Enter).

Содержание контекстного меню зависит от объекта, для которого оно вызвано. Например, контекстное меню рабочего стола содержит команды: Упорядочить значки, Выстроить значки. Меню значка программы – Открыть, Вырезать, Копировать, Переименовать, Удалить.

Почти любой объект имеет в своем контекстном меню команду Свойства. Команда Свойства позволяет просматривать характеристики объектов и изменять многие из них, такие как даты и атрибуты файлов, размеры и цвета экранных элементов, параметры работы программ и многое др.

В левом нижнем углу экрана, на панели задач находится кнопка Пуск. (Кнопка перемещается по экрану вместе с панелью задач.) Кнопка Пуск открывает Главное меню Windows. Главное меню содержит команды доступа к прикладным и служебным программам, системе помощи Windows и находящимся в работе документам.

Пункт главного меню Программы открывает доступ к прикладным и служебным программам. В пункте Программы обычно имеются следующие группы: Стандартные, Автозагрузка, Microsoft Office, др. группы, создаваемые в процессе инсталляции программ.

В число стандартных входят программы, выбранные при установке Windows: Блокнот, Калькулятор, Графический редактор Paint, текстовый редактор WordPad. Среди стандартных про-

грамм две подгруппы: Служебные программы и Мультимедиа. В первой программе наибольший интерес представляет программа ScanDisk – аналог Norton Disk Doctor, а в группе Мультимедиа – Универсальный проигрыватель, способный воспроизводить видео- и звуковые фрагменты, хранящиеся в файлах.

Группа Документы содержит список имен последних 20 документов, открывавшихся пользователем при работе с Windows, для того, чтобы любой из них мог быть легко открыт вновь.

Группа Настройка содержит папки Панель управления и Принтеры. Здесь необходимо сказать, что Windows – система с большой избыточностью. Одной и той же цели можно достичь совершенно разными путями, по крайней мере, начиная движение и1080 из разных точек. Так, папку Панель управления можно найти в папке Мой компьютер, что на рабочем столе, так и через Главное меню, в группе Настройка. Свойства панели задач можно изменить, как вызвав контекстное меню и выбрав строчку Свойства, так и через Главное меню, группу Настройка, выбрав строчку Панель задач.

Папки на Рабочем столе и группы программ в Главном меню существуют независимо друг от друга. Одна и та же программа может быть запущена как с помощью ее значка на Рабочем столе или в одной из папок Рабочего стола, так и с помощью Главного меню.

1.20.5. Создание значков программ в Windows

Значки программ в Windows называются ярлычками. Чтобы создать значок, соответствующий программе или документу и разместить его (значок) на Рабочем столе, щелкните правой кнопкой мышки на свободном пространстве Рабочего стола и в открывшемся контекстном меню выберите строку Создать. Задержите курсор мышки в этой строке на доли секунды и откроется подменю. Осторожно перемещайте курсор мышки по строке Создать до самого подменю, после чего поместите его в строку Ярлык и щелкните левой кнопкой. Откроется диалоговое окно Создание ярлыка.

Создать ярлычок можно и из папки Мой компьютер. Откройте папку двойным щелчком мышки. Выберите и откройте двойным щелчком окно со списком файлов на устройстве С:. В меню этого последнего окна (сразу под строкой заголовка) выберите последовательно пункт Файл, строку Создать и еще строку Ярлык. Раскроется то же диалоговое окно Создание ярлыка.

В строке, которая так и озаглавлена Командная строка, наберите команду, которая будет запускать вашу программу. Как правило, это полное имя файла с расширением .exe или .com. Если Вы испытываете трудности в наборе команды, или вам просто лень набирать длинный список каталогов и подкаталогов, щелкните по кнопке Обзор, чтобы получить доступ к списку каталогов на диске. Вход в каталоги осуществляется двойным щелчком мышки, а выбор файла – одним щелчком. Имя файла должно появиться в строке Имя файла, после чего можно щелкнуть по кнопке Открыть. Полное имя файла появится в Командной строке.

Щелкните по кнопке Далее и вам будет предложено указать название создаваемого ярлычка, т.е. подпись под значком. По умолчанию, компьютер предлагает в качестве подписи имя файла. Вы можете ввести любое имя, любыми буквами, однако, постарайтесь, чтобы оно не было слишком длинным.

Щелкните по кнопке Далее еще раз, и вам будет предложено выбрать значок из числа стандартных. Значки видны на экране, а доступ к тем значкам, которые не видны можно осуществить с помощью полосы прокрутки. Выбор значка осуществляется щелчком мышки. Но если вам понравится другой значок, то вы можете изменить свой выбор, щелкнув мышкой на новом значке.

Если ни один значок вас не устраивает, выберите любой. Чуть позже у вас появится возможность сменить его. Щелкните по кнопке Готово и на этом создание ярлычка заканчивается. Он появляется на рабочем столе.

Теперь у вас появляется возможность улучшить его, сменить значок, название, изменить некоторые параметры окна, в котором будет открываться программа. Щелкните по значку правой кнопкой мыши и в открывшемся контекстном меню вы-

берите строку Свойства. Или выделите значок с помощью клавиш управления курсором и нажмите Alt+Enter. И в том и в другом случае откроется окно Свойства В заголовке окна будет указано, свойства ярлычка какой именно программы вы сейчас изменяете.

Удалить непонравившийся вам значок можно, выделив его с помощью клавиш управления курсором, или щелкнув по нему мышкой, и нажав клавишу Delete. Windows запросит подтверждения удаления значка – щелкните по кнопке Да.

Вы можете удалить значок, перетащив его мышкой к значку Корзина. В этом случае подтверждения не потребуется.

1.20.6. Создание новой папки

Создать новую папку можно, так же как значок, щелкнув правой кнопкой мышки в поле рабочего стола и выбрав в контекстном меню строку Создать, в подменю строку Папка. На рабочем столе появится значок папки и под ним рамочка, в которой вы можете ввести название папки. Как только вы нажмете Enter создание папки будет завершено.

Создать папку, соответствующую каталогу на диске можно, дважды щелкнув по значку Мой компьютер, дважды щелкнув по значку одного из доступных вам устройств, например, по значку диска С: и выбрав последовательно пункты меню Файл>Создать>Папка. Введите с клавиатуры имя новой папки и нажмите Enter.

Чтобы создать подкаталог, войдите предварительно в одну из папок на диске С:, дважды щелкнув по ней мышкой. Команда создания новой папки доступна только в тех папках, которые соответствуют обычным дисковым каталогам. Папки Мой компьютер, Корзина и некоторые другие таковыми не являются.

1.20.7. Работа с объектами

Перетаскивание. Любые операции, связанные с переносом значков объектов, лучше всего выполнять правой кнопкой мыши.

Когда Вы отпустите правую кнопку, появится меню. В нем можно выбрать нужное действие или отказаться от операции.

Например, при перетаскивании правой кнопкой мыши значков файлов из одной папки в другую будет по умолчанию выбрана команда Переместить. Если надо скопировать файлы, выбирается команда Копировать. Для того, чтобы создать ярлык – команда Создать ярлык(и) .

Копирование объектов. Существует несколько способов копирования выделенных объектов в другую папку.

– Выбрать меню окна папки или Проводника *Правка>Копировать*, либо пункт контекстного меню

выделенных объектов *Копировать*, либо нажать *Ctrl+C*. После этого перейти в целевую папку и выбрать меню *Правка>Вставить* (клавиши *Ctrl+V*).

– Перетащить выделенные объекты в окно целевого каталога, удерживая *правую* кнопку мыши. Можно подтащить их к кнопке целевого каталога на панели задач и дождаться, пока откроется его окно. Отпустив кнопку мыши, выбрать из появившегося меню команду *Копировать*.

– Удерживая *Ctrl*, перетащить выделенные объекты в окно (или подтащить к кнопке) целевого каталога *левой* кнопкой мыши.).

– Если исходная и целевая папки находятся *на разных дисках*, можно перетаскивать объекты *левой* кнопкой мыши (не удерживая *Ctrl*)

Перенос объектов:

– Выбрать меню окна папки или Проводника *Правка>Вырезать (Edit>Cut)*, либо пункт контекстного меню выделенных объектов *Вырезать (Cut)*, либо нажать *Ctrl+X* или кнопку с ножницами инструментальной панели. После этого перейти в целевую папку и выбрать меню *Правка>Вставить (Edit>Paste)* (в контекстном меню, *Ctrl+V*, кнопка с буфером).

– Перетащить выделенные объекты в окно целевого каталога, удерживая *правую* кнопку мыши. Можно подтащить их к кнопке целевом каталога на панели задач и дождаться, пока откроется его окно. Отпустив кнопку мыши, выбрать из появившегося меню команду *Переместить (Move Here)*.

– Удерживая Shift, перетащить выделенные объекты в окно (или подтащить к кнопке) целевого каталога левой кнопкой мыши.

– Если исходная и целевая папки находятся на одном диске, можно перетаскивать объекты левой кнопкой мыши (не удерживая Shift).

Удаление объектов:

– Выбрать меню окна папки или Проводника *Файл>Удалить (File>Delete)*, либо пункт контекстного меню выделенных объектов *Удалить (Delete)*, либо нажать Del или кнопку с крестиком инструментальной панели. Подтвердить операцию удаления в появившемся диалоговом окне.

– Перетащить выделенные объекты в Корзину, удерживая *правую кнопку* мыши. Отпустив кнопку мыши, выбрать из появившегося меню команду *Переместить*.

– Перетащить выделенные объекты в Корзину, *левой* кнопкой мыши. Если вы удалили таким образом файлы с жесткого диска, то содержимое Корзины по-прежнему занимает дисковое пространство, но зато файлы можно вернуть. Для того, чтобы удалить файлы безвоз-

вратно, следует дать команду меню окна Корзина, или Проводника *Файл>Очистить корзину*, или контекстного меню *Очистить корзину*. Файлы с дискет удаляются раз и навсегда.

Переименование объектов. Для переименования объекта следует, выделив его, выбрать меню окна папки или Проводника *Файл>Переименовать*, либо пункт контекстного меню объекта *Переименовать* или щелкнуть мышью подпись выделенного объекта. После этого можно просто редактировать подпись. Переименовать сразу несколько объектов нельзя. Для того, чтобы переименовать любой экранный объект, удобнее всего щелкнуть мышью сначала по значку, а затем по его подписи. Значок сразу переключится в режим редактирования подписи.

Не забывайте о том, что длина имени объекта в Windows может составлять до 255 символов.

Отмена операций. Операции, совершенные с объектами средствами оболочки Windows, можно отменять. Для этого служит команда меню папки или Проводника *Правка>Отменить*,

либо команда контекстного меню *Отменить*, либо кнопка с круговой стрелкой влево на панели инструментов.

Для того чтобы отменить операцию удаления определенных объектов, проще всего выделить эти объекты в папке Корзина и выбрать команду меню папки или Проводника *Файл>Восстановить*, либо команду *Восстановить* контекстного меню.

1.20.8. Проводник Windows

Известные по работе в ОС DOS программы для работы с папками, файлами Norton Commander (Volkov Commander) не позволяют поддерживать некоторые специальные режимы Windows, например не поддерживают "длинные имена" (имена с числом символов больше 8). Кроме того, специально для Windows была написана другая Norton-образная программа Far (работа с FAR практически не отличается от работы с Norton Commander. Она не только поддерживает длинные имена файлов (до 255 символов), но даже удаляет их в Корзину. Для работы с файлами именно Far и Проводник используется наиболее часто. Так как программа Проводник (в отличие от FAR, которую приобретают отдельно) интегрирована в ОС Windows, рассмотрим работу с ней более подробно.

Проводник можно запустить из Главного меню, щелкнув по кнопке Пуск на панели задач и последовательно выбрав строки Программы, Проводник. Окно Проводника имеет все характерные атрибуты любого окна Windows. Кроме того, можно добавить еще два. Выберите пункт меню Вид (меню находится вверху окна, сразу под строкой заголовка).

Щелкнув мышкой, поставьте галочки в строках Строка состояния и Панель инструментов. В строке состояния, которая появится внизу окна вы, узнаете, сколько файлов и подкаталогов содержится в текущем каталоге, а также, что всегда полезно, сколько свободного места осталось на диске.

Окно Проводника соответствует лишь одному устройству. Если Вам необходимо работать с несколькими устройствами, то для каждого из них вам придется запустить свой Проводник. При этом устанавливать размеры и форму окон придется вруч-

ную. Окно Проводника также разделено на две части. Дерево каталогов в левой половине окна включает в себя устройства, а корнем дерева является Рабочий стол. Чтобы перейти с одного устройства на другое средствами Проводника, достаточно щелкнуть мышкой по значку соответствующего устройства на дереве каталогов в левой половине окна.

При работе с файлами в Windows следует помнить, что:

- допустимы имена файлов и папок длиной до 255 символов;
- имена могут содержать пробелы и любые символы за исключением? \ * " < > |;
- в локальной версии (русифицированной) в именах можно использовать буквы русского алфавита;
- можно использовать составные расширения, например, Документ.7896.doc;
- прописные и строчные буквы не различаются;
- можно использовать символы шаблона * и ?.

Для выполнения операций над файлом (файлами) предварительно надо выбрать его (их) одним из следующих способов:

- щелкнуть левой кнопкой мыши для выбора одного;
- щелкнуть левой кнопкой мыши на первом файле и, нажав и не отпуская клавишу Shift, щелкнуть на последнем файле смежного диапазона (выделится группа);
- щелкать мышью на каждом нужном файле удерживая нажатой клавишу Ctrl для выделения группы несмежных файлов;
- поставить курсор мыши на свободное место за файлами, которые необходимо выбрать, нажать правую кнопку мыши и растягивать появившуюся рамку, охватывая диапазон (в этом случае также можно использовать клавиши Shift и Ctrl для тех же целей).

При отпуске правой кнопки мыши появится контекстное меню для выбора нужной операции. В принципе для выделения можно использовать и левую кнопку мыши. Выбранные файлы выделяются темной подсветкой.

Копирование и перемещение файлов с помощью Проводника выполняется в следующей последовательности:

- 1 Выбрать файлы.

2 Если открыто только одно окно, то можно отбуксировать выбранные файлы из правого подокна в левое, совместив их со значком папки или диска, куда их надо скопировать или переместить. При этом по умолчанию предполагается выполнение операции копирования, если используются разные диски, и операция перемещения – на одном. Изменить назначение по умолчанию, можно используя клавишу Ctrl, удерживая которую при выполнении буксировки, изменяют выполняемую операцию.

Можно использовать при копировании и перемещении буфер обмена (clipboard). Для этого выделенные файлы сначала копируются в буфер обмена, а затем вставляются в новое место. При этом можно применять команды Вырезать, Копировать и Вставить из меню окна или же воспользоваться соответствующими кнопками панели инструментов.

Удаление файлов в Проводнике осуществляется нажатием клавиши Del для выделенного файла. Удаленный файл на самом деле не удаляется, а перемещается в папку называемую Корзиной (Recycled Bin). При этом он продолжает занимать место на диске. Чтобы действительно удалить файл, вернитесь на Рабочий стол и откройте папку Корзина двойным щелчком мышки. В открывшемся окне вы увидите

список удаленных файлов из разных каталогов. Если какой-нибудь из них вас заинтересует, и его удаление покажется вам преждевременным, то вы всегда можете восстановить его в прежнем каталоге, выбрав в меню Правка строку Отменить, или перетащить его в другой каталог любым известным вам способом. Чтобы окончательно удалить файлы из Корзины, выберите в меню Файл строку Очистить Корзину. Окно пустой Корзины можно закрыть, так же как любое окно.

Чтобы найти необходимый файл выберите последовательно пункты меню u1057 Сервис, Найти, Файлы и папки и в открывшемся диалоговом окне в строке Имя наберите искомое имя или маску. Потом щелкните мышкой по кнопке Найти. Если вы хотите ограничить область поиска, то в строке Папка необходимо предварительно указать имя каталога, в котором вести поиск.

1.20.9. Операции с дисками. Сжатие информации на диске

Копирование дискет. Для того, чтобы скопировать все содержимое одной дискеты на другую дискету той же емкости, можно воспользоваться меню *Файл>Копировать диск* или командой контекстного меню *Копировать диск*. Появится диалоговое окно. Выбрав исходный и целевой дисководы, нужно нажать кнопку *Начать* окна копирования. Имейте в виду, что операции копирования дискет создает точные копии дискет. Это значит, в частности, что все данные, которые были на целевой дискете до копирования, будут безвозвратно уничтожены.

Форматирование дисков. Диски необходимо форматировать для того, чтобы ими можно было пользоваться. Иногда дискеты продают уже форматированными. Кроме того, дискеты следует время от времени реформатировать, чтобы поддерживать их в рабочем состоянии.

Для форматирования диска служит меню *Файл>Форматировать* или команда *Форматировать* контекстного меню. Окно форматирования предоставляет на выбор три способа форматирования:

- Быстрое (очистка оглавления диска) – быстрое форматирование, очищающее дискету от содержащихся на ней данных.
- Полное – полная переразметка дорожек и секторов дискеты, занимает около минуты.

Информацию на диске можно обрабатывать с помощью специальных программ таким образом, чтобы она занимала меньший объем.

Существуют различные методы сжатия информации. Некоторые из них ориентированы на сжатие текстовых файлов, другие – графических, и т.д. Однако во всех них используется общая идея, заключающаяся в замене повторяющихся последовательностей бит более короткими кодами.

Сжатие информации используют, если объем жесткого диска недостаточен для хранения требуемого объема информации, если какая-то информация не используется длительное время, но удалять ее нецелесообразно, поскольку она может потребоваться позже, или если какую-то информацию, занимающую большой

объем, хотят перенести на другую ЭВМ с помощью небольшого количества дискет.

Сжатие всего диска используют редко, поскольку, во-первых, оно замедляет работу (при любом обращении к диску информацию нужно или сжимать при записи или возвращать к нормальному состоянию при считывании), во-вторых, информация на таком диске сложнее восстановить при каких-либо сбоях, например при заражении вирусами. Архивацию, т.е. выборочное сжатие определенных файлов, применяют гораздо чаще. Эти программы часто позволяют достичь высокой степени сжатия информации – в два раза и более. Наиболее часто для архивации используют программы WinZip, WinRar. Работа с этими программами достаточно проста. Наиболее удобно (если установлена одна из названных программ) использовать команды контекстного меню (при нажатии правой клавиши мышки, указав на подлежащий архивированию или разархивированию объект).

Встроенная в Windows функция архивации данных доступна, например, при нажатии правой клавишей мышки на один из логических дисков (C:, D:) в папке Мой компьютер.

1.20.10. Стандартные приложения Windows

В состав Windows включен ряд стандартных приложений, которые устанавливаются во время инсталляции системы или же могут добавляться по мере необходимости (Панель управления>Установка и удаление программ). В состав этих приложений включены: простейший текстовый редактор Блокнот (Notepad), текстовый редактор WordPad, графический редактор Paint, Программа просмотра изображений Imaging, Калькулятор, служебные программы и др. Доступ к этим программам осуществляется командой Пуск> Программы>Стандартные.

Рассмотрим некоторые из них более подробно. А для остальных приведем лишь краткий обзор возможностей стандартных программ (так как работа с ними интуитивно понятна и, кроме того, всегда имеется возможность просмотра помощи по каждой программе).

Текстовый редактор WordPad.

Приложение WordPad является полноценным текстовым процессором, который позволяет форматировать страницы и абзацы – устанавливать размеры страницы и ее ориентацию, границы текста, абзацный отступ, выравнивать текст, изменять шрифты, но и вставлять в документ графические объекты, звуковые фрагменты и видеоклипы, так как он поддерживает технологию внедрения и связывания объектов (OLE).

WordPad обеспечивает сохранение и чтение документов в различных форматах (Word, RTF – Rich Text Format, поддерживаемом большинством текстовых процессоров). Подробно описывать работу с программой WordPad не имеет смысла, так как она аналогична работе с текстовым процессором MS Word (работа с ним будет рассмотрена ниже), имеющим более широкие возможности.

Графический редактор Paint.

Графический редактор Paint пришел на смену Paintbrush из Windows 3.x, но в него добавлены новые функции, расширяющие возможности работы с графическими объектами: масштабирование рисунка, растягивание и вращение изображения, сохранение изображения в виде обоев рабочего стола и др.

Рабочую область окна Paint обычно называют *холстом*. Рисование на нем осуществляется с помощью мыши. Зафиксировав левую кнопку мыши и перемещая ее по холсту, можно рисовать прямые и кривые линии, а используя графические примитивы, встроенные в редактор (прямая, кривая, прямоугольник, овал, многоугольник, прямоугольник со скругленными углами), строить сложные фигуры.

Помимо этого предоставляется возможность использовать различные способы оформления изображения (заливка, регулируемой ширины распылитель, различной формы кисточка). Графический редактор позволяет создавать и текстовые фрагменты, применять при этом различные шрифты и стили (курсив, жирный, подчеркивание).

Управление редактором осуществляется с помощью меню (Файл, Правка, Вид, Рисунок, Параметры), а также панелей инструментов (инструментов для рисования и работы с текстом). Инструменты следующие:

- *Выделение области* (произвольной  или прямоугольной );
- *Замена цвета* на цвет фона (ластик ) и *Заливка* основным цветом ;
- *Копирование цвета*  и *Изменение масштаба* изображения ;
- *Карандаш* с изменяемой толщиной линии  и *Кисточка*  с изменяемой формой;
- *Распылитель* основного цвета  и *Ввод* текстовой информации  (действует только при масштабе изображения 1:1);
- *Рисование* прямой линии под любым наклоном и *Рисование* кривой линии  ;
- *Рисование* эллипса и *Рисование* прямоугольника со скругленными углами  .

Приемы работы в графическом редакторе Paint.

1 Для рисования правильной фигуры необходимо выбрать на панели инструментов ее вид и, удерживая клавишу Shift, выполнить формирование изображения, растягивая его в нужном направлении до получения фигуры требуемого размера.

2 Для заливки изображения задать на палитре нужный цвет, затем на панели инструментов выбрать *Заливка* и щелкнуть мышью внутри замкнутого контура фигуры (если контур будет не замкнутый, то краска как бы выплеснется и зальет весь холст).

3 При использовании инструмента *Многоугольник* можно рисовать любое количество сегментов линии, для получения закрашенного объекта необходимо ее замкнуть, соединив начальную и конечную точки.

4 Для копирования или перемещения части изображения необходимо сначала выделить нужную область с помощью инструмента *Выделить*, а затем:

- либо выполнить команды *Правка*> *Копировать* или *Правка*>*Вырезать* (выделенное изображение копируется или перемещается в буфер обмена), *Правка*>*Вставить*;
- либо щелкнуть мышью внутри выделенного объекта и при нажатой кнопке мыши перетащить его на новое место. Если

при этом держать нажатой клавишу Ctrl, то выполнится копирование.

5 Для изменения размера холста, единицы измерения и типа используемой палитры (цветная или черно-белая) используется команда Рисунок-Атрибуты.

6 Для формирования нового рисунка можно использовать следующие способы:

- в меню Рисунок выбрать команду Очистить;
- в меню Файл выбрать команду Создать.

7 Если необходимо изменить отдельные точки (пиксели) изображения, можно увеличить масштаб части рисунка, выполнив команду Масштаб одним из следующих способов:

- Вид>Масштаб>Крупный;
- выбрать инструмент Масштаб на панели.

8 Для удобства работы в режиме крупного изображения можно установить отображение сетки.

Для этого выполнить команду Вид>Масштаб>Показать сетку.

9 Для создания пользователем своих собственных цветов путем их смешивания используется один из способов:

- дважды щелкнуть на палитре;
- вызвать команду Параметры>Изменить палитру.

10 Созданную палитру можно сохранить на диске и использовать при следующих сеансах работы.

11 Сформированный рисунок можно поворачивать (Рисунок-Отразить>повернуть), растягивать (Рисунок>Растягивание>наклон) или обрабатывать его цвета с помощью команды Рисунок>Обратить цвета.

12 Сохранение созданного рисунка осуществляется по командам Файл>Сохранить или Файл>Сохранить как. Пользователю предоставляется возможность сохранить изображение в графическом формате BMP (монохромном, если рисунок не содержит цвета, 16-цветном – по умолчанию, 256-цветном – для рисунков с большим количеством цветов, 24-разрядном – для многоцветных сканированных изображений).

Программа Kodak Imaging.

Эта программа позволяет просматривать, аннотировать и выполнять основные задачи обработки графических документов, включая факсы и отсканированные изображения.

Программа Калькулятор.

Калькулятор имеет два режима работы: обычный, предназначенный для простейших вычислений, и инженерный, который обеспечивает доступ ко многим математическим (в том числе и статистическим) функциям.

Программа Портфель.

Портфель удобен при работе с несколькими копиями одних и тех же документов на разных компьютерах. Он позволяет, например, согласовать между собой копии документов на переносном и настольном компьютерах.

Программа связи (HyperTerminal) позволяет установить с помощью модема связь с удаленным компьютером даже в том случае, если на нем не загружена операционная система Windows. Эта программа предназначена также для отправки и приема файлов, подключения к электронным доскам объявлений и другим интерактивным службам.

При работе программы ведется протокол связи, который можно сохранить и затем распечатать.

Программа проверки диска относится к служебным программам и позволяет проверить жесткий диск на наличие логических и физических ошибок. После этого поврежденные области могут быть исправлены.

Программа дефрагментации относится к служебным программам и позволяет ускорить выполнение программ за счет перераспределения файлов и неиспользуемого объема на жестком диске.

Программа проверки системных файлов относится к служебным программам. Ее используют для проверки целостности файлов операционной системы, их восстановления в случае повреждения, а также для распаковки сжатых файлов (таких как драйверы) с установочных дисков. Программа проверки системных файлов позволяет также резервировать существующие файлы перед восстановлением исходных файлов. Кроме того, поль-

зователь имеет возможность задать настройку условий поиска по именам папок и расширениям имен файлов.

Программа *Сведения о системе* относится к служебным программам и выполняет сбор сведений о конфигурации системы. Программа содержит меню, позволяющее открывать связанные разделы описания системы. Сотрудникам служб u1090 технической поддержки для устранения неполадок в конфигурации системы требуется определенная информация о компьютере. Окно *Сведения о системе* позволяет быстро собрать данные, необходимые для устранения неполадок.

Системный монитор относится к служебным программам. Программу используют для наблюдения за быстродействием компьютера или сети. Каждый выбранный показатель отображается на диаграмме, которая обновляется через 5 секунд.

Ответы на возникающие вопросы при работе с операционной системой Windows легко получить, воспользовавшись справочной системой (команда Пуск>Справка).

1.20.11. Сетевые и коммуникационные возможности Windows

Наряду с локальными сетями, сетями внутри организаций (*интронет*), существует *глобальная компьютерная сеть Интернет*. Интернет был создан довольно давно и развивался как ведомственная сеть, принадлежащая министерству обороны США. Однако, он достаточно быстро стал доступным рядовым пользователям, а начиная с 1990 г., когда стало резко расти число его пользователей, и, особенно, с 1993 г., когда была изобретена система WWW (англ. World Wide Web, всемирная паутина), Интернет превратился в явление совершенно иного рода. Интернет – это огромный объем информации, доступный с любого компьютера, подключенного к сети, это новое средство общения и массовой информации, отличающееся от привычных открытостью, доступностью и демократичностью [1,2,10,22].

В основе устройства Интернета лежит система клиент-сервер. Информация в сети находится на огромном множестве серверов, разбросанных по всему миру. Для обращения к ним и просмотра получаемой оттуда информации на компьютерах у

пользователей сети устанавливаются специальные программы – клиенты (*браузеры*).

Каждый сервер Интернет имеет свой электронный адрес. Эти адреса называют *доменными*, поскольку они состоят из названий доменов (англ. Domain – область, регион). Имена доменов состоят из сегментов, названия которых записываются справа налево и разделяются точками, т.е. слева находится имя компьютера, справа имя домена первого уровня, соответствующего стране или, реже, какой-либо другой большой группе серверов. Например, *tstu.ru* имя домена *ru* означает Россия, *tstu* – имя компьютера-сервера Тамбовского государственного технического университета.

Чтобы использовать возможности работы в сети необходимо специальное программное обеспечение. В Windows встроены сетевые функции, которые обеспечивают функционирование системы в среде с несколькими сетевыми операционными системами и сетевыми протоколами. Доступ к другим компьютерам сети осуществляется с помощью папки Сетевое окружение. Окно папки Сетевое окружение содержит папки и файлы совместного использования на удаленном компьютере. Для обращения к файлам на другом компьютере используются так называемые сетевые имена, которые строятся по следующей схеме: \\имя компьютера в сети\диск:\путь\имя и расширение файла. Для обеспечения защиты информации от несанкционированного доступа и изменений сетевой администратор устанавливает права доступа к ресурсам сети. Могут быть определены следующие права: чтение файлов без права их изменения; чтение и изменение файлов; чтение и изменение защищенных паролем файлов при вводе пароля. Наивысшим приоритетом обладает администратор сети, который имеет права открывать все файлы и распоряжаться ресурсами по своему усмотрению.

Подключившись к Интернету и используя встроенную в ОС программу Обозреватель Internet Explorer, можно найти и просмотреть любую информацию в этой глобальной сети.

Проводник Windows и Internet Explorer позволяют объединить ресурсы Web в едином представлении. Рабочий стол Active Desktop делает возможной настройку рабочего стола, запуск

программ, переключение между файлами и отслеживание последних мировых новостей за счет объединения Web и рабочего стола пользователя. Active Desktop позволяет преобразовывать элементы Web в элементы рабочего стола и обновлять их в любое время.

Панель управления Windows включает программу Outlook Express, которая предоставляет защищенные средства для личной электронной почты и подключения к группам новостей. Для запуска Outlook Express достаточно нажать кнопку Пуск и выбрать команду Программы>Outlook Express.

Программа NetMeeting позволяет вести разговоры по цифровым каналам связи с родственниками, друзьями и деловыми партнерами по всему миру без больших расходов. Кроме того, NetMeeting делает возможной совместную работу группы пользователей любых приложений для Windows с помощью общей доски, текстовых сообщений и передачи файлов. Если на компьютере установлено необходимое оборудование, становятся возможными телеконференции в режиме реального времени.

1.20.12. Окончание работы Windows

Windows – достаточно сложная операционная система, чтобы можно было просто выключить компьютер. Для окончания работы закройте все программы, с которыми вы работали. (Если вы не сделаете этого, чуть позже вам все равно будет предложено это сделать, причем отдельно по каждой программе).

Щелкните мышкой по кнопке Пуск в панели задач. В открывшемся меню выберите строку Завершение работы и в открывшемся окне на вопрос Сейчас следует выключить компьютер? щелкните по кнопке "Да". (Для перезагрузки ОС прежде чем щелкать мышкой по кнопке "Да", нажмите и держите клавишу Shift. Тогда перезагрузка Windows осуществится быстрее и без дополнительных вопросов). Подождите пока "Идет подготовка к выключению компьютера" и только когда появится сообщение "Теперь можете выключить компьютер" – действительно выключайте.

1.21. Защита информации

При работе с операционными системами семейства Windows и перед тем, как рассматривать приложения семейства Microsoft Office, обязательно следует сначала коснуться основных положений, способов и методов, касающихся защиты информации, чтобы при изучении и работе с разного рода программами не совершать досадных ошибок, которые могут привести к потере информации.

Хорошо известно, что в современном мире информация имеет определенную, а часто и очень высокую ценность. Как и любую ценность ее нужно защищать. Защита необходима, например, от потерь из-за случайного удаления, сбоев, вирусов, несанкционированного доступа к информации.

Под *мероприятиями по защите от несанкционированного доступа* имеются в виду те, что связаны с секретностью информации. К их числу относятся самые разнообразные способы защиты, начиная от простейших, но очень эффективных защит паролем (при рассмотрении программ, имеющих такие возможности, на это будет указано дополнительно) до использования сложнейших технических систем.

Как показывает практика, вероятность взлома современных средств защиты информации гораздо ниже, чем вероятность доступа к секретной информации в их обход. Поэтому особое внимание следует обращать не только на системы защиты, но и на различные организационные вопросы – подбор людей, допускаемых к секретной информации, тщательное соблюдение правил работы с ней и т.д.

На сегодняшнее время никакая система защиты не обеспечивает стопроцентную надежность. Достаточно надежной считается такая система защиты информации, которая обеспечивает ее защиту в течении необходимого периода времени. Иными словами, система защита информации должна быть такой, чтобы на ее взлом потребовалось больше времени, чем время, которое эта информация должна оставаться секретной.

В случае если информация не является секретной наиболее простым и универсальным способом защиты информации является ее *резервное дублирование*. Действительно, если имеется ре-

зервная копия какого-то файла, например на дискете, магнитной ленте или магнитооптическом диске, то в случае порчи или потери основного файла его можно будет легко заменить. Очевидно, что резервное дублирование требует дополнительных устройств хранения информации, а значит определенных (а часто – весьма больших) материальных затрат. Поэтому резервное копирование, как правило, применяют только для наиболее ценной информации, потеря которой приведет к серьезным последствиям. Если вся информация, с которой ведется работа очень ценная, на компьютере иногда устанавливают два диска винчестера – один точная копия другого. Такое дублирование обеспечивает технология Rade, при которой информация одновременно дублируется на два винчестера. Кроме материальных затрат важно иметь в виду и человеческий фактор. Очень часто пользователь экономит несколько минут на создание резервной копии важного файла, а в результате теряет часы и дни работы на восстановление потерянной информации.

Большой вред информации наносят *компьютерные вирусы* – небольшие программы, которые без ведома пользователя, приписывая себя к другим файлам, проникают на компьютер через приносимые на компьютер диски или по компьютерной сети, распространяются на нем и производят нежелательные действия – уничтожают или модифицируют данные, нарушают работу программ.

Так как вирус – это программа, то заражение вирусом может произойти, если на компьютере хотя бы раз выполнена зараженная программа, которая, например, принесена с другого компьютера.

Мероприятия по защите от вирусов следующие.

- Аккуратность и внимательность при работе.
- Размещение ценной информации на защищенных от записи носителях.
- Своевременное удаление ненужных файлов и рациональное размещение файлов по каталогам во избежание неразберихи. С течением времени на диске появляется все больше и больше файлов, диск забивается. Постепенно пользователь забывает, что в каком файле находится и в каких каталогах (папках) содержит-

ся нужная информация. В результате, когда возникнет необходимость освободить место на диске, могут быть удалены файлы, содержащие ценную информацию. Поэтому необходимо периодически приводить диски в порядок.

– Быстрое восстановление ошибочно удаленных файлов при помощи специальных программ. Дело в том, что при удалении файла информация с диска не стирается, просто на его место разрешается запись другой информации. Если пользователь быстро обнаружил свою ошибку, у него остаются шансы восстановить случайно удаленную информацию, причем, если после удаления он не копировал, не перемещал другие файлы, не запускал другие программы или не перезапускал компьютер, эти шансы будут выше. Для восстановления ошибочно удаленных файлов существуют специальные программы, например Undelete в составе Norton Utilities. В ОС Windows копии удаленных файлов автоматически помещаются в специальную папку (каталог) – "корзину", откуда в случае необходимости их можно восстановить (при условии правильной предварительной ее настройки).

Мероприятия по защите от сбоев в работе устройств могут быть следующие.

– Наличие источников бесперебойного питания, что позволит корректно завершить работу компьютера при отключении напряжения и сохранить важную информацию.

– Использование современных защищенных операционных систем (Windows 2000, XP и т.д.), которые дают меньше сбоев при работе и имеют возможности по восстановлению поврежденных системных файлов.

– Периодическая проверка исправности оборудования (в частности – поверхности жесткого диска) при помощи специальных программ: Disk Doctor из состава Norton Utilities, ScanDisk из ОС Windows и др. Подобные программы позволяют обнаружить дефектные участки на поверхности диска и соответствующим образом их пометить, чтобы при записи информации эти участки были обойдены.

– Периодическая оптимизация (дефрагментация) диска для оптимального размещения файлов на нем, ускорения работы и уменьшения его износа. При записи на диск части файла могут

оказаться записанными в разных, удаленных друг от друга секторах диска, что связано с тем, что информация может быть записана только в свободные сектора. Для того, чтобы объединить эти фрагменты файлов и, тем самым уменьшить износ диска и затраты времени на считывание информации, следует периодически производить оптимизацию (дефрагментацию) диска при помощи соответствующих программ, например, Speed Disk из состава Norton Utilities, утилиты дефрагментации диска в Windows.

– Наличие системной дискеты, с которой можно запустить компьютер (т.е. загрузить операционную систему) в случае сбоя с основным системным диском. Напомним, что для того, чтобы компьютер заработал, необходимо загрузить в оперативную память операционную систему, основная часть которой находится в виде файлов на одном из дисков, называемом системным. Если с системным диском или с какой-то его частью, где находятся файлы операционной системы, что-то произошло, запустить компьютер с него не удастся, поэтому и нужно иметь резервный системный диск – дискету с соответствующими файлами.

В случае обнаружения заражения вирусами также следует перезапустить компьютер с резервной системной дискеты, поскольку операционная система на основном системном диске также может оказаться зараженной и, следовательно, при каждом включении компьютера и загрузке с основного системного диска операционной системы в оперативной памяти будут находиться вирусы. В такой ситуации борьба с вирусами, например возможна с использованием антивирусных программ запускаемых не только со своего компьютера, но и с дисков или дискет, а также с компьютеров локальной сети (в крайнем случае, можно снять винчестер и подключить его на другом компьютере для проверки). Весьма распространенными и удобными в эксплуатации являются антивирусные программы AVP Касперского, Доктор WEB, Nod, Avira. Использование этих программ (при условии своевременного обновления антивирусных баз, например через Интернет) значительно обезопасит Вас от потери информации из-за заражения вирусами.

1.22. Текстовый редактор документов Microsoft Word

Интегрированная программа *Microsoft Word* [2,3] предназначена для создания документов, включающих форматированный текст, таблицы, списки, формулы и другие объекты (рис. 1.17).

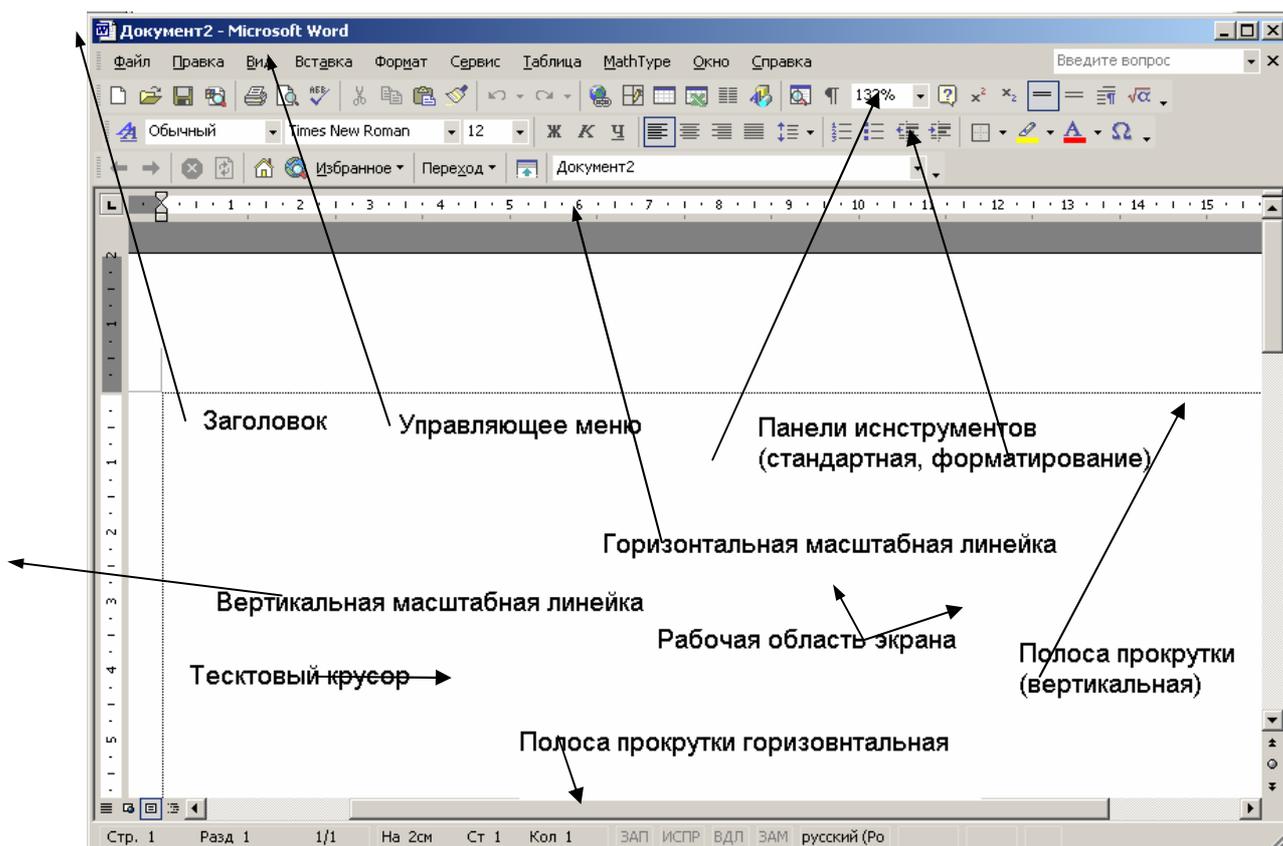


Рис. 1.17. Внешний вид программы Microsoft Word

Внешний вид экрана *Word*, как и любого окна *Windows*, включает рабочую область экрана, заголовок, управляющее меню, панели инструментов, полосы прокрутки и масштабные линейки. Основную площадь окна занимает рабочая область экрана, в которой выводится редактируемый документ. Основные команды по редактированию документа собраны на панелях инструментов. Различают стандартную панель, панели форматирования, рисования, таблиц и границ и др. Для выбора и подключения необходимой панели инструментов используется вкладка

управляющего меню **Вид**, категории **Панели Инструментов** (рис. 1.18).

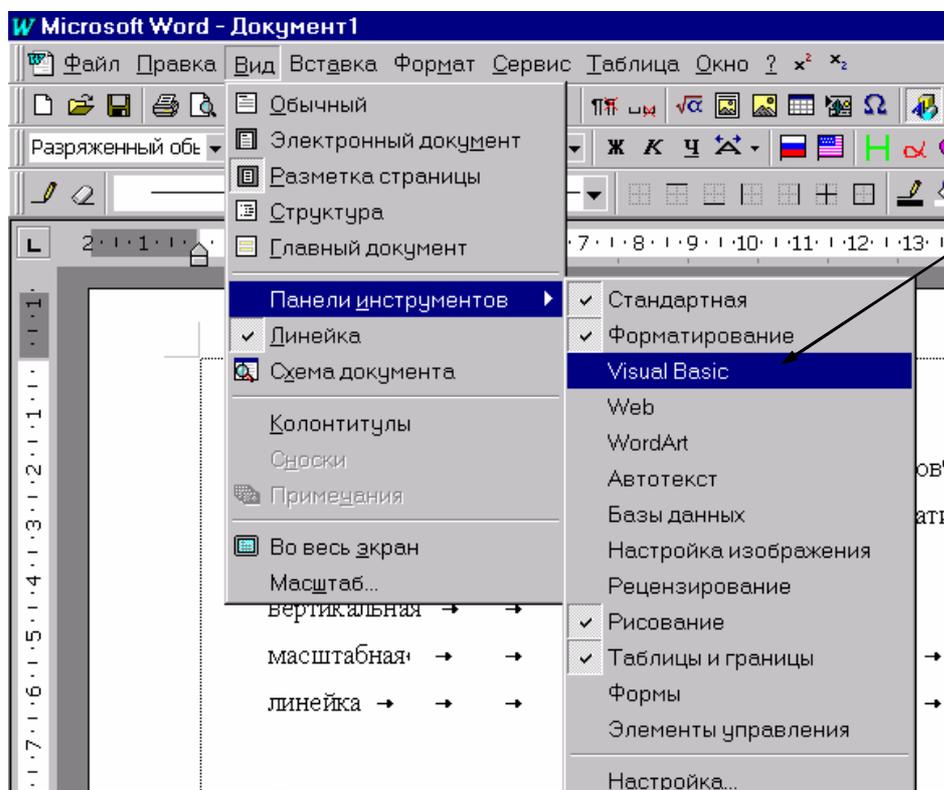


Рис. 1.18. Подключение панелей инструментов

Содержимое **Панели Инструментов** можно настраивать добавлением или удалением кнопок. Например, для вынесения команд нижний (x_2) и верхний (x^2) индекс необходимо проделать следующую последовательность действий. В категории **Вид**, вкладке **Панели Инструментов** выбрать пункт **Настройка**. В появившемся окошке выбрать вкладку **Команды**. Появится окошко (рис. 1.19), которое разделено на два поля: слева представлены все категории меню редактора Word, а справа команды, соответствующие выбранной категории. Для настройки панели инструментов (добавление команд) необходимо левой кнопкой мыши выбрать требуемую команду и, удерживая кнопку нажатой, переместить команду на панель инструментов. Для удаления команды с **Панели Инструментов** необходимо проделать обратную операцию, то есть перетащить удаляемую команду на окошко **Настройка** и команда удаляется.

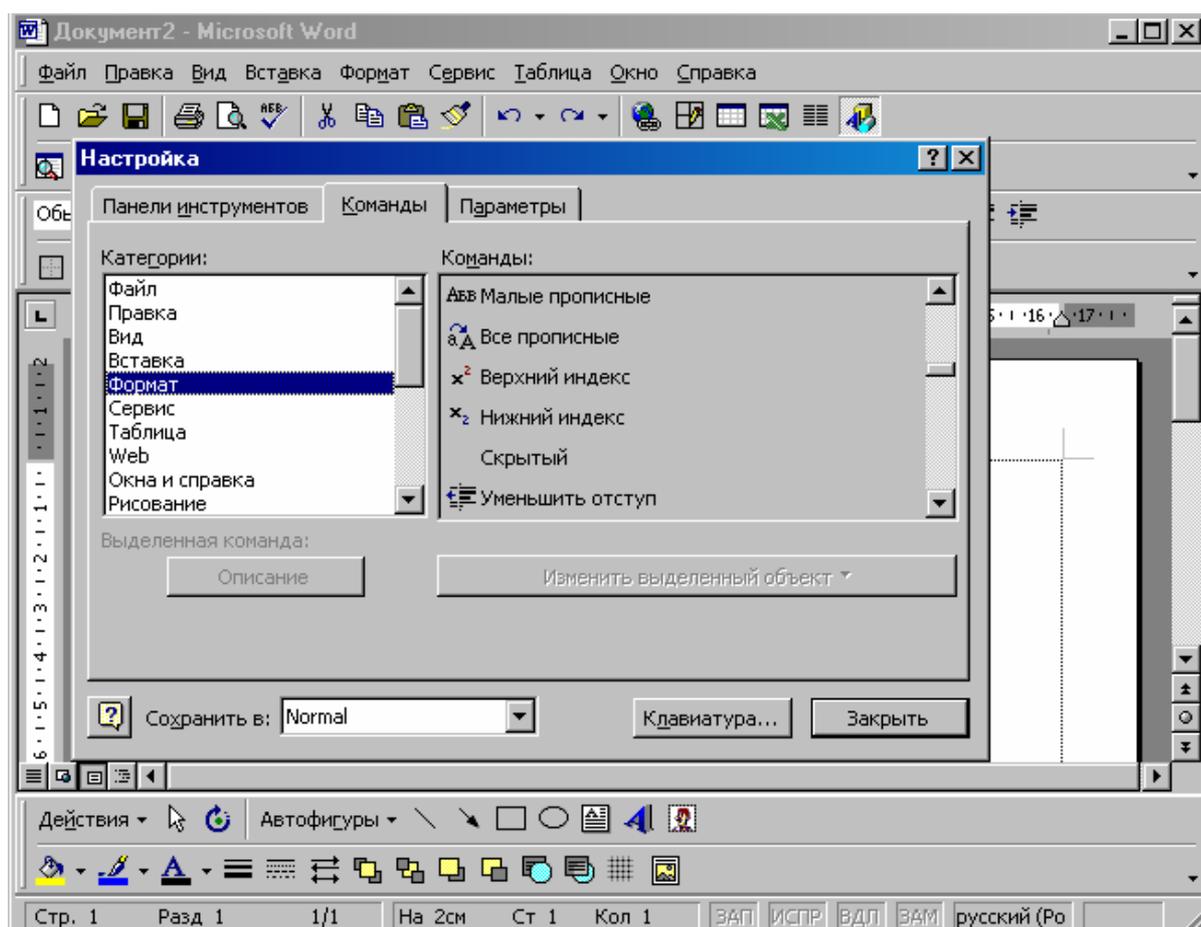


Рис. 1.19. Настройка панели инструментов

Word – это многооконный текстовый редактор, позволяющий редактировать несколько документов. Переключение между документами выполняется через вкладку **Окно** управляющего меню, в котором отображаются все имена файлов типа .doc, открытые на данный момент.

Ввод текста в *Word* осуществляется построчно с автоматическим переходом на новую строку. После нажатия клавиши «Enter» завершается текущий абзац и начинается новый абзац. Конец абзаца отмечается специальным служебным символом ¶ (рис. 1.20), который не отображается в документе при печати, то есть является непечатаемым символом. *Word* использует набор служебных непечатаемых символов для организации документа и облегчения ввода и редактирования текста. Просмотр непечат-

ных символов осуществляется нажатием кнопки ¶ на панели инструментов либо нажатием сочетания клавиш «Ctrl+*».

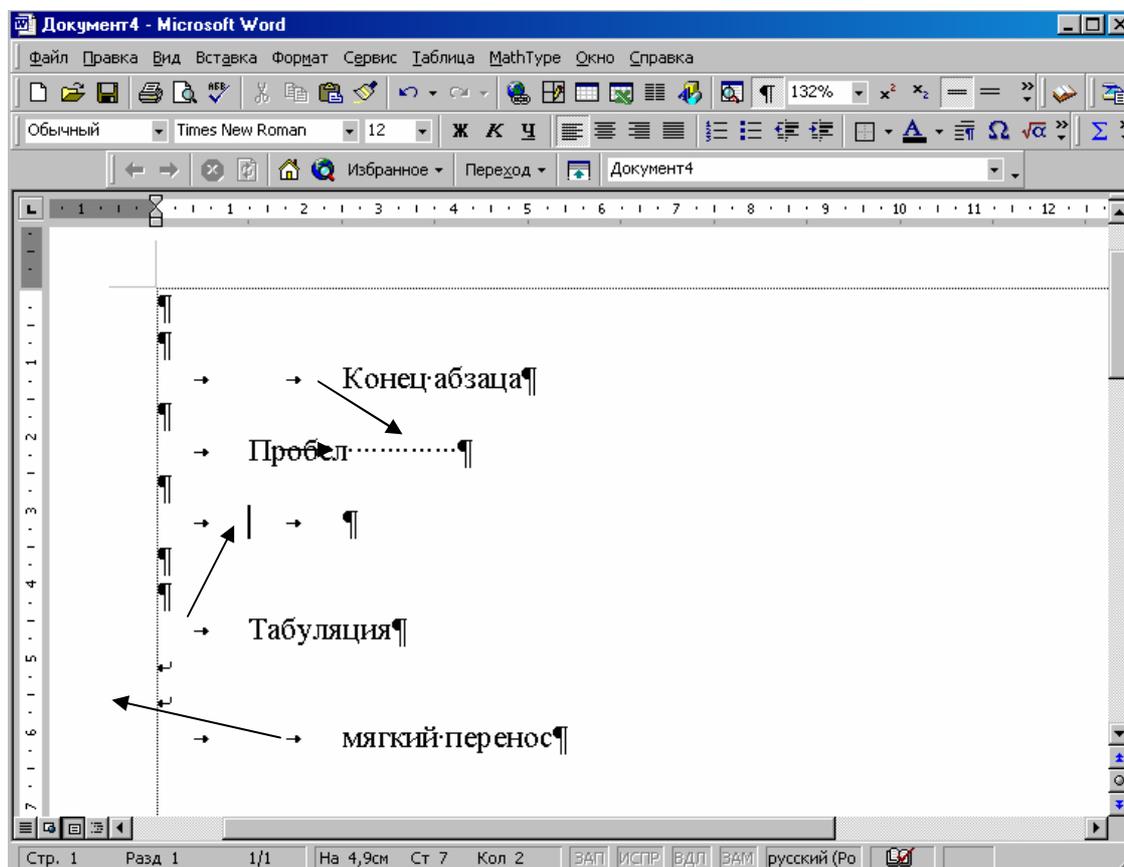


Рис 1.20. Пример непечатаемых символов

Основное назначение Word – создание форматируемых документов. Word предоставляет для организации работы несколько возможностей, а именно: использование шаблонов документов, создание и использование стилей, установление параметров форматирования вручную на текущий сеанс работы и т.д.

Шаблон – это файл, содержанием которого является образец оформления текущего документа. В шаблон входят набор панелей инструментов вместе со стилями, пользовательское меню, макросы (набор команд для выполнения некоторого множества операций). *Стиль* – это часть шаблона, регламентирующая применение совокупности форматов. Любой текст вводится на основе стиля, принятого по умолчанию. *Формат* – это составляющая стиля, регламентирующая правила создания объекта (символ, слово, строка, абзац и т.д.). В простейшем случае соз-

дание форматированного документа связано с установкой форматов:

- размер шрифта;
- тип шрифта;
- способ начертания;
- тип выравнивания;
- отступы левой и правой границ области ввода текста;
- отступ первой строки абзаца;
- межстрочное расстояние;
- ориентация бумаги;
- уровень вводимого текста.

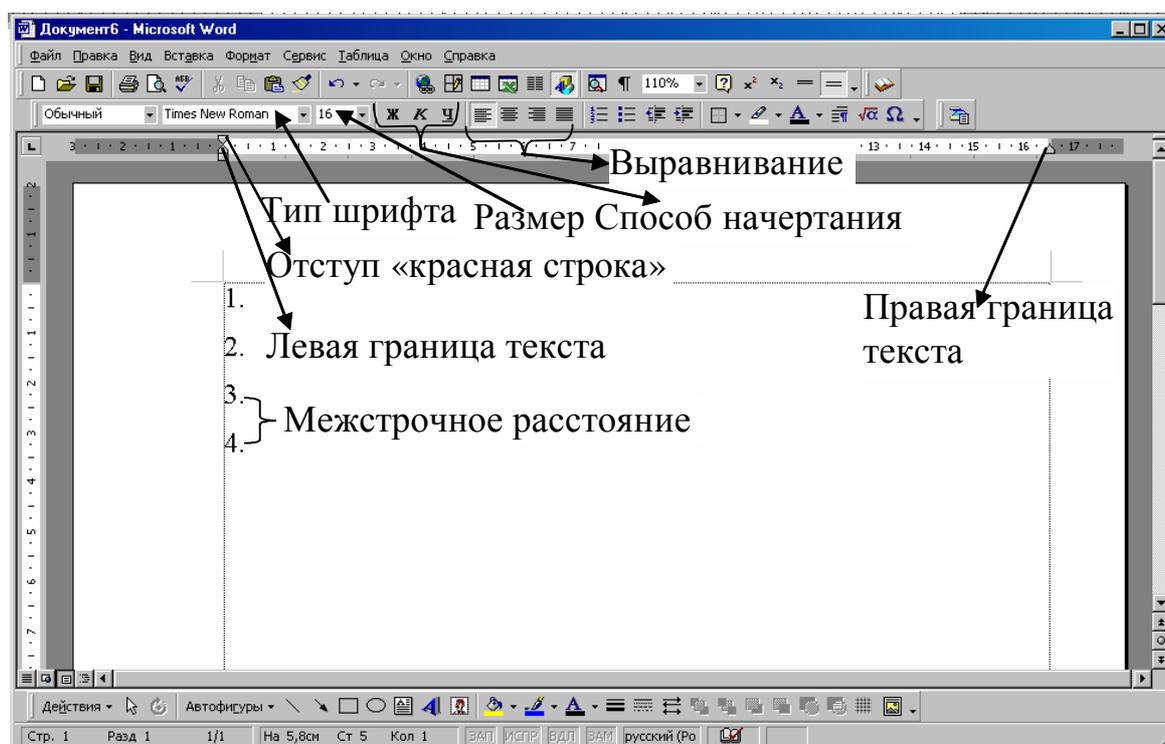


Рис. 1.21. Команды для форматирования текста

Большинство форматов имеет графическое представление в виде значков, расположенных на панели инструментов **Форматирование** (рис. 1.21). Формат **Тип шрифта** определяет написание символов. Стандартным шрифтом, соответствующим шрифту печатной машинки, считается Times New Roman. Стан-

дартный **размер шрифта** составляет 14 или 16 пунктов. Формат **Способ начертания** (представлен тремя значками **Ж**, **К**, **Ч**) определяет выделенное (**Ж**), наклонное (**К**) начертание или начертание с подчеркиванием (**Ч**). Формат **Тип выравнивания** определяет вид вводимого текста по правой и левой границам рабочей области экрана.

Вместе с тем отдельные форматы описываются в управляющем меню во вкладках **Формат** и **Файл**. Вкладка **Формат** в категории **Абзац** содержит такие способы оформления текста, как **межстрочное расстояние**, **первая строка**, **уровень текста**. В категории **Параметры** страницы вкладки **Файл** содержатся форматы **отступов полей печати** и варианты **ориентации бумаги** (рис. 1.22). Вкладка **Шрифт** категории **Формат** содержит способы оформления текста, такие как вид шрифта, размер, цвет, интервалы, варианты видоизменения (верхний/нижний индекс, зачеркнутый, контур и т.д.).

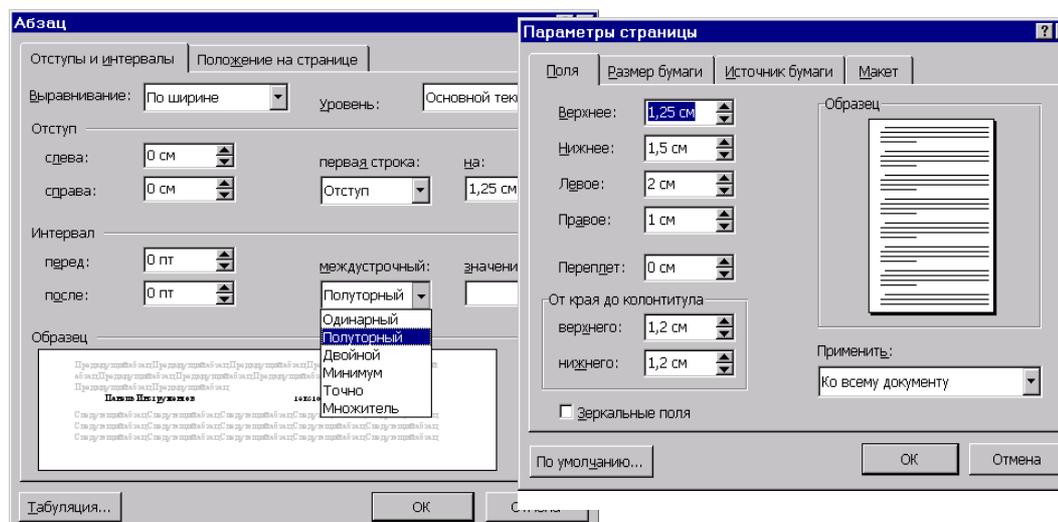


Рис. 1.22. Форматирование абзаца

Особенность задания форматов определяется тем, что действие форматов распространяется либо на вводимый текущий текст, либо на выделенный текст, введенный раньше.

Для работы с файлами в управляющем меню существует вкладка **Файл**, в которой расположены команды **Создать**, **Открыть файл**, **Сохранить**, **Сохранить как**. Первая команда предназначена для создания нового файла. Команда **Открыть**

файл осуществляет выбор существующего файла для последующего редактирования. Выбор выполняется обычным способом, принятым в Windows. По умолчанию выбор файла ограничен установленным фильтром (тип файлов) *документы Word*, который может быть изменен (рис. 1.23).

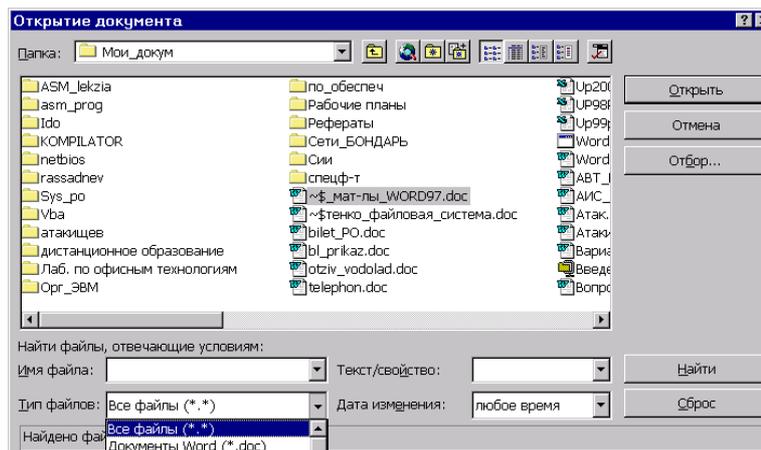


Рис. 1.23. Операция открытия документа типа Word

Команды **С**охранить, **С**охранить **к**ак предназначены для сохранения редактируемого документа в файл (рис. 1.24). Отличие между ними состоит в том, что последняя команда позволяет сохранить документ под новым именем, причем исходная версия сохранится под прежним именем.

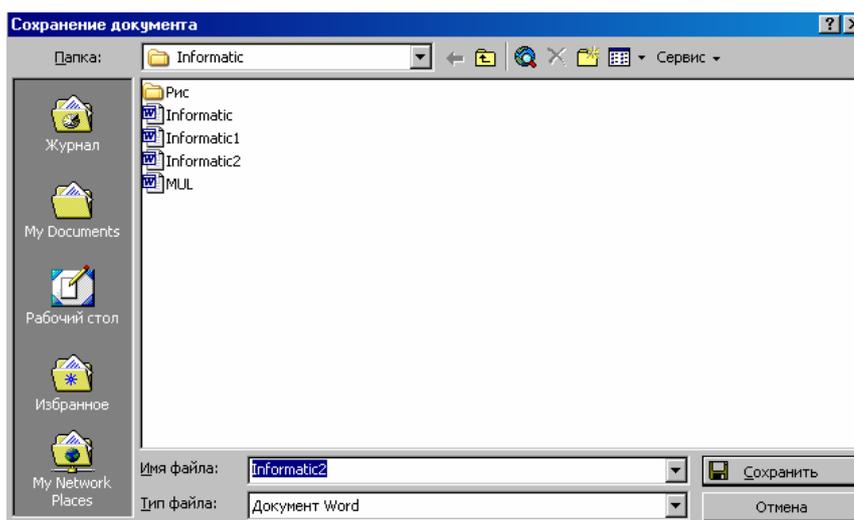


Рис. 1.24. Операция сохранения файлов

Создание списков облегчает восприятие перечислений. Списки подразделяются на нумерованные, маркированные и

многоуровневые. Для создания списков можно использовать команду управляющего меню **Формат\Список** или значки панели инструментов . Причем, при использовании значков создания списков, список формируется ранее созданного формата. Используя окошко **Список** (рис. 1.25) пользователь может изменять параметры списка.

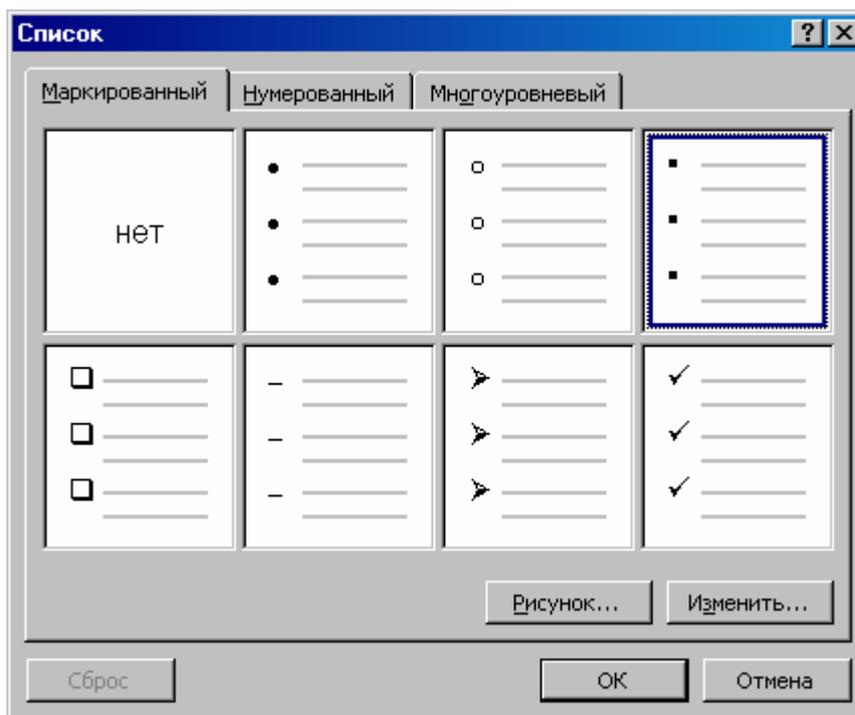


Рис. 1.25. Создание списков

Создание многоуровневого списка (рис. 1.26) выполняется в 2 этапа: сначала задается шаблон будущего списка; затем вводятся значения списка. Создание шаблона выполняется командой **Формат\Список\Многоуровневый\Изменить** и состоит в определении маркера списка для каждого уровня, его положения и отступа текста. Место элемента списка в многоуровневой иерархии определяется величиной отступа, которая задается командой **Увеличить отступ**, **Уменьшить отступ** .

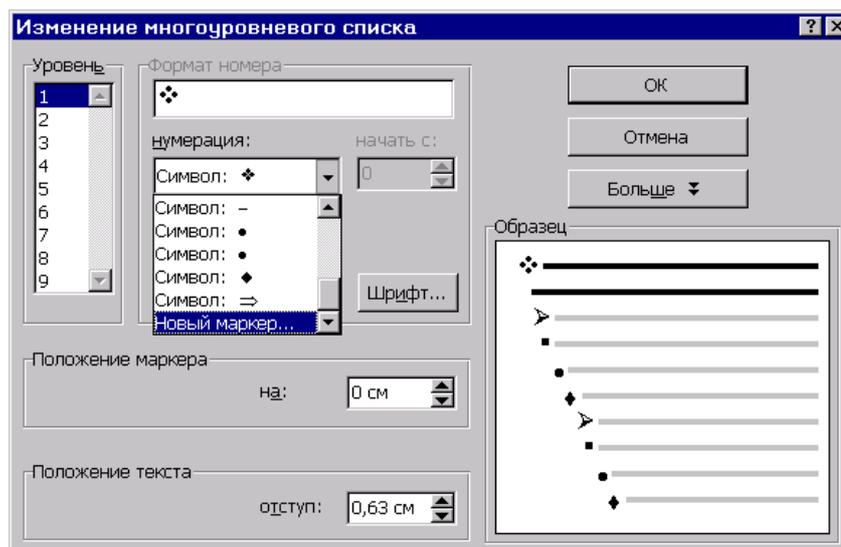


Рис. 1.26. Создание многоуровневого списка

Для работы с таблицами в управляющем меню имеется вкладка **Таблица**, содержащая все команды по созданию таблиц, редактированию данных и форматированию ячеек. Существует 2 способа создания таблиц (рис. 1.27):

- 1) создание стандартной таблицы командой **Добавить таблицу**;
- 2) создание нестандартной таблицы командой **Нарисовать таблицу**.

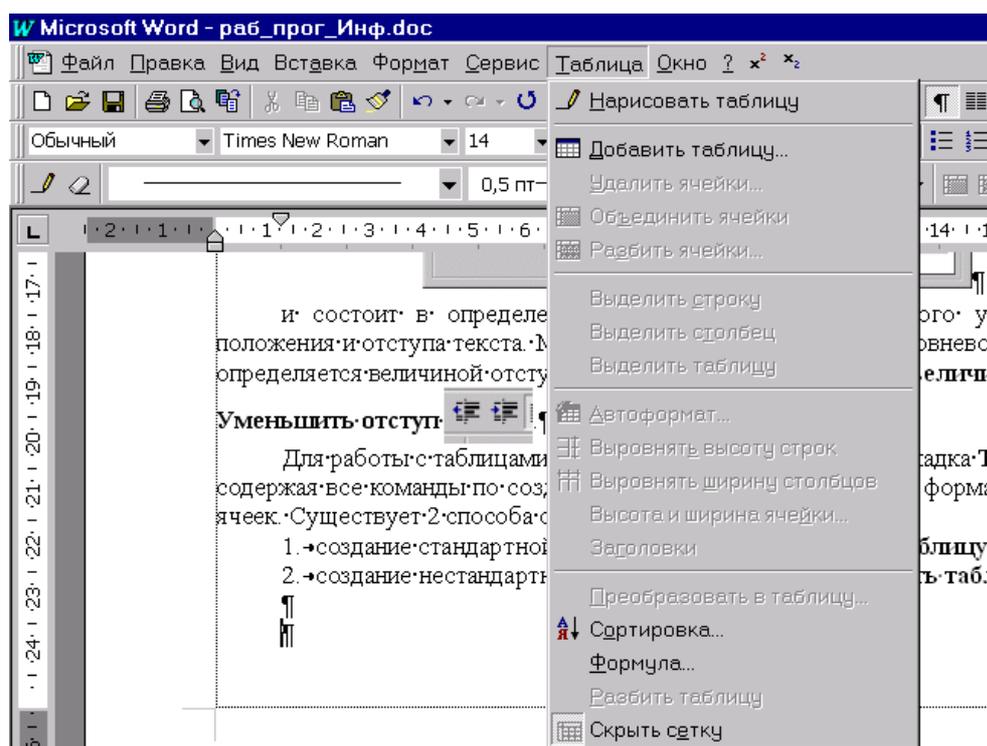


Рис. 1.27. Создание таблицы

Под *стандартной* таблицей условимся понимать таблицу из фиксированного количества ячеек (рис. 1.28).

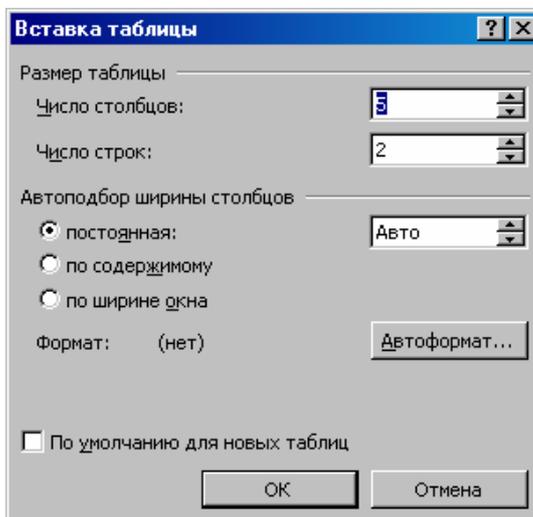


Рис. 1.28. Добавление стандартной таблицы

Под *нестандартной* таблицей понимают таблицу без фиксированного числа горизонтальных и вертикальных ячеек. Выбрав команду **Нарисовать таблицу**, указатель мыши приобретает форму карандаша, с помощью которого сначала рисуются внешние границы таблицы (прямоугольник), а затем произвольные вертикальные и горизонтальные разграничительные линии (рис. 1.29).

ЭТО НЕСТАНДАРТНАЯ ТАБЛИЦА				

Рис. 1.29. Создание нестандартной таблицы

Для работы с нестандартной таблицей используется панель инструментов **Таблицы и границы** (рис. 1.30), в которой, в частности, имеются команды вертикального выравнивания текста в ячейках и изменения направления ввода текста.



Рис. 1.30. Панель инструментов Таблицы и границы

Таблица состоит из ячеек, которые можно заполнять текстом или графикой. Текст вводится внутри в каждой ячейки с автоматическим переходом на новую строку точно так же, как и при вводе в обычный документ. Переход между ячейками таблицы осуществляется с помощью клавиши **<Tab>** для перемещения слева–направо, либо **<Shift+Tab>** для перемещения справа–налево. Добавление (удаление) строк (столбцов) производится с помощью меню **Таблица** вкладки **Добавить (Удалить)**. В этих вкладках необходимо выбрать определенное действие.

Работа с графикой в Word сводится к импорту графических изображений или созданию рисунков с помощью встроенного графического редактора. Вставка рисунка из файла выполняется командой **Вставка\Рисунок\Из файла**. В открывшемся диалоговом окне с помощью обычных средств следует вывести список графических файлов и выбрать нужный файл. В состав Word входят стандартные комплекты графических файлов, располагаемых в каталоге CLIPART.

Для создания графических рисунков используется панель инструментов **Рисование** (рис. 1.31).

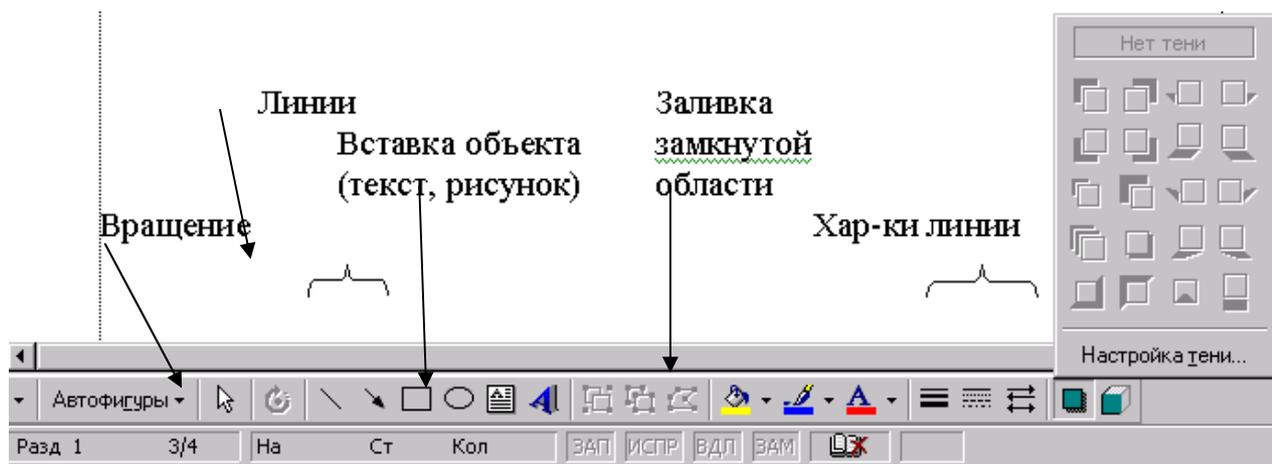


Рис. 1.31. Панель инструментов Рисование

Элементом панели рисование является инструмент WordArt (рис. 1.32), предназначенный для оформления заголовков и создания спецэффектов со строками. Вызов WordArt связан с выбором из коллекции надписей нужного варианта.

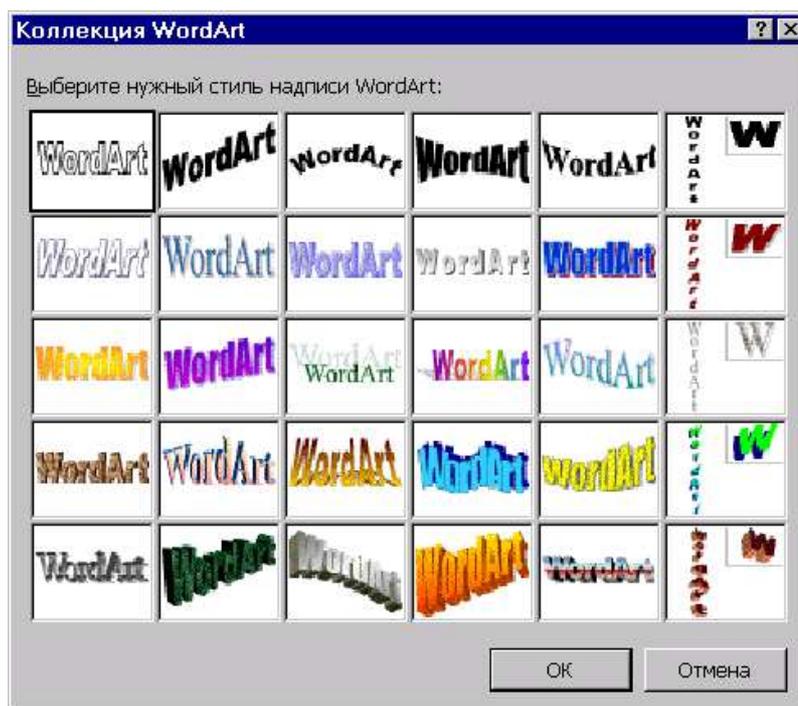


Рис. 1.32. Инструмент WordArt

В дальнейшем необходимо задать текст и отформатировать его. Например, применение WordArt позволит получить например, заголовок, изображенный на рис. 1.33.

Рис. 1.33. Текст, полученный с помощью инструмента WordArt

Таким образом, в настоящем пункте учебного пособия были рассмотрены основные функции и возможности текстового редактора *Microsoft Word*, позволяющие подготавливать, редактировать и оформлять текстовую документацию с использованием таблиц, списков, графики и рисунков.

Таким образом, были рассмотрены основные функции и возможности текстового процессора *Microsoft Word*, позволяющий подготавливать, редактировать и оформлять текстовую документацию с использованием таблиц, списков, графиков и рисунков.

1.23. Электронные таблицы Microsoft Excel

Программа обработки данных, обеспечивающая взаимодействие с пользователем при помощи выводимых на экран дисплея прямоугольных таблиц и работающая в диалоговом режиме называется **электронной таблицей** [4,5].

Данные, входящие в таблицы, можно автоматически представлять в виде графиков, диаграмм, гистограмм и т.д.

Пользователь работает в диалоге со специальной программой, которая позволяет заполнять ячейки нужным ему содержанием (текстами, числами или формулами для расчетов), очищать их, копировать и удалять, сортировать (т.е. располагать клетки, а также строки и столбцы в определенном порядке), производить вычисления над всей таблицей или ее частью, сохранять таблицу на диске и распечатывать частично или полностью на бумагу и т.д.

После запуска Excel экран содержит пять областей: окно книги, которое занимает большую часть экрана, строку меню, две или несколько панелей инструментов, строку формул и строку состояния. Вместе эти пять областей называются *рабочей областью Excel*.

Окно книги (рис. 1.34) составляет основную часть рабочей

области. **Рабочая книга** – документ приложения Excel. Книга состоит из нескольких рабочих листов, каждый из которых является электронной таблицей с широкими возможностями обработки данных. В нижней части окна книги размещаются ярлычки листов и кнопки их прокрутки, а в верхней части – строка заголовка. Кроме того, окно содержит листы и полосы прокрутки. Новая книга, показанная на рисунке, первоначально содержит три отдельных листа.

В нижней части окна книги находится несколько кнопок, с помощью которых вы можете переходить от одного листа к другому.

Когда видны не все ярлычки, для просмотра содержания книги можно использовать четыре кнопки, расположенные в нижнем левом углу окна. Две средние кнопки служат для прокрутки на один ярлычок влево или вправо. Крайние кнопки выполняют прокрутку к первому или последнему ярлычку книги. Можно изменить количество видимых на экране ярлычков, перетаскив маркер разделения ярлычков. Чтобы восстановить исходное положение маркера разделения ярлычков, достаточно дважды щелкнуть на нем.

Перечисленные кнопки прокрутки и маркер разделения ярлычков не активизируют листы книги. Чтобы сделать лист активным, вы должны после прокрутки ярлычков щелкнуть на ярлычке нужного листа.

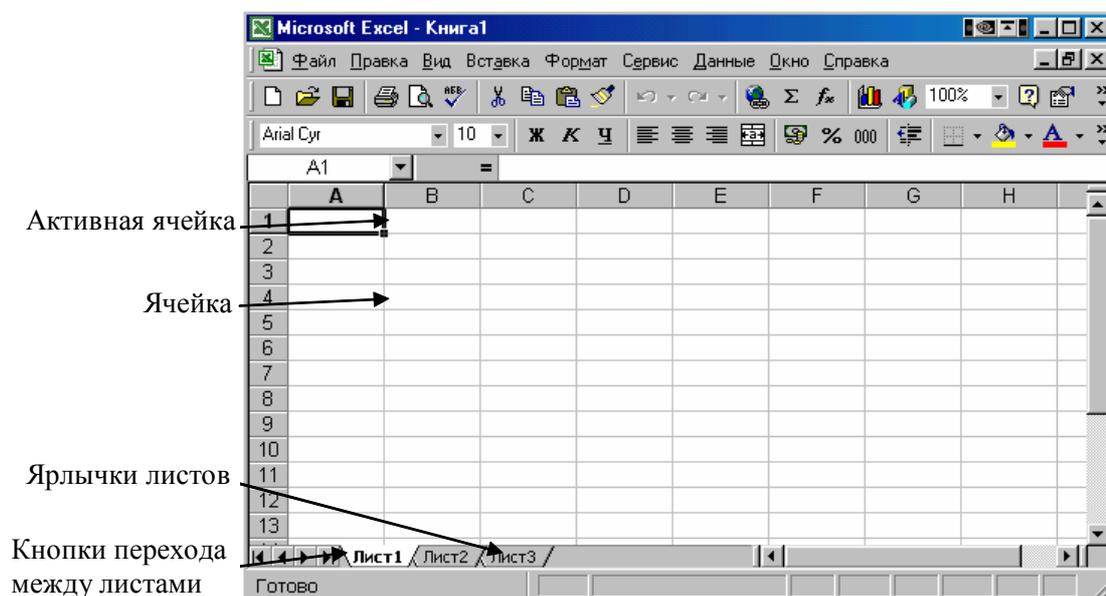


Рис. 1.34. Внешний вид окна Microsoft Excel

Рабочий лист книги разделен линиями сетки на строки и столбцы. Каждому столбцу соответствует определенная буква, которая появляется в качестве его заголовка. Заголовки столбцов могут принимать значения в диапазоне от А до IV (после столбца Z идет столбец AA, после AZ идет BA и так далее вплоть до IV). Каждой строке назначается целое число, которое выводится слева от сетки листа в качестве заголовка строки. Номера строк могут изменяться от 1 до 65 536.

На пересечении строки и столбца находится *ячейка (cell)*. Ячейки являются основными строительными блоками любого рабочего листа. Каждая ячейка занимает свое место на листе, где можно хранить и отображать информацию, и имеет уникальные координаты, которые называются *адресом ячейки* или *ссылкой*. Например, ячейка, находящаяся на пересечении столбца А и строки 1, имеет адрес А1. Ячейка на пересечении столбца Z и строки 100, имеет адрес Z100. Выделенную ячейку называют *активной ячейкой (active cell)*. Адрес активной ячейки выводится в поле имени, которое находится в левом конце строки формул.

Полосы прокрутки

Для просмотра содержимого листа можно использовать полосы прокрутки, расположенные вдоль правой и нижней сторон окна книги. Полосы прокрутки имеются только у активного окна книги, то есть у окна, в котором вы работаете.

Стрелки прокрутки на концах полос прокрутки позволяют за один раз перемещать лист на одну строку или один столбец. Щелчок на стрелке ▲ или стрелке ▼ на вертикальной полосе прокрутки сдвигает лист на одну строку вверх или вниз соответственно. Аналогично, щелчок по стрелке ► или стрелке ◀ на горизонтальной полосе прокрутки перемещает лист на один столбец вправо или влево. Прокрутка при нажатой клавише <Shift> позволяет в большом листе быстро перемещаться к столбцам или строкам, находящимся за пределами активной области.

Строка формул

Информацию можно вводить непосредственно в ячейку или с помощью строки формул. Содержимое активной ячейки всегда появляется в строке формул, это особенно важно, когда ячейка содержит формулу, как это показано на рис. 1.35.

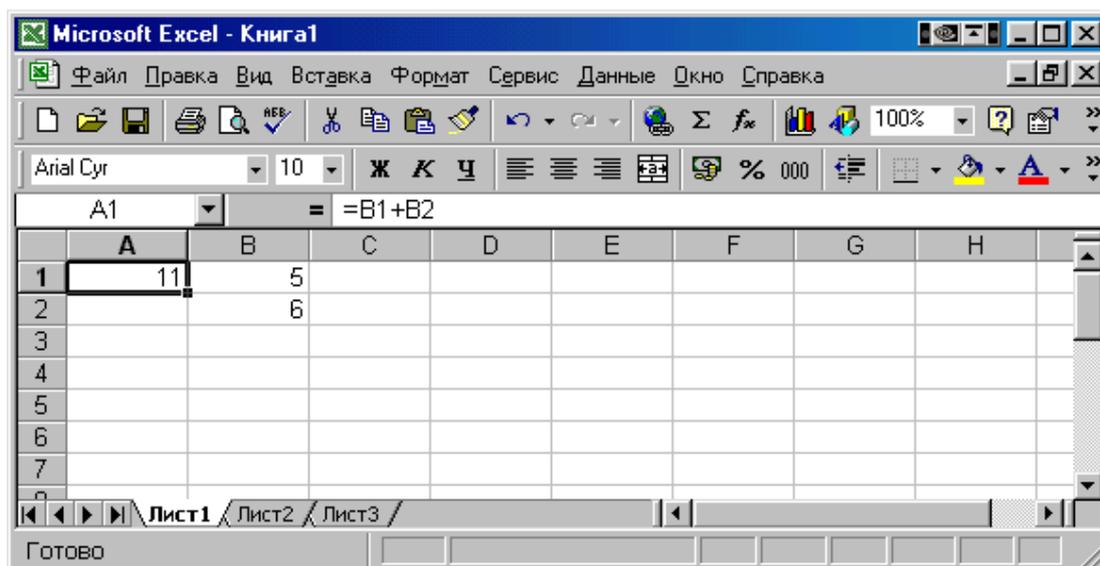


Рис. 1.35. Операции с формулами в Microsoft Excel

Кнопка  (Изменить формулу) отображается в строке формул только в случае ввода или редактирования данных в ячейке.

Хотя вы можете вводить информацию непосредственно в ячейку, использование строки формул имеет некоторые преимущества. Когда вы устанавливаете указатель на строке формул и нажимаете кнопку мыши, в строке появляются три кнопки. Кнопка с символом  называется кнопкой отмены, а кнопка  - кнопкой ввода. При нажатии кнопки ввода Excel «фиксирует» введенную в строке формул информацию и переносит ее в лист. Нажатие кнопки ввода аналогично нажатию клавиши <Enter> за исключением того, что нажатие <Enter> обычно дополнительно активизирует ячейку, находящуюся непосредственно ниже той, в которую вы вводили данные. Для удаления ошибочно введенной информации нажмите кнопку отмены или клавишу <Esc>.

Прежде чем выполнять какие-либо операции с ячейками, необходимо выделить ячейку или группу ячеек. Для выделения одной ячейки необходимо установить на ней указатель и нажать левую кнопку мыши. Вокруг ячейки появится рамка, показывающая, что данная ячейка является активной, а в поле имени

будет выведен ее адрес.

Выделение диапазонов осуществляется несколькими способами. Вместо перетаскивания указателя мыши по ячейкам, с которыми предстоит работать, необходимо указать любые два противоположных (по диагонали) угла диапазона. Этот прием известен как *расширение выделения*. В окне можно увидеть только часть рабочего листа, величина которой зависит от размеров экрана и его разрешения. Если необходимо выделить диапазон, выходящий за пределы окна книги, необходимо перетащить указатель за границы окна.

При необходимости можно перетащить указатель влево или вправо за границы окна книги, чтобы вывести на экран дополнительные столбцы и сделать нужное выделение.

Выделение больших диапазонов путем перетаскивания указателя занимает много времени. Существует другой способ выделения диапазона. Чтобы выделить, к примеру, диапазон A1:M38, проделайте следующие действия:

- 1) щелкните на ячейке A1;
- 2) пользуясь полосой прокрутки, выведите на экран ячейку M38;
- 3) нажмите клавишу <F8> (или <Shift>);
- 4) щелкните на ячейке M38.

При выделении нескольких диапазонов с помощью мыши используется клавиша <Ctrl>.

Чтобы выделить столбец или строку целиком, следует щелкнуть на заголовке столбца или строки. Первая видимая ячейка становится активной.

Чтобы ввести числовое значение, следует выделить ячейку и ввести с клавиатуры число. Вводимые цифры отображаются в строке формул и в активной ячейке. Мигающая вертикальная черта, которая появляется в строке формул и в активной ячейке, называется *точкой вставки* (insertion point).

По окончании ввода значения необходимо его зафиксировать, чтобы оно постоянно хранилось в ячейке. Простейшим способом фиксации ввода является нажатие клавиши <Enter>. После этого точка вставки исчезает, и Excel сохраняет введенное значение в ячейке.

Ниже дается описание символов, которые в Excel имеют специальное значение.

1. Если вы начинаете ввод числа со знака «плюс» (+) или «минус» (-), Excel опускает «плюс» и сохраняет «минус», в последнем случае интерпретируя введенное значение как отрицательное число.

2. Символ E или e используется при вводе чисел в экспоненциальном представлении. Например, Excel интерпретирует 1Е6 как 1000000 (единица, умноженная на десять в шестой степени).

3. Числовые значения, заключенные в круглые скобки, Excel интерпретирует как отрицательные. (Такая запись отрицательных чисел используется в бухгалтерском учете.) Например, значение (100) воспринимается Excel как -100.

4. Можно использовать десятичную запятую, как обычно. Кроме того, допускается вставлять пробел для отделения сотен от тысяч, тысяч от миллионов и т.д. Если ввести числа с пробелами, разделяющими группы разрядов, то в ячейках они появляются вместе с пробелами, а в строке формул - без пробелов.

5. Если начать ввод числа со знака доллара (\$), Excel применяет к ячейке денежный формат.

6. Если ввод числового значения заканчивается знаком процента (%), Excel применяет к ячейке процентный формат.

7. Если при вводе числового значения используется наклонная черта (/) и введенная строка символов не может быть интерпретирована как дата, то Excel рассматривает введенное значение как дробь.

Числовое значение отображается в ячейке не более чем 15 цифрами. Если вводится длинное числовое значение, которое не может быть выведено в ячейке, то Excel использует экспоненциальное представление числа. При этом точность значения выбирается такой, чтобы число можно было отобразить в ячейке.

Значения, которые появляются в ячейке, называются *выводимыми* или *отображаемыми* значениями (displayed values). Значения, которые хранятся в ячейках и появляются в строке формул, называются *хранимыми* значениями (underlying values). Количество выводимых цифр зависит от ширины столбца. Если

эта ширина недостаточна для вывода числа, то в зависимости от используемого формата отображения Excel может вывести либо округленное значение, либо строку символов #. Для отображения числа необходимо расширить столбец.

Ввод текста аналогичен вводу числовых значений. Чтобы ввести текстовое значение, необходимо выделить ячейку, набрать на клавиатуре текст и зафиксировать ввод, нажав клавишу <Enter> или кнопку ввода. Чтобы отказаться от ввода, следует нажать клавишу <Esc> или кнопку отмены.

Если текст не может быть полностью отображен в одной ячейке, Excel может вывести его, перекрывая соседние ячейки. Но текст при этом хранится в одной ячейке. Когда вводится текст в ячейку, которая перекрыта содержимым другой ячейки, то перекрывающий текст выглядит обрезанным.

Для переноса текста следует выделить ячейку, выбрать в меню **Формат (Format)** команду **Ячейки (Cells)** и на вкладке **Выравнивание (Alignment)** открывшегося окна диалога установить флажок **Переносить по словам (Wrap Text)**, затем нажать кнопку ОК.

Использование формул

Для ввода формул необходимо сначала выделить пустую ячейку, затем ввести знак "=", после чего ввести требуемую формулу, например "=10+5*2". Далее следует нажать клавишу <Enter>. В ячейке A10 появится значение 20, но при выделении ячейки A10 в строке формул будет выведена только что введенная формула.

Приоритет операторов

Термин «приоритет» (precedence) обозначает последовательность, в которой Excel обрабатывает операторы в формуле. Excel руководствуется следующими правилами:

- в первую очередь вычисляются выражения внутри круглых скобок;
- умножение и деление выполняются раньше сложения и вычитания;
- операторы с одинаковым приоритетом выполняются слева направо.

Использование ссылок в формулах

Ссылка (cell reference) является идентификатором ячейки или группы ячеек в книге. Создавая формулу, содержащую ссылки на ячейки, вы связываете формулу с ячейками книги. Значение формулы зависит от содержимого ячеек, на которые указывают ссылки, и оно изменяется при изменении содержимого этих ячеек.

В качестве упражнения введите формулу, которая содержит ссылку на ячейку. Выделите ячейку A1 и введите следующее:

= 10*2

Теперь выделите ячейку A2 и введите формулу:

=A1

Значения в обеих ячейках равны 20. Если вы измените значение в ячейке A1, значение в ячейке A2 также изменится.

Теперь выделите ячейку A3 и введите:

=A1+A2

Excel выведет значение 40. В дальнейшем вы убедитесь, что ссылки чрезвычайно полезны при создании и использовании сложных формул.

Ввод ссылок с использованием мыши

Например, чтобы ввести в ячейку B10 формулу со ссылкой на ячейки A9 и A10, выполните следующие действия:

- 1) выделите ячейку B10 и введите знак равенства;
- 2) щелкните на ячейке A9 и введите знак «плюс»;
- 3) щелкните на A10 и нажмите клавишу <Enter>.

Вводить значение в активную ячейку можно даже тогда, когда она не видна на экране. При вводе формулы можно прокручивать лист без изменения активной ячейки и щелкать на ячейках, расположенных в дальних областях листа. Какая бы часть листа ни была видна на экране, в строке формул выводится содержимое активной ячейки.

Относительная ссылка (Relative reference) указывает на ячейку, основываясь на ее положении относительно ячейки, в которой находится формула, например, «на две строки выше». Ссылки именно такого типа мы использовали в предыдущих примерах. **Абсолютная ссылка** использует для указания на ячейку ее фиксированное положение на листе, например «ячейка находится в столбце A строки 2». **Смешанная ссылка** содержит

относительную и абсолютную ссылку: например, «ячейка находится в столбце А и выше на две строки». Абсолютные и смешанные ссылки особенно полезны при копировании формулы из одного места листа в другое.

Относительная ссылка на ячейку А1 записывается так: =А1, а абсолютная ссылка на ячейку А1 имеет следующий вид: =\$А\$1.

Комбинируя абсолютные и относительные ссылки на ячейку А1, можно создать следующие смешанные ссылки: =\$А1, =А\$1.

Если символ доллара стоит перед буквой, то координата столбца абсолютная, а строки – относительная. Если символ доллара стоит перед числом, то, напротив, координата столбца относительная, а строки – абсолютная.

Формулы и функции

Все вычисляемые функции делятся на формулы и встроенные функции (рис. 1.36). Как уже отмечалось ранее, ввод формул и функций выполняется в строке формул или непосредственно в ячейке и всегда должен начинаться с «=».

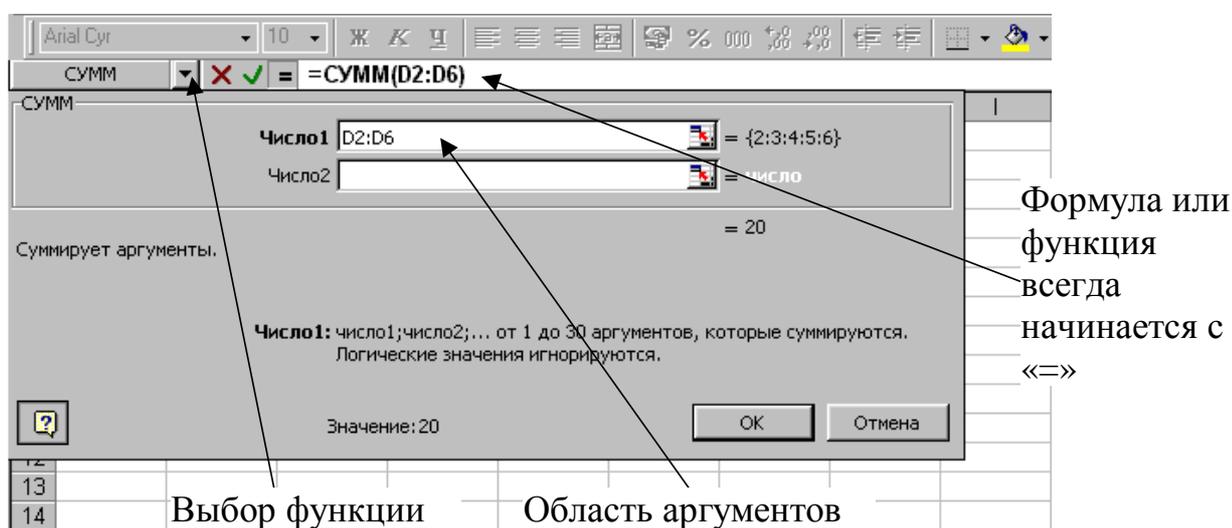


Рис. 1.35. Использование формул и функций в Excel

Операции, используемые в формулах

Основные операции, которые используются при создании формул, приведены в таблице 1.3.

Таблица 1.3.
Основные операции создания формул

Операция	Обозначение	Назначение
Арифметические	+, -, *, /, %, ^	
Текстовые	&	сцепление
Сравнения	=, <, >, <=, >=, <>	
Адресные	двоеточие (:), запятая (,) пробел ()	: - диапазон , - объединение СУММ (A12, F4)

Условное форматирование

Под «условным форматированием» (рис. 1.36) понимается написание данных от их значения (ФОРМАТ\УСЛОВИЕ–ФОРМАТИРОВАНИЕ). Преимущества условного форматирования следующие:

- не требует знания числового значения;
- содержит большое количество элементов форматирования;
- имеет возможность задания трех логических условий;
- в качестве условия использует логические формулы.

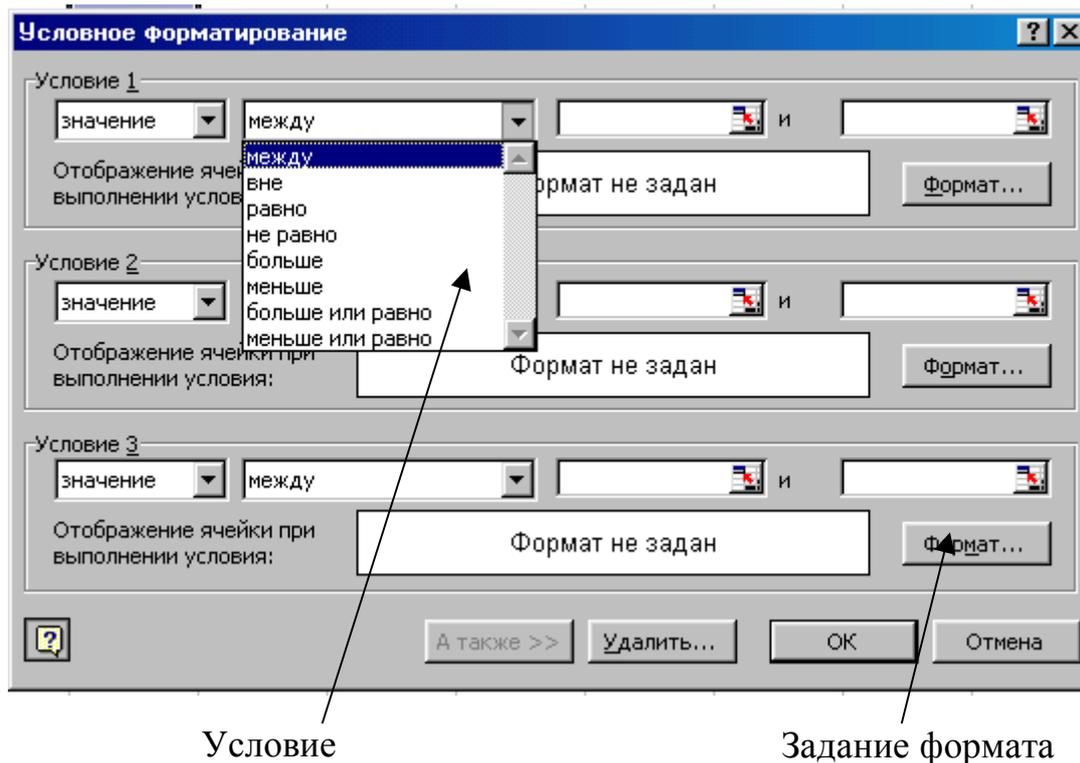


Рис. 1.36. Использование мастера условного форматирования в Excel

Список некоторых функций

ЕСЛИ (условие, знач. ДА, знач. НЕТ)

СУММ (ЧИСЛО 1, число2, ...)

СУММЕСЛИ (диапазон, условие) - суммирование ячеек при выполнении условия

СРЗНАЧ (число1, число2, ...) - вычисление среднеарифметического аргумента

СЧЕТ (значение1, значение2, ...) - подсчет количества чисел среди аргументов

СЧЕТА (значение1, значение2, ...) - подсчет количества непустых значений среди аргументов

СЧЕТЕСЛИ (диапазон, условие) - подсчет кол-ва ячеек в диапазоне, соответствующем условию

ТЕКСТ (значение, строка_формат) - преобразование числового значения в текст с указанным форматом

ДЛСТР (текст) - подсчет длины строки

МАКС (ЧИСЛО 1, число2, ...)

МИН (ЧИСЛО 1, число2, ...),

НД() - возврат значения ошибки #Н/Д, трактуемое как «нет доступного значения». Эту функцию следует использовать в ячейках, от которых зависят вычисления. В случае отсутствия значения вместо формулы будет выведено #н/д.

Форматирование листов

Команда **Ячейки (Cells)** меню **Формат (Format)** контролирует большинство форматов, применяемых к ячейкам рабочего листа. Выполнить форматирование ячеек очень легко: просто выделите ячейку или диапазон и выберите соответствующую команду в меню **Формат**. Например, для форматирования ячеек B2:G8, показанных на рис. 1.37, следует выполнить описанные ниже действия.

	A	B	C	D	E	F	G	H	I
1		Фактор 1	Фактор 2	Фактор 3	Фактор 4	Фактор 5			
2	Выборка 1	23	45	95	25	36	44,8		
3	Выборка 2	35	87	34	66	45	53,4		
4	Выборка 3	45	38	97	47	35	52,4		
5	Выборка 4	67	43	43	26	47	45,2		
6	Выборка 5	34	34	65	17	85	47		
7	Выборка 6	23	68	57	14	97	51,8		
8	Общее	227	315	391	195	345	294,6		
9									
10									
11									

Рис. 1.37. Пример форматирования ячеек при создании таблиц

1. Выделите ячейки B2:G8.
2. В меню **Формат** выберите команду **Ячейки**.
3. В открывшемся окне диалога щелкните на вкладке **Число (Number)**, если она не является активной.
4. В списке **Числовые форматы** выберите числовой формат.

5. Установите в поле **Число десятичных знаков** значение 1.

6. Нажмите кнопку **ОК**, чтобы вернуться в лист.

Форматирование с помощью кнопки <Формат> по образцу

Чтобы скопировать форматы в другое место листа, выполните описанные ниже действия.

1. Выделите ячейку или ячейки, из которых вы хотите скопировать форматы.

2. Нажмите кнопку  (Формат по образцу). Рядом с указателем появится значок маленькой кисти.

3. Выделите ячейку или ячейки, в которые вы хотите скопировать форматы.

Чтобы применить формат с помощью кнопки  панели инструментов, выделите ячейку или диапазон и затем нажмите кнопку, пользуясь мышью. Для удаления формата нажмите эту кнопку снова.

Форматирование чисел и текста

Команды меню **Формат** позволяют управлять отображением числовых значений и изменять вывод текста. Выберите в этом меню команду **Ячейки** (или просто нажмите клавиши <Ctrl+I>) и затем в открывшемся окне диалога **Формат ячеек** перейдите на вкладку **Число**, показанную на рис. 1.38.

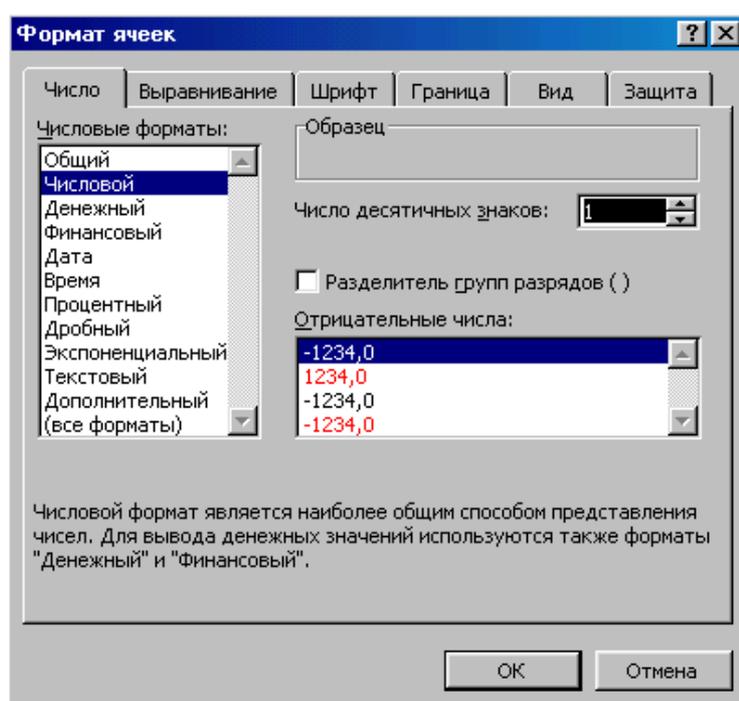


Рис. 1.38. Форматирование ячеек

Никогда не забывайте о различии между хранимыми и отображаемыми значениями. На хранимые числовые или текстовые значения в ячейках форматы не воздействуют. Например, при вводе числа с шестью десятичными знаками в ячейку, которая отформатирована с двумя десятичными знаками, число отображается только с двумя десятичными знаками, но хранимое значение, используемое в вычислениях, не изменяется.

Формат **Общий** - это первая категория в списке **Числовые форматы**. Если вы явно не измените формат ячейки, Excel отображает любое введенное текстовое или числовое значение в формате **Общий**. За тремя исключениями формат **Общий** отображает точно то, что вы ввели в ячейку. Например, при вводе 123,45 в ячейке будет выведено 123,45. Ниже приведены три исключения:

1) длинные числовые значения отображаются в экспоненциальной записи или округляются. Например, в ячейке со стандартной шириной формат **Общий** отобразит целое число *12345678901234* как *1.23457E+13*. Если ввести значение *123456,7812345* в ячейку со стандартной шириной и применить формат **Общий**, то будет выведено число *123456,8*;

2) формат **Общий** не отображает незначащие нули. Например, число *723,0* выводится как *123*;

3) десятичная дробь, введенная без числа слева от десятичной запятой, выводится с нулем. Например, *,123* выводится как *0,123*.

Выравнивание содержимого ячеек

Вкладка **Выравнивание** окна диалога **Формат ячеек** (рис. 1.39), контролирует расположение текста и чисел в ячейках. Эту вкладку можно также использовать для создания многострочных надписей, повторения ряда символов в одной или нескольких ячейках, изменения ориентации текста.

При выравнивании по левому краю вы можете изменять величину отступа, которая по умолчанию принимается равной нулю. При увеличении отступа на одну единицу значение в ячейке смещается на ширину одного символа вправо. Максимальная величина отступа составляет 15 единиц.

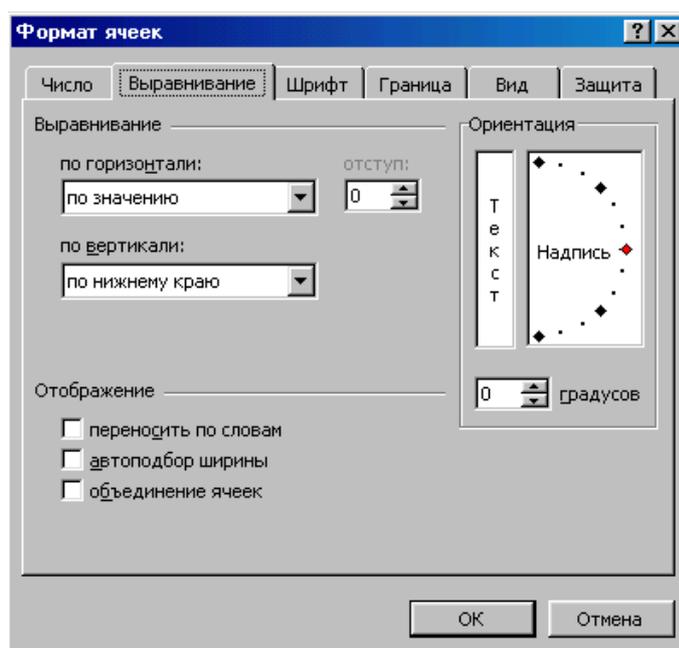


Рис. 1.39. Вкладка **Выравнивание** окна **Формат ячеек**

Формат **По центру выделения** позволяет центрировать текст от ячейки вдоль всех выделенных справа пустых ячеек до края выделения или до следующей непустой ячейки в выделении.

Область Ориентация позволяет размещать значения горизонтально (по умолчанию), вертикально сверху вниз или наклонно под углом до 90 градусов по часовой или против часовой стрелки. Excel автоматически настраивает высоту строки при вертикальной ориентации в том случае, если вы сами ранее или впоследствии не установите высоту строки вручную. Чтобы вернуть выделенные ячейки к их обычной ориентации, установите в горизонтальное положение ползунок **Надпись** в области **Ориентация**.

Флажок **Автоподбор ширины** в области **Отображение** на вкладке **Выравнивание** уменьшает размер символов в выделенной ячейке так, чтобы ее содержимое полностью помещалось в столбце. Это может быть полезно при работе с листом, в котором настройка ширины столбца по длинному значению имеет нежелательный эффект для остальных данных, или в том случае, когда использование вертикального или наклонного текста, переноса по словам является неприемлемым решением.

Изменение шрифта

Понятие *шрифт* включает в себя *гарнитуру*, например Helvetica, и ряд других атрибутов (размер, цвет и т. д.). В Excel выбор шрифта осуществляется на вкладке **Шрифт** окна диалога **Формат ячеек**. В рабочем листе шрифты используются для наглядного оформления информации различного типа и, в частности, для выделения заголовков. Чтобы задать шрифт для ячейки или диапазона, сначала выделите эту ячейку или диапазон. В меню **Формат** выберите команду **Ячейки** (или нажмите клавиши <Ctrl+I>) и затем перейдите на вкладку **Шрифт**.

Таким образом, применяя представленную информацию из первой главы, вы сможете качественно оформлять текстовую документацию на компьютере, используя при этом основные возможности прикладной программы Microsoft Word, а также приобретете навыки обработки чисел в электронных таблицах Microsoft Excel и представлять полученную информацию наглядно в виде графиков различных типов.

Таким образом, применяя представленную в данной главе информацию, вы ознакомитесь с понятием и смыслом термина «информатика», научитесь измерять полученную информацию в зависимости от того, к какой области она относится, узнаете, как происходит преобразование полученного сообщения в сигнал, передача его получателю и дальнейшей расшифровкой, ознакомитесь с фазами информационного цикла, его моделями, информационным ресурсом и его составляющими, а также с видами обработки данных. Кроме того, ознакомившись с материалами первой главы, вы сможете правильно и корректно работать, следить и управлять своими документами, не допускать их случайной порчи либо потери, защищать важную информацию, стараться не допускать проникновения вирусов с возможной дальнейшей полной либо частичной порчи ваших данных в наиболее популярной в настоящее время операционной системе Microsoft Windows. Также, вы научитесь корректно и в соответствии с современными требованиями оформлять текстовую документацию на компьютере, используя при этом возможности наиболее популярной прикладной программы Microsoft Word, а также при-

обрести навыки обработки чисел в электронных таблицах Microsoft Excel.

Контрольные вопросы

1. Дайте понятие термина информатика.
2. Какие научные дисциплины могут входить в термин «информатика» и почему? Обоснуйте свое мнение.
3. Что понимают под социальными, правовыми и этическими аспектами информатики?
4. Какими способами можно измерить информацию, содержащуюся в сообщении?
5. Дайте понятие первичного кода и поясните его смысл.
6. Какие бывают фазы информационного цикла? Поясните свой ответ на примерах.
7. Каковы составляющие информационного ресурса?
8. Какие бывают особенности в информационных технологиях?
9. Какие бывают виды и технологии обработки данных?
10. Опишите функциональную и структурную организацию компьютера.
11. Что значит равномерные и неравномерные коды? Приведите примеры таких кодов.
12. Какие вы знаете устройства хранения, управления и обработки информацией?
13. Как загрузить файл с диска?
14. Как перемещаться по тексту?
15. Как выделить фрагменты?
16. Как применяется поиск и замена?
17. Что такое автозамена и автотекст?
18. Как установить переносы?
19. Как просмотреть исправление?
20. Как вставить кадр? надписи?
21. Как скопировать текст или рисунок?
22. Как выровнять текст?
23. Как уменьшить или увеличить отступ?
24. Как поставить верхний или нижний индекс?
25. Как вставить автофигуру?

26. Как выполняется заливка?
27. Как пронумеровать страницы текста?
28. Для чего нужна табуляция?
29. Каким образом можно перейти с одного листа на другой?
30. Как переименовать рабочий лист другим именем?
31. Как добавить в документе еще один лист или удалить ненужный?
32. Как с помощью пользовательского меню можно изменить высоту строк и ширину столбцов?
33. Как промаркировать (выделить) одну или несколько ячеек с помощью: а) комбинации клавиш; б) мышки?
34. Как отменить последнее действие с помощью: а) клавиш; б) мышки?
35. Как скопировать содержимое групп ячеек с помощью: а) комбинации клавиш; б) мышки;
36. Как удалить содержимое группы ячеек и поместить его на новое место в таблице с помощью: а) комбинации клавиш; б) мышки;
37. Как изменить формат представления чисел?
38. Как осуществить выравнивание содержимого ячеек?
39. Как изменить шрифт и размер в определенной группе ячеек?
40. Какой знак должен предшествовать вводу формул?
41. Как просуммировать содержимое группы ячеек?
42. Как вычислить среднее значение содержимого ячеек?
43. Как можно редактировать содержимое ячейки?
44. Как осуществить обращение к значению ячейки, расположенной на другом листе?
45. Как построить диаграмму по данным таблицы?
46. Как изменить надписи на осях?
47. Что нужно сделать, чтобы добавить к диаграмме дополнительный объект?
48. Каким образом можно вставить рисунок в таблицу?

1. Постройте таблицу Microsoft Excel.

2. Постройте по данным таблицы диаграмму.

3. Измените тип диаграммы.

4. Осуществите удаление данных, используя: а) пользовательское меню; б) нажатие клавиши; в) нажатие правой клавиши мыши.

2. ОБЩИЕ СВЕДЕНИЯ О ПРЕДСТАВЛЕНИИ ИНФОРМАЦИИ В ЭВМ

Объектом передачи и преобразования в ЭВМ является *дискретная информация*. Для представления её применяется так называемый алфавитный способ, основой которого является использование фиксированного конечного набора символов любой природы, называемого алфавитом [6, 13,18].

Примерами таких алфавитов могут служить и алфавиты естественных человеческих языков, совокупность десятичных цифр, любая другая упорядоченная совокупность знаков, предназначенная для образования и передачи сообщений. Символы из набора алфавита называются *буквами*, а любая конечная последовательность букв – *словом* в алфавите. При этом не требуется, чтобы слово обязательно имело языковое смысловое значение. Например, словами являются последовательности символов, составленные из алфавита, включающего буквы и цифры:

2006 AAAA CARCASS SENTENCED ПАСКАЛЬ

Все процессы, происходящие в вычислительной системе, связаны непосредственно с различными физическими *носителями информационных сообщений*, а все узлы и блоки этой системы являются физической средой, в которой осуществляются информационные процессы. Особенности носителя информации накладывают определённые ограничения на используемый для её представления алфавит. При подготовке к решению на ЭВМ исходная формулировка, описание метода решения, задание конкретных исходных данных осуществляются на математическом языке, алфавит которого, наряду с буквами других языков, составляют специальные символы знаков математических операций и другие знаки.

В процессе ввода, хранения, вывода и обработки информации в ЭВМ осуществляется неоднократное её преобразование из одной формы представления в другую. При этом с каждой из используемых форм представления информации связаны различ-

ные алфавиты. Для представления информации на перфокартах и перфолентах используется, например, алфавит, включающий всего две буквы (есть пробивка, нет пробивки). Таким образом, процесс преобразования информации часто требует представлять буквы одного алфавита средствами (буквами, словами) другого алфавита. Такое представление называется *кодированием*.

Декодированием называется процесс обратного преобразования информации относительно ранее выполненного кодирования.

Для представления информации в ЭВМ преимущественное распространение получило *двоичное кодирование*, при котором символы вводимой в ЭВМ информации представляются средствами *двоичного алфавита*, состоящего из двух букв.

Двоичный алфавит по числу входящих в него символов является минимальным, поэтому при двоичном кодировании алфавита, включающего большее число букв, каждой букве ставится в соответствие последовательность нескольких двоичных знаков или двоичное слово. Такие последовательности называют *кодowymi комбинациями*.

Полный набор кодовых комбинаций, соответствующих двоичному представлению всех букв кодируемого алфавита, называется *кодом*.

Различают коды равномерные и неравномерные. Кодовые комбинации равномерных двоичных кодов содержат одинаковое число двоичных знаков, неравномерных – не одинаковое.

Примером неравномерного двоичного кода может служить азбука Морзе, в которой для каждой буквы и цифры определена двоичная последовательность коротких и длинных сигналов. В азбуке Морзе буква Е, например, соответствует один короткий сигнал (точка), а букве Ш – четыре длинных сигнала (четыре тире). Неравномерное кодирование позволяет повысить скорость передачи сообщений за счёт того, что наиболее часто встречающимся в передаваемых текстах символам (к ним относится и буква Е) назначается для её представления более короткая комбинация.

В технике наибольшее применение нашли равномерные коды, как более удобные для реализации. Например, во втором

международном телеграфном коде символы передаваемого алфавита кодируются последовательностями из пяти «токовых или безтоковых посылок».

В математике широко используется две формы записи чисел: *естественная* и *нормальная*.

При *естественной* форме число записывается в естественном натуральном виде, например: 12 560 – целое число, 0.003572 – правильная дробь, 4.89760 – неправильная дробь.

При *нормальной* форме запись одного числа может быть различной в зависимости от ограничений, накладываемых на её форму. Например, число 12560 может быть записано так:

$$12\ 560 = 1.256 \cdot 10^4 = 0.1256 \cdot 10^5 = 125\ 600 \cdot 10^1 \text{ и т.д.}$$

Так как числа бывают положительные и отрицательные, то в разрядной сетке при их машинном представлении один разряд отводится под знак числа, а остальные образуют поле числа. В знаковый разряд, который может располагаться как в начале, так и в конце числа, записывается информация о знаке числа. Примем, что знак положительного числа «+» изображается символом 0, а знак отрицательного числа «-» – символом 1. Если поле числа включает n разрядов, то диапазон представления целых чисел в этом случае ограничивается значениями $-(2^n-1)$ и (2^n-1) .

Представление чисел в ЭВМ в нормальной форме называют представлением числа в форме с плавающей запятой, так как положение запятой в записи числа в этом случае не является однозначным.

2.1. Числа конечной точности

Когда люди выполняют какие-либо арифметические действия, их не волнует вопрос, сколько десятичных разрядов занимает то или иное число. Физики, к примеру, могут вычислить, что во Вселенной присутствует 10^{78} электронов, и их не волнует тот факт, что полная запись этого числа потребует 79 десятичных разрядов. Никогда не возникает проблема нехватки бумаги для записи числа.

С компьютерами дело обстоит иначе. В большинстве компьютеров количество доступной памяти для хранения чисел фиксированно и зависит от того, когда был разработан этот ком-

пьютер. Если приложить усилия, программист сможет представлять числа в два, три и более раз большие, чем позволяет размер памяти, но это не меняет природы данной проблемы. Память компьютера ограничена, поэтому мы можем иметь дело только с такими числами, которые можно представить в фиксированном количестве разрядов. Такие числа называются *числами конечной точности*.

Рассмотрим ряд положительных целых чисел, которые можно записать тремя десятичными разрядами без десятичной запятой и без знака. В этот ряд входит ровно 1000 чисел: 000, 001, 002, 003, ..., 999. При таком ограничении невозможно выразить определенные типы чисел. Сюда входят:

- 1) числа больше 999;
- 2) отрицательные числа;
- 3) дроби;
- 4) иррациональные числа;
- 5) комплексные числа.

Одно из свойств набора всех целых чисел – замкнутость по отношению к операциям сложения, вычитания и умножения. Другими словами, для каждой пары целых чисел i и j числа $i+j$, $i-j$ и $i \times j$ – то же целые числа. Ряд целых чисел не замкнут относительно деления, поскольку существуют такие значения i и j , для которых i/j не выражается в виде целого числа (например, $7/2$ или $1/3$).

Числа конечной точности не замкнуты относительно всех четырех операций. Ниже приведены примеры операций над трехразрядными десятичными числами:

$$600+600=1200 \text{ (слишком большое число);}$$

$$003-005=-2 \text{ (отрицательное число);}$$

$$050 \times 050=2500 \text{ (слишком большое число);}$$

$$007/002=3,5 \text{ (не целое число).}$$

Отклонения можно разделить на два класса: 1) операции, результат которых превышает самое большое число ряда (ошибка переполнения) или меньше, чем самое маленькое число ряда (ошибка из-за потери значимости), и 2) операции, результат которых не является слишком маленьким или слишком большим, а просто не является членом ряда. Из четырех примеров, приве-

денных выше, первые три относятся к первому классу, а четвертый – ко второму классу.

Поскольку размер памяти компьютера ограничен и он должен выполнять арифметические действия над числами конечной точности, результаты определенных вычислений будут неправильными с точки зрения классической математики. Вычислительное устройство, которое выдает неправильный ответ, может показаться странным на первый взгляд, но ошибка в данном случае – это только следствие его конечной природы. Некоторые компьютеры содержат специальное аппаратное обеспечение, которое обнаруживает ошибки переполнения.

Алгебра чисел конечной точности отличается от обычной алгебры. В качестве примера рассмотрим ассоциативный закон $a+(b-c)=(a+b)-c$.

Вычислим обе части выражения для $a=700$, $b=400$ и $c=300$. В левой части сначала вычислим значение $(b-c)$. Оно равно 100. Затем прибавим это число к a и получим 800. Чтобы вычислить правую часть, сначала вычислим $(a+b)$. Для трехразрядных целых чисел получится переполнение. Результат будет зависеть от компьютера, но он не будет равен 1100. Вычитание 300 из какого-то числа, отличного от 1100, не даст результата 800. Ассоциативный закон не имеет силы. Порядок операций важен.

Другой пример – дистрибутивный закон:

$$a \times (b - c) = a \times b - a \times c.$$

Сосчитаем обе части выражения для $a=5$, $b=210$ и $c=195$. В левой части $5 \times 15 = 75$. В правой части 75 не получается, поскольку $a \times b$ выходит за пределы ряда.

Система счисления, в которой значение цифры не зависит от ее положения в числе, называется *непозиционной*. Здесь веса всех одноименных цифр, где бы они не располагались в числе, будут одинаковы.

Наиболее известными представителями непозиционных систем счисления являются иероглифические и алфавитные. Иероглифические – это такие системы счисления, у которых каждая цифра представлена своим символом, значком или иероглифом. Наиболее известной из них является римская система счисления.

Значение числа, записанного в римской системе счисления, определяется как сумма записанных подряд цифр, причем, если слева от цифры стоит меньшая цифра, то значение последней принимается со знаком минус, например: IX=9₍₁₀₎; XI=11₍₁₀₎, то есть здесь существует отклонение от правила независимости значения цифры от положения в числе. В настоящее время римская система используется в основном для целей нумерации. Запись числа в алфавитных системах строится по такому же принципу.

К основным недостаткам непозиционных систем счисления можно отнести:

- 1) отсутствие нуля;
- 2) необходимость содержания бесконечного количества символов;
- 3) сложность арифметических действий над числами.

Систему счисления называют *позиционной*, если значение цифры определяется ее положением в числе. Веса цифр в позиционной системе счисления различны и значение веса цифры зависит от номера ее позиции в числе. В общем случае вес $G_i=f(p,i)$, где i – номер позиции в числе, p – целое число, отличное от нуля и называемое *основанием системы счисления*.

Десятичная система счисления наиболее распространена в вычислительной практике. Своему распространению она обязана наличию у человека на руках десяти пальцев.

2.2. Диапазоны представления чисел

Арифметические действия над числами приводят к взаимодействию между разрядами, поэтому используются позиционные системы счисления с простейшим постоянным соотношением между весами соседних разрядов. Веса разрядов подчиняются закону геометрической прогрессии. Знаменатель прогрессии $p=const$ определяется основанием позиционной системы счисления (ССЧ).

Основание p определяет возможное количество цифр в любом из разрядов, то есть одно и то же число может быть представлено различным количеством разрядов. Чем меньше основа-

ние P , тем больше разрядов требуется для изображения одного и того же числа. Например, $31_{(10)} = 11111_{(2)} = 37_{(8)}$.

Если длина разрядной сетки задана, то это ограничивает максимальное по абсолютному значению число, которое может быть записано.

Пусть длина разрядной сетки равна любому положительному числу, например N . Тогда максимальное число, которое можно представить в разрядной сетке заданной длины:

$$A_{(p)max} = p^N - 1. \quad (2.1)$$

Однако, если задано максимальное абсолютное значение числа, то длина разрядной сетки

$$N = \log_{(p)} (A_{(p)max} + 1). \quad (2.2)$$

Интервал числовой оси, заключенный между максимальным и минимальным числами, называют *диапазоном представления (ДП) чисел* в данной системе счисления для заданного ограничения на длину разрядной сетки

$$-A_{(p)max} \leq \text{ДП} \leq A_{(p)max}. \quad (2.3)$$

Если предположить, что затраты на каждый разряд эквивалентны затратам на изображение одной цифры, то наименьшие затраты окажутся при использовании троичной системы счисления ($p=3$). Десятичная система по затратам не оптимальна, и для уменьшения затрат основание P следует уменьшить. В ВМ используются в основном двоичная ССЧ с основанием $p=2$. Она позволяет наиболее простым способом изображать и хранить в разрядах любую из двух цифр на основе двухпозиционных схем и передавать информацию по линиям связи в форме наличия (одна цифра) или отсутствия (другая цифра) сигнала. Кроме того, двоичная ССЧ самым оптимальным способом изображает логику высказывания: истина – 1, ложь – 0. Логика высказывания является основой построения логических устройств и широко используется в программах.

В двоичных ССЧ могут быть использованы как положительные, так и отрицательные цифры, например комбинации цифр: 00, $\bar{1}\bar{1}$, $0\bar{1}$, где $\bar{1}$ – отрицательная цифра. Но в подавляющем большинстве случаев используются цифры 01.

В зависимости от принятых весов разрядов, изображаемые числа могут быть целыми, дробными или смешанными, содер-

жащими целую и дробную части. Обычно ограничиваются или только целыми, или только дробными. Веса разрядов нигде не записываются, а только подразумеваются. У дробных чисел вес самого младшего разряда равен 2^{-m} (m – число разрядов). Вес самого старшего разряда $2^{-m} * 2^{m-1} = 2^{-1}$. У целых чисел соответственно равен 2^0 – вес младшего разряда; 2^{m-1} – вес старшего разряда.

Максимальное значение изображаемого дробного числа

$$\sum_{i=-m}^{-1} 2^i = 2^0 - 2^{-m}, \quad (2.4)$$

целого числа

$$\sum_{i=0}^{m-1} 2^i = 2^m - 2^0. \quad (2.5)$$

Если к максимальному значению изображаемого числа добавить одно минимальное дискретное значение – единицу младшего разряда, то изображение станет нулевым, то есть арифметические действия в вычислительных устройствах выполняются из условия сравнимости по модулю 2^m или 2^0 (это неизбежно, потому что существуют ограничения разрядности чисел).

2.3. Позиционные системы счисления

Любая позиционная система счисления характеризуется основанием. *Основание (базис)* естественной позиционной системы счисления – число знаков или символов, используемых для изображения цифр в данной системе. Таким образом, основание позиционной системы счисления – это количество различных знаков или символов, используемых для изображения цифр в данной системе. За основание системы можно принять любое натуральное число – два, три, четыре и т.д. Возможно бесчисленное множество позиционных систем, так как за основание можно принять любое число, образовав новую систему. Например, возможна шестнадцатеричная система счисления, запись чисел в которой может производиться с помощью следующих знаков: 0, 1, ..., 9, A, B, C, D, E, F.

Обычное десятичное число состоит из цепочки десятичных разрядов и иногда десятичной запятой. Общая форма записи показана на рис. 2.1. Десятка выбрана в качестве основы возведе-

ния в степень (основание системы счисления), поскольку мы используем 10 цифр.

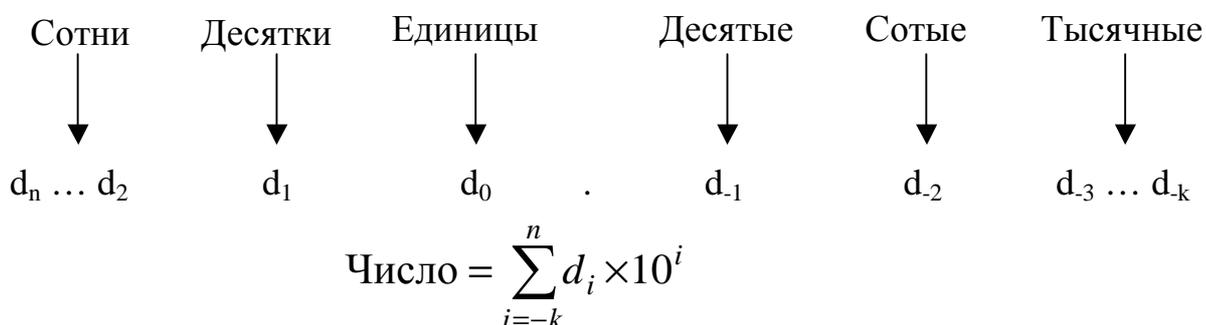


Рис. 2.1. Общая форма записи чисел

В компьютерах удобнее иметь дело с другими основаниями системы счисления. Самые важные из них – 2, 8 и 16. Соответствующие системы счисления называются двоичной, восьмеричной и шестнадцатеричной соответственно.

k -ричная система требует k различных символов для записи разрядов с 0 по $k-i$. Десятичные числа строятся из 10 десятичных цифр:

0123456789

Двоичные числа, напротив, строятся только из двух двоичных цифр:

01

Восьмеричные числа состоят из восьми цифр:

01234567

Шестнадцатеричные числа строятся из следующих цифр и символов:

0123456789ABCDEF

Двоичный разряд (то есть 1 или 0) обычно называют *битом*. На рисунке 2.2 десятичное число 2001 представлено в двоичной, восьмеричной и шестнадцатеричной системе. Число 7B9 очевидно шестнадцатеричное, поскольку символ B встречается только в шестнадцатеричных числах. А число 111 может быть в любой из четырех систем счисления. Чтобы избежать двусмысленности, нужно использовать индекс для указания основания системы счисления.

Двоичное число	1 1 1 1 1 0 1 0 0 0 1 $1 \times 2^{10} + 1 + 1 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ 1024 + 512 + 256 + 128 + 64 + 0 + 16 + 0 + 0 + 0 + 1
Восьмеричное число	3 7 2 1 $3 \times 8^3 + 7 \times 8^2 + 2 \times 8^1 + 1 \times 8^0$ 1536 + 448 + 16 + 1
Десятичное число	2 0 0 1 $2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 1 \times 10^0$ 2000 + 0 + 0 + 1
Шестнадцатеричное число	7 D 1 $7 \times 16^2 + 13 \times 16^1 + 1 \times 16^0$ 1792 + 208 + 1

Рис. 2.2. Примеры представления чисел в двоичной, восьмеричной и шестнадцатеричной системах счисления

В таблице 2.1 в качестве примера ряд неотрицательных целых чисел представлен в каждой из четырех систем счисления.

Таблица. 2.1

Десятичные числа и их двоичные, восьмеричные и шестнадцатеричные эквиваленты

Десятичная ₁₀	Двоичная ₂	Восьмеричная ₈	Шестнадцатеричная ₁₆
0	00000	000	00
1	00001	001	01
2	00010	002	02

3	00011	003	03
4	00100	004	04
5	00101	005	05
6	00110	006	06
7	00111	007	07
8	01000	010	08
9	01001	011	09
10	01010	012	A
11	01011	013	B
12	01100	014	C
13	01101	015	D
14	01110	016	E
15	01111	017	F
16	10000	020	10
17	10001	021	11
18	10010	022	12
19	10011	023	13
20	10100	024	14
21	10101	025	15
22	10110	026	16
23	10111	027	17
24	11000	030	18
25	11001	031	19
26	11010	032	1A
27	11011	033	1B
28	11100	034	1C
29	11101	035	1D
30	11110	036	1E
31	11111	037	1F
40	101000	50	28
50	110010	62	32
60	111100	74	3C
70	1000110	106	46
80	1010000	120	50
90	1011010	132	5A
100	1100100	144	64

	0		
1000	1111101 000	1750	3E8
2989	1011101 01101	5655	BAD

Запись чисел в каждой из систем счисления с основанием p означает сокращенную запись выражения

$$a_{n-1}p^{n-1} + a_{n-2}p^{n-2} + \dots + a_1p^1 + a_0p^0 + a_{-1}p^{-1} + \dots + a_{-m}p^{-m}, \quad (2.6)$$

где a_i – цифры системы счисления; n и m – число целых и дробных разрядов соответственно.

В каждой системе счисления цифры упорядочены в соответствии с их значениями: 1 больше 0, 2 больше 1 и т.д.

В любой системе счисления число изображается в виде набора знаков (цифр)

$$a_{m-1}, a_{m-2}, \dots, a_1, a_0. \quad (2.7)$$

В этих наборах цифра a_i определяет значение разряда i . При ручном счете количество разрядов не регламентируется; в вычислительных устройствах оно заранее устанавливается исходя из требований точности вычислений и допустимому объему аппаратуры. Память компьютера ограничена, поэтому мы можем иметь дело только с такими числами, которые можно представить в фиксированном количестве разрядов. Напомним, что такие числа называют *числами конечной точности*.

Каждый разряд i характеризуется своим весом p_i

$$A = a_{m-1}p_{m-1} + a_{m-2}p_{m-2} + \dots + a_1p_1 + a_0p_0 = \sum_{i=0}^{m-1} a_i p_i, \quad (2.8)$$

где p_i – целочисленное значение весов разрядов; a_i – значение цифр разрядов, определяющих сколько раз надо повторить веса, чтобы определить долю данного разряда в значении числа.

В вычислительных устройствах, как и при ручном счете, используются только позиционные системы счисления, позволяющие строго разграничивать аппаратуру по разрядам, что позволяет ее унифицировать.

2.4. Однородные и неоднородные системы счисления

Различают однородные и неоднородные системы счисления. В *неоднородных* системах счисления G_i не зависят друг от

друга и могут принимать любые значения. Эти системы еще называют системами со смешанным основанием. В неоднородных системах счисления в каждом i -м разряде количество допустимых символов может быть различно, при этом $0 \leq a_i \leq p_i - 1$, где p_i – основание системы в i -м разряде.

В общем виде число A может быть представлено следующим образом:

$$A = a_n p_{n-1} \dots p_1 + a_{n-1} p_{n-2} \dots p_1 + \dots + a_2 p_1 + a_1, \quad (2.9)$$

где a_i – цифра i -го разряда числа, причем $a_i = \overline{0, p_{j-1}}$ есть база системы счисления;

$$p_i = \prod_0^i p_j \text{ – вес } i\text{-го разряда числа.}$$

Примером неоднородной системы счисления может служить система счисления времени, для которой $p_0=1$ с; $p_1=60$ с; $p_2=60$ мин; $p_3=24$ ч; $p_4=365$ сут. Например, время в 2 года 25 суток 14 часов 35 мин 48 секунд, выраженное в единицах младшего разряда – секундах, определится по формуле (2.9):

$$A = 2 \cdot 365 \cdot 24 \cdot 60 \cdot 60 \cdot 1 + 25 \cdot 24 \cdot 60 \cdot 60 \cdot 1 + 14 \cdot 60 \cdot 60 \cdot 1 + 35 \cdot 60 \cdot 1 + 48 \cdot 1.$$

Специально для применения в ЭВМ была создана неоднородная двоично-пятеричная система, в которой в нечетных разрядах основание $p_1=5$ ($a_i=0-4$), а в четных разрядах основание $p_2=2$ ($a_i=0,1$). Так как произведение весов двух соседних (четного и нечетного) разрядов равно десяти, то двумя двоично-десятичными разрядами можно кодировать одну десятичную цифру (табл. 2.2).

Таблица. 2.2
Кодирование десятичных цифр
в двоично-пятеричной системе

$a_{(10)}$	$a_{(2-5)}$	$a_{(10)}$	$a_{(2-5)}$
0	00	5	10
1	01	6	11
2	02	7	12
3	03	8	13
4	04	9	14

Пример. Записать число $748_{(10)}$ в двоично-пятеричной системе счисления.

Решение. Исходя из значений $7_{10}=12_{(2-5)}$, $4_{10}=04_{(2-5)}$, $8_{10}=13_{(2-5)}$, представленных в таблице 2.2, получим $A_{(10)} = 12\ 04\ 13_{(2-5)}$.

Произведем обратное преобразование. Здесь $n = 6$, основания $p_1 = 5$; $p_2 = 2$; $p_3 = 5$; $p_4 = 2$; $p_5 = 5$; $p_6 = 2$, а цифры $a_1 = 3$; $a_2 = 1$; $a_3 = 4$; $a_4 = 0$; $a_5 = 2$; $a_6 = 1$. Для вычисления количественного эквивалента числа A подставим эти значения в формулу (2.9).

$$A = 1 \cdot 5 \cdot 2 \cdot 5 \cdot 2 \cdot 5 + 2 \cdot 2 \cdot 5 \cdot 2 \cdot 5 + 0 \cdot 5 \cdot 2 \cdot 5 + 4 \cdot 2 \cdot 5 + 1 \cdot 5 + 3 = 500 + 200 + 0 + 40 + 5 + 3 = 748_{(10)}.$$

На практике используют сокращенную запись чисел

$$A_{(a)} = a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m}. \quad (2.10)$$

В настоящее время, в основном используются позиционные системы счисления, в которых $p_i = p_j$ при всех i и j . В них веса отдельных разрядов представляют собой степень основания, равную номеру позиции, то есть однородные системы счисления.

2.5. Свойства систем счисления

Системы счисления имеют следующие свойства:

1. Отношение весов соседних разрядов равно основанию системы счисления, то есть основание системы счисления p показывает, во сколько раз значение одной цифры $(i+1)$ -го разряда превышает значение одной цифры i -го разряда.

2. Для любой системы счисления справедливо, что ее основание изображается символами 10 в своей системе, то есть любое число в своей системе p можно записать символами этой системы в виде

$$A_{(p)} = a_{n-1} \cdot 10^{n-1} + a_{n-2} \cdot 10^{n-2} + \dots + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + \dots + a_{-m} \cdot 10^{-m}.$$

3. Умножение (деление) числа на целую степень, равную k , основания системы счисления p приводит к сдвигу запятой вправо (влево) на k разрядов, например:

$$A \cdot p = a_{n-1} a_{n-2} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m} \cdot p^1 = a_{n-1} a_{n-2} \dots a_1 a_0 a_{-1}, a_{-2} \dots a_{-m} = a_{n-1} a_{n-2} \dots a_1, a_0 a_{-1} a_{-2} \dots a_{-m} \cdot p^0.$$

4. Общее количество различных n -разрядных p -ричных чисел равно $N = p^n$.

5. Определение наибольшего значения числа A , осуществляется следующим путем: в формулу (2.9) подставляют максимальное значение цифры:

$$A_{max} = (p-1) \cdot p^{n-1} + (p-1) \cdot p^{n-2} + \dots + (p-1) \cdot p + (p-1) + (p-1) \cdot p^{-1} + \dots + (p-1) \cdot p^{-m} = p^n - 1 + 1 - p^{-m} = p^n - p^{-m}.$$

2.6. Выбор системы счисления

Выбор числа десять в качестве основания системы счисления исторически связан с числом пальцев на руках человека. Однако десятичная система счисления не является наиболее удобной с точки зрения ее реализации в ЭВМ. При анализе систем счисления на предмет их применения в ЭВМ учитываются следующие факторы:

1. Наличие физических элементов, способных изобразить символы системы.

2. Экономичность системы, то есть количество элементов, необходимое для представления многоразрядных чисел.

3. Трудоемкость выполнения арифметических операций в ЭВМ.

4. Быстродействие вычислительных устройств.

5. Наличие формального математического аппарата для анализа и синтеза вычислительных устройств.

6. Удобство работы человека с машиной.

7. Наибольшую помехоустойчивость кодирования цифр на носителях информации.

Наличие физических элементов, способных изобразить символы системы. Любой из символов, применяемых для записи чисел, должен в ЭВМ изображаться в виде одного или нескольких состояний какого-то физического элемента. Очевидно, что элемент будет тем проще, чем меньше состояний ему требуется иметь, то есть чем меньше основание системы счисления. Например, для реализации двоичной системы можно использовать любой простой элемент с двумя устойчивыми состояниями. Таковыми элементами являются: реле, конденсаторы, магнитные, полупроводниковые элементы, триггерные схемы и т.п. В настоящее время имеющиеся элементы с более чем двумя устойчивыми состояниями имеют существенные недостатки по основным па-

раметрам (надежность, быстродействие, габариты, стоимость). Таким образом, по этому критерию наиболее пригодной для ЭВМ является двоичная система счисления.

Экономичность системы счисления оценивается числом цифроразрядов, необходимых для изображения чисел в машине. Для представления в ЭВМ любого n -разрядного числа в системе счисления с основанием p можно использовать:

- n физических элементов с p устойчивыми состояниями;
- nk физических элементов с двумя устойчивыми состояниями, где k – минимально необходимое число двоичных разрядов, необходимых для кодирования любой p -ичной цифры.

Оценка экономичности той или иной системы счисления показала, что по критерию экономичности наиболее приемлемой является система счисления с основанием $p = 3$. Затем следуют системы с $p = 2$ и $p = 4$, которые уступают ей на 5,8%. Однако ввиду того, что троичный элемент менее надежен, чем двоичный, приходится оборудование для одного троичного разряда, как правило, увеличивать в два раза, то есть хранить троичный разряд в двух двоичных. С учетом этого наиболее экономичной оказывается двоичная система.

Трудоемкость выполнения арифметических операций. По этому критерию наиболее эффективной является двоичная система, так как чем меньше цифр участвуют в арифметических операциях, тем проще их выполнять.

Быстродействие вычислительных устройств. Этот критерий находится в прямой зависимости от простоты арифметических операций. Очевидно также, что с увеличением числа цифр в системе счисления быстродействие ЭВМ при прочих равных условиях будет падать. Исследование показали, что ЭВМ, работающая в двоичной системе счисления, характеризуется более высоким быстродействием относительно троичной на 26,2% и относительно десятичной – в 2,7 раза.

Наличие формального математического аппарата для анализа и синтеза вычислительных устройств. Таким аппаратом является алгебра логики. Наибольшее развитие и закончен-

ность изучения, вследствие своей простоты и широкого практического применения получила двоичная логика.

Удобство работы человека с машиной. Безусловно, самой удобной по этому критерию является десятичная система счисления. Но решить, какая система находится на втором месте, сложнее, так как все они требуют перевода чисел. Очевидно, наиболее удобной для человека будет система, в которой проще всего выполняются арифметические действия, то есть двоичная.

Наибольшая помехоустойчивость кодирования цифр. Исходя из условия равных технических возможностей при реализации любой системы счисления, будем считать, что диапазон изменения носителя информации для всех систем остается одинаковым. Это значит, что при наложении некоторой помехи на основной сигнал, изображающий цифру, наибольшая ошибка возможна в устройстве, использующим систему счисления с самым большим основанием. Следовательно, с позиций наибольшей помехоустойчивости предпочтение следует отдать двоичной системе счисления.

Таким образом, исходя из перечисленных критериев, наиболее приемлемой для ЭВМ является *двоичная система счисления*. Однако в некоторых случаях при синтезе вычислительного устройства какому-либо критерию придается большее значение, чем остальным. Тогда для применения выбирается система счисления, оптимальная по выбранному критерию.

При использовании двоичной системы счисления необходимо выполнить преобразование десятичных чисел в двоичные. Однако, учитывая то обстоятельство, что многие математические задачи требуют сравнительно малый объем исходных данных по сравнению с объемом вычислений, этот недостаток становится несущественным.

Существенным недостатком двоичной системы счисления является то, что для записи двоичного числа требуется примерно в 3,3 раза больше разрядов, чем для записи того же числа в десятичной системе. Поэтому двоичную систему применяют, как правило, «для внутренних» нужд машины, а для целей коммуникации человека с машиной выбирают двоично-кодированные

системы счисления: восьмеричную, шестнадцатеричную и двоично-кодированную десятичную.

2.7. Преобразование чисел в системах счисления

При подготовке вычислений на ЭВМ необходимы прямые и обратные преобразования между двоичными и десятичными числами. Эти преобразования различны для целых и дробных чисел.

Пусть целое десятичное число A разложено по другому основанию ССЧ p :

$$A = a_{m-1}p^{m-1} + a_{m-2}p^{m-2} + \dots + a_1p^1 + a_0p^0. \quad (2.11)$$

Если A разделить на p , то получится число $A_1 = Ap^{-1}$, и в остатке a_0 , равное значению младшего разряда числа A и записанное в новой p -ричной ССЧ.

При этом

$$A_1 = a_{m-1}p^{m-2} + a_{m-2}p^{m-3} + \dots + a_1p^0. \quad (2.12)$$

Структура A_1 такая же, как A в формуле (2.11), поэтому путем повторных преобразований можно последовательно определить значения разрядов $a_1, a_2, a_3, \dots, a_{m-1}$.

Таким образом, первый способ перевода основан на делении числа на 2. Частное записывается непосредственно под исходным числом, а остаток (0 или 1) записывается рядом с частным. То же делается с полученным частным. Процесс повторяется до тех пор, пока не останется 0. В результате должны получиться две колонки чисел – частных и остатков. Двоичное число можно считать из колонки остатков снизу вверх. На рисунке 2.3 показано, как происходит преобразование из десятичной в двоичную систему с помощью операции деления.

Преобразование десятичного числа 1492 в двоичное производится путем последовательного деления (сверху вниз). Например, 93 делится на 2, получается 46 и остаток 1. Остаток записывается в строку внизу.

Второй способ вытекает из определения двоичных чисел. Самая большая степень двойки, меньшая, чем число, вычитается из этого числа. Та же операция делается с полученной разностью. Когда число разложено по степеням двойки, двоичное

число может быть получено следующим образом. Единички ставятся в тех позициях, которые соответствуют полученным степеням двойки, а нули – во всех остальных позициях.

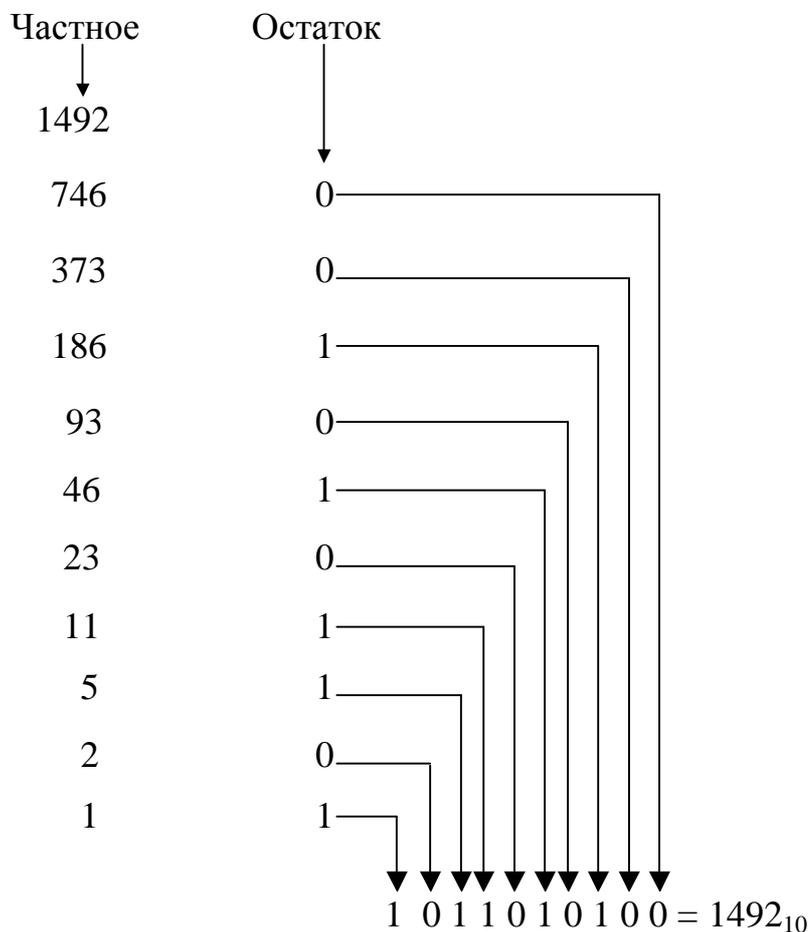


Рис. 2.3. Преобразование десятичного числа в двоичное

Соответствующие степени числа два представлены следующим рядом:

$$2^0 = 1 \quad 2^2 = 4 \quad 2^4 = 16 \quad 2^6 = 64 \quad 2^8 = 256 \quad 2^{10} = 1024$$

$$2^1 = 2 \quad 2^3 = 8 \quad 2^5 = 32 \quad 2^7 = 128 \quad 2^9 = 512 \quad \text{и т.д.}$$

Например, с помощью подбора целых степени двойки перевести десятичное число 293 в соответствующее двоичное можно следующим образом:

$$A^1 = 293 - 2^8 = 37$$

$$A^2 = 37 - 2^5 = 5$$

$$A^3 = 5 - 2^2 = 1$$

$$A^4 = 1 - 2^0 = 0$$

$$293_{10} = [2^8 + 2^5 + 2^2 + 2^0 + 2^8] = 100100101_2$$

Чтобы преобразовать двоичное число в восьмеричное, нужно разделить его на группы по три бита, причем три бита непосредственно слева от двоичной запятой формируют одну группу, следующие три бита слева от этой группы формируют вторую группу и т. д. Каждую группу по три бита можно преобразовать в один восьмеричный разряд со значением от 0 до 7 (см. табл. 2.1). Чтобы дополнить группу до трех битов, нужно спереди приписать один или два нуля. Преобразование из восьмеричной системы в двоичную тоже тривиально. Каждый восьмеричный разряд просто заменяется эквивалентным 3-битным числом. Преобразование из 16-ричной в двоичную систему, по сути, сходно с преобразованием из 8-ричной в двоичную систему, только каждый 16-ричный разряд соответствует группе из четырех битов, а не из трех. На рисунке 2.4 приведены примеры преобразований из одной системы в другую.

Шестнадцатеричное число	0 3 2 9 1 6 В
Двоичное число	0001100101001000101101100
Восьмеричное число	0 1 4 5 1 0 5 5 4
Шестнадцатеричное число	1 E E 8 E F 1
Двоичное число	01111011101000111011110001
Восьмеричное число	1 7 3 5 0 7 3 6 1

Рис. 2.4. Примеры преобразования двоичных чисел

Двоичные числа можно преобразовывать в десятичные двумя способами. Первый способ – суммирование степеней двойки, которые соответствуют биту 1 в двоичном числе. Например:

$$10110 = 24 + 22 + 21 = 16 + 4 + 2 = 22$$

Второй способ. Двоичное число записывается вертикально по одному биту в строке, крайний левый бит находится внизу. Самая нижняя строка — это строка 1, затем идет строка 2 и т. д. Десятичное число строится напротив этой колонки. Сначала обозначим строку 1. Элемент строки n состоит из удвоенного элемента строки $n-1$ плюс бит строки n (0 или 1). Элемент, полученный в самой верхней строке, и будет ответом. Метод проиллюстрирован на рис. 2.5.

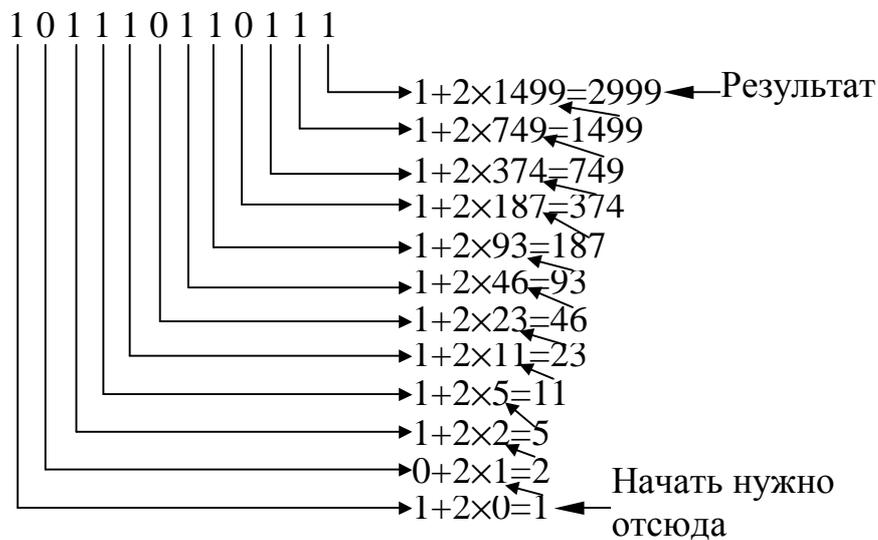


Рис. 2.5. Преобразование двоичного числа в десятичное с помощью последовательного умножения

На рис. 2.5 двоичное число 101110110111 преобразуется в десятичное путем последовательного удваивания снизу вверх. В каждой следующей строке удваивается значение предыдущей строки и прибавляется соответствующий бит. Число 374 умножается на 2 и прибавляется бит соответствующей строки (в данном случае 1). В результате получается 749 и т.д.

Преобразование из десятичной в восьмеричную или шестнадцатеричную систему можно выполнить либо путем преобразования сначала в двоичную, а затем в нужную нам систему, либо путем вычитания степеней 8 или 16.

Для перевода правильных дробей из системы счисления с основанием p_1 в систему с основанием p_2 используется метод, базирующийся на умножении переводимой правильной дроби на

основание p_2 новой системы счисления. Правильная дробь $A_{(p1)}$ в системе с основанием p_2 может быть записана в виде

$$A_{(p2)} = a_{-1} \cdot p_2^{-1} + a_{-2} \cdot p_2^{-2} + \dots + a_{-m} \cdot p_2^{-m}. \quad (2.13)$$

Если правую часть выражения умножить на p_2 , то найдем неправильную дробь, в целой части которой будет число a_{-1} . Умножив затем оставшуюся дробную часть на величину основания p_2 , получим дробь, в целой части которой будет a_{-2} и т.д. Повторяя процесс умножения m раз, найдем все m цифр дробной части числа в новой системе счисления. При этом все действия должны выполняться по правилам арифметики и, следовательно, в целой части получающихся дробей будут появляться эквиваленты цифр новой системы счисления, записанные в исходной системе счисления.

При $p=2$ алгоритм преобразования десятичного дробного в двоичное число сводится к последовательному умножению дробных частей произведений на два и к последовательному формированию двоичных разрядов начиная со старших по значению частей произведений.

Пример. $A_{10}=0.752$ перевести с точностью до 6 двоичных разрядов.

$$0.752 \cdot 2 = 1.504 \quad A_{-1} = 1$$

$$0.504 \cdot 2 = 1.008 \quad A_{-2} = 1$$

$$0.008 \cdot 2 = 0.016 \quad A_{-3} = 0$$

$$0.016 \cdot 2 = 0.032 \quad A_{-4} = 0$$

$$0.032 \cdot 2 = 0.064 \quad A_{-5} = 0$$

$$0.064 \cdot 2 = 0.128 \quad A_{-6} = 0$$

$$0.752_{10} \approx 0.110000_2$$

При переводе правильных дробей из одной системы счисления в другую можно получить дробь в виде бесконечного числа или расходящегося ряда. Процесс перевода можно закончить, если появится дробная часть, имеющая во всех разрядах нули, или будет достигнута заданная точность перевода (получено требуемое число разрядов результата). Последнее означает, что при переводе дроби необходимо указать число разрядов в случае

ее представления в новой системе счисления. Естественно, что при этом возникает погрешность перевода чисел. В ЭВМ точность перевода обычно ограничивается длиной разрядной сетки, отведенной для представления чисел.

Двоичные числа неудобны в обращении из-за большой разрядности, особенно при вводе в ЭВМ. Поэтому в ВТ находят широкое применение в качестве промежуточных 2^p -ричные ССЧ ($p=2, p=3, p=4$), в каждой из которых могут быть записаны любые из 2^p цифр. В свою очередь, любая 2^p -ричная цифра может быть изображена комбинацией из p двоичных цифр. Таким образом, группируя двоичное число по p разрядам, можно элементарным способом переходить из двоичной системы в 2^p -ричную и наоборот. Наиболее широко используются восьмеричная ($p=3$)

$p=3$ 0,1,2,3,4,5,6,7

и шестнадцатеричная ($p=4$)

$p=4$ 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

системы счисления, которые очень удобны в качестве промежуточных при переводе десятичных чисел в двоичные.

Представление чисел в восьмеричной и шестнадцатеричной системе счисления мы рассмотрим в следующем пункте.

2.8. Восьмеричная и шестнадцатеричная системы

Напомним, что восьмеричная и шестнадцатеричная системы счисления относятся к двоично-кодированным системам, когда основание системы счисления p представляет целую степень двойки:

$p = 8 = 2^3$ – для восьмеричной,

$p = 16 = 2^4$ – для шестнадцатеричной.

База *восьмеричной системы счисления* использует для изображения чисел восемь цифр: 0, 1, 2, 3, 4, 5, 6, 7, то есть $a_i = \overline{0,7}$. Основание $p = 8_{(10)} = 10_{(8)}$. Если восьмеричное число записать в развернутом виде в виде суммы значений цифр и выполнить арифметические действия по правилам десятичной системы, то получим десятичный эквивалент восьмеричного числа. Например, $125,4_{(8)} = 1 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0 + 4 \cdot 8^{-1} = 85,5_{(10)}$.

В *шестнадцатеричной системе* для записи чисел используются цифры 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 и прописные латинские бу-

квы A, B, C, D, E, F , имеющие значение десятичных чисел 10, 11, 12, 13, 14, 15 соответственно. Поэтому шестнадцатеричное число может иметь, например, вид $3E5, C_{(16)}$. Представляя это число в развернутом виде, получим

$$3E5, C_{(16)} = 3 \cdot 16^2 + E \cdot 16^1 + 5 \cdot 16^0 + C \cdot 16^{-1}.$$

Выполняя арифметические операции по правилам десятичной системы и принимая во внимание, что $E=14, C=12$, получим $3E5, C_{(16)} = 560,75_{(10)}$.

Большим достоинством восьмеричной и шестнадцатеричной систем счисления является, во-первых, возможность более компактно представить запись двоичного числа, во-вторых, сравнительно просто осуществлять преобразование чисел из двоичной в восьмеричную и шестнадцатеричную системы, и наоборот. Действительно, так как для восьмеричного числа каждый разряд представляется группой из трех двоичных разрядов, а для шестнадцатеричного – группой из четырех двоичных разрядов, то для такого преобразования достаточно объединить двоичные цифры в группы по 3 и 4 бита соответственно, продвигаясь от разделяющей запятой вправо и влево. При этом в случае необходимости добавляют нули в начале и в конце числа и каждую такую группу – триаду или тетраду – заменяют эквивалентной восьмеричной или шестнадцатеричной цифрой.

Приведем примеры:

а) перевод двоичного числа 1101111001,1101 в восьмеричное:

$$\begin{array}{cccccc} 001 & 101 & 111 & 001, & 110 & 100_{(2)} & = 1571,64_{(8)} \\ 1 & 5 & 7 & 1 & 6 & 4 \end{array}$$

б) перевод двоичного числа 1111111011,100111 в шестнадцатеричное:

$$\begin{array}{cccccc} 0111 & 1111 & 1011, & 1001 & 1100_{(2)} & = 7FB,9C_{(16)} \\ 7 & F & B & 9 & C \end{array}$$

Восьмеричная и шестнадцатеричная системы счисления используются в текстах программ для более короткой и удобной записи двоичных кодов команд, адресов и операндов. Особенно удобно использовать шестнадцатеричную систему, когда разрядность чисел и команд выбрана кратной байту, при этом каж-

дый двоичный код байта записывается в виде двухразрядного шестнадцатеричного числа. Кроме того, эти системы применяются в ЭВМ при некоторых формах представления чисел.

2.9. Отрицательные двоичные числа

На протяжении всей истории цифровых компьютеров для репрезентации отрицательных чисел использовались четыре различные системы. Первая из них называется системой со знаком. В такой системе крайний левый бит – это знаковый бит (0 – это «+», а 1 – это «-»), а оставшиеся биты показывают абсолютное значение числа.

Во второй системе, которая называется дополнением до единицы, тоже присутствует знаковый бит (0 – это плюс, а 1 – это минус). Чтобы сделать число отрицательным, нужно заменить каждую 1 на 0 и каждый 0 на 1. Это относится и к знаковому биту. Система дополнения до единицы уже устарела.

Третья система, дополнение до двух, содержит знаковый бит (0 – это «+», а 1 – это «-»). Отрицание числа происходит в два этапа. Сначала каждая единица меняется на 0, а каждый 0 – на 1 (как и в системе дополнения до единицы). Затем к полученному результату прибавляется 1. Двоичное сложение происходит точно так же, как и десятичное, только перенос совершается в том случае, если сумма больше 1, а не больше 9. Например, рассмотрим преобразование числа 6 в форму с дополнением до двух:

00000110 (+6);

11111001 (-6 в системе с дополнением до единицы);

11111010 (-6 в системе с дополнением до двух).

Если нужно совершить перенос из крайнего левого бита, он просто отбрасывается.

В четвертой системе, которая для m -битных чисел называется *excess* 2^{m-1} , число представляется как сумма этого числа и 2^{m-1} . Например, для 8-битного числа ($m=8$) система называется *excess* 128, а число сохраняется в виде суммы исходного числа и 128. Следовательно, -3 превращается в $-3+128=125$, и это число (-3) представляется 8-битным двоичным числом 125 (01111101). Числа от -128 до $+127$ выражаются числами от 0 до 255 (все их

можно записать в виде 8-битного положительного числа). Отметим, что эта система соответствует системе с дополнением до двух с обращенным знаковым битом. В таблице 2.3 представлены примеры отрицательных чисел во всех четырех системах.

Таблица 2.3
Отрицательные 8-битные числа
в четырех различных системах

N - деся-	N дво- ичное	-N в системе со знаком	-N до- полнение до единицы	-N дополне- ние до	-N excess 128
1	0000	100000	11111110	1111	011
2	0000	100000	11111101	1111	011
3	0000	100000	11111100	1111	011
4	0000	100001	11111011	1111	011
5	0000	100001	11111010	1111	011
6	0000	100001	11111001	1111	011
7	0000	100001	11111000	1111	011
8	0000	100010	11110111	1111	011
9	0000	100010	11110110	1111	011
1	0000	100010	11110101	1111	011
2	0001	100101	11101011	1110	011
3	0001	100111	11100001	1110	011
4	1010	101010	11010111	1101	010
5	0011	101100	11001101	1100	010
6	0011	101111	11000011	1100	010
7	0100	110001	10111001	1011	001
8	0101	110100	10101111	1011	001
9	0101	110110	10100101	1010	001
1	0110	111001	10011011	1001	000
1	0111	111111	10000000	1000	000
1 28	Не сущест.	Не су- щест.	Не су- щест.	1000 0000	000 00000

В системах со знаком и с дополнением до единицы есть два представления нуля: $+0$ и -0 . Такая ситуация нежелательна. В системе с дополнением до двух такой проблемы нет, поскольку здесь плюс ноль это всегда плюс ноль. Но зато в этой системе есть другая особенность. Набор битов, состоящий из 1, за которым следуют все нули, является дополнением самого себя. В результате ряд положительных и отрицательных чисел несимметричен – существует одно отрицательное число без соответствующего ему положительного.

2.10. Двоично-кодированная десятичная система счисления (D -коды)

Непосредственное изображение десятичных чисел приводит к необходимости двоичного кодирования десятичных цифр. Устройствам, выполняющим арифметические преобразования с десятичными числами, присваивается специальный термин «десятичная арифметика». Такие устройства должны иметь максимальное сходство с обычными двоичными устройствами.

Десятичная арифметика включается в состав аппаратурных средств высокопроизводительных систем с целью исключения преобразований исходных данных в двоичную форму и результатов в десятичную.

Двоично-кодированная десятичная система является комбинированной системой счисления, которая обладает достоинствами двоичной и удобством десятичной системы.

D -код – это двоично-кодированное представление десятичного числа, в котором каждая десятичная цифра представляется тетрадой из двоичных символов.

Количество различных двоичных тетрад $N = 2^4 = 16$. Для кодирования двоичных цифр из них используется только десять. Наличие избыточных комбинаций позволяет иметь различные D -коды. В ЭВМ наибольшее применение нашли системы кодирования 8421 – D_1 , 2421 – D_2 , $(8421+3)$ – D_4 . Появляющаяся избыточность приводит к множеству кодирования десятичных цифр, из которых следует выбирать оптимальную.

Код 8421 (табл. 2.4) называется *кодом с естественными весами*, где цифры 8,4,2,1 – веса двоичных разрядов тетрад. Любая десятичная цифра в этом коде изображается ее эквивалентом в двоичной системе счисления. Этот код нашел наибольшее применение при кодировании десятичных чисел в устройствах ввода-вывода и при построении операционных устройств десятичной арифметики.

Особенность кодов D_2 и D_4 $(8421+3)$ или кода с избытком 3 в том, что кодирование любой десятичной цифры и дополнительной к ней цифры до 9 осуществляется взаимно дополняющими тетрадами. Эта особенность дает простой способ получения дополнения до 9 путем инвертирования двоичных цифр тет-

рады. Такие коды удобно использовать для организации операции вычитания при построении десятичных сумматоров.

Таблица 2.4

Примеры кодирования десятичных цифр тетрадами

Десятичная цифра	Эквиваленты в D -кодах		
	D_1 (8421)	D_2 (2421)	D_4 (8421+3)
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0011	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

Приведем пример кодирования десятичного числа $A = 8371$ в двоично-кодированной десятичной системе счисления:

$$D_1: A = 1000\ 0011\ 0111\ 0001_{(2/10)};$$

$$D_2: A = 1110\ 0011\ 1101\ 0001_{(2/10)};$$

$$D_4: A = 1011\ 0110\ 1010\ 0100_{(2/10)}.$$

Оптимальность кодирования определяется шестью требованиями, которым должен удовлетворять десятичный код.

1. *Однозначность.* Каждой десятичной цифре должен соответствовать определенный, отличающийся от других, двоичный код.

Невыполнение данного требования приводит к неоднозначности результатов.

2. *Упорядоченность.* Большим десятичным цифрам должны соответствовать большие тетрады десятичного кода и, наоборот, меньшим – меньшие тетрады.

Выполнение данного требования необходимо для организации количественного сравнения цифр в десятичных разрядах.

3. *Четность.* Четным цифрам должны соответствовать четные тетрады, нечетным цифрам – нечетные тетрады. Соответствие может быть отмечено любым способом.

Выполнение данного требования необходимо для выполнения округления результата.

4. *Дополнительность*. Если x_1 и x_2 – такие две цифры, для которых $x_1 + x_2 = 9$ и цифре x_1 сопоставляется тетрада $\alpha_3\alpha_2\alpha_1\alpha_0$, то цифре x_2 , если удовлетворяется требование дополнительной, должна сопоставляться тетрада $\alpha_3\alpha_2\alpha_1\alpha_0$, получаемая путем инверсии двоичных разрядов кода цифры x_1 .

Требование дополнительной необходимо для упрощения реализации дополнительных и обратных кодов десятичных чисел.

5. *Весомозначность*. Должны существовать четыре целых положительных числа: p_3, p_2, p_1, p_0 , называемых весами, с помощью которых можно определить десятичную цифру x по значению двоичной тетрады $\alpha_3\alpha_2\alpha_1\alpha_0$, сопоставленной x , по формуле

$$x = \alpha_3 p_3 + \alpha_2 p_2 + \alpha_1 p_1 + \alpha_0 p_0.$$

Выполнение данного требования способствует декодированию.

6. *Непрерывность*. Непрерывной последовательности изменений значения цифр должна соответствовать непрерывная последовательность изменений значения тетрад.

Ни один из десятичных кодов не удовлетворяет одновременно всем шести перечисленным требованиям.

Наибольшее распространение в ВТ нашел код прямого за-мещения с весом разрядов 8421. Этот код самый наглядный и удобный, так как в соответствии с названием кода десятичная цифра в нем соответствующим значением двоичного кода. Однако код 8421 не удовлетворяет требованию дополнительной, поэтому действия в этом коде с изменением знака десятичного числа связаны с инверсией разрядов или взятия дополнения, то есть требуют дополнительных коррекций и/или временных затрат.

Достоинствами двоично-кодированной десятичной системы счисления относительно двоичной являются:

- отсутствие необходимости перевода исходных данных и результатов из одной системы счисления в другую;

- удобство контроля промежуточных результатов путем вывода их на индикацию для внутреннего наблюдения;
- более широкие возможности для автоматического контроля из-за наличия в D -кодах избыточных комбинаций.

D -коды применяют для решения экономических задач, которые характеризуются большим объемом исходных данных, сравнительной простотой и малым объемом выполняемых над ними преобразований и большим количеством результатов вычислений. Эта система широко используется в калькуляторах и персональных микроЭВМ.

2.11. Формы представления чисел в ЭВМ

ЭВМ оперирует с числами, содержащими конечное число разрядов. Количество разрядов ограничено длиной разрядной сетки машины.

Под разрядной сеткой понимается совокупность двоичных разрядов, предназначенных для хранения и обработки машинных слов (двоичных кодов).

Для представления чисел используются две формы с фиксированной запятой (естественная) и с плавающей запятой (нормальная). Для любой формы представления чисел их знаки – плюс и минус – кодируются соответственно цифрами 0 и 1. При этом знак числа располагается перед старшим разрядом кода числа.

Форма представления чисел с фиксированной запятой предполагает, что положение запятой, отделяющей целую часть от дробной, фиксировано в разрядной сетке машины. (Хотя запятая и фиксируется, но никак не выделяется в коде числа, а только подразумевается.)

Количество двоичных разрядов и положение запятой определяют такие важные характеристики ЭВМ, как точность и диапазон представления чисел.

Обычно в ЭВМ используются два способа расположения запятой (рис. 2.6): перед старшим разрядом (дробные числа) или после младшего (целые числа).

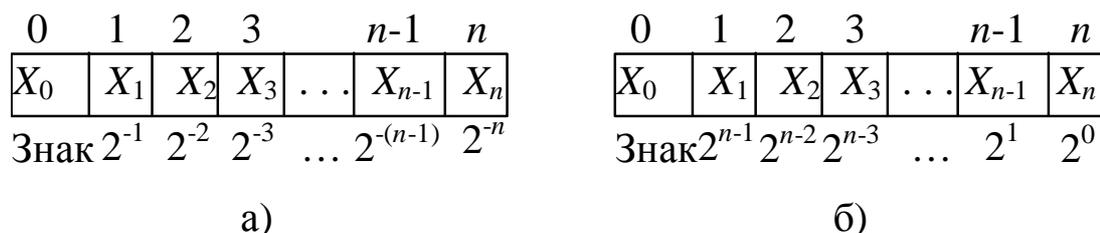


Рис. 2.6. Разрядные сетки ЭВМ для представления чисел с фиксированной запятой: а – для правильных дробей; б – для целых чисел

Диапазоны представления модулей для этих разрядных сеток имеют вид:

а) для дробных чисел:

$$|X_{max}| = 0,111\dots11_{(2)} = 1 - 2^{-n};$$

$$|X_{min}| = 0,000\dots01_{(2)} = 2^{-n};$$

$$1 - 2^{-n} \geq |X| \geq 2^{-n}; \quad (2.14)$$

б) для целых чисел:

$$|X_{max}| = 111\dots11_{(2)} = 2^n - 1;$$

$$|X_{min}| = 000\dots01_{(2)} = 1;$$

$$2^n - 1 \geq |X| \geq 1. \quad (2.15)$$

Важной характеристикой является *динамический диапазон* модулей представляемых чисел:

$$d = \frac{|X_{max}|}{|X_{min}|}. \quad (2.15)$$

Для дробных чисел

$$d = \frac{1 - 2^{-n}}{2^{-n}} = 2^n - 1; \quad (2.16)$$

для целых чисел

$$d = \frac{2^n - 1}{2^{-n}} = 2^n - 1. \quad (2.17)$$

Величина динамического диапазона d не зависит от положения запятой, а определяется только длиной разрядной сетки. В обоих случаях диапазон представления чисел в машине с фиксированной запятой

$$d = \frac{|X_{max}|}{|X_{min}|} = \frac{1 - 2^{-n}}{2^{-n}} = \frac{2^n - 1}{1} \cong 2^n. \quad (2.18)$$

Если при выполнении вычислений значения чисел выйдут за пределы допустимого диапазона, то возникнет ошибка. Чтобы этого избежать, исходные данные необходимо масштабировать:

$$X = [X]K_x, \quad (2.19)$$

где K_x – масштабный коэффициент.

Длину разрядной сетки с фиксированной запятой в операционных устройствах (ОУ) современных ЭВМ принято выбирать кратной байту (8 бит). Для мини- и микроЭВМ использует 1, 2 или 4 байта, для ЭВМ общего назначения длина разрядной сетки составляет 4 байта (32 бита) или 8 байт (64 бита). Длина слова в памяти ОУ также кратна 1 байту.

По сложившимся в вычислительной технике традициям нумерация разрядов в разрядной сетке в машинах общего назначения ведётся слева направо, а в мини- и микроЭВМ, микропроцессорах – справа налево.

Форма представления чисел с плавающей запятой позволяет избежать трудоёмкого масштабирования исходных чисел и значительно увеличить диапазон и точность представляемых чисел [9,10].

Представление чисел в форме с плавающей запятой в общем виде определяются выражением

$$X = \pm M_x Q^p, \quad (2.20)$$

где M_x – мантисса числа;

Q^p – характеристика числа;

Q – основание системы счисления ($Q = 2$ или $Q = 16$);

p – порядок числа.

Мантисса и порядок задаются в системе счисления с основанием Q . Знак числа совпадает со знаком числа мантиссы.

Мантисса в ЭВМ обычно представляется правильной дробью в нормализованном виде, то есть первая цифра справа от запятой должна быть отличной от нуля:

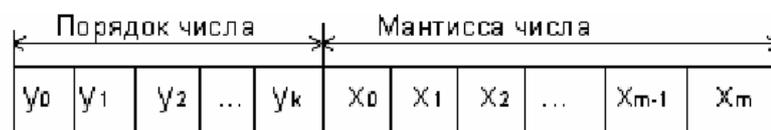
$$M_x = x_0, x_1, x_2, \dots, x_m,$$

где x_0 – код знака числа.

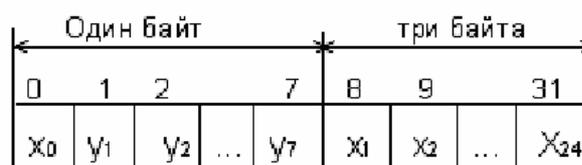
Для $Q = 2$: $M_x = x_0, 1, x_2, \dots, x_m$; для $Q = 16$: $M = x_0, x_1^*, x_2^* \dots x_m^* x_{m/4}^*$, где x_i^* – шестнадцатеричная цифра, $x_1^* > 0$.

Порядок числа p – целое число со знаком – имеет смысл указателя истинного положения запятой в числе.

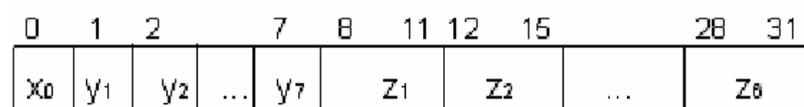
Форматы чисел (разрядная сетка) для плавающей запятой в двоичной системе счисления с указанием веса разрядов представлены на рис. 2.7.



а)



б)



в)

Рис. 2.7. Формат чисел с плавающей запятой:
 а – изображение порядков положительными и отрицательными числами; y₀ – код знака порядка; б – изображение кода числа со смещенным порядком в формате слова; в – изображение шестнадцатеричного числа в формате слова со смещенным порядком

В памяти компьютера, как правило, число с плавающей точкой занимает 4 ячейки памяти. Причем внутренне представление вещественного числа сводится представлению пары целых чисел: мантиссы и порядка. В старшем бите 1-го байта храниться знак числа: 0 обозначает плюс, 1 – минус. Оставшиеся 7 бит 1-го байта содержат машинный порядок. В следующих трех байтах охраняться значащие цифры мантиссы (24 разряда). Схематично это можно представить с виде таблицы 2.5

Таблица 2.5. Представление числа с плавающей точкой в памяти ЭВМ

±	МАНТИССА		
1-й	2-й байт	3-й байт	4-й байт

байт			
------	--	--	--

В данном случае в семи двоичных разрядах помещаются двоичные числа в диапазоне от 0000000 до 1111111. Значит, машинный порядок изменяется в диапазоне от 0 до 127 в десятичной системе счисления, то есть всего 128 значений. Порядок, может быть как положительным так и отрицательным и поэтому разумно эти 128 значений разделить поровну между положительным и отрицательным значениями порядка: от -64 до 63.

Машинный порядок смещен относительно математического и имеет только положительные значения. Смещение выбирается так, чтобы минимальному математическому значению порядка соответствовал нуль.

Связь между машинным порядком (M_p) и математическим (p) в данном случае выражается формулой:

$$M_p = p + 64 \quad (2.21)$$

Полученная формула записана в десятичной системе, а в двоичной системе формула имеет вид:

$$M_{p_2} = p_2 + 1000000_2 \quad (2.22)$$

Таким образом для записи внутреннего представления вещественного числа в 4-х байтовой ячейке необходимо:

- 1) перевести модуль данного числа в двоичную систему счисления с 24 значащими цифрами;
- 2) нормализовать двоичное число;
- 3) найти машинный порядок в двоичной системе счисления;
- 4) учитывая знак числа, выписать его представление в 4-х байтовом машинном слове.

Пример 1. Записать внутреннее представление числа 250,1875 в форме с плавающей точкой.

Решение.

- 1) Приведем исходное число в двоичную систему счисления с 24 значащими цифрами с помощью стандартных алгоритмов и способов перевода (п. 2.7):

$$250,1875_{10} = 1111\ 1010,0011\ 0000\ 0000\ 0000_2.$$

- 2) Запишем в форме нормализованного двоичного числа с плавающей точкой: $0,1111\ 1010\ 0011\ 0000\ 0000\ 0000 \cdot 10_2^{1000}$. Здесь мантисса, то есть основание системы счисления ($2_{10} = 10_2$) и порядок ($8_{10} = 1000_2$) записаны в двоичной системе.

3) Вычислим машинный порядок в двоичной системе счисления: $Mp_2 = 1000 + 100\ 0000 = 100\ 1000$.

4) Запишем представление числа в 4-х байтовой ячейке памяти с учетом знака числа:

0	1001000	11111010	00110000	00000000
---	---------	----------	----------	----------

Пример2.

По шестнадцатеричной форме внутреннего представления числа в форме с плавающей точкой C9811000 восстановить само число.

Решение.

1. Перейдем к двоичному представлению числа в 4-х байтовой ячейке, воспользовавшись стандартными средствами перевода (п.2.7):

1100 1001 1000 0001 0001 0000 0000 0000				
1	1001001	10000001	00010000	00000000

2. Из полученного представления можно заметить, что получен код отрицательного числа, поскольку в старшем разряде с номером 31 записана 1. Отсюда можно получить порядок числа с помощью представленных выше средств. В результате получим:

$$p = 1001001_2 - 1000000_2 = 1001_2 = 9_{10}. \quad (2.23)$$

3. Запишем в форме нормализованного двоичного числа с плавающей точкой с учетом знака числа:

$$-0,1000\ 0001\ 0001\ 0000\ 0000\ 0000 \cdot 2^{1001}$$

4. Тогда число, записанное в двоичной системе счисления будет иметь следующий вид:

$$-100000010,001_2.$$

5. Переведем число в десятичную систему счисления: $-100000010,001_2 = -(1 \cdot 2^8 + 1 \cdot 2^1 + 1 \cdot 2^{-3}) = -258,125_{10}$

Диапазон представления нормализованных чисел с плавающей запятой, когда порядок может принимать как положительное, так и отрицательное значение, определяется так (для $Q = 2$):

$$P_{\max} = 2^k - 1; \quad |M_{\max}| = 1 - 2^{-m};$$

$$|X_{\max}| = 2^{P_{\max}} |M_{\max}| = 2^{2^{1-k}-1} (1 - 2^{-m});$$

$$P_{\min} = -(2^k - 1); |M_{\min}| = 2^{-1}; |X_{\min}| = 2^{-(2^1-1)} \cdot 2^{-1} = 2^{-2^1};$$

$$2^{2^1-1}(1 - 2^{-m}) \geq |X| \geq 2^{-2^1}. \quad (2.23)$$

Динамический диапазон

$$d = \frac{|X_{\max}|}{|X_{\min}|} = \frac{2^{2^1-1}}{2^{-2^1}} \cong 2^{2^{1+1}-1}. \quad (2.24)$$

Сравнивая выражения (2.18) и (2.24) можно показать, что при одной и той же длине разрядной сетки диапазон представляемых чисел с плавающей запятой значительно шире, а относительная погрешность представления $e=1/d$ значительно меньше, чем для чисел с фиксированной запятой. Однако устройства, реализующие операции с числами, представленными в форме с плавающей запятой, характеризует большая сложность и меньшее быстродействие.

В некоторых ЭВМ для упрощения операций над порядками используют форму с плавающей запятой со смещенным порядком. В этом случае порядок представляется целым k -разрядным положительным числом, а разряд знака порядка добавляется к мантиссе (см. рис. 2.7). При таком кодировании число, равное нулю, будет содержать нули в разрядной сетки как для мантиссы, так и для порядка.

Для представления чисел с плавающей запятой широко используется шестнадцатеричная система счисления ($Q = 16$), при этом смещенный порядок числа представляется двоичным целым числом, а мантисса – шестнадцатеричной дробью (см. рис. 2.7в). Тогда выражение (2.20) примет вид

$$X = \pm 16^p M_x. \quad (2.25)$$

В этом случае при одинаковом числе двоичных разрядов разрядной сетки несколько уменьшается точность представления чисел, по сравнению с точностью представления чисел в двоичной системе счисления, однако увеличивается диапазон представляемых чисел в машине.

2.12. Машинный нуль и переполнение разрядной сетки

В общем случае при любом способе представления чисел при решении задачи на машине возможны два вида выхода результатов за пределы диапазона представления.

В первом случае числа по абсолютной величине становятся меньше наименьшего значащего числа, которое можно записать в разрядную сетку: $|X| < X_{\min}$, а во втором случае они могут стать больше наибольшего представимого числа $|X| > X_{\max}$.

При ситуации, соответствующей первому случаю, получаем число, состоящее в пределах разрядной сетки из одних нулей (машинный нуль). При выполнении операций над числами с плавающей запятой возможно получение числа, имеющего порядок меньше допустимого и нормализованную мантиссу, или числа, имеющего нулевую мантиссу и допустимый порядок. Эти числа рассматриваются как машинные нули.

Во втором случае описывается возникновение переполнения разрядной сетки. Различают два вида переполнения: положительное, когда $X > +X_{\max}$ и отрицательное, если $X < -X_{\max}$.

В случае представления чисел с плавающей запятой переполнение разрядной сетки выражается в том, что результат какой-либо операции имеет порядок больше допустимого.

При любой форме представления чисел переполнение разрядной сетки приводит к аварийной ситуации, требующей остановки решения задачи.

2.13. Точность представления чисел в ЭВМ

Представление чисел характеризуется абсолютной и относительной погрешностями.

Абсолютная погрешность – это разность между истинным значением входной величины X и её значением, полученным из машинного изображения $[X]$, то есть

$$\Delta[X] = X - [X]. \quad (2.26)$$

Усреднённая абсолютная погрешность представления чисел в машине с фиксированной запятой определяется как среднее арифметическое между минимальным представимым числом и его минимальной потерей:

$$\Delta = \frac{X_{\min} + 0}{2} = \frac{2^{-n} + 0}{2} = 2^{-(n+1)}, \quad (2.27)$$

то есть в машинах с фиксированной запятой абсолютная погрешность постоянна и равна половине младшего разряда.

Относительная погрешность представления определяется как отношение усреднённой абсолютной погрешности к самому числу $\varepsilon = |X|$.

Так как само число X меняется в пределах $X_{\min} = 2^{-n} \leq |X| \leq 1 - 2^{-n} = X_{\max}$, то и относительная погрешность является величиной переменной, меняющейся соответственно в пределах $\varepsilon_{\min} \leq \varepsilon \leq \varepsilon_{\max}$.

Для машин с фиксированной запятой она определяется следующим образом:

$$\begin{aligned} \varepsilon_{\min} &= \left| \frac{\Delta}{X_{\max}} \right| = \frac{2^{-(n+1)}}{1 - 2^{-n}} \approx 2^{-(n+1)}; \\ \varepsilon_{\max} &= \left| \frac{\Delta}{X_{\min}} \right| = \frac{2^{-(n+1)}}{2^{-n}} \approx 2^{-1}. \end{aligned} \quad (2.28)$$

Таким образом, относительная погрешность для машин с фиксированной запятой зависит от величины числа и колеблется в пределах от $2^{-(n+1)}$ для больших чисел до 2^{-1} – для малых; Причём при $A \rightarrow 2^{-n}$ относительная погрешность может достигать 100%.

В машинах с плавающей запятой абсолютная погрешность представления числа определяется следующим образом:

$$\Delta = \Delta M \cdot 2^p, \quad (2.29)$$

где ΔM – погрешность представления мантиссы, которая определяется так же, как абсолютная погрешность в машинах с фиксированной запятой;

$$\begin{aligned} \Delta M &= 2^{-(n+1)}; \\ p &\text{ – порядок числа, который изменяется в пределах} \\ &-(2^k - 1) \leq p \leq (2^k - 1). \end{aligned} \quad (2.30)$$

Следовательно, в отличие от машин с фиксированной запятой в машинах с плавающей запятой абсолютная погрешность представления чисел зависит от порядка числа: минимальная при

наибольшем отрицательном p и максимальная при наибольшем положительном p . Δ_{\min} и Δ_{\max} определяются следующим образом:

$$\begin{aligned}\Delta_{\min} &= 2^{-(n+1)} \cdot 2^{-(2^1-1)} = 2^{-(n+2^1)}; \\ \Delta_{\max} &= 2^{-(n+1)} \cdot 2^{2^1-1} = 2^{-n+2^1-2}.\end{aligned}\tag{2.31}$$

Относительная погрешность представления чисел в машинах с плавающей запятой определяется по общему правилу

$$\varepsilon = \left| \frac{\Delta}{X} \right| = \frac{\Delta M \cdot 2^p}{M \cdot 2^p} = \frac{\Delta M}{M},\tag{2.32}$$

то есть не зависит от порядка числа и изменяется в следующих пределах:

$$\begin{aligned}\varepsilon_{\min} &= \frac{\Delta X}{X_{\max}} = \frac{2^{-(n+1)}}{1-2^{-n}} = 2^{-(n+1)}; \\ \varepsilon_{\max} &= \frac{\Delta X}{X_{\min}} = \frac{2^{-(n+1)}}{2^{-1}} = 2^{-n}.\end{aligned}\tag{2.33}$$

Следовательно, в машинах с плавающей запятой, в отличие от машин с фиксированной запятой, относительная погрешность изображения чисел во всём диапазоне представления практически постоянна и для чисел с нормализованной мантиссой зависит от количества разрядов мантиссы: чем их больше, тем меньше погрешность представления.

2.14. Формы представления двоичных и десятичных чисел в ЭВМ

Знаковый разряд двоичных чисел весом 2^m для целых и 2^0 для дробных чисел участвует совместно с числовыми разрядами в арифметических операциях. Знаковый разряд так же, как цифровые разряды, принимает значение 1 (это знак «−») и 0 (это знак «+»).

Для машинного представления отрицательных чисел используют прямой, обратный и дополнительный коды. При этом знаки чисел кодируются двоичными цифрами: «+» цифрой 0, а «−» цифрой 1.

Любой код положительного числа совпадает с самим числом.

Прямой код (ПК) числа – простейший код, в котором к абсолютной величине числа слева приписывается знаковый признак.

Для целого n -разрядного двоичного числа X связь между числом X и его изображением в прямом коде имеет вид

$$[X]_n = \begin{cases} X, & \text{если } X \geq 0; \\ 2^n + |X|, & \text{если } X \leq 0. \end{cases} \quad (2.34)$$

Для правильной двоичной дроби прямой код получим исходя из следующего соотношения:

$$[X]_n = \begin{cases} X, & \text{если } X \geq 0; \\ 1 + |X|, & \text{если } X \leq 0. \end{cases} \quad (2.35)$$

Пример. Найти прямой код для отрицательного числа -3 (1011) и -0.3 (0.0011) Пусть разрядная сетка имеет 8 разрядов и один разряд отводится для знака.

Примечание. Здесь и далее для целых чисел условно точкой будем отделять знаковый разряд от цифровой части числа.

На основании (2.33) и (2.34) получаем:

$X_{\text{ПК}}=1.0000011$ – для целых чисел;

$X_{\text{ПК}}=1,0000011$ – для правильных дробей.

Важная особенность прямого кода в том, что цифру знакового разряда и цифровую часть числа нельзя рассматривать как единое целое. Поэтому для этого кода при умножении и делении чисел операции со знаком и модулями исходных чисел выполняются отдельно. Выполнять операции сложения и вычитания в прямом коде возможно в том случае, если оба числа, участвующие в арифметических операциях имеют одинаковые знаковые разряды (оба положительные либо оба отрицательные). Операция сложения и вычитания в прямом коде неудобна, поскольку прямой код не обеспечивает замену вычитания чисел сложением их кодов. Поэтому перед выполнением операции алгебраического сложения числа преобразуются в один из инверсных кодов (обратный или дополнительный). Прямой код широко используется в ЗУ и устройствах ввода-вывода.

Обратный код (ОК) является дополнением модуля исходного числа до наибольшего числа без знака, помещающегося в разрядную сетку. Обратный код обладает определенными преимуществами. Он полностью симметричен, так как изображению максимального по абсолютному значению отрицательного числа соответствует $2^m + 2^0 / 2^0 + 2^{-m}$ сопоставляется изображение такого же максимального положительного числа $2^m - 2^0 / 2^0 - 2^{-m}$. Изображение положительных и отрицательных чисел взаимно дополняют друг друга, то есть до последовательности единиц во всех двоичных разрядах, то есть для получения ОК отрицательного числа необходимо взять инверсию всех двоичных разрядов.

Для n – разрядной сетки имеем для *целых чисел*:

$$[\overline{X}] = \begin{cases} \overline{X}, & \text{при } \overline{X} \geq 0; \\ 2^n - 1 - |\overline{X}|, & \text{при } \overline{X} \leq 0. \end{cases} \quad (2.36)$$

Для *правильной двоичной дроби*

$$[\overline{X}] = \begin{cases} \overline{X}, & \text{при } \overline{X} \geq 0; \\ 2 - 2^{-n} - |\overline{X}|, & \text{при } \overline{X} \leq 0. \end{cases} \quad (2.37)$$

Из соотношений (2.36) и (2.37) видно, что положительное число не меняет своего изображения в обратном коде, а обратный код отрицательного n -разрядного двоичного числа определяется из равенства:

$$\begin{aligned} [X]_0 &= 1. \overline{x_1} \overline{x_2} \dots \overline{x_n} \text{ — для целых чисел;} \\ [X]_0 &= 1, \overline{x_1} \overline{x_2} \dots \overline{x_n} \text{ — для правильных дробей,} \end{aligned} \quad (2.38)$$

где $\overline{X} = 1 - x_i$.

Пример. Найти обратный код для отрицательных чисел $X = -1011$ и $Y = -0,1011$. Результат представить 8-битным числом.

Решение. На основании выражений (2.38) для X и Y получим:

$$X_{\text{ОК}} = 1.1110100; \quad Y_{\text{ОК}} = 1,0100111.$$

В обратном коде можно изображать максимальное положительное число $X_{\text{max}} = 0,11\dots11 = 1 - 2^{-n}$ и наибольшее отрицательное число $X_{\text{min}} = -0,11\dots11 = -(1 - 2^{-n})$:

$$X_{\text{maxОК}} = 0,11\dots11; \quad X_{\text{minОК}} = 1,00\dots00.$$

Нуль в обратном коде имеет два изображения:

$$[+0]_0 = 0,00\dots00;$$

$$[-0]_0 = 1,11\dots11.$$

Изображение чисел в *дополнительном* коде (ДК) наиболее распространенное и не требует каких-либо дополнительных аппаратных дополнений. Изображение положительных чисел равно значению самих чисел. Например, изображение положительного числа +3 в дополнительном коде будет выглядеть следующим образом:

$$X_{\text{ДК}} = 0.0000011$$

Изображение отрицательных чисел представляет собой дополнение до 2^{m-1} для целых чисел и до 2^1 – для дробных. Поэтому сумма изображений двух одинаковых по абсолютному значению чисел разного знака равна $2^{m-1}/2^1$. В общем виде функцию изображения ДК для целых чисел можно представить следующим образом:

$$A_{\text{ДК}} = \begin{cases} A, & \text{при } 0 \leq A < 2^m; \\ 2^{m+1} - |A|, & \text{при } -2^m \leq A < 0; \end{cases} \quad (2.39)$$

для дробных

$$A_{\text{ДК}} = \begin{cases} A, & \text{при } 0 \leq A < 2^0; \\ 2^1 - |A|, & \text{при } -2^0 \leq A < 0. \end{cases} \quad (2.40)$$

Следует обратить внимание на несимметричность ДК в связи с тем, что изображению $A = -2^m / -2^0$ не сопоставляется изображение $A = +2^m / +2^0$. Необходимо помнить, что чем больше изображение отрицательного числа, тем меньше абсолютное значение самого изображаемого числа.

Существует два способа перевода чисел из прямого кода в дополнительный. Первый способ заключается в следующем:

- 1) записать число в прямом коде;
- 2) для ПК найти соответствующий обратный код;
- 3) к ОК числа прибавить к младшему разряду единицу;
- 4) полученное число будет дополнительным кодом заданного числа.

Например, найдем дополнительный код для отрицательного числа -6 (1.0110).

Для перевода числа по первому способу необходимо сделать следующую последовательность действий:

$X_{ПК}=1.0000110$ – прямой код числа -6 ;

$X_{ОК}=1.1111001$ – обратный код числа -6 ;

$+1$;

$X_{ДК}=1.1111010$ – дополнительный код числа -6 .

Второй способ перевода заключается в следующем:

1) записать число в прямом коде;

2) найти, просматривая с младших разрядов, первую встретившуюся единицу и все разряды слева от нее перевести в обратный код (за исключением знаковой);

3) все разряды справа от найденной единицы, включая найденную, оставить в прежнем виде.

По второму способу для того же числа получаем следующую последовательность действий:

$X_{ПК}=1.00001|10$

$X_{ДК}=1.1111010$

Важное достоинство дополнительного и обратного кодов заключается в том, что при выполнении арифметических операций сложения и вычитания цифру знакового разряда и цифровую часть числа можно рассматривать как единое целое и обращаться со знаковым разрядом так же, как и с разрядами цифровой части числа.

Во второй главе учебного пособия рассмотрели общие сведения о представлении информации в ЭВМ, основные вопросы, касающиеся информации, систем счисления и чисел, такие как представление информации в цифровых автоматах (ЦА); позиционные системы счисления; методы перевода чисел; форматы представления чисел с плавающей запятой.

Контрольные вопросы

1. Какие из операций с плавающей запятой считаются наиболее сложными? Ответ обоснуйте на конкретных примерах.

2. По каким причинам наибольшее распространение в ЭВМ получила двоичная система счисления?

3. Переведите числа 329; 701; 0,280 из десятичной системы счисления в двоичную, троичную и пятеричную.

4. В чем суть организации циклического переноса и циклического займа при выполнении арифметических операций в обратном коде?

5. Обоснуйте необходимость игнорирования выходных переносов и входных займов при арифметических операциях в дополнительном коде.

6. Можно ли предугадать одну из двух форм изображения нулевого результата арифметических операций в обратном коде?

7. Переведите числа 329; 701; 0,280 из десятичной системы счисления в двоичную в полулогарифмической форме с основанием порядка $X = 16$, изображая порядок в смещенном дополнительном коде, а мантиссе — в обычном дополнительном.

8. Объясните необходимость коррекции первичной суммы на плюс шесть и первичной разности на минус шесть при выполнении соответствующих операций с числами, представленными в двоично-десятичном коде прямого замещения (8421).

9. Какие арифметические действия (сдвиги, инкременты, декременты, сложения, вычитания, умножения, деления), в какой последовательности и взаимосвязи необходимо выполнить при реализации четырех основных арифметических операций над числами, представленными в двоичном коде и в полулогарифмической форме (с порядками)? Выполните примеры на сложение, умножение и деление.

10. Какие из следующих цепочек символов являются шестнадцатеричными числами: BED, CAB, DEAD, DECADE, ACCEDED, BAG, DAD?

11. Сколько различных положительных чисел можно выразить в k разрядах, используя числа с основанием системы счисления g ?

12. Большинство людей с помощью пальцев на руках могут сосчитать до 10. Однако компьютерщики способны на большее. Представим, что каждый палец соответствует одному двоичному разряду. Пусть вытянутый палец означает 1, а загнутый - 0. До скольки вы сможете сосчитать, используя пальцы обеих рук? А если рассматривать пальцы на руках и ногах? Представим, что большой палец левой ноги - это знаковый бит для чисел с дополнением до двух. Сколько чисел можно выразить таким спосо-

бом?

13. Система с дополнением до десяти аналогична системе с дополнением до двух. Отрицательное число в системе с дополнением до десяти получается путем прибавления 1 к соответствующему числу с дополнением до девяти без учета переноса. По какому правилу происходит сложение в системе с дополнением до десяти?

14. Получить шестнадцатеричную форму внутреннего представления числа в формате с плавающей точкой в 4-х байтовой ячейке памяти: $A=26.28125$

15. Получить шестнадцатеричную форму внутреннего представления числа в формате с плавающей точкой в 4-х байтовой ячейке памяти: $A=-139.375$

16. По шестнадцатеричной форме внутреннего представления вещественного числа в 4-х байтовой ячейке восстановить само число: $A=C5DB0000$

17. По шестнадцатеричной форме внутреннего представления вещественного числа в 4-х байтовой ячейке восстановить само число: $A=45ED0000$

3. ДВОИЧНАЯ АРИФМЕТИКА

Чтобы овладеть любой системой счисления, надо уметь складывать и умножать в ней любые числа. Арифметические действия в двоичной системе счисления выполняют по тем же правилам, что и в десятичной системе, с той лишь разницей, что основание системы равно двум [10].

3.1. Правила двоичной арифметики

Сложение и вычитание двоичных чисел основаны на правилах этих действий в пределах одного разряда и правилах учета межразрядных переносов и займов.

Для операций сложения, вычитания и умножения используются правила, приведенные в таблице 3.1.

Таблица 3.1

Правила арифметических операций

Сложение	Вычитание	Умножение
$0+0=0$	$0-0=0$	$0 \cdot 0 = 0$
$0+1=1$	заем $\rightarrow 0-1=1$	$0 \cdot 1 = 0$
$1+0=1$	$1-0=1$	$1 \cdot 0 = 0$
$1+1=1 \rightarrow$ перенос	$1-1=0$	$1 \cdot 1 = 1$

Перенос, возникающий в i -м разряде, передается в следующий $(i+1)$ -разряд с увеличенным вдвое весом и уменьшенным вдвое значением.

Заем из $(i+1)$ -го разряда передается в i -й разряд с уменьшенным вдвое весом и увеличенным вдвое значением.

Приведем пример сложения двух двоичных чисел. Справа показано сложение тех же чисел в десятичной системе счисления. Следует обратить внимание на то, что перенос в соседний (старший) разряд возникает в том случае, если сумма цифр данного разряда больше или равна основанию системы счисления.

$$\begin{array}{r}
 \text{переносы } 1 \quad 1 \ 1 \ 1 \ 1 \\
 \quad 1 \ 0 \ 0 \ 1 \ 1, \ 1 \ 1 \\
 + \quad 1 \ 1 \ 0 \ 0 \ 1, \ 0 \ 1 \\
 \hline
 1 \ 0 \ 1 \ 1 \ 0 \ 1, \ 0 \ 0
 \end{array}
 \qquad
 \begin{array}{r}
 1 \ 1 \ 1 \\
 1 \ 9, \ 7 \ 5 \\
 + 2 \ 5, \ 2 \ 5 \\
 \hline
 4 \ 5, \ 0 \ 0
 \end{array}$$

При вычитании двоичных чисел (см. табл. 3.1) в данном разряде при необходимости занимается единица из соседнего (старшего) разряда. Эта занимаемая единица равна двум едини-

цам данного разряда. Заем производится каждый раз, когда цифра в разряде вычитаемого больше цифры в том же разряде уменьшаемого. Например, при вычитании:

$$\begin{array}{r} 10 \\ 10\cancel{1}0011 \\ + 1001 \\ \hline 1001010 \end{array}$$

единица из разряда с весом 2^4 была занята в разряд с весом 2^3 ; эта единица стала там двойкой, и в разряде с весом 2^3 выполнилось вычитание $10-1=1$; на месте разряда с весом 2^4 в уменьшаемом фактически остался нуль.

Распространение займа сразу на несколько более старших разрядов можно проследить на примере вычитания чисел $101110,001_{(2)}$ и $101,011_{(2)}$. Записав числа друг под другом:

$$\begin{array}{r} 101110,001 \\ - 101,011 \\ \hline \end{array}$$

нетрудно заметить, что в разряде с весом 2^{-2} в результате вычитания должен произойти заем из разряда с весом 2^1 . Перепишем пример с учетом фактического расположения цифр после заема и выполним вычитание. Вместо зачеркнутых цифр необходимо использовать в качестве уменьшаемого надписанные цифры. Окончательный результат (разность) составляет $101000,110_{(2)}$.

$$\begin{array}{r} 1011\overset{\cdot}{1}\overset{\cdot}{0},\overset{\cdot}{0}\overset{\cdot}{0}1 \\ - 101,011 \\ \hline 101000,110 \end{array}$$

Пример. Уменьшаемое $1000000_{(2)}$, вычитаемое $1_{(2)}$, разность составляет

$$\begin{array}{r} 1000000 \\ - 1 \\ \hline 0111111 \end{array}$$

В соответствии с правилами можно эффективно организовать последовательное умножение множимого на разряды множителя. При каждом умножении на разряд множителя, равный 1, множимое передается в сумматор с накапливающим регистром; если разряд множителя равен 0, передача множимого в сумматор

блокируется. Каждый раз при передаче множимого в сумматор должен быть учтен вес очередного разряда множителя путем сдвига накапливаемого частичного произведения или множимого. Таким образом, основу устройства умножения составляет устройство сложения, к которому добавляются регистры множителя и множимого, а также цепи сдвига частичных произведений и множимого.

Операция деления выполняется путем последовательных вычитаний делителя из промежуточных остатков, а устройство деления состоит из вычитателя с накапливающим регистром, регистра частного и регистра делителя с цепями сдвига остатков или делителя.

В основном арифметические операции выполняются на одном общем устройстве, называемом арифметико-логическим устройством (АЛУ).

Старшие разряды сумматоров с наименьшими весами разрядов участвуют в операциях сложения как обычные числовые разряды, но дополнительно они выполняют функции знаковых разрядов.

3.2. Арифметические операции в двоичной системе счисления

Рассмотрим основные операции (сложение и вычитание) с использованием ОК и ДК. Для примера возьмем числа 25, 31 и проведем с ними все возможные операции сложения и вычитания:

$$1) 31 + 25 = 56$$

Данный вариант самый простой и не требует никаких преобразований перед сложением, поэтому необходимо просто перевести числа в двоичную систему счисления и сложить с помощью обычной операции сложения:

$$\begin{array}{r}
 31 \rightarrow \quad 0.0011111 \\
 25 \rightarrow \quad 0.0011001 \\
 \hline
 0.0111000 \rightarrow 32+16+8=56
 \end{array}$$

$$2) 31 + (-25) = +6$$

В данном случае перед сложением необходимо отрицательное число перевести предварительно либо в ОК, либо в дополнительный код, после чего произвести обычную операцию сложения. В качестве примера переведем ПК в ДК:

$$\begin{array}{r} -25_{\text{ПК}} \rightarrow \quad 1.0011001 \\ -25_{\text{ДК}} \quad \quad 1.1100111 \end{array}$$

После перевода числа в дополнительный код проводим основной цикл сложения:

$$\begin{array}{r} +31 \rightarrow \quad 0.0011111 \\ \quad \quad \quad + \\ -25_{\text{ДК}} \rightarrow \quad 1.1100111 \\ \hline 0.0000110 \rightarrow +6 \end{array}$$

так как «0» в знаковом разряде, то $C_{\text{доп}} = C_{\text{ПК}}$.

В случае появления знакового разряда со значением «1», результат представлен в ДК и необходим перевод в ПК для получения результата.

$$3) -31 + 25 = -6$$

В данном случае отрицательное число необходимо предварительно перевести либо в ДК, либо в обратный код, после чего провести сложение. В случае получения отрицательного числа результат необходимо проинвертировать (заменить единицу на ноль, а ноль на единицу) для получения ПК. В примере воспользуемся переводом в ОК:

$$\begin{array}{r} -31_{\text{ПК}} \quad \quad 1.0011111 \\ -31_{\text{ОК}} \quad \quad 1.1100000 \\ -31 \rightarrow \quad \quad 1.1100000 \\ \quad \quad \quad + \\ 25 \rightarrow \quad \quad 0.0011001 \\ \hline 1.1111001 \end{array}$$

После основного сложения, так как результат отрицательный, производим его преобразования в прямой код, основываясь на методике, описанной в п. 2.14. В результате получаем ПК=1.0000110, то есть число -6.

Сумматор ОК отличается от других сумматоров тем, что в них имеет место цепь обратной связи (циклический перенос). При появлении переноса из знакового разряда к результату необходимо добавить «+1» в младший разряд.

$$4) 31 + (-25) = +6$$

В данном случае отрицательное число необходимо перевести в обратный код. После перевода получим $-25_{\text{ОК}} = 1.1100110$.

Производим операцию сложения в ОК:

$$\begin{array}{r}
 31 \rightarrow \quad 0.0011111 \\
 \quad \quad \quad + \\
 -25_{\text{ОК}} \rightarrow \quad 1.1100110 \\
 \quad \quad \quad \text{-----} \\
 \quad \quad \quad \leftarrow 0.0000101 \\
 \quad \quad \quad + \\
 \quad \quad \quad \quad \quad \quad +1 \\
 \quad \quad \quad \quad \quad \quad \text{-----} \\
 \quad \quad \quad \quad \quad \quad 0.0000110 \rightarrow +6
 \end{array}$$

$$5) 31 - (-25) = +56$$

По правилам арифметики $A - (-B) = A + B$. В данном случае два отрицания превращаются в обычное сложение, и поэтому операция сложения в данном случае проводится аналогично примеру 1.

$$\begin{array}{r}
 31 \rightarrow \quad 0.0011111 \\
 \quad \quad \quad + \\
 25 \rightarrow \quad 0.0011001 \\
 \quad \quad \quad \text{-----} \\
 \quad \quad \quad 0.0111000 \rightarrow 32 + 16 + 8 = 56
 \end{array}$$

$$6) -31 + (-25) = -56$$

В данном случае перед сложением необходимо оба числа сначала перевести в обратный код, после чего проводить операцию сложения:

$$\begin{array}{r}
 -31 \rightarrow \quad 1.1100000 \\
 -25 \rightarrow \quad 1.1100110 \\
 \hline
 \leftarrow 1.1000110 \\
 \quad \quad \quad +1 \\
 \hline
 1.0111000
 \end{array}$$

Полученный результат представлен в обратном коде, поэтому для получения ПК его необходимо проинвертировать. Получим $1.0111000 \rightarrow -56$.

Таким образом, используя обратный и дополнительный коды можно выполнить операции сложения и вычитания чисел с различными знаками в двоичной системе счисления.

3.3. Выполнение операции сложения в D -кодах

3.3.1. Сложение двоичных чисел

Сложение двоичных чисел в коде прямого замещения (D_1). Так как наибольшее десятичное одноразрядное число равно 9, то, с учетом переноса в данный разряд, значение результата разрядного суммирования лежит в пределах от 0 до 19. При этом единица во втором разряде представляет собой десятичный перенос в следующую тетраду, а сумма получается в двоичном коде, отличном от требуемого десятичного представления, то есть требует коррекции. В связи с этим при сложении могут возникнуть следующие случаи:

1) если $a_i + b_i + p_i < 10$, то при выполнении действий над разрядами тетрады по правилам двоичной арифметики сразу получается правильный результат;

2) если $a_i + b_i + p_i \geq 10$, то возникает тетрадный перенос, который следует рассматривать как десятичный. Сумму следует скорректировать на +6 (+0110), так как перенос передан в старший разряд с весом 16 вместо 10. Признаком необходимости

коррекции является выходной перенос в старшую тетраду (межтетрадный перенос);

3) если $15 \geq a_i + b_i + p_i \geq 10$, то необходима такая же коррекция из-за превышения допустимого значения суммы (т.е. появляется запрещенная комбинация).

Запрещенные комбинации – это следующие значения:

$$10 = 1010$$

$$11 = 1011$$

$$12 = 1100$$

$$13 = 1101$$

$$14 = 1101$$

$$15 = 1111$$

Признаком необходимости коррекции является единица в старшем разряде и хотя бы по одной единице в 2-х соседних кодах первоначальной суммы.

На первом шаге проводим основной цикл сложения, после этого производим коррекцию в тетрадах, имеющих или тетрадный перенос, или в которых возникает в результате сложения запрещенная комбинация.

Пример: $184 + 298 = 482$ (код 8421)

0001 1000 0100

+

0010 1001 1000

0100 ← 0001 1100

+

0000 0110 0110

0100 1000 0010

В данном примере в младшей тетраде присутствует запрещенная комбинация 1100, значит ее необходимо скорректировать на +6. Из второй тетрады в третью был межтетрадный перенос, поэтому к тетраде, из которой был перенос, прибавим +6 для коррекции результата.

Прямое вычитание десятичных разрядов всегда меньше 10, поэтому разность необходимо скорректировать на минус шесть только при возникновении шестнадцатеричного займа, так как

десятичный разряд приобретает в данном случае лишних шесть единиц.

Пример: произвести вычитание чисел в коде прямого за-
мещения (код 8421)

$$615 - 396 = 219$$

0110 0001 0101

-

0011 1001 0110

0010 → 0111 → 1111

-

0000 0110 0110

0010 0001 1001

В данном случае поправки делаются для тех тетрад, для ко-
торых был сделан заем.

Сложение двоичных чисел в коде 8421+3 (D_4). В данном
коде перед сложением необходимо представить десятичное чис-
ло в двоичном виде, основываясь на следующем правиле:

$$a'_i = a_i + 3; \quad b'_i = b_i + 3,$$

то есть для получения разряда двоичного числа необходимо
к исходному десятичному числу прибавить цифру 3 и записать
результат в виде тетрады.

В данном коде при сложении могут возникнуть следующие
случаи:

1) если $a_i + b_i + p_i \leq 15$, то результат необходимо скорректи-
ровать на величину -3 (-0011). Так как операция вычитания про-
изводится через операцию сложения, то -3 (-0011) заменяем на
 1101 ($+13$) с блокировкой межтетрадного переноса;

2) если $a_i + b_i + p_i > 15$, то при переходе в старшую тетраду
(возник межтетрадный перенос) меняет свой вес с 16 на 10. По-
этому требуется коррекция на величину $+3$ (0011).

В качестве примера рассмотрим сложение чисел $184 + 298 =$
 482 .

0100 1011 0111

0101 1100 1011

1010←-1000←-0010

-0011 +0011+0011

0111 1011 0101

В данном случае в двух младших тетрадах возникли межтетрадные переносы, поэтому их необходимо скорректировать на величину +3. В старшей тетраде появилось число меньше 15, поэтому эту тетраду необходимо скорректировать на величину -3 (0011).

3.3.2. Сложение в коде D_1 с использованием обратного кода

Как и в обычном двоичном сложении чисел со знаком, перед сложением двоично-десятичных чисел необходимы их предварительные преобразования. Рассмотрим способы сложения с применением обратного кода (ОК).

Существует два способа нахождения обратного кода. В первом случае для получения обратного кода числа, записанного в виде тетрад, необходимо для каждой тетрады найти ее дополнение до числа 9. После этого необходимо провести сложение по правилам сложения чисел в коде прямого замещения с учетом необходимых поправок. После получения результата, если число отрицательное, необходимо снова найти дополнение каждой тетрады до числа 9 для получения необходимого результата (кода ПК). В качестве примера рассмотрим сложение чисел:

$$-298 + 127 = -171$$

Предварительно перед сложением представим отрицательное число в обратном коде.

$$-298_{\text{ПК}}: 1. 0010 1001 1000$$

$$-298_{\text{ОК}}: 1. 0111 0000 0001$$

В данном случае для дополнения младшей тетрады до числа 9 необходимо число 1, для дополнения второй тетрады необходимо ноль, а для дополнения старшей тетрады – число 7. После перевода отрицательного числа в ОК проводим основной цикл сложения:

$$\begin{array}{r}
 1. 0111\ 0000\ 0001 \\
 0. 0001\ 0010\ 0111 \\
 \hline
 1. 1000\ 0010\ 1000
 \end{array}$$

Получили результат, записанный в обратном коде. Для получения конечного результата находим его дополнение до числа 9. Аналогично, дополняя разряды до девяти, получаем:

$$1. 0001\ 0111\ 0001 \quad \text{или} \quad -171.$$

При этом необходимо учитывать, что если существует перенос из знакового разряда, то эту единицу прибавляем к младшей тетраде.

Во втором способе последовательность действий следующая. Первоначально необходимо записать исходное число в виде тетрад в прямом коде. После этого прибавляем к каждой тетраде число +6 (0110), после чего результат инвертируем. Это и будет обратный код исходного числа. После этого проводим обычное сложение, как и в первом способе, после чего при получении в результате сложения отрицательного числа переведем его в ПК путем добавления +6 и инверсии.

$$\begin{array}{r}
 1. 0010\ 1001\ 1000 \\
 + \\
 0. 0110\ 0110\ 0110 \\
 \hline
 1. 1000\ 1111\ 1110 \\
 1 \quad 0111\ 0000\ 0001
 \end{array}$$

Полученное число и будет обратным кодом. Как видно, оно полностью совпадает с обратным кодом, полученным в первом случае. Теперь проводим основной цикл сложения.

$$\begin{array}{r}
 1. 0111\ 0000\ 0001 \\
 + \\
 0. 0001\ 0010\ 0111
 \end{array}$$

$$\begin{array}{r}
 \text{-----} \\
 1. 1000\ 0010\ 1000 \\
 + \\
 \quad 0110\ 0110\ 0110 \\
 \text{-----} \\
 1.1110\ 1000\ 1110 \\
 1.0001\ 0111\ 0001 \rightarrow -171
 \end{array}$$

После основного цикла сложения находим прибавление +6 с последующей инверсией.

3.3.3. Сложение в коде D_4 с использованием кода ДК

Для получения дополнительного кода, записанного в коде D_4 , необходимо найти дополнение всех тетрад, кроме младшей, до числа 15, а младшую тетраду – до числа 16. После основного цикла сложения необходимо произвести коррекции. В случае возникновения межтетрадных переносов тетрады, из которых был перенос, необходимо скорректировать на число +3 (0011). Остальные тетрады корректируются на число +13 (1101) с блокировкой межтетрадных переносов. После коррекции, если число отрицательное, то для него также необходимо найти дополнение до 15- и 16-ти, как и в начале вычислений.

В качестве примера возьмем числа -298 и $+127$ и выполним операцию сложения.

Сначала необходимо отрицательное число перевести в дополнительный код с использованием описанной выше методики. Получаем:

$$\begin{array}{r}
 -298_{\text{ПК}} \quad 1\ 0101\ 1100\ 1011 \\
 -298_{\text{ДК}} \quad 1\ 1010\ 0011\ 0101
 \end{array}$$

Затем производим основной цикл сложения:

```

1. 1010 0011 0101
0. 0100 0101 1010
-----
1. 1110 1000 1111
0. 1101 1101 1101   коррекция
-----
1. 1011 0101 1100   результат

```

После основного цикла сложения получаем отрицательное число, которое представлено в дополнительном коде. Для получения прямого кода необходимо найти дополнения младшей тетрады до 16-ти, а всех остальных до 15-ти. Получаем следующий результат:

```
1. 0100 1010 0100
```

Учитывая, что полученное число записано в коде D_4 , после вычитания из каждой тетрады цифры 3 получим значение -171 .

3.3.4. Алгоритм сложения с избытком шесть

Данный алгоритм сложения использует избыточные на плюс шесть значения разрядов одного из слагаемых. Если в разряде суммы возникнет перенос, то значение суммы правильное, так как оно было скорректировано заранее. Если переноса нет, то предварительная коррекция была не нужна, и избыток на плюс шесть необходимо компенсировать вычитанием шести или эквивалентным ему сложением с десятью и с игнорированием выходного переноса (десять есть дополнение шести до шестнадцати).

Для пояснения данного алгоритма выполним сложение $184+298=482$ в коде прямого замещения с предварительной коррекцией первого слагаемого на плюс шесть. Получаем следующие тетрады для числа 184:

```
0111 1110   1010
```

Второе слагаемое записываем в виде тетрад без изменений.

0010 1001 1000

Проводим основной цикл сложения:

0111 1110 1010

+

0010 1001 1000

1010 ←1000 ←0010

После основного цикла сложения видно, что в первой и второй тетрадах был перенос. Это означает, что эти тетрады были скорректированы удачно и коррекция не нужна. В третьей тетраде никаких переносов не было, и это означает, что предварительная коррекция была не нужна и необходимо выполнить коррекцию. Для этого прибавим к этой тетраде число 10_{10} с блокировкой выходного переноса:

1010 1000 0010

+

1010 0000 0000

0100 1000 0010 → 482

То есть для выполнения операции сложения с избытком шесть, при выполнении представленного выше примера, необходимо выполнить коррекцию только первой тетрады, так как при использовании традиционного алгоритма сложения требовалось бы корректировать две тетрады.

3.4. Выполнение операции умножения в двоичной системе счисления

По сравнению со сложением и вычитанием, умножение – более сложная операция, как при программном, так и при аппаратном воплощении. В ЭВМ применяются различные алгоритмы реализации операции умножения и, соответственно, несколько

схем построения операционных блоков, обеспечивающих выполнение операции умножения.

При точном умножении двух чисел количество значащих цифр произведения может в пределе достичь двойного количества значащих цифр сомножителей. Правила приближенных вычислений рекомендуют оставлять в произведении столько же значащих цифр, сколько их содержится в наименее точном из сомножителей.

Наиболее просто операция умножения выполняется в ПК, при этом на первом этапе определяется знак произведения путем сложения знаковых цифр сомножителей по модулю два.

Традиционная схема умножения похожа на известную из школьного курса процедуру записи «в столбик». Вычисление произведения $P(p_{2n-1}, p_{2n-2}, \dots, p_1, p_0)$ двух n -разрядных двоичных чисел без знака $A(a_{2n-1}, a_{2n-2}, \dots, a_1, a_0)$ и $B(b_{2n-1}, b_{2n-2}, \dots, b_1, b_0)$ сводится к формированию частичных произведений (ЧП) W_i (по одному на каждую цифру множителя) с последующим суммированием полученных ЧП. Перед суммированием каждое частичное произведение должно быть сдвинуто на один разряд относительно предыдущего согласно весу цифры множителя, которой это ЧП соответствует. Поскольку операндами являются двоичные числа, вычисление ЧП упрощается: если цифра множителя b_i равна 0, то W_i тоже равно 0, а при $b_i = 1$ частичное произведение равно множимому ($W_i = A$). Перемножение двух n -разрядных двоичных чисел $P = A \times B$ приводит к получению результата, содержащего $2n$ битов. Таким образом, алгоритм умножения предполагает последовательное выполнение двух операций – сложения и сдвига (рис. 3.1).

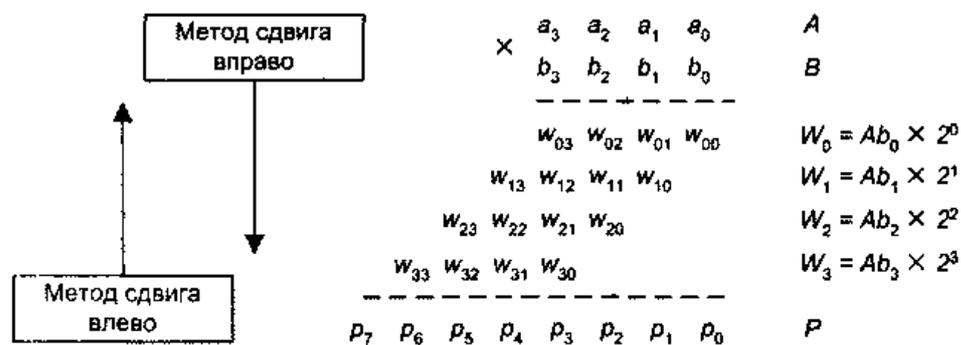


Рис. 3.1. Общая схема умножения со сдвигом суммы

частичных произведений влево или вправо

Суммирование ЧП обычно производится не на завершающем этапе, а по мере их получения. Это позволяет избежать необходимости хранения всех ЧП, то есть сокращает аппаратные издержки. Согласно данной схеме устройство умножения предполагает наличие регистров множимого, множителя и суммы частичных произведений, а также сумматора ЧП и, возможно, схем сдвига, если операция сдвига не реализована иным способом, например, за счет «косой» передачи данных между узлами умножителя.

В зависимости от способа получения суммы частичных произведений (СЧП) возможны четыре варианта реализации «традиционной» схемы умножения [10]:

1) начиная с младших разрядов множителя, со сдвигом суммы частичных произведений вправо и при неподвижном множимом;

2) начиная со старших разрядов множителя, при сдвиге суммы частичных произведений влево и неподвижном множимом;

3) начиная с младших разрядов множителя, при сдвиге множимого влево и неподвижной сумме частичных произведений;

4) начиная со старших разрядов множителя, со сдвигом множимого вправо и при неподвижной сумме частичных произведений.

3.4.1. Умножение младшими разрядами множителя со сдвигом СЧП вправо при неподвижном множимом

Полученное выражение можно представить в виде схемы Горнера для вычисления полиномов

$$C = (((...((0 + A + Ab_n)2^{-1} + Ab_{n-1})2^{-1} + \dots + Ab_{n-1})2^{-1} + \dots + Ab_{n-1})2^{-1} + Ab_{n-1})2^{-1}. \quad (3.1)$$

Это выражение может быть сведено к n -кратному выполнению цикла $C_{i+1} = (C_i + Ab_{n-1})2^{-1}$ при начальных условиях $i=0$; $C_0=0$.

В каждом цикле множимое либо добавляется к СЧП (если $b_i=1$), либо нет (если $b_i=0$), после этого сумма умножается на 2^{-1} , то есть сдвигается на один разряд вправо. После окончания n -го цикла образуется искомое произведение, то есть $C_n=C=AB$.

Очередную цифру множителя, управляющую суммированием частичных произведений, удобнее всего снимать с младшего разряда регистра множителя, в котором в каждом цикле производится сдвиг содержимого на один разряд вправо.

```

12×13=156
0.0001100 множимое
0.0001101 множитель
0.0000000 СЧП
0|0001100
  0000000
  0001100 1 шаг
 00001100 сдвиг
000001100 2 шаг + 0 → сдвиг
 0001100
 000111100 3 шаг + А
 0000111100 сдвиг
 0001100
 0010011100 4 шаг + А
 00010011100 сдвиг
000010011100 + 0 → сдвиг
0000010011100 + 0 → сдвиг
00000010011100 + 0 → сдвиг
128+16+8+4 =156

```

Знак результата получается путем сложения знаковых разрядов сомножителей (если знаковые разряды не участвуют в операции умножения) и присваивается результату до операций сложения и сдвига.

3.4.2. Умножение младшими разрядами множителя со сдвигом множимого влево при неподвижном СЧП

Выражение представим в следующем виде:

$$C = 2^{-n} (Ab_n + 2Ab_{n-1} + \dots + 2^i Ab_{n-i} + \dots + 2^{n-1} Ab_{n_1}). \quad (3.2)$$

Вычисление этого выражения сводится к n -кратному выполнению цикла $C_{i+1} = C_i + A_i b_{n-i}$, где $A_i = 2A_{i-1}$ при начальных значениях $i=0$, $C_0=0$, $A_0=A$.

В каждом цикле умножения множимое сдвигается на один разряд влево и либо прибавляется к СЧП (при $b_i=1$), либо нет (при $b_i=0$).

Например, выполним вычисление $12 \times 13 = 156$:

0.0001100 множимое

0.0001101 множитель

0.0000000 СЧП

0001100

0000000 1 шаг СЧП + А

0001100 СЧП

00011000 сдвиг А на один разряд

000110000 сдвиг А на один разряд ($b_i=0$)

0001100 СЧП

000111100 СЧП

0001100000 сдвиг А на один разряд

0010011100 СЧП+А ($b_i=1$)

00011000000 сдвиг А на один разряд

000110000000 сдвиг А на один разряд ($b_i=0$)

0001100000000 сдвиг А на один разряд ($b_i=0$)

00011000000000 сдвиг А на один разряд ($b_i=0$)

В результате имеем $128+16+8+4 = 156$.

3.4.3. Умножение старшими разрядами множителя со сдвигом СЧП влево при неподвижном множимом

Выражение преобразуется к виду:

$$C = 2^{-(n+1)} ((\dots((0 + Ab_1)2 + Ab_2)2 + \dots + Ab_i)2 + \dots + Ab_{n-1})2 + Ab_n)2 \quad (3.3)$$

При этом умножение сводится к n -кратному повторению цикла $C_{i+1} = (C_i + Ab_{i+1})2, i = 0, C_0 = 0$.

Тогда управление умножением будет производиться цифрами множителя, начиная со старших разрядов. СЧП в каждом цикле будет сдвигаться на один разряд влево.

Например, выполним вычисления $12 \times 13 = 156$:

```

0.0001100 множимое
0.0001101 множитель
0000000 СЧП
00000000 сдвиг СЧП влево (bi=0)
000000000 сдвиг СЧП влево (bi=0)
0000000000 сдвиг СЧП влево (bi=0)
0001100 + А, т.к. (bi=1)
0000001100 СЧП
00000011000 сдвиг СЧП влево
0001100 + А, т.к. (bi=1)
00000100100
000001001000 сдвиг СЧП влево
0000010010000 сдвиг СЧП влево (bi=0)
0001100 + А, т.к. (bi=1)
0000010011100 СЧП
Получаем результат 128+16+8+4 = 156.

```

3.4.4. Умножение старшими разрядами множителя со сдвигом множимого вправо при неподвижном СЧП

Выражение представляется в виде

$$C = A2^{-1}b_1 + A2^{-2}b_2 + \dots + A2^{-i}b_i + \dots + A2^{-n}b_n. \quad (3.4)$$

Тогда вычисление произведения может быть сведено к n -кратному выполнению цикла:

$$A_{i+1} = A_i 2^{-1};$$

$$C_{i+1} = C_i + A_{i+1}b_{i+1}, \quad i = 0, A_0 = A, C_0 = 0,$$

то есть в каждом цикле множимое сдвигается на один разряд вправо и в зависимости от значения управляющего разряда множителя либо прибавляется к СЧП, либо нет.

Как и в предыдущих случаях, выполним вычисление $12 \times 13 = 156$. После выполнения преобразований в двоичную систему счисления получаем:

```

0001100 множимое
0001101 множитель
0000000 СЧП
00001100 сдвиг А на разряд вправо, т.к. разряд  $b_i=0$ 
000001100 сдвиг А на разряд вправо, т.к. разряд  $b_i=0$ 
0000001100 сдвиг А на разряд вправо, т.к. разряд  $b_i=0$ 
0000000      + СЧП
0000001100 СЧП
00000001100 сдвиг А на разряд вправо
0000001100      СЧП
00000100100 СЧП
000000001100 сдвиг А на разряд вправо
0000000001100 сдвиг А на вправо, т.к. разряд  $b_i=0$ 
00000100100
000001001100 СЧП → 156.

```

3.5. Умножение чисел со знаком

Несколько сложнее обстоит дело с умножением чисел со знаком, когда n -разрядные сомножители содержат знак (в старшем разряде слова) и $(n-1)$ значащую цифру. В дальнейшем условимся отделять знаковый разряд точкой, не забывая однако, что знаковый разряд участвует в операции наряду с цифровыми разрядами.

Наиболее очевидная мысль — получить абсолютные значения операндов и перемножить их как числа без знака. Справедливость такого решения видна из примера, приведенного на рис. 3.2 (показан процесс умножения чисел $+13$ и $+10$).

$$\begin{array}{r}
 \times \quad 0.1101 \quad +13 \\
 \quad \quad 0.1010 \quad +10 \\
 \hline
 \quad \quad 0.0000 \\
 \hline
 + \quad 0.0000 \\
 \quad \quad 0.0000 \\
 \Rightarrow \quad 0.00000 \\
 \hline
 + \quad 0.1101 \\
 \quad \quad 0.11010 \\
 \hline
 \Rightarrow \quad 0.011010 \\
 \hline
 + \quad 0.0000 \\
 \quad \quad 0.011010 \\
 \hline
 \Rightarrow \quad 0.0011010 \\
 \hline
 + \quad 0.1101 \\
 \quad \quad 1.0000010 \\
 \hline
 \Rightarrow \quad 0.10000010 \quad +130
 \end{array}$$

Рис. 3.2. Пример умножения

Во всех вычислительных машинах (ВМ) общепринято представлять числа со знаком в форме с фиксированной запятой в дополнительном коде (ДК). Положительные числа в этом представлении не отличаются от записи в прямом коде, а отрицательные записываются в виде $2^n - x$, где x — фактическое значение числа. В двоичной системе запись отрицательного числа в дополнительном коде сводится к инвертированию всех цифровых разрядов числа, представленного в прямом коде, и прибавлению единицы к младшему разряду получившегося после инвертирования обратного кода.

По этой причине более предпочтительны варианты, не требующие преобразования сомножителей и обеспечивающие вычисления непосредственно в дополнительном коде. Первая из особенностей умножения проявляется при выполнении операции арифметического сдвига вправо для суммы частичных произведений — освободившиеся при сдвиге цифровые позиции должны заполняться не нулем, а значением знакового разряда сдвигаемого числа. Здесь, однако, следует учитывать, что это правило заполнения освободившихся цифровых разрядов начинает действовать лишь с момента, когда среди анализируемых разрядов множителя появляется первая единица.

3.5.1. Множимое произвольного знака

Пример для *положительных сомножителей* ($A > 0, B > 0$) был рассмотрен выше, в случае же отрицательного множимого процедура умножения протекает аналогично (рис. 3.3), но с учетом сделанного выше замечания об арифметическом сдвиге СЧП.

$$\begin{array}{r}
 \times \quad 1.0011 \quad -13 \\
 \quad \quad 0.1010 \quad +10 \\
 \hline
 \quad \quad 0.0000 \\
 \hline
 + \quad 0.0000 \\
 \quad \quad 0.0000 \\
 \Rightarrow 0.00000 \\
 \hline
 + \quad 1.0011 \\
 \quad \quad 1.00110 \\
 \Rightarrow 1.100110 \\
 \hline
 + \quad 0.0000 \\
 \quad \quad 1.100110 \\
 \Rightarrow 1.1100110 \\
 \hline
 + \quad 1.0011 \\
 \quad \quad 0.1111110 \\
 \Rightarrow 1.01111110 \quad -130 \\
 \hline
 \end{array}$$

Рис. 3.3. Пример умножения с сомножителями разного знака

Поскольку результат умножения отрицательный, он получается в ДК, и для получения правильного результата его необходимо перевести в ПК.

3.5.2. Множимое произвольного знака. Множитель отрицательный

Так как множитель отрицателен, он записывается в дополнительном коде; $[B]_д = 2^n - |B|$, и в цифровых разрядах кода будет представлено число $2^{n-1} - |B|$. При типовом умножении (как в случае $B > 0$) получим $P' = A \times (2^{n-1} - |B|) = -|B| \times A + A \times 2^{n-1}$.

Псевдопроизведение P' больше истинного произведения P на величину $A \times 2^{n-1}$, что и необходимо учитывать при формировании окончательного результата. Для этого перед последним сдвигом из полученного псевдопроизведения необходимо вычесть избыточный член. На рисунках 3.4 и 3.5 приведены примеры умножения положительного и отрицательного множимого на отрицательный множитель, в которых видна упомянутая коррекция результата.

×	0.1101	+13
	1.0110	-10
	0.0000	
+	0.0000	
	0.0000	
	⇒ 0.00000	
+	0.1101	
	0.11010	
	⇒ 0.011010	
+	0.1101	
	1.001110	
	⇒ 0.1001110	
+	0.0000	
	0.1001110	
	⇒ 0.01001110	
+	1.0011	Коррекция
	1.01111110	-130

Рис. 3.4. Пример умножения положительного множимого на отрицательный множитель

$$\begin{array}{r}
 \times \quad 1.0011 \quad -13 \\
 \quad 1.0110 \quad -10 \\
 \hline
 \quad 0.0000 \\
 \hline
 + \quad 0.0000 \\
 \quad 0.0000 \\
 \Rightarrow 0.00000 \\
 \hline
 + \quad 1.0011 \\
 \quad 1.00110 \\
 \hline
 \Rightarrow 1.100110 \\
 \hline
 + \quad 1.0011 \\
 \quad 0.110010 \\
 \hline
 \Rightarrow 1.0110010 \\
 \hline
 + \quad 0.0000 \\
 \quad 1.0110010 \\
 \hline
 \Rightarrow 1.10110010 \\
 \hline
 + \quad 0.1101 \quad \text{Коррекция} \\
 \quad 0.10000010 \quad +130 \\
 \hline
 \end{array}$$

Рис. 3.5. Пример умножения отрицательного множимого на отрицательный множитель

Процесс умножения, представленный на рис. 3.4 и 3.5, протекает аналогично описанному в п/п 3.5 и 3.5.1, за исключением последнего этапа, где необходимо проводить коррекцию. Для коррекции необходимо вычесть избыточный член, которым является множимое.

В примере, представленном на рис. 3.4, множимое положительное, поэтому при вычитании оно имеет отрицательный знак. Для вычитания его необходимо перевести в дополнительный код. В результате перевода образуется число 1.0011, которое прибавляется к псевдопроизведению. После сложения образуется требуемое произведение.

В примере, показанном на рис. 3.5, исходное множимое отрицательное, поэтому при коррекции при вычитании образуется положительный знак, так как $-(-13)$ по правилам арифметики дает положительный знак. Следовательно, необходимо прибавить число 0.1101. В результате сложения с псевдопроизведением получаем требуемое произведение.

3.5.3. Умножение целых чисел и правильных дробей

Рассмотренные алгоритмы относились к представлению чисел с фиксированной запятой, то есть, как это принято в боль-

Как видно, младший разряд произведения целых чисел, имеющий вес 2^0 , размещается в позиции двойного слова, соответствующей весу 2^1 . Таким образом, для правильного расположения произведения в разрядной сетке двойного слова необходим дополнительный сдвиг вправо. Такой сдвиг можно учесть как в аппаратуре умножителя, так и программным способом.

С другой стороны, при перемножении правильных дробей дополнительный сдвиг не нужен. Это обстоятельство необходимо принимать во внимание при построении умножителя для чисел в форме с плавающей запятой, где участвующие в операции мантиссы представлены в нормализованном виде, то есть правильными дробями.

При умножении правильных дробей часто ограничиваются результатом, имеющим одинарную длину. В этом случае может применяться либо отбрасывание лишних разрядов, либо округление.

3.6. Методы ускорения операции умножения

Условно методы ускорения умножения можно разделить на *аппаратные* и *логические*. Те и другие требуют дополнительных затрат оборудования, которые возрастают с увеличением разрядности сомножителей. Аппаратные способы приводят к усложнению схемы умножителя, но не затрагивают схемы управления. Дополнительные затраты оборудования при реализации логических методов не зависят от разрядности операндов, но схема управления умножителя при этом утяжеляется. На практике ускорение умножения часто достигается комбинацией аппаратных и логических методов.

Логические подходы к убыстрению умножения можно подразделить на две группы:

- 1) методы, позволяющие уменьшить количество сложений в ходе умножения;
- 2) методы, обеспечивающие обработку нескольких разрядов множителя за шаг.

Реализация и тех и других требует введения дополнительных цепей сдвига в регистры. Рассмотрим первую группу логических методов.

3.6.1. Алгоритм Бута

В основе алгоритма Бута [7,9] лежит следующее соотношение, характерное для последовательностей двоичных цифр:

$$2^m + 2^{m-1} + \dots + 2^k = 2^{m+1} - 2^k, \quad (3.4)$$

где m и k — номера крайних разрядов в группе из последовательных единиц. Например, $011110 = 2^5 - 2^1$. Это означает, что при наличии в множителе групп из нескольких единиц (комбинаций вида 011,110), последовательное добавление к СЧП множимого с нарастающим весом (от 2^k до 2^m) можно заменить вычитанием из СЧП множимого с весом 2^k и прибавлением к СЧП множимого с весом 2^{m+1} .

Как видно, алгоритм предполагает три операции: сдвиг, сложение и вычитание. Помимо сокращения числа сложений (вычитаний), у алгоритма есть еще одно достоинство — он в равной степени применим к числам без знака и со знаком.

Алгоритм Бута сводится к перекодированию множителя из системы $(0, 1)$ в избыточную систему $(-1, 0, 1)$, из-за чего его часто называют перекодированием Бута (Booth recoding). В записи множителя в новой системе: 1 означает добавление множимого к сумме частичных произведений, -1 — вычитание множимого и 0 не предполагает никаких действий. Во всех случаях после очередной итерации производится сдвиг множимого влево или суммы частичных произведений вправо. Реализация алгоритма предполагает последовательный в направлении справа налево анализ пар разрядов множителя — текущего b_i и предшествующего b_{i-1} (b, b_{i-1}). Для младшего разряда множителя ($i = 0$) считается, что предшествующий разряд равен 0, то есть имеет место пара b_00 . На каждом шаге i ($i = 0, 1, \dots, n-1$) анализируется текущая комбинация b, b_{i-1} .

Комбинация 10 означает начало цепочки последовательных единиц, и в этом случае производится вычитание множимого из СЧП.

Комбинация 01 соответствует завершению цепочки единиц, и здесь множимое прибавляется к СЧП.

Комбинация 00 свидетельствует об отсутствии цепочки единиц, а 11 — о нахождении внутри такой цепочки. В обоих случаях никакие арифметические операции не производятся.

По завершении описанных действий осуществляется сдвиг множимого влево либо суммы частичных произведений вправо, и цикл повторяется для следующей пары разрядов множителя.

Описанную процедуру рассмотрим на примерах (используется вариант со сдвигом множимого влево). В приведенных примерах операция вычитания, как это принято в реальных умножителях, выполняется путем сложения с множителем, взятым с противоположным знаком и представленным в дополнительном коде. Для удлинения кода до нужного числа разрядов в дополнительные позиции слева заносится значение знакового разряда.

Пример: $0110 \times 0011 = 00010010$ (в десятичном виде $6 \times 3 = 18$). После перекодирования Бута множитель $(0,0,1,1)$ приобретает вид $(0,1,0,-1)$.

Вначале сумма частичных произведений принимается равной нулю: 00000000 . Полагается, что младшему разряду множителя предшествовал 0. Дальнейший процесс поясняет рис. 3.7.

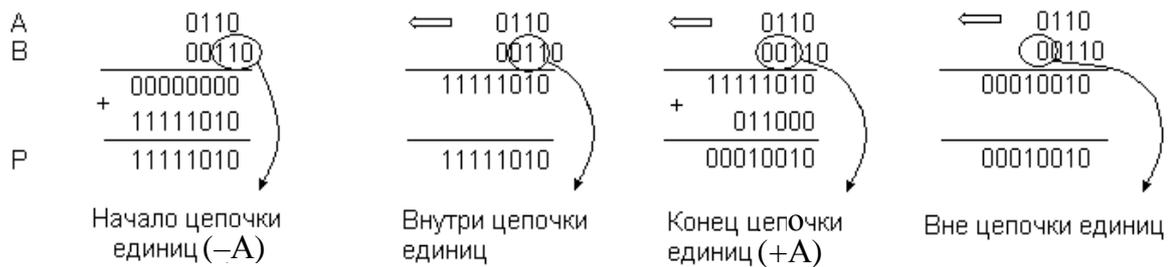


Рис. 3.7. Умножение (6×3) в соответствии с алгоритмом Бута

Пример: В двоичной системе провести умножение чисел $1100 \times 0011 = 11110100$ или в десятичной записи $-4 \times 3 = -12$. Процесс вычисления проиллюстрируем на рис. 3.8.

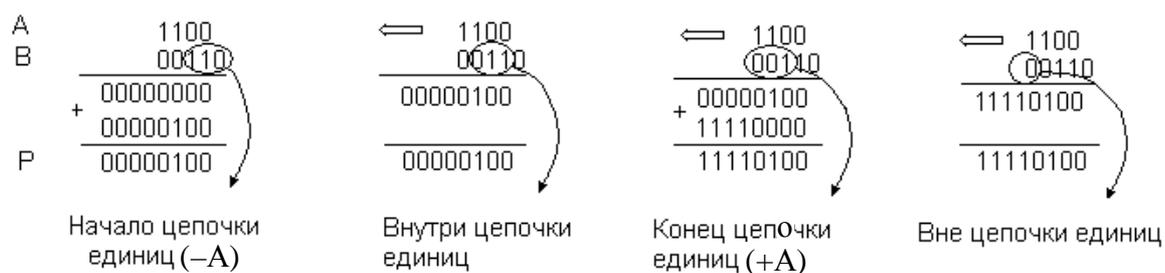


Рис. 3.8. Умножение (-4×3) в соответствии с алгоритмом Бута

Процесс умножения, представленного на рис. рис. 3.7 происходит следующим образом. Анализ разрядов множителя начинается с младшего разряда, первая, встречающаяся комбинация в множителе является 10, поэтому производится вычитание множимого из суммы частных произведений, после этого появляется цепочка единиц, поэтому никаких действий не производится. Следующей появляется комбинация 01, поэтому происходит прибавление множимого к сумме частных произведений. На последнем этапе появляется комбинация 00, поэтому никаких действий не происходит. В результате получается искомое произведение.

Умножение чисел, показанных на рис. 3.8 происходит аналогично умножению, показанному на рис. 3.7. Первоначально появляется комбинация 10 и происходит вычитание множимого из суммы частных произведений, затем следует цепь единиц и не производится никаких действий, после этого цепочка единиц заканчивается и производится прибавление множимого к последнему варианту суммы частных произведений. Последней остается комбинация нулей и поэтому не производится никаких действий. Последний вариант суммы частных произведений есть требуемое произведение. На рисунках представлены конечные результаты после выполнения всех промежуточных вычислений.

При наиболее благоприятном сочетании цифр множителя количество суммирований равно $n/2$, где n — число разрядов множителя.

3.6.2. Модифицированный алгоритм Бута

На практике большее распространение получила модификация алгоритма Бута, где количество операций сложения при любом сочетании единиц и нулей в множителе всегда равно $n/2$. В модифицированном алгоритме производится перекодировка цифр множителя из стандартной двоичной системы (0,1) в избыточную систему (-2,-1,0,1,2), где каждое число представляет собой коэффициент, на который умножается множимое перед добавлением к СЧП. Одновременно анализируются три разряда множителя $b_{i+1}b_i b_{i-1}$ (два текущих и старший разряд из предыдущей тройки) и, в зависимости от комбинации 0 и 1 в этих разрядах, выполняется прибавление или вычитание множимого, прибавление или вычитание удвоенного множимого, либо никакие действия не производятся (табл. 3.2).

Таблица 3.2

Логика модифицированного алгоритма Бута

			Код (-2-1	Выполняемые действия
			0	Не выполнять никаких дей-
			1	Прибавить к СЧП множи-
			1	Прибавить к СЧП множи-
			2	Прибавить к СЧП удвоенное множимое
			-2	Вычесть из СЧП удвоенное множимое
			-1	Вычесть из СЧП удвоенное множимое
			-1	Вычесть из СЧП удвоенное множимое
			0	Не выполнять никаких дей-

Пример вычисления произведения $011001 \times 101110 = 011000111110$ (в десятичном виде $25 \times (-18) = -450$) показан на рис. 3.9.

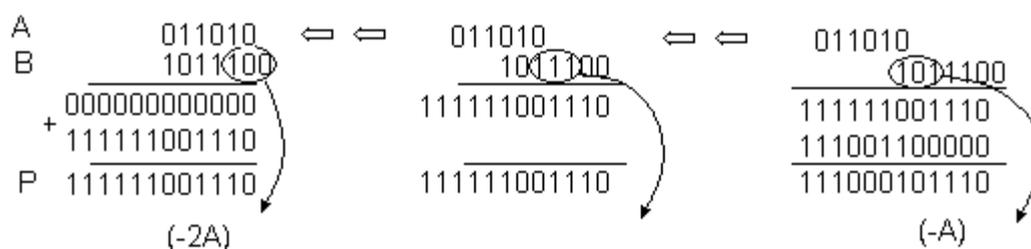


Рис. 3.9. Пример умножения ($18 \times (-25)$) в соответствии с модифицированным алгоритмом Бута

На рисунке 3.9 представлены примеры операций без представления перевода из ПК в ДК (алгоритм требует проведения операций вычитания множимого и необходим перевод в ДК). Первоначально производится вычитание из СЧП удвоенного произведения множимого (разряды множителя равны 100), после этого не происходит никаких действий, так как разряды множителя равны 111 и на последнем шаге снова происходит вычитание из СЧП удвоенного произведения множимого (разряды множителя равны 101). Как видно из примера, ускорение операции умножения достижимо за счет логического анализ разрядов множителя и сокращения количества операций суммирования, что широко используется для сокращения времени операции умножения при проектировании различных вычислительных устройств.

3.6.3. Алгоритм Лемана

Еще большее сокращение количества сложений может дать модификация, предложенная м. Леманом [8]. Здесь, даже при наименее благоприятном сочетании цифр множителя, количество операций сложения не превышает величины $n/2$, а в среднем же оно составляет $n/3$. Суть модификации заключается в следующем:

- если две группы нулей разделены единицей, стоящей в k -й позиции, то вместо вычитания в k -й позиции и сложения в $(k+1)$ -й позиции достаточно выполнить только сложение в k -й позиции;

• если две группы единиц разделены нулем, стоящим в k -й позиции, то вместо сложения в k -й позиции и вычитания в $(k+1)$ -й позиции достаточно выполнить только вычитание в k -й позиции.

Из второй группы логических методов остановимся на умножении с обработкой за шаг двух разрядов множителя (IBM 360/370). Анализ множителя начинается с младших разрядов. В зависимости от входящей двухразрядной комбинации предусматриваются следующие действия:

• 00 – ростей сдвиг на два разряда вправо суммы частичных произведений (СЧП);

• 01 – к СЧП прибавляется одинарное множимое, после чего СЧП сдвигается на 2 разряда вправо;

• 10 – к СЧП прибавляется удвоенное множимое, и СЧП сдвигается на 2 разряда вправо;

• 11 – из СЧП вычитается одинарное множимое, и СЧП сдвигается на 2 разряда вправо. Полученный результат должен быть скорректирован на следующем шаге, что фиксируется в специальном триггере признака коррекции.

Так как в случае пары 11 из СЧП вычитается одинарное множимое вместо прибавления утроенного, то для корректировки результата к СЧП перед выполнением сдвига надо было бы прибавить учетверенное множимое. Но после сдвига на два разряда вправо СЧП уменьшается в четыре раза, так что на следующем шаге достаточно добавить одинарное множимое. Это учитывается при обработке следующей пары разрядов множителя: путем обработки пары 00 как 01, пары 01 – как 10, 10 — как 11, а 11 — как 00. В последних двух случаях фиксируется признак коррекции (табл. 3.3).

Таблица 3.3

Формирование признака коррекции

Па ра раз- рядов	+1 из предыдущей пары	+1 в сле- дующую пару	Зна- к дейст- вия	Крат- ность мно- жимому
00	0	0		0
01	0	0	+	1
10	0	0	+	2

11	0	1	-	1
00	1	0	+	1
01	1	0	+	2
10	1	1	-	1
11	1	1	+	0

После обработки каждой комбинации содержимое регистра множителя и сумматора частичных произведений сдвигается на 2 разряда вправо. Данный метод умножения требует корректировки результата, если старшая пара разрядов множителя равна 11 или 10 и состояние признака коррекции единичное. В этом случае к полученному произведению должно быть добавлено множимое.

3.6.4. Умножение чисел в D -кодах

Подобный вид умножения сводится к последовательному суммированию частных произведений, получаемых при умножении множимого на очередную цифру множителя. При этом умножение сопровождается расшифровкой значения очередной 1-й тетрады множителя и сдвигом множителя на четыре разряда сразу. Самым простым приемом расшифровки тетрады является последовательное вычитание единицы из значения тетрады до получения нуля и, соответственно, прибавление множимого к СЧП на каждом такте. Так как при умножении множимого на тетраду возможно переполнение разрядной сетки СЧП, то в ней необходимо предусмотреть дополнительную тетраду для учета возникающих переносов. Применяется способ умножения младшими разрядами множителя со сдвигом СЧП вправо. Умножение обычно производится в прямом коде.

Пример. Умножить на сумматоре прямого кода (код D_1) числа

$$[A]_{np} = 1,1000 \ 0110;$$

$$[B]_{np} = 1,0011 \ 0011.$$

Решение: При умножении используются сумматор прямого кода для кода D_1 на три тетрады и регистры на две тетрады (табл. 3.4). СЧП на первом шаге алгоритма обнуляется. В приве-

денном примере при анализе младшей тетрады (0011) выполнено три прибавления множимого к СЧП. Затем при появлении в результате вычитания нуля производятся сдвиги СЧП и множителя на 4 разряда вправо.

Таблица 3.4
Пример операции умножения чисел в D-кодах

СЧП	Регистр В	Примечание
0000 0000	0011	Анализ младшей тетрады множителя.
0000 0000	0011	
+ 1000 0110	—	Результат на каждом шаге суммирования СЧП и множимого представлен после проведения коррекции
0000 1000	1	
0000 0110	0010	
+ 1000 0110	—	
0001 0111	1	Конец анализа младшей тетрады. Сдвиг на четыре разряда
0010 0010	0001	
+ 1000 0110	—	Анализ следующей за младшей тетрады множителя
0010 0101	1	
0010 1000	0000	
0000 0010	1000	
0000 0101	0011	
+ 1000 0110	—	
0001 0001	1	Конец анализа младшей тетрады. Сдвиг на четыре разряда
0001 0001	0001	
+ 1000 0110	—	
0001 1001	1	
0001 0111	0000	
+ 1000 0110	0011	
0010 1000	1000	

	0011		
0000	0010		
	1000		

Если сомножители имеют по n десятичных разрядов, то в регистре множимого должно быть n разрядов (тетрад) справа от запятой, а в регистре СЧП должно быть n основных разрядов, одна тетрада переполнений и один $(n+1)$ -й дополнительный десятичный разряд, который предназначен для округления результата. При необходимости сохранить все $2n$ разрядов произведения справа от регистра СЧП должен быть еще сдвиговый регистр для младших разрядов произведения. Регистр множителя должен иметь n тетрад справа от запятой. Причем, если младшая тетрада этого регистра выполнена в виде реверсивного счетчика, (т.е. счетчика, который может прибавлять или вычитать единицу), то операцию умножения можно ускорить в среднем почти в 2 раза. В этом случае необходимо иметь специальный разряд для записи 1, если при суммировании в счетчике–тетраде появится код 10_{10} .

Такой состав оборудования объясняется следующим образом. Если очередная цифра множителя, находящаяся в младшей тетраде регистра множителя, равна или меньше 5_{10} , то производится многократное прибавление множимого к СЧП. При каждом суммировании множимого вычитается 1 из счетчика. Такие действия выполняются до тех пор, пока на счетчике не появится код нуля. Если же очередная цифра множителя равна или больше 6_{10} , то производится многократное вычитание (сложение в дополнительном D -коде) множимого из СЧП. При каждом вычитании множимого к содержимому счетчика прибавляется 1. Вычитания продолжаются до тех пор, пока в счетчике не появится код 10_{10} , при этом в специальный разряд записывается 1.

Вторая тетрада множителя при этом приписывается на место младшей тетрады. Затем производится анализ второй тетрады множителя путем вычитания единицы по алгоритму приведенному выше. После сдвигов СЧП и множителя вправо на 4 разряда на этапе анализа второй тетрады для приведенного примера нет больше тетрад множителя, которые необходимо анали-

зировать. В результате использования алгоритма умножения получен результат равный 0, 0000 0010 1000 0011 1000 (+2838₁₀). Знак результата определяется с использованием логической функции «сумма по модулю 2».

При умножении в *D*-кодах также можно использовать еще один способ ускорения операции умножения, основанный на следующих формулах:

$$AB = 2A \frac{B-1}{2} + A, \quad \text{если } B \text{ – нечетное число;}$$

$$AB = 2A \frac{B}{2}, \quad \text{если } B \text{ – четное число.}$$

При двоично-десятичном представлении чисел удвоение числа означает сдвиг влево, а деление на 2 – сдвиг вправо, причем так как сдвиги производятся над двоичными кодами, требуется коррекция тетрад на каждом шаге. Корректирующие поправки определяются для каждого *D*-кода. Коррекция выполняется тогда, когда происходит сдвиг единицы из крайнего разряда данной тетрады в соседнюю тетраду. Для кода *D*₁ корректирующая поправка равна +0110 для тетрад множимого и –0011 (или +1101) для тетрад множителя.

Пример. Умножить ускоренным способом (1) на сумматоре прямого кода числа (табл. 3.5):

$$[A]_{np} = 1,1000 \ 0110;$$

$$[B]_{np} = 1,0011 \ 0011.$$

На примере, представленном в табл. 3.5 приведена операция умножения числа -86 на -33, записанных потетрадно в двоичном коде. Здесь в колонке *B* показан множитель, то есть число -33, в колонке *A* записано множимое, то есть число -86, колонка *C* иллюстрирует процесс накопления произведения в СЧП. В примере показаны числа после требуемых коррекций. На первом шаге множитель отрицательный (33), следовательно, его необходимо добавить к СЧП. После этого производится сдвиг множителя на разряд вправо, а множимого – на разряд влево, при этом в множителе происходит межтетрадный перенос, в младшей тетраде множимого появляется запрещенная комбинация (число

1100), а во второй – произошел межтетрадный перенос. Следовательно разряды A и B требуют коррекции, младшую тетраду B следует скорректировать на $+13$ с блокировкой межтетрадного переноса, а тетрады A на $+0110$. После коррекции число B положительно, поэтому добавления множимого к C не происходит, а производится очередная операция сдвигов множимого и множителя. На каждом шаге производится анализ необходимости коррекции в тетрадах множимого и множителя, и добавления скорректированного множимого к C . В C в случае необходимости необходимо производить коррекции на $+0110$. Операция сдвигов прекращается, когда тетрада множителя станет равной 0001 , при этом происходит последнее добавления A к C (значение 0001 является нечетным).

Таблица 3.5

Пример ускоренной операции умножения

B	A	C	Примечание
-----	-----	-----	------------

0011 0011	0000 1000 0110	0000 0000 1000 0110	<i>B</i> – нечетное
0001 1001	0001 0000 1100		Сдвиг
+	+		
1101	0110 0110		Поправки
0001 0110	0001 0111 0010		<i>B</i> – четное
0000 1011	0010 1110 0100		Сдвиг
+	+		
1101	0110 .		Поправки
0000 1000	0011 0100 0100		<i>B</i> – четное
0000 0100	0110 1000 1000		Сдвиг, <i>B</i> – четное
0000 0010	1101 0001 0000		Сдвиг
	+		
	0110 0110 0110		Поправки
	0001 0011 0111 0110		<i>B</i> – четное
0000 0001	0010 0110 1110 1100		Сдвиг
	+		
	0110 0110		Поправки
	0010 0111 0101 0010	0010 0111 0101 0010	<i>B</i> – нечетное
		0010 0111 1101 1000	
		+	
		0110	Поправка
		0010 1000 0011 1000	

3.6.5. Аппаратные методы ускорения умножения

Традиционный метод умножения за счет сдвигов и сложений, даже при его аппаратной реализации, не позволяет достичь высокой скорости выполнения операции умножения. Связано это, главным образом, с тем, что при добавлении к СЧП очередного частичного произведения перенос должен распространиться от младшего разряда СЧП к старшему. Задержка из-за распространения переноса относительно велика, причем она повторяется при добавлении каждого ЧП.

Один из способов ускорения умножения состоит в изменении системы кодирования сомножителей, за счет чего можно сократить количество суммируемых частичных произведений. Примером такого подхода может служить алгоритм Бута.

Еще один ресурс повышения производительности умножителя – использование более эффективных способов суммирования ЧП, исключающих затраты времени на распространение переносов. Достигается это за счет представления ЧП в избыточной форме, благодаря чему суммирование двух чисел не связано с распространением переноса вдоль всех разрядов числа. Наиболее употребительной формой такого избыточного кодирования является так называемая форма *с сохранением переноса*. В ней каждый разряд числа представляется двумя битами cs , известными как перенос (c) и сумма (s). При суммировании двух чисел в форме s с сохранением переноса перенос распространяется не далее, чем на один разряд. Это делает процесс суммирования значительно более быстрым, чем в случае сложения с распространением переноса вдоль всех разрядов числа.

Наконец, третья возможность ускорения операции умножения заключается в параллельном вычислении всех частичных произведений. Если рассмотреть общую схему умножения (рис. 3.10), то нетрудно заметить, что отдельные разряды ЧП представляют собой произведения вида $a_i b_j$, то есть произведение определенного бита множимого на определенный бит множителя. Это позволяет вычислить все биты частичных произведений одновременно, с помощью n^2 схем И. При перемножении чисел в дополнительном коде отдельные разряды ЧП могут иметь вид $\overline{a_i b_j}$, $\overline{a_i} b_j$ или $\overline{\overline{a_i b_j}}$. Тогда элементы И заменяются элементами, реализующими соответствующую логическую функцию.

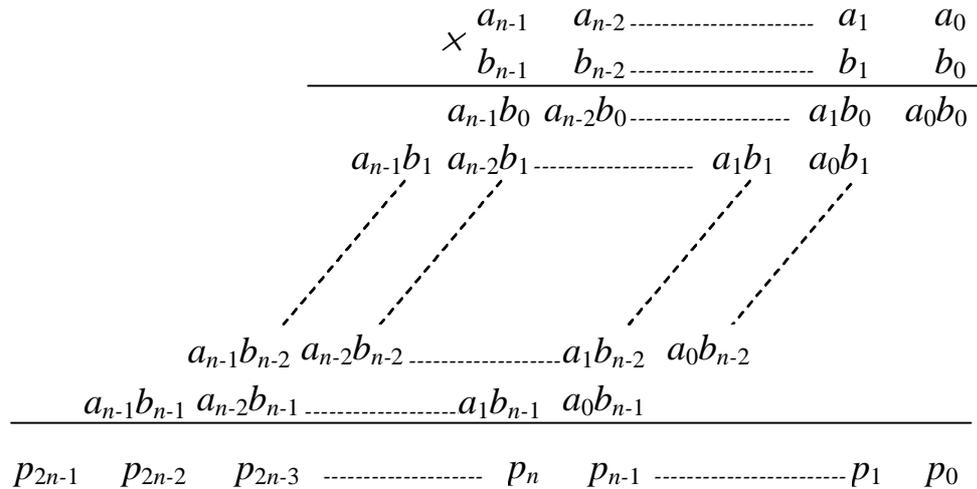


Рис. 3.10. Схема перемножения n -разрядных чисел без знака

Таким образом, аппаратные методы ускорения умножения сводятся:

- к параллельному вычислению частичных произведений;
- к сокращению количества операций сложения;
- к уменьшению времени распространения переносов при суммировании частичных произведений.

Все три подхода в любом их сочетании обычно реализуются с помощью комбинационных устройств.

Параллельное вычисление ЧП имеет место практически во всех рассматриваемых ниже схемах умножения. Различия проявляются в основном в способе суммирования полученных частичных произведений, и с этих позиций используемые схемы умножения можно подразделить на *матричные* и *с древовидной структурой* [9]. В обоих вариантах суммирование осуществляется с помощью массива взаимосвязанных одноразрядных сумматоров. В матричных умножителях сумматоры организованы в виде матрицы, а в древовидных они реализуются в виде дерева того или иного типа.

Различия в рамках каждой из этих групп выражаются в количестве используемых сумматоров, их виде и способе распространения переносов, возникающих в процессе суммирования.

В *матричных умножителях* суммирование осуществляется матрицей сумматоров, состоящей из последовательных линеек

(строк) одноразрядных сумматоров с сохранением переноса (ССП). По мере движения данных вниз по массиву сумматоров каждая строка СПП добавляет к СЧП очередное частичное произведение. Поскольку промежуточные СЧП представлены в избыточной форме с сохранением переноса, во всех схемах, вплоть до последней строки, где формируется окончательный результат, распространения переноса не происходит. Это означает, что задержка в умножителях отталкивается только от «глубины» массива (числа строк сумматоров) и не зависит от разрядности операндов, если только в последней строке матрицы, где формируется окончательная СЧП, не используется схема с последовательным переносом.

Наряду с высоким быстродействием важным достоинством матричных умножителей является их регулярность, что особенно существенно при реализации таких умножителей в виде интегральной микросхемы. С другой стороны, подобные схемы занимают большую площадь на кристалле микросхемы, причем с увеличением разрядности сомножителей эта площадь увеличивается пропорционально квадрату числа разрядов. Вторая проблема с матричными умножителями – это низкий уровень утилизации аппаратуры. По мере движения СЧП вниз каждая строка задействуется лишь однократно, когда ее пересекает активный фронт вычислений. Это обстоятельство, однако, может быть затребовано для повышения эффективности вычислений путем конвейеризации процесса умножения, при которой по мере освобождения строки сумматоров последняя может быть использована для умножения очередной пары чисел.

3.7. Выполнение операции деления

Деление несколько более сложная операция, чем умножение, но базируется на тех же принципах. Основу составляет общепринятый способ деления с помощью операций вычитания или сложения и сдвига (рис. 3.11).

$$\begin{array}{cccccccc|cccc}
 z\text{-делимое} & z_7 & z_6 & z_5 & z_4 & z_3 & z_2 & z_1 & z_0 & d_3 & d_2 & d_1 & d_0 & D\text{-делимое} \\
 -Dq_3 \times 2^3 & & & & & r_3 & r_2 & r_1 & r_0 & \hline
 -Dq_3 \times 2^2 & & & & & r_3 & r_2 & r_1 & r_0 & q_3 & q_2 & q_1 & q_0 & Q\text{-частное} \\
 -Dq_3 \times 2^1 & & & & & r_3 & r_2 & r_1 & r_0 & & & & & \\
 -Dq_3 \times 2^0 & & & & & & & & & & & & & \\
 & & & & & \hline
 & & & & & s_3 & s_2 & s_1 & s_0 & & & & & S\text{-остаток}
 \end{array}$$

Рис. 3.11. Общая схема операции деления

Задача сводится к вычислению частного Q и остатка S :

$$Q = \text{int}\left(\frac{Z}{D}\right), S = Z - QD, S < D. \tag{3.5}$$

Деление выражается как последовательность вычитаний делителя сначала из делимого, а затем из образующихся в процессе деления частичных остатков (ЧО). Делимое $Z(z_{2n-1}z_{2n-2} \dots z_1z_0)$ обычно представляется двойным словом ($2n$ разрядов), делитель $D(d_{2n-1}d_{2n-2} \dots d_1d_0)$, частное $Q(q_{2n-1}q_{2n-2} \dots q_1q_0)$ и остаток $S(s_{2n-1}s_{2n-2} \dots s_1s_0)$ имеют разрядность n .

Операция выполняется за n итераций и может быть описана следующим образом:

$$S^{(i)} = 2S^{(i-1)} - q_{n-i}(2^n D), \text{ при } S^{(0)} = Z \text{ и } S^{(n)} = 2^n S; \tag{3.6}$$

$$q_{n-i} = \begin{cases} 1, & \text{если } (2S^{(i-1)} - 2^n D) \geq 0, \\ 0, & \text{если } (2S^{(i-1)} - 2^n D) < 0. \end{cases} \tag{3.7}$$

После n итераций получается

$$S^{(n)} = 2^n S^{(0)} - Q(2^n D) - 2^n [Z - (Q \times D)] = 2^n S. \tag{3.8}$$

Частное от деления $2n$ -разрядного числа на n -разрядное может содержать более, чем n разрядов. В этом случае возникает переполнение, из-за чего перед выполнением деления необходима проверка условия

$$Z < (2^n - 1)D + D = 2^n D. \tag{3.9}$$

Из выражения следует, что переполнения не будет, если число, содержащееся в старших n разрядах делимого, меньше делителя.

Помимо этого требования, перед началом операции необходимо исключить возможность ситуации деления на 0.

Реализовать деление можно двумя основными способами:

1) с неподвижным делимым и сдвигаемым вправо делителем;

2) с неподвижным делителем и сдвигаемым влево делимым.

Недостатком первого способа является потребность иметь в устройстве деления сумматор и регистр двойной длины. Вторым способом позволяет строить делитель с сумматором одинарной длины. Неподвижный делитель D хранится в регистре одинарной длины, а делимое Z , сдвигаемое относительно D , находится в двух таких же регистрах. Образующиеся цифры частного Q заносятся в освобождающиеся при сдвиге Z разряды одного из регистров Z .

Ниже на примере чисел без знака рассматриваются два основных алгоритма целочисленного деления.

Наиболее очевидный алгоритм носит название алгоритма *деления с неподвижным делителем и восстановлением остатка*. В учебном пособии он представлен в силу того, что очень похож на общепринятый способ деления столбиком. Данный алгоритм может быть описан следующим образом:

1) исходное значение частичного остатка полагается равным старшим разрядам делимого;

2) частичный остаток удваивается путем сдвига на один разряд влево. При этом в освобождающийся при сдвиге младший разряд ЧО заносится очередная цифра частного.

3) из сдвинутого ЧО вычитается делитель и анализируется знак результата вычитания;

4) очередная цифра модуля частного равна единице, когда результат вычитания положителен, и нулю, если отрицателен. В последнем случае значение остатка восстанавливается до того значения, которое было до вычитания;

5) пункты 2-4 последовательно выполняются для получения всех цифр модуля частного.

На рисунке 3.12 показан процесс деления с восстановлением остатка (здесь число 41 делится на 7).

	Делимое		Делитель 7_{10}	
Исходное значение ЧО	0 0 1 0 1 0 0 1		0 1 1 1	Частное 5_{10}
Сдвиг ЧО влево	0 0 1 0 1 0 0 1		0 1 0 1	
Вычитание делителя	- 0 1 1 1			
Результат < 0	1 1 1 1 0 0 0 1 0			
Восстановление ЧО	+ 0 1 1 1			
Восстановленный ЧО	0 0 1 0 1 0 0 1			
Сдвиг ЧО влево	0 0 1 0 1 0 0 1			
Вычитание делителя	- 0 1 1 1			
Результат > 0	0 0 0 0 1 1 0 0			
Сдвиг ЧО влево	0 0 0 0 1 1 0 0			
Вычитание делителя	- 0 1 1 1			
Результат < 0	1 1 1 1 1 0 0 0			
Восстановленный ЧО	+ 0 1 1 1			
Восстановленный ЧО	0 0 0 0 1 1 0 0			
Сдвиг ЧО влево	0 0 0 0 1 1 0 0			
Вычитание делителя	- 0 1 1 1			
Результат > 0	0 0 0 0 1 1 0 0			
	0 1 1 0			Остаток 6_{10}

Рис. 3.12 Пример деления с восстановлением остатка

На первом шаге алгоритма деления производится операция сдвига делимого на один разряд влево (рис. 3.12), при этом в освобождающийся разряд записывается ноль. После этого происходит вычитания из сдвинутого делимого числа разрядов делителя. Затем анализируется старший разряд частичного остатка. Если он равен 1, то есть число меньше нуля, то в разряд частного записывается 0, если – 0 (число больше нуля), то в частное записываем 1. В нашем случае он равен 1, поэтому в частное записываем ноль и производим восстановление остатка с помощью добавления делителя. Результат суммирования сдвигаем на разряд влево. На этом первый шаг деления закончен и для получения очередной цифры частного необходимо снова вычесть разряды делителя. В примере на рис. 3.12 результат вычитания больше нуля (старший разряд равен 0), поэтому в частное записываем 1. операцию необходимо продолжить до получения количества

разрядов, равных количеству разрядов делителя. В итоге получаем значение частного равное $0101 (5_{10})$ и в остатке 0110 .

Недостаток алгоритма деления с восстановлением остатков заключается в необходимости выполнения на отдельных шагах дополнительных операций сложения для восстановления частичного остатка. Это увеличивает время выполнения деления, которое в этом случае может меняться в зависимости от конкретного сочетания кодов операндов. В силу указанных причин реальные делители строятся на основе алгоритма *деления с неподвижным делителем без восстановления остатка*. Приведем описание этого алгоритма:

1) исходное значение частичного остатка полагается равным старшим разрядам делимого;

2) частичный остаток удваивается путем сдвига на один разряд влево, при этом в освобождающийся при сдвиге младший разряд ЧО заносится очередная цифра частного;

3) из сдвинутого частичного остатка вычитается делитель, если остаток положителен, и к сдвинутому частичному остатку прибавляется делитель, если остаток отрицательный;

4) очередная цифра модуля частного равна единице, когда результат вычитания положителен, и нулю, если он отрицателен;

5) пп. 2-4 последовательно выполняются для получения всех цифр модуля частного.

Как видим, пп. 1, 2, 5 полностью совпадают с соответствующими пунктами предыдущего алгоритма деления.

Процесс деления без восстановления остатка для ранее рассмотренного примера демонстрируется на рис. 3.13.

Алгоритм деления без восстановления остатка аналогичен алгоритму деления с восстановлением остатков. Отличие состоит в том, что после появления отрицательного числа после вычитания производится сдвиг на единицу влево, после чего к сдвинутому остатку добавляется делитель. Полученный частичный остаток анализируем на значение больше нуля или меньше нуля, и в зависимости от этого аналогично предыдущему алгоритму записываем единицу или ноль в разряд частного. В остальном, процесс деления полностью аналогичен алгоритму деления с восстановлением остатков.

	Делимое 41_{10}	0 0 1 0	1 0 0 1	0 1 1 1	Делитель 7_{10}
Исходное значение ЧО		0 0 1 0	1 0 0 1	0 1 0 1	Частное 5_{10}
Сдвиг ЧО влево	0	0 1 0 1	0 0 1 0		
Вычитание делителя	-	0 1 1 1			
Результат < 0		1 1 1 1 0	0 0 1 0		
Сдвиг ЧО влево		1 1 1 1 0 0	0 1 0		
Прибавление делителя	+	0 1 1 1			
Результат > 0		0 0 0 0 1 1	0 1 0 0		
Сдвиг ЧО влево		0 0 0 0 1 1 0	1 0 0 0		
Вычитание делителя	-	0 1 1 1			
Результат < 0		1 1 1 1 1 1 1	1 0 0 0		
Сдвиг ЧО влево		1 1 1 1 1 1 1 0	0 0 0 0		
Прибавление делителя	+	0 1 1 1			
		0 0 0 0 0 1 1 0	0 1 0 0		
		0 1 1 0			Остаток 6_{10}

Рис. 3.13. Пример деления без восстановления остатка

3.7.1. Деление меньшего числа на большее с восстановлением остатков

Данный вид операции деления аналогично описанному выше проще всего выполнять в прямом коде. Знак частного при делении определяется как сумма по mod2 знаковых цифр делимого и делителя, и присваивается частному в конце операции деления.

Частное определяется путем деления модулей исходных чисел. При этом во избежание переполнения разрядной сетки должно выполняться условие $|A| < |B|$.

Цифры частного определяются последовательно начиная со старшего разряда. Допустим, что в результате выполнения i циклов получены старшие i разрядов частного Y_i . В следующем $(i+1)$ -м цикле будет получено значение $(i+1)$ -го разряда частного. Исходными данными для этого цикла являются текущее значение частного Y_i и значение текущего остатка R_i .

Цифра y_{i+1} может иметь одно из двух значений: 1 или 0.

Если $y_{i+1} = 0$, то

$$Y_{i+1} = Y_i + y_{i+1} 2^{-(i+1)}$$

$$R_{i+1} = (A - BY_{i+1}) 2^{i+1} = 2R_i,$$

т.е. в частном записывается 0 при условии

$$0 \leq R_{i+1} = 2R_i < B$$

Если $y_{i+1} = 1$, то

$$Y_{i+1} = Y_i + 2^{-(i+1)}$$

$$R_{i+1} = (A - BY_{i+1}) 2^{i+1} = 2R_i - B,$$

т.е. цифра частного равна 1, если выполняется условие $0 \leq R_{i+1} = 2R_i - B < B$ или $B \leq 2R_i < 2B$.

Т.к. всегда выполняется одно из приведенных условий, то для определения текущей цифры частного достаточно проверить одно из них. Обычно проверяют первое. Очередной остаток перед началом следующего шага всегда является положительным. Для проверки правой части неравенства сравним разность $(2R_i - B)$ с нулем. Если эта разность окажется отрицательной, то в $(i+1)$ разряд частного записываем 0 и для подготовки исходных данных для $(i+2)$ цикла определим R_{i+1} следующим образом:

$$R_{i+1} = (2R_i - B) + B = 2R_i$$

Если разность $(2R_i - B)$ окажется положительной, то запишем в $i+1$ разряд частного 1, а в качестве исходного значения для $(i+2)$ -го цикла используем вычисленную разность $R_{i+1} = (2R_i - B)$.

Исходными данными явл. $Y_0 = 0$

$$R_0 = (A - BY_0) 2^0 = A < B$$

Правило деления с восстановлением остатков формулируется следующим образом.

Делитель вычитается из делимого и определяется знак нулевого остатка. Если остаток положительный, то в псевдознаковом разряде частного проставляется 1, при появлении которой формируется признак переполнения разрядной сетки и операция деления прекращается. Если остаток отрицательный, то в псевдознаковом разряде частного записывается 0, а затем производится восстановление делимого путем добавления к остатку делителя. Далее производится сдвиг восстановленного делимого на

один разряд влево и повторное вычитание делителя. Знак полученного таким образом остатка определяет первую значащую цифру частного. Если остаток положительный, то в первом разряде частного записывается 1, если отрицательный – 0. Далее если остаток положительный он сдвигается влево на 1 разряд и из него вычитается делитель для определения следующей цифры частного.

Если остаток отрицательный, то к нему прибавляется делитель для восстановления предыдущего остатка, затем восстановленный остаток сдвигается на 1 разряд влево и от него вычитается делитель для определения следующей цифры частного и т.д. до получения необходимого количества разрядов частного с учетом одного разряда для округления.

Рассмотрим пример операции деления с восстановлением остатков числа 5 на число 6 (рис. 3.14) с точностью до 5 значащих разрядов. То сеть необходимо $5:6=0.8(3)$ или в двоичном виде $0.0101:0.0110$.

0.0101	делимое
- 0.0110	делитель
1.0001	т.к. остаток <0, в знаковый разряд пишем 0
+ 0.0110	восстановление остатка
0.0101	восстановленный остаток
- 0.1010	сдвиг влево на один разряд
0.0110	вычитаем делитель
0.0100	т.к. остаток >0, в частное пишем 1
0.1000	сдвиг влево на один разряд
- 0.0110	вычитаем делитель
0.0010	т.к. остаток >0, в частное пишем 1
0.0100	сдвиг влево на один разряд
- 0.0110	вычитаем делитель
1.0010	т.к. остаток <0, в знаковый разряд пишем 0
+ 0.0110	восстановление остатка
0.0100	восстановленный остаток
0.1000	сдвиг влево на один разряд
- 0.0110	вычитаем делитель
0.0010	т.к. остаток >0, в частное пишем 1
0.0100	сдвиг влево на один разряд
- 0.0110	вычитаем делитель
1.0010	т.к. остаток <0, в знаковый разряд пишем 0

Рис. 3.14 Пример деления с восстановлением остатков
меньшего числа на большее

В примере на рис. 3.14 не показаны промежуточные операции перевода отрицательных чисел в обратные либо дополнительные коды, так как это было показано в п. 3.2 настоящего пособия.

В результате операции деления числа 5 на число 6 получаем результат 0.11010. Данный результат является приближенным, а для перевода данного кода в десятичный эквивалент необходимо умножить каждый разряд числа на соответствующую степень двойки с учетом того, число является дробным. Например, старший разряд числа следует умножить на 2^{-1} , следующий разряд – на 2^{-2} и т.д. до конца двоичного числа. В результате получаем число 0,8125. Т.е. результат вычислений правильный с учетом того, что деление выполнялось с точностью до 5 значащих разрядов.

3.7.2. Деление меньшего числа на большее без восстановления остатков

Рассмотренный в 3.10 процесс деления с восстановлением остатков является аритмичным процессом с переменным числом шагов.

Операцию можно упростить и получить каждую цифру за два шага.

Если в предыдущем пункте остаток $R_i < 0$, то в п. 3.10 необходимо было бы выполнить следующие операции:

1) восстановление остатка

$$R'_i = R_i + |B| = |2R_{i-1} - B| + |B| = 2R_{i-1}$$

2) сдвиг восстановленного остатка влево

$$\overline{R}'_i = 2R'_i = 2R_{i-1} \cdot 2 = 4R_{i-1}$$

3) вычитание модуля делителя из восстановленного и сдвинутого влево остатка для определения следующего остатка

$$R_{i+1} = 4R_{i-1} - |B|$$

Если не восстанавливать остаток, а сразу сдвинуть отрицательный R_i на один разряд влево, то получим

$$R_{i+1} = 2R_i = 2(2R_{i-1} - |B|) = 4R_{i-1} - 2|B|$$

Результат в данном случае отличается от действительного на величину $|+B|$. Поэтому в качестве второго шага необходимо произвести коррекцию результата на эту величину:

$$R_{i+1} = 4R_{i+1} - 2|B| + |B| = 4R_{i-1} - |B|$$

т.е. в результате получаем требуемую величину последующего остатка за 2 шага.

Делитель вычитается из делимого и определяется знак нулевого остатка. Если остаток положительный, то в псевдознаковом разряде частного проставляется 1, при появлении которой формируется признак переполнения разрядной сетки и операция деления прекращается. Если остаток отрицательный, то в разряд частного записывается 1 и далее происходит сдвиг текущего остатка влево на 1 разряд, а затем необходимо алгебраически прибавить к нему модуль делителя, которому приписывается знак, противоположный знаку текущего остатка. Знак, полученного таким образом остатка и определяет следующую цифру частно-

го: если он положителен, то в частном записывается 1, если отрицателен, то 0. Операция сдвигов и алгебраических сложений повторяется до тех пор, пока в частном не получится требуемое количество цифр.

Рассмотрим пример деления двух чисел таких же как и в п. 3.10, то есть разделим 5 на 6. аналогично частное равно 0.8(3) или в двоичном виде 0.0101:0.0110 (рис. 3.15).

0.0101	делимое
0.0110	делитель
1.0001	т.к. остаток <0, в знаковый разряд пишем 0
1.0010	сдвиг влево на один разряд
+0.0110	прибавление модуля делителя
0.0100	т.к. остаток >0, в частное пишем 1
0.1000	сдвиг влево на один разряд
0.0110	вычитаем делитель
0.0010	т.к. остаток >0, в частное пишем 1
0.0100	сдвиг влево на один разряд
0.0110	вычитаем делитель
1.0010	т.к. остаток <0, в знаковый разряд пишем 0
1.0100	сдвиг влево на один разряд
+0.0110	прибавление модуля делителя
0.0010	т.к. остаток >0, в частное пишем 1
0.0100	сдвиг влево на один разряд
0.0110	вычитаем делитель
1.0010	т.к. остаток <0, в знаковый разряд пишем 0

Рис. 3.15 Деления без восстановления остатка меньшего числа на большее

В результате операции деления получили результат аналогичный полученному в п. 3.10 – 0.11010. Аналогично в приведенном примере на рис. 3.15 не оказан перевод отрицательных чисел в дополнительный и обратный код.

3.7.3. Деление чисел со знаком

Как и в случае умножения, деление чисел со знаком может быть выполнено путем перехода к абсолютным значениям делимого и делителя с последующим присвоением частному знака

«плюс» при совпадающих знаках делимого и делителя либо «минус» – в противном случае.

Деление чисел, представленных в дополнительном коде, можно осуществлять, не переходя к модулям. Рассмотрим необходимые для этого изменения в алгоритме без восстановления остатка.

Так как делимое и делитель не обязательно имеют одинаковые знаки, то действия с частичным остатком (прибавление или вычитание делителя) зависят от знаков остатка и делителя (табл. 3.4).

- если знак остатка совпадает со знаком делителя, то очередная цифра частного – 1, иначе – 0;
- если $Z > 0$ и $D < 0$, частное необходимо увеличить на 1;
- если $Z < 0$ и $D > 0$, то при ненулевом остатке от деления частное нужно увеличить на единицу;
- если $Z < 0$ и $D < 0$, то при нулевом остатке от деления частное нужно увеличить на единицу.

Таблица 3.4.

Операция, выполняемая в очередной итерации деления

Знак	Знак де-	Действие
+	+	Вычитание дели-
+	-	Прибавление де-
-	+	Прибавление де-
-	-	Вычитание дели-

Остаток всегда приводится к положительному числу, то есть, если по завершении деления он отрицателен, к нему следует прибавить модуль делителя.

3.7.4. Деление в избыточных системах счисления

Наиболее распространенные методы ускорения операции деления основаны на применении алгоритмов, где частное представляется в системе счисления, отличной от двоичной. Это означает, что цифры частного могут иметь больше, чем два значения, например $\{-1,0,1\}$, как это было в алгоритме умножения Бута, или $\{-2,-1,0,1,2\}$. В таких системах одно и то же число может быть записано несколькими способами, из-за чего системы на-

зывают избыточными. Очередная цифра частного в избыточной системе счисления, в зависимости от базы этой системы, соответствует двум или более цифрам в двоичном представлении частного, и для нужного количества двоичных цифр частного и остатка требуется меньше итераций. В то же время реализация такого подхода ведет к усложнению аппаратуры делителя, в частности, надстраивается логика определения операции, выполняемой в очередной итерации. Для этой цели в состав устройства деления включается специальная память, хранящая таблицу, определяющую необходимые действия в зависимости от текущей комбинации цифр в частичном остатке и делителе. Тем не менее выигрыш в быстродействии оказывается решающим моментом. Так, в микропроцессорах Pentium при делении мантисс чисел с плавающей запятой используется алгоритм SRT с базой 4, то есть частное сначала вычисляется с использованием цифр -2, -1, 0, 1, 2 с последующим преобразованием результата к стандартному двоичному представлению. В этом варианте выбор очередной цифры частного производится с помощью таблицы, состоящей из отдельных секций. Конкретную секцию определяют четыре старшие цифры делителя (после его нормализации). Входом в секцию служат шесть старших цифр частичного остатка. ЧО в каждой итерации сдвигается не на один, а на два разряда, то есть число итераций сокращается вдвое. Известны варианты делителей, где берется еще большее основание системы счисления, в частности 8 и 16. В этом случае логика работы устройства существенно усложняется.

3.7.5. Замена деления умножением на обратную величину

Операцию деления на D можно заменить умножением на

$$\frac{1}{D} : Q = \frac{Z}{D} = Z \times \frac{1}{D}, \quad (3.10)$$

где D – это делитель

В этом случае проблема сводится к эффективному вычислению $1/D$. Обычно задача решается одним из двух методов: с помощью ряда Тейлора или метода Ньютона–Рафсона [9]. В обоих случаях основное время расходуется на умножение, по-

этому рассматриваемый метод ускорения деления имеет смысл при наличии быстрых схем умножения.

При реализации первого метода делитель D представляется в виде: $D = 1 + X$. Тогда для двоичного представления D можно записать:

$$\frac{1}{D} = (1 - X) \times (1 + X^2) \times (1 + X^4) \times (1 + X^8) \times (1 + X^{16}) \dots$$

Метод был использован в модели 91 вычислительной машины IBM 360 для вычисления 32-разрядной величины $1/D$. Возможные значения сомножителей в правой части выражения извлекались из таблицы емкостью 28 байт, хранящейся в памяти. Операция вычисления $1/D$ требует шести умножений.

Вычисление величины $1/D$ методом Ньютона–Рафсона сводится к нахождению корня уравнения

$$f(X) = \frac{1}{X} - D = 0, \quad (3.11)$$

то есть $X = 1/D$. Решение может быть получено с привлечением рекуррентного соотношения $X_{i+t} = X_i (2 - X_i D)$. Количество итераций определяется требуемой точностью вычисления $1/D$. Реализация метода для n -разрядных чисел требует $2 \cdot \text{int}(\log_2 n) - 1$ операций умножения.

В общем, замена операции деления на умножение более характерна для чисел с плавающей запятой.

3.7.6. Ускорение операции деления

В основе методов ускорения операции деления лежит так называемый алгоритм SRT [11,12]. Свое название алгоритм получил по фамилиям авторов (Sweeney, Robertson, Tocher), разработавших его независимо друг от друга приблизительно в одно и то же время. Этот алгоритм представляет собой модификацию деления без восстановления остатка. В стандартной процедуре на каждом шаге помимо сдвига частичного остатка производится прибавление либо вычитание делителя. В SRT-алгоритме сдвиг частичных остатков (ЧО) также имеется в каждой итерации, однако сложение или вычитание, в зависимости от получающегося ЧО, на отдельных шагах может не выполняться, что, естественно, позитивно влияет на быстродействие деления.

Алгоритм был ориентирован на операции над мантиссами чисел с плавающей запятой и опирается на то обстоятельство, что мантиссы в таких числах нормализованы. Впервые SRT-алгоритм был реализован в модели 91 вычислительной машины IBM 360. В настоящее время он широко применяется в блоках обработки чисел с плавающей запятой, в частности в микропроцессорах фирмы Intel.

Сначала рассмотрим алгоритм применительно к положительным целым числам. Делимое представляется $(2n+1)$ -разрядным числом, а делитель – n -разрядным. Процедура деления начинается с удаления в делителе всех нулей, предшествующих старшей единице, то есть с операции, аналогичной нормализации мантиссы в числах с плавающей запятой. Будем условно называть эту операцию нормализацией. Исключение k предшествующих нулей реализуется за счет сдвига делителя влево на k разрядов. На аналогичное число разрядов влево сдвигается и делимое. Далее выполняются n итераций, в которых вычисляются цифры частного и частичные остатки. Действия, выполняемые на i -й итерации, можно описать следующим образом:

$$q_i = \begin{cases} 1, & \text{если } 2S^{(i-1)} \geq D; \\ 0, & \text{если } -D \leq 2S^{(i-1)} < D; \\ -1, & \text{если } 2S^{(i-1)} < -D, \end{cases} \quad (3.12)$$

$$S^{(i)} = 2S^{(i-1)} - q_i D.$$

Частное представляется в системе счисления, отличной от двоичной. Это означает, что цифры частного могут иметь больше чем два значения 0 и 1. В рассматриваемом случае 1,0,1.

По завершении всех n итераций, если последний остаток отрицателен, выполняется коррекция этого остатка и полученного частного, для чего к остатку прибавляется делитель, а из частного вычитается единица с весом младшего разряда.

Последний этап в алгоритме – преобразование частного из системы $\{-1, 0, 1\}$ в систему $\{0, 1\}$, то есть в обычную двоичную систему.

На практике используется следующая процедура. Делимое и делитель, представленные в дополнительном коде, размещают-

ся в регистре делимого (РДМ) и делителя (РДТ) соответственно. Дальнейшие действия можно описать следующим образом:

1) если в делителе D имеются k предшествующих нулей (при $D > 0$) или предшествующих единиц (при $D < 0$), то производится предварительный сдвиг содержимого РДМ и РДТ влево на k разрядов;

2) для i от 0 до $n-1$:

- если три старших цифры частичного остатка в РДМ совпадают, то $q_i = 0$ и производится сдвиг содержимого РДМ на один разряд влево;

- если три старших цифры частичного остатка в РДМ не совпадают, а сам ЧО отрицателен, то $q_i = -1$, выполняется сдвиг содержимого РДМ на один разряд влево и к ЧО прибавляется делитель;

- если три старших цифры частичного остатка в РДМ не совпадают, а сам ЧО положителен, то $q_i = 1$, выполняется сдвиг содержимого РДМ на разряд влево и из ЧО вычитается делитель;

3) если после завершения п. 2 остаток отрицателен, то производится коррекция (к остатку прибавляется делитель, а из частного вычитается единица);

4) остаток сдвигается вправо на k разрядов.

Рассмотрим пример реализации данного алгоритма (рис. 3.16).

000001000	0011	
000100000	1100	
000100000		
001000000		
001000000		
010000000		
10100		
111000000		
111000000		
110000000		
100000000		
01100		
111000000		
111000000		
01100		
0100		
01000		
00010		
		← Нормализация делителя и ЧО
		0 ← Три старшие цифры ЧО совпадают
		Сдвиг ЧО влево
		01 ← Три старшие цифры ЧО не совпадают, ЧО > 0
		Сдвиг ЧО влево
		Вычитание делителя
		010 ← Три старшие цифры ЧО совпадают
		Сдвиг ЧО влево
		010 - 1 ← Три старшие цифры ЧО не совпадают, ЧО < 0
		Сдвиг ЧО влево
		Прибавление делителя
		Остаток < 0
		010 - 1 ← Коррекция остатка и частного
		-1
		01 - 10 ← Денормализация остатка и преобразование
		00 10 ← частного в двоичную форму.

Делимое (ЧО)	Делитель	Частное	
000001000	0011		
000100000	1100		
000100000		0 ←	Нормализация делителя и ЧО
001000000			Три старшие цифры ЧО совпадают
001000000			Сдвиг ЧО влево
001000000		01 ←	Три старшие цифры ЧО не совпадают. ЧО > 0
010000000			Сдвиг ЧО влево
10100			Вычитание делителя
111000000			
111000000		010 ←	Три старшие цифры ЧО совпадают
110000000			Сдвиг ЧО влево
100000000		010 - 1 ←	Три старшие цифры ЧО не совпадают. ЧО < 0
01100			Сдвиг ЧО влево
111000000			Прибавление делителя
111000000			Остаток < 0
01100		010 - 1	Коррекция остатка и частного
0100		-1	
01000		01 - 10	Денормализация остатка и преобразование
00010		00 10	частного в двоичную форму.

Рис. 3.16. Пример ускорения операции деления

На первом шаге для удаления предшествующих нулей делитель сдвигается на два разряда влево. Аналогично поступают и с ЧО, который вначале совпадает с делимым. Далее выполняется процедура, описанная выше в п. 2. Операция вычитания D обеспечивается прибавлением делителя с противоположным знаком. Поскольку по завершении операции остаток отрицателен, производится его коррекция путем прибавления D . Одновременно частное уменьшается на единицу (эта операция показана в системе $\{-1, 0, 1\}$, в которой представлено частное). Наконец, на последнем шаге форма представления частного меняется, и переходят к представлению в стандартной двоичной системе.

В стандартном алгоритме деления без восстановления остатка помимо сдвига в каждой итерации выполняется операция сложения или вычитания. В варианте SRT, в зависимости от кодов операндов в отдельных итерациях, достаточно только сдвига, что ускоряет процесс деления. Согласно статистическим данным, в среднем число сложений и вычитаний при использовании этого алгоритма сокращается в 2,67 раза.

В главе рассмотрели основные вопросы арифметики ЭВМ, различные варианты и способы организации операций сложения, вычитания, умножения и деления, применяемых в компьютерах в разных системах счисления. Способы выполнения арифметических операций подробно рассмотрены на примерах, приведенных в данной главе учебного пособия.

Контрольные вопросы

1. Чем обусловлена специфика целочисленного сложения и вычитания? Ответ поясните на примерах.

2. Какую роль играет в целочисленном сложении и вычитании дополнительный код? Ответ поясните на примерах.

3. К чему бы привел отказ от дополнительного кода при целочисленном сложении и вычитании? Ответ поясните на примерах.

4. Как выявляется переполнение при целочисленном сложении и вычитании?

5. Сформулируйте достоинства, недостатки и область применения четырех вариантов целочисленного «традиционного» умножения. Как учитываются знаки сомножителей?

6. Охарактеризуйте суть двух групп логических методов ускорения умножения.

7. Парно сравните алгоритм Бута, модифицированный алгоритм Бута, алгоритм Лемана.

10. Разработайте алгоритм умножения с обработкой за шаг трех разрядов множителя.

11. Поясните суть аппаратных методов ускорения умножения, выделив три возможных подхода.

12. Сравните организацию целочисленного деления с восстановлением остатка и без восстановления остатка. Как учитываются при делении знаки операндов?

13. Обоснуйте возможность совмещения структур множителя и делителя. Опишите объединенную структуру.

14. Сформулируйте четыре пути ускорения целочисленного деления. Сравните между собой их возможную реализацию.

15. Создайте структуру операционного блока для выполнения как сложения/вычитания, так и базового набора логических

операций. Обоснуйте каждый элемент этой структуры.

16. Составьте таблицы умножения для чисел системы счисления с основанием 3.

17. Перемножьте двоичные числа 0111 и 0011.

18. Выполните следующие вычисления над 8-битными числами с использованием ДК:

00101101	11111111	00000000	11110111
+01101111	+11111111	-11111111	-11110111

19. Выполните те же вычисления с использованием ОК.

20. За счет чего операция деления без восстановления остатков короче операции деления с восстановлением остатков?

ЗАКЛЮЧЕНИЕ

В данном учебном пособии излагается основная информация, касающаяся практического применения ЭВМ, в виде базовых сведений о наиболее популярных в настоящее время программных пакетах семейства Microsoft Windows и Microsoft Office: Microsoft Word и Microsoft Excel, их основных возможностей по использованию, подготовке, редактированию и оформлению текстовой документации, графиков, диаграмм и рисунков; обработке числовых данных в электронных таблицах, позволяющих качественно оформлять отчетную документацию при обучении в университете

Рассмотренные в данном учебном пособии общие сведения о представлении информации в ЭВМ, такие как системы счисления, формы представления чисел в ЭВМ, преобразования чисел в системах счисления позволяют студентам разобраться в многообразии представлений информации в цифровой технике и научиться преобразовывать информацию, поступающую в ЭВМ. Также приведены примеры и пояснения по выполнению основных арифметических операций, таких как операции сложения, вычитания, умножения и деления в различных системах счисления. На их основе повышается качество изучения таких дисциплин как «Теория автоматов» и «Организация ЭВМ и систем», в частности, рассмотренная информация оказывает существенную помощь при проектировании процессора ЭВМ в курсовой работе по дисциплине «Организация ЭВМ и систем».

Данное пособие носит пояснительный характер и предназначено для ознакомления с базовыми основами функционирования ЭВМ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Григорьев, В.Л. Самоучитель по операционной системе. М.: Энергоатомиздат. – 1992. 311 с.
2. Фигурнов, В.Э. IBM PC для пользователя [Текст] / В.Э. Фигурнов. Изд 6-е перераб. и доп. М.: ИНФРА-М. 1996. 287 с.
3. Жмакин, А.П., Бабкин Е.А. Текстовый редактор Word-97. [Текст] / А.П. Жмакин, Е.А. Бабкин; Курск. гос. тех. ун-т, 1999. 118 с.
4. Персон, Р. Microsoft Excel в подлиннике [Текст] Т.1 / Р. Персон. СПб.: ВHV–С-Петербург. – 1997. 651 с.
5. Персон, Р. Microsoft Excel в подлиннике [Текст] Т.2. / Р. Персон. СПб.: ВHV–С-Петербург. – 1997. 1272 с.
6. Острейковский, В.А. Информатика [Текст] / В.А. Острейковский. М.: Высшая школа. – 2000. 512 с.
7. Booth, A.D. A signal binary multiplication technique [Text] / A.D. Booth // Mech. Appl. Math. 1951. 4, part 2. P. 236-240.
8. Lehman, M. High-speed digital multiplication. [Text] / M. Lehman // IRE Transaction on electronic computers. 1957. V. EC-6–6, №3б.
9. Цилькер, Б.Я. Организация ЭВМ и систем [Текст]: учебник для вузов / Б.Я. Цилькер, С.А. Орлов. СПб.: Питер, 2004. 668 с.
10. Танэнбаум, Э. Архитектура компьютера [Текст] / Э. Таненбаум. 4-е изд. СПб.: Питер, 2003. 704 с.
11. Cocks J. High Speed Arithmetic in a Parallel Device [Text] / J. Cocks, D.W. Sweney // Technical Report IBM. 1957. feb.
12. Robertson, J.E. A new class of digital division methods [Text] / J.E. Robertson // IEEE Transactions on Computers, Electronics Computers, EC–7. 1958, Sep. p. 218-222.
13. Хохлов, А.Е.. Основы информатики: Конспект лекций [Текст] / А.Е. Хохлов. Пенза: Изд-во Пенз. гос. ун-та, 2003. – 103 с.

14. Советов, Б.Я. Информационные технологии [Текст]: Учебник / Б.Я. Советов, В.В. Цехановский. – М.: Высш. шк., 2003. – 263 с.
15. Рагулин, П.Г. Информационные технологии. Электронный учебник [Текст] /П.Г. Рагулин. Владивосток: ТИДОТ, Дальневост. ун-т, 2004. - 208 с.
16. Информационные системы в экономике [Текст]: Учебник. / Под ред. В.В. Дика. – М.: Финансы и статистика, 1996. – 272 с.
17. Бройдо, В.Л. Вычислительные системы, сети и телекоммуникации [Текст]: Учебник. – 2-е изд. / В.Л. Бройдо. – СПб.: Питер, 2003. – 704 с. – (Серия «Учебники для вузов»).
18. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.
19. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.
20. Информационные технологии управления [Текст]: Учеб. пособие / Под ред. Г.А. Титоренко. – М.: ЮНИТИ-ДАНА, 2002. – 280 с.
21. Борзов, Д.Б. Процедурно–ориентированное программирование на С++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватутин; Курск гос. тех. ун-т. Курск, 2007. 209 с.
22. Столингс, В. Современные компьютерные сети [Текст] / В. Столингс. СПб.: Питер, 2003, . 783 с.
23. Баскаков С.И. Радиотехнические цепи и сигналы: Учебник для вузов. - М.: Высшая школа, 1988. - 462 с.
24. Дмитриев В.И. Прикладная теория информации: Учебник для вузов. - М.: Высшая школа, 1989.- 325 с.
25. Лосев А.К. Линейные радиотехнические цепи: Учебник для вузов. - М.: Высшая школа, 1971.- 560 с.
26. Сергиенко А.Б. Цифровая обработка сигналов. / Учебник для вузов. – СПб.: Питер, 2003203. – 608 с.

Борзов Дмитрий Борисович
Чернецкая Ирина Евгеньевна

ИНФОРМАТИКА

Учебное издание

Редактор
Компьютерная верстка и макет ...

ИД №06430 от 10.12.01.

Подписано в печать . Формат 60×84 1/16. Бумага офсетная.
Усл. печ. п. . Уч.-изд.л. . Тираж экз. Заказ .
Курский государственный технический университет.
Издательско-полиграфический центр Курского государственного
технического университета. 305040, г. Курск, ул. 50 лет Октября, 94.