Н.В.Максимов Т.Л.Партыка И.И.Попов

АРХИТЕКТУРА ЭВМ И ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

АРХИТЕКТУРА ЭВМ
И.И.Попов
Т.Л.Партыка
Н.В.Максимов

ФОРУМ
профессиональное образование

МАКСИМОВ Николай Вениаминович -
доктор технических наук, доцент, заведующий кафедрой Информационных ресурсов Российского государственного гуманитарного университета.

ПАРТЫКА Татьяна Леонидовна -
кандидат экономических наук, доцент кафедры проектирования автоматизированных информационных систем Российской экономической академии им. Г.В.Плеханова.

ПОПОВ Игорь Иванович -
доктор технических наук, профессор, заведующий кафедрой Информатики Российской экономической академии им. Г.В.Плеханова.

ISBN 5-8199-0160-6

ФОРУМ
профессиональное образование

Н. В. Максимов, Т. Л. Партыка,
И. И. Попов

АРХИТЕКТУРА ЭВМ И ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Рекомендовано Министерством образования Российской Федерации в качестве учебника для студентов учреждений среднего профессионального образования, обучающихся по группе специальностей 2200 Информатика и вычислительная техника

Москва
ФОРУМ - ИНФРА-М
2005

УДК 004.2(075.32)
ББК 32.973-02я723
М17

Рецензенты:
к т. н , доцент кафедры «Проектирование АИС»
РЭА им. Г. В. Плеханова *Ю. Г Бачинин*,
доктор экономических наук, профессор, декан факультета «Информатика»
в НОУ «Московский международный институт эконометрики,
информатики, финансов и права» (ММИЭИФП) *А А. Емельянов*

Максимов Н. В., Партыка Т. Л., Попов И. И.

М17 Архитектура ЭВМ и вычислительных систем: Учебник. — М.: ФОРУМ: ИНФРА-М, 2005. — 512 с.: ил. — (Профессиональное образование).

ISBN 5-8199-0160-6 (ФОРУМ)
ISBN 5-16-002257-0 (ИНФРА-М)

Рассмотрены вопросы организации и функционирования вычислительных устройств, машин и систем. Описываются логические, информационные, алгоритмико-вычислительные основы построения систем. Значительное внимание уделено архитектурам вычислительных машин и систем, их классификациям, составным компонентам — информационно-вычислительным средам и коммутационно-коммуникационным средам. В качестве примера подробно представлены технические, структурные, архитектурные компоненты персональных машин и средства их комплексирования.

Для студентов учреждений среднего профессионального образования, обучающихся по группе специальностей 2200 Информатика и вычислительная техника.

УДК 004.2(075.32)
ББК 32.973-02я723

ISBN 5-8199-0160-6 (ФОРУМ)
ISBN 5-16-002257-0 (ИНФРА-М)

Введение

Человек научился добывать и поддерживать огонь 30—40 тысяч лет назад. Однако только примерно в середине XVIII столетия он смог использовать это умение для создания первых паровых машин. Промышленная революция конца XVIII — начала XIX в. самым радикальным образом преобразовала производство. Руки и физическую силу человека постепенно заменили механизмы, машины, станки. Научное и промышленное применение электричества, разработка и использование приборов и средств автоматизации позволили уже в XX в. не только механизировать, но и автоматизировать многие технологические процессы. Одновременно возрастание объема информации, связанное с бурным развитием науки, усложнением техники и технологии, ускорением темпов развития производства и общественной жизни, привело к такому же увеличению затрат нервной энергии и умственного труда. В ряде случаев, особенно в сфере управления производством, экономикой и социальными процессами, уже невозможно стало обходиться без совершенных технических средств, способных взять на себя часть интеллектуальной работы. Необходимо было передать автоматам значительную долю информационной деятельности человека. Таким универсальным «информационным автоматом» стал компьютер (электронная вычислительная машина, ЭВМ).

Человечество прошло большой и трудный исторический путь перед тем, как достичь современного уровня развития и применения компьютеров, информационных и сетевых технологий.

В далеком доисторическом прошлом люди считали на пальцах или делали насечки на костях. Древнейшим «счетным инструментом», который был представлен самой природой в распоряжение человека, была его собственная рука. На заре человеческой цивилизации были изобретены различные системы счисления, позволяющие осуществлять торговые сделки, рассчитывать астрономические циклы, проводить другие вычисления. Спустя несколько тысячелетий появились первые ручные вычислительные средства.

В наши дни сложнейшие вычислительные задачи, как и множество других операций, казалось бы, не связанных с числами, решаются именно с помощью компьютеров. Это еще один тип машин, построенных для того, чтобы увеличить эффективность и качество выполняемых работ и повысить производительность труда. Он по существу обладает единственной способностью — обрабатывать с высокой быстротой импульсы электрического поля. Истинное величие заключено в человеке, его гении, который нашел способ преобразовать разнообразную информацию, поступающую из реального мира, в последовательность нулей и единиц двоичного кода, т. е. выразить ее на цифровом языке, идеально подходящем для электронных схем компьютера. Однако ни одна другая машина в истории не принесла в наш мир столь быстрых и глубоких изменений (посадка аппаратов на поверхность Луны и исследование планет Солнечной системы, управление медицинской аппаратурой в операционных, решение сложных экономических и управленческих задач, принятие управленческих решений, управление телефонными станциями и многое др.).

Каждому, знакомому с современными компьютерами, механические счетные машины и приборы покажутся, пожалуй, забавными и неуклюжими устройствами. Однако, ознакомившись с историей развития счетных машин, можно поразиться изобретательности, хитроумию и настойчивости их создателей. Уместно вспомнить слова Б. Паскаля о том, что для создания «арифметической машины» ему потребовалось все ранее приобретенные знания по геометрии, физике и механике.

Закладка фундамента компьютерной революции происходила медленно и далеко не гладко. Отправной точкой этого процесса можно считать изобретение счетов (более 1500 лет назад). Они оказались очень эффективным инструментом и вскоре распространились по всему миру (в некоторых странах применяются и по сей день). В XVII в. европейские мыслители были увлечены идеей создания счетных устройств.

Работы, выполняемые в 40-х годах XX в. по созданию вычислительных машин с программным управлением, были тесно связаны с появлением новой фундаментальной области науки — кибернетики, или науки об управлении и коммуникации. Судьба этой науки в нашей стране (в бывшем СССР) была трудной. Долгое время она считалась буржуазной «вульгарной лженаукой». Только в начале 50-х гг. прошлого века появились первые советские вычислительные машины. Несмотря на это, роль отечественных ученых в области кибернетики и вычислительной техники неоценима.

Современный компьютер — это универсальное многофункциональное электронное автоматическое устройство для работы с информацией. Компьютеры в современном обществе взяли на себя значительную часть работ, связанных с информацией. По историческим меркам компьютерные технологии обработки информации еще очень молоды и находятся в самом начале своего развития, однако уже сегодня они преобразуют или вытесняют традиционные процессы обработки информации.

Содержанием предлагаемого учебника является попытка краткого изложения основных сведений, отражающих этапы развития, классы, структуры, функции и технические средства вычислительных машин и систем.

В первой главе рассмотрены основные события в истории развития вычислительных устройств, машин и приборов, различные принципы классификации вычислительных машин. Рассмотрены также основные аспекты основ вычислительной техники — информационные (представление, кодирование, обработка информации в ЭВМ), логические основы (основные функции, операции, элементы и узлы) и алгоритмические основы (структура и основные элементы программ и алгоритмов).

Во второй главе речь идет об основных представлениях об архитектурных и структурных компонентах вычислительных машин и систем. Рассмотрены базовые представления об архитектуре ЭВМ, типы, устройство, структура и функционирование процессоров, а также основные направления повышения их производительности, организация оперативной памяти и интерфейсов ЭВМ. Здесь же содержатся краткий обзор и характеристики таких внешних устройств, как накопители (магнитные ленты, магнитные, оптические и другие типы дисковых устройств), средства диалогового (терминалы, манипуляторы), а также пакетного (принтеры, плоттеры, сканеры, дигитайзеры) ввода-вывода информации.

Третья глава посвящена проблематике вычислительных систем. Рассмотрены различные подходы (М. Флинн и другие исследователи) к их классификации, примеры некоторых архитектур вычислительных систем, абстрактные представления об архитектуре вычислительных машин, систем и сетей, использующие такие понятия, как процессорные среды, запоминающие (среды памяти), а также коммутационные и коммуникационные среды. В связи с этим вкратце рассмотрены перспективные типы процессоров ЭВМ, системы памяти, коммуникационные среды, коммутаторы вычислительных систем, а также приведены характеристики некоторых вычислительных систем и суперкомпьютеров.

В четвертой главе рассмотрены некоторые приложения принципов организации и функционирования ЭВМ и систем к наиболее популярному их представителю — персональному компьютеру (ПК, ПЭВМ). Вкратце рассмотрены история, общая структура и устройство ПК на процессорах INTEL, эволюция процессоров Intel и их аналогов, режимы процессора, система команд реального режима процессоров i80x86 (в терминах макроассемблера — MASM). Рассмотрены также особенности защищенного режима процессора и вопросы, связанные с BIOS и ее настройкой.

Настоящий учебник базируется на материалах, которые авторы накопили в процессе практической, исследовательской, а также преподавательской (МИФИ, МИСИ, МГУ, РГГУ, РЭА им. Г. В. Плеханова) деятельности. Авторы выражают благодарность коллегам, принявшим участие в обсуждении материала: А. Г. Романенко (РГГУ), К. И. Курбакову (РЭА им. Г. В. Плеханова), П. Б. Храмцову (РНИЦ «Курчатовский институт»), рецензентам, а также студентам РГГУ, РЭА им. Г. В. Плеханова за предоставленные иллюстративные материалы.

Глава 1
ВЫЧИСЛИТЕЛЬНЫЕ ПРИБОРЫ И УСТРОЙСТВА. АЛГОРИТМЫ И ВЫЧИСЛЕНИЯ

Вплоть до XVII в. деятельность общества в целом и каждого человека в отдельности была направлена на овладение веществом, т. е. есть познание свойств вещества и изготовление сначала примитивных, а потом все более сложных орудий труда, вплоть до механизмов и машин, позволяющих изготовлять потребительские ценности.

Затем в процессе становления индустриального общества на первый план вышла проблема овладения энергией — сначала тепловой, затем электрической, наконец, атомной. Овладение энергией позволило освоить массовое производство потребительских ценностей и, как следствие, повысить уровень жизни людей и изменить характер их труда.

В то же время человечеству свойственна потребность выразить и запомнить информацию об окружающем мире — так появились письменность, книгопечатание, живопись, фотография, радио, телевидение. В истории развития цивилизации можно выделить несколько информационных революций — преобразование общественных отношений из-за кардинальных изменений в сфере обработки информации, информационных технологий. Следствием подобных преобразований являлось приобретение человеческим обществом нового качества.

В конце XX в. человечество вступило в новую стадию развития — стадию построения информационного общества. Информация стала важнейшим фактором экономического роста, а уровень развития информационной деятельности и степень вовлеченности и влияния ее на глобальную информационную инфраструктуру превратились в важнейшее условие конкурентоспособности страны в

мировой экономике. Понимание неизбежности прихода этого общества наступило значительно раньше. Австралийский экономист К. Кларк еще в 40-е годы говорил о приближении эпохи общества информации и услуг, общества новых технологических и экономических возможностей. Американский экономист Ф. Махлуп выдвинул предположение о наступлении информационной экономики и превращении информации в важнейший товар в конце 50-х гг. В конце 60-х гг. Д. Белл констатировал превращение индустриального общества в информационное. Что касается стран, ранее входивших в "СССР, то процессы информатизации в них развивались замедленными темпами.

Информатика меняет всю систему общественного производства и взаимодействия культур. С наступлением информационного общества начинается новый этап не только научно-технической, но социальной революции. Меняется вся система информационных коммуникаций. Разрушение старых информационных связей между отраслями экономики, направлениями научной деятельности, регионами, странами усилило экономический кризис конца века в странах, которые уделяли развитию информатизации недостаточное внимание. Важнейшая задача общества — восстановить каналы коммуникации в новых экономических и технологических условиях для обеспечения четкого взаимодействия всех направлений экономического, научного и социального развития как отдельных стран, так и в глобальном масштабе.

Современный компьютер — это универсальное, многофункциональное, электронное автоматическое устройство для работы с информацией Компьютеры в современном обществе взяли на себя значительную часть работ, связанных с информацией. По историческим меркам компьютерные технологии обработки информации еще очень молоды и находятся в самом начале своего развития. Компьютерные технологии сегодня преобразуют или вытесняют старые технологии обработки информации.

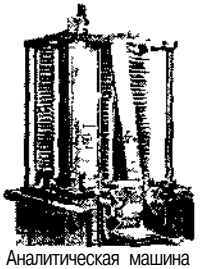
1.1. Вычислительные устройства и приборы, история вопроса («Время — события — люди»)

Рассмотрим историю развития вычислительных средств и методов «в лицах» и объектах (табл. 1 1)

Таблица 1 1 **Основные события в истории развития вычислительных методов, приборов, автоматов и машин**

 Джон Непер (1550-1617)	Шотландец Джон Непер в 1614 м г опубликовал «Описание удивительных таблиц логарифмов» Он обнаружил, что сумма логарифма чисел *a* и *b* равна логарифму произведения этих чисел По этому действие умножения сводилось к простой операции сложения Также им разработан инструмент перемножения чисел — «костяшки Непера» Он состоял из набора сегментированных стерженьков, которые можно было располагать таким образом, что, складывая числа в прилегающих друг к другу по горизонтали сегментах, получали результат их умножения «Костяшки Непера» вскоре были вытеснены другими вычислительными устройствами (в основном механического типа) Таблицы Непера, расчет которых требовал очень много времени, были позже «встроены» в удобное устройство, ускоряющее процесс вычисления, - логарифмическую линейку (Р Биссакар, конец 1620 г)
 Вильгельм Шиккард (1592-1636)	Считалось, что первую механическую счетную машину изобрел великий французский математик и физик Б Паскаль в 1642 г. Однако в 1957 г Ф Гаммер (ФРГ, директор Кеплеровского научного центра) обнаружил доказательства создания механической вычислительной машины приблизительно за два десятилетия до изобретения Паскаля Вильгельмом Шиккардом Он назвал ее «часы для счета» Машина предназначалась для выполнения четырех арифметических действий и состояла из частей суммирующее устройство, множительное устройство, механизм для промежуточных результатов Суммирующее устройство состояло из зубчатых передач и представляло простейшую форму арифмометра Предложенная схема механического счета считается классической Однако эту простую и эффективную схему пришлось изобретать заново, так как сведения о машине Шиккарда не стали всеобщим достоянием
 Блэз Паскаль (1623-1662)	В 1642 г , когда Паскалю было 19 лет, была изготовлена первая действующая модель суммирующей машины Через несколько лет Блэз Паскаль создал механическую суммирующую машину («паскалина»), которая позволяла складывать числа в десятичной системе счисления В этой машине цифры шестизначного числа задавались путем соответствующих поворотов дисков (колесиков) с цифровыми делениями, результат операции можно было прочитать в шести окошках - по одному на каждую цифру Диск единиц был связан с диском десятков, диск десятков - с диском сотен и т д Другие операции выполнялись с помощью довольно неудобной процедуры повторных сложений, и в этом заключался основной недостаток «паскалины» Всего приблизительно за десятилетие он построил более 50 различных вариантов машины Изобретенный Паскалем принцип связанных колес явился основой, на которой строилось большинство вычислительных устройств на протяжении следующих трех столетий

Продолжение табл 1 1

 Готфрид Вильгельм Лейбниц (1646-1716)	В 1672 г , находясь в Париже, Лейбниц познакомился с голландским математиком и астрономом Христианом Гюйгенсом Видя как много вычислении приходится делать астроному Лейбниц решил изобрести механическое устроиство для расчетов В 1673 г он завершил создание механического калькулятора Развив идеи Паскаля, Лейбниц использовал операцию сдвига для поразрядного умножения чисел Сложение производилось на нем по существу так же, как и на «паскалине», однако Лейбниц включил в конструкцию движущуюся часть (прообраз подвижной каретки будущих настольных калькуляторов) и ручку, с помощью которой можно было крутить ступенчатое колесо или - в последующих вариантах машины - цилиндры, расположенные внутри аппарата
 Жозеф Мари Жаккар (1775--1834)	Развитие вычислительных устройств связано с появлением перфорационных карт и их применением Появление же перфорационных карт связано с ткацким производством В 1804 г инженер Жозеф-Мари Жаккар построил полностью автоматизированный станок (станок Жаккара), способный воспроизводить сложнейшие узоры Работа станка программировалась с помощью колоды перфокарт, каждая из которых управляла одним ходом челнока Переход к новому рисунку происходил заменой колоды перфокарт
 Чарльз Бэббидж (1791–1871)  Аналитическая машина Ч Бэббиджа	Он обнаружил погрешности в таблицах логарифмов Непера, которыми широко пользовались при вычислениях астрономы, математики, штурманы дальнего плавания В 1821 г приступил к разработке своей вычислительной машины, которая помогла бы выполнить более точные вычисления В 1822 г была построена разностная машина (пробная модель), способная рассчитывать и печатать большие математические таблицы Это было очень сложное большое устройство и предназначалось для автоматического вычисления логарифмов Работа модели основывалась на принципе известном в математике как «метод конечных разностей» при вычислении многочленов используется только операция сложения и не выполняется умножение и деление которые значительно труднее поддаются автоматизации В последующем он пришел к идее создания более мощной - аналитической машины Она не просто должна была решать математические задачи определенного типа а выполнять разнообразные вычислительные операции в соответствии с инструкциями, задаваемыми оператором По замыслу это не что иное, как первый универсальный программируемый компьютер Аналитическая машина в своем составе должна была иметь такие компоненты, как «мельница» (арифметическое устройство по современной терминологии) и «склад» (память) Инструкции (команды) вводились в аналитическую машину с помощью перфокарт (использовалась идея программного управления Жаккара с помощью перфокарт) Шведский издатель, изобретатель и переводчик Пер Георг Шойц воспользовавшись

Продолжение таол 1 1

	советами Бэббеджа построил видоизмененный вариант этой машины В 1855 г машина Шоица была удостоена золотой медали на Всемирной выставке в Париже В дальнейшем один из принципов, лежащих в основе идеи аналитической машины, - использование перфокарт - нашел воплощение в статистическом табуляторе построенном американцем Германом Холлеритом (для ускорения обработки результатов переписи населения в США в 1890 г)
 Огаста Ада Байрон (графиня Лавлейс) (1815-1852)	Графиня Огаста Ада Лавлейс, дочь поэта Байрона, совместно с Ч Бэббиджем работала над созданием программ для его счетных машин Ее работы в этой области были опубликованы в 1843 г Однако в то время считалось неприличным для женщины издавать свои сочинения под полным именем, и Лавлейс поставила на титуле только свои инициалы В материалах Бэббиджа и комментариях Лавлейс намечены такие понятия, как «подпрограмма» и «библиотека подпрограмм», «модификация команд» и «индексный регистр», которые стали употребляться только в 50 х гг XX в Сам термин «библиотека» был введен Бэббиджем, а термины «рабочая ячейка» и «цикл» предложила А Лавлейс «Можно с полным основанием сказать, что аналитическая машина точно так же плетет алгебраические узоры, как ткацкий станок Жаккара воспроизводит цветы и листья», - писала графиня Лавлейс Она фактически была первой программисткой (в ее честь был назван язык программирования Ада)
 Джордж Буль (1815 –1864)	Дж Буль по праву считается отцом математической логики Его именем назван раздел математической логики - булева алгебра В 1847 г написал статью «Математический анализ логики» В 1854 г Буль развил свои идеи в работе под названием «Исследование законов мышления» Эти труды внесли революционные изменения в логику как науку Дж Буль изобрел своеобразную алгебру - систему обозначений и правил, применяемую к всевозможным объектам, от чисел и букв до предложений Пользуясь этой системой, Буль мог закодировать высказывания (утверждения) с помощью своего языка, а затем манипулировать ими подобно тому, как в математике манипулируют обычными числами Три основные операции системы - это И, ИЛИ и НЕ
 Пафнутий Львович Чебышев (1821-1894)	Им была разработана теория машин и механизмов, написан ряд работ, посвященных синтезу шарнирных механизмов Среди многочисленных изобретенных им механизмов имеется несколько моделей арифмометров, первая из которых была сконструирована не позднее 1876 г Арифмометр Чебышева для того времени был одной из самых оригинальных вычислительных машин В своих конструкциях Чебышев предложил принцип непрерывной передачи десятков и автоматический переход каретки с разряда на разряд при умножении Оба эти изобретения вошли в широкую практику в 30-е гг XX в в связи с применением электропривода и распространением полуавтоматических и автоматических клавишных вычислительных машин С появлением этих и других изобретений стало возможно значительно увеличить скорость работы механических счетных устройств

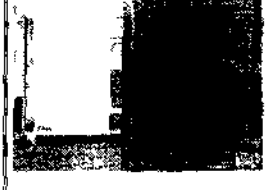
Продолжение табл 1 1

 Алексей Николаевич Крылов (1863-1945)	Русский кораблестроитель механик математик, академик АН СССР В 1904 г он предложил конструкцию машины для интегрирования обыкновенных дифференциальных уравнений В 1912 г такая машина была построена Это была первая интегрирующая машина непрерывного действия, позволяющая решать дифференциальные уравнения до четвертого порядка
 Вильгодт Теофил Однер (1845–1905)	Выходец из Швеции Вильгодт Теофил Однер в 1869 г приехал в Петербург Некоторое время он работал на заводе «Русский дизель» на Выборгской стороне, на котором в 1874 г был изготовлен первый образец его арифмометра Созданные на базе ступенчатых валиков Лейбница первые серийные арифмометры имели большие размеры в первую очередь потому, что на каждый разряд нужно было выделять отдельный валик Однер вместо ступенчатых валиков применил более совершенные и компактные зубчатые колеса с меняющимся числом зубцов - колеса Однера В 1890 г Однер получает патент на выпуск арифмометров и в этом же году было продано 500 арифмометров (очень большое количество по тем временам) Арифмометры в России назывались «Арифмометр Однера», «Оригинал Однер», «Арифмометр системы Однер» и др В России до 1917 г было выпущено примерно 23 тыс арифмометров Однера После революции производство арифмометров было налажено на Сущевском механическом заводе им Ф Э Дзержинского в Москве С 1931 г они стали называться арифмометры «Феликс» Далее в нашей стране были созданы модели арифмометров Однера с клавишным вводом и электроприводом
 Герман Холлерит (1860-1929)	После окончания Колумбийского университета поступает на работу в контору по переписи населения в Вашингтоне В это время США приступили к исключительно трудоемкой (длившейся семь с половиной лет) ручной обработке данных, собранных в ходе переписи населения в 1880 г К 1890 г Холлерит завершил разработку системы табуляции на базе применения перфокарт На каждой карте имелось 12 рядов, в каждом из которых можно было пробить по 20 отверстий, они соответствовали таким данным, как возраст, пол, место рождения, количество детей, семейное положение и прочим сведениям включенным в вопросник переписи Содержимое заполненных формуляров переносилось на карты путем соответствующего перфорирования Перфокарты загружались в специальные устройства, соединенные с табуляционной машиной, где они нанизывались на ряды тонких игл, по одной игле на каждую из 240 перфорируемых позиций на карте Когда игла попадала в отверстие, она замыкала контакт в соответствующей электрической цепи машины Полный статистический анализ результатов занял два

Продолжение табл ! 1

	с половиной года (втрое быстрее по сравнению с предыдущей переписью) Впоследствии Холлерит организовал фирму «Computer Tabulating Recording» (CTR) Молодой коммивояжер этой компании Том Уотсон первым увидел потенциальную прибыльность продажи счетных машин американским бизнесменам на основе перфокарт Позднее он возглавил компанию и в 1924 г переименовал ее в корпорацию «International Business Machines» (IBM)
 Ванневар Буш (1890-1974)	В 1930 г построил механическое вычислительное устройство - дифференциальный анализатор Это была машина, на которой можно было решать сложные дифференциальные уравнения Однако она обладала многими серьезными недостатками, прежде всего, гигантскими размерами Механический анализатор Буша представлял собой сложную систему валиков, шестеренок и проволок, соединенных в серию больших блоков, которые занимали целую комнату При постановке задачи машине оператор должен был вручную подбирать множество шестереночных передач На это уходило обычно 2-3 дня Позднее В Буш предложил прототип современного гипертекста - проект MEMEX (MEMory EXtention - расширение памяти) как автоматизированное бюро, в котором человек хранил бы свои книги, записи, любую получаемую им информацию таким образом, чтобы в любой момент воспользоваться ею с максимальной быстротой и удобством Фактически это должно было быть сложное устройство, снабженное клавиатурой и прозрачными экранами, на которые бы проецировались тексты и изображения, хранящиеся на микрофильмах В MEMEX устанавливались бы логические и ассоциативные связи между любыми двумя блоками информации В идеале речь идет о громадной библиотеке, универсальной информационной базе
 Джон ВинсентАтанасофф (1903-1995)	Профессор физики, автор первого проекта цифровой вычислительной машины на основе двоичной, а не десятичной системы счисления Простота двоичной системы счисления в сочетании с простотой физического представления двух символов (0, 1) вместо десяти (0, 1, , 9) в электрических схемах компьютера перевешивала неудобства, связанные с необходимостью перевода из двоичной системы в десятичную и обратно Кроме того, применение двоичной системы счисления способствовало уменьшению размеров вычислительной машины и снизила бы ее себестоимость В 1939 г Атанасофф построил модель устройства и стал искать финансовую помощь для продолжения работы Машина Атанасоффа была практически готова в декабре 1941 г , но находилась в разобранном виде В связи с началом Второй мировой войны все работы по реализации этого проекта прекратились Лишь в 1973 г приоритет Атанасоффа как автора первого проекта такой архитектуры вычислительной машины был подтвержден решением федерального суда США

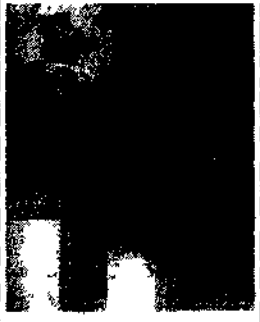
Продолжение табл 1 1

 Говард Айкен  «Марк-1»	В 1937 г Г Айкен предложил проект большой счетной машины и искал людей согласных профинансировать эту идею Спонсором выступил Томас Уотсон, президент корпорации IBM его вклад в проект составил около 500 тыс долларов США Проектирование новой машины «Марк-1» основанной на электромеханических реле, началось в 1939 г в лабораториях Нью Йоркского филиала IBM и продолжалось до 1944 г Готовый компьютер содержал около 750 тыс деталей и весил 35 т Машина оперировала двоичными числами до 23 разрядов и перемножала два числа максимальной разрядности примерно за 4 с Поскольку создание «Марк-1» длилось достаточно долго, пальма первенства досталась не ему, а релейному двоичному компьютеру Z3 Конрада Цузе, построенному в 1941 г Стоит отметить, что машина Z3 была значительно меньше машины Айкена и к тому же дешевле в производстве
 Конрад Цузе (1910-1995)  Машина Z3	В 1934 г , будучи студентом технического вуза (в Берлине), не имея ни малейшего представления о работах Ч Бэббиджа, К Цузе начал разрабатывать универсальную вычислительную машину, во многом подобную аналитической машине Бэббиджа В 1938 г он завершил постройку машины, занимавшую площадь 4 кв м , названную Z1 (по немецки его фамилия пишется как Zuse) Это была полностью электромеханическая программируемая цифровая машина Она имела клавиатуру для ввода условий задач Результаты вычислений высвечивались на панели с множеством маленьких лампочек Ее восстановленная версия хранится в музее Verker und Technik в Берлине Именно Z1 в Германии называют первым в мире компьютером Позднее Цузе стал кодировать инструкции для машины, пробивая отверстия в использованной 35-миллиметровой фотопленке Машина, работавшая с перфорированной лентой, получила название Z2 В 1941 г Цузе построил программно-управляемую машину, основанную на двоичной системе счисления - Z3 Эта машина по многим своим характеристикам превосходила другие машины, построенные независимо и параллельно в иных странах В 1942 г Цузе совместно с австрийским инженером электриком Хельмутом Шрайером предложили создать компьютер принципиально нового типа - на вакуумных электронных лампах Эта машина должна была работать в тысячу раз быстрее, чем любая из машин, имевшихся в то время в Германии Говоря о потенциальных сферах применения быстродействующего компьютера, Цузе и Шрайер отмечали возможность его использования для расшифровки закодированных сообщений (такие разработки уже велись в различных странах)

Продолжение табл 1 1

 Алан Тьюринг (1912-1954)	Английский математик, дал математическое определение алгоритма через построение, названное машиной Тьюринга В период Второй мировой войны немцы использовали аппарат «Enigma» для шифровки сообщений Без ключа и схемы коммутации (немцы их меняли три раза в день) расшифровать сообщение было невозможно С целью раскрытия секрета британская разведка собрала группу блестящих и несколько эксцентричных ученых Среди них был математик Алан Тьюринг В конце 1943 г группа сумела построить мощную машину (вместо электромеханических реле в ней применялись около 2000 электронных вакуумных ламп) Машину назвали «Колосс» Перехваченные сообщения кодировались, наносились на перфоленту и вводились в память машины Лента вводилась посредством фотоэлектрического считывающего устройства со скоростью 5000 символов в секунду Машина имела пять таких считывающих устройств В процессе поиска соответствия (расшифровки) машина сопоставляла зашифрованное сообщение с уже известными кодами «Enigma» (по алгоритму работы машины Тьюринга) Работа группы до сих пор остается засекреченной О роли Тьюринга в работе группы можно судить по следующему высказыванию члена этой группы математика И Дж Гуда «Я не хочу сказать, что мы выиграли войну благодаря Тьюрингу, но беру на себя смелость сказать, что без него мы могли бы ее и проиграть» Машина «Колосс» была ламповая (крупный шаг вперед в развитии вычислительной техники) и специализированная (расшифровка секретных кодов)
 Джон Мочли (1907-1980)  Преспер Экерт (род в 1919)	Первой ЭВМ считается машина ЭНИАК (ENIAC, Electronic Numeral Integrator and Computer - электронный цифровой интегратор и вычислитель) Ее авторы, американские ученые Дж Мочли и Преспер Экерт, работали над ней с 1943 по 1945 гг Она предназначалась для расчета траекторий полетов снарядов, и представляла собой сложнейшее для середины XX в инженерное сооружение длиной более 30 м, объемом 85 куб м, массой 30 т В ЭНИАКе были использованы 18 тыс электронных ламп, 1500 реле, машина потребляла около 150 кВт Далее возникла идея создания машины с программным обеспечением, хранимым в памяти машины, что изменило бы принципы организации вычислений и подготовило почву для появления современных языков программирования (ЭДВАК - Электронный Автоматический Вычислитель с дискретными переменными, EDVAC - Electronic Discret Variable Automatic Computer) Эта машина была создана в 1950 г В более емкой внутренней памяти содержались и данные, и программа Программы записывались электронным способом в специальных устройствах - линиях задержки Самое главное было то, что в ЭДВАКе данные кодировались не в десятичной системе, а в двоичной (сократилось количество используемых электронных ламп) Дж Мочли и П Экерт после создания своей собственной компании задались целью создать универсальный компьютер для широкого коммерческого применения - ЮНИВАК (UNIVAC, Universal Automatic Computer - универсальный автоматический компьютер) Примерно за год до того, как первый

Продолжение табл 1 1

 ЭНИАК	ЮНИВАК вступил в эксплуатацию в Бюро переписи населения в США, партнеры оказались в тяжелом финансовом положении и вынуждены были продать свою компанию фирме «Ремингтон Рэнд» Однако ЮНИВАК не стал первым коммерческим компьютером Им стала машина ЛЕО (LEO, Lyons Electronic Office), которая применялась в Англии для расчета зарплаты работникам чайных магазинов (фирма «Лайонс») В 1973 г федеральный суд США признал их авторские права на изобретение электронного цифрового компьютера недействительными, а идеи - заимствованными у Дж Атанасоффа
 Джон фон Нейман (1903–1957)	Работая в группе Дж Мочли и П Экерта, фон Нейман подготовил отчет - «Предварительный доклад о машине ЭДВАК», в котором обобщил планы работы над машиной Это была первая работа го цифровым электронным компьютерам, с которой познакомились определенные круги научной общественности (по соображениям секретности работы в этой области не публиковались) С этого момента компьютер был признан объектом, представлявшим научный интерес В своем докладе фон Нейман выделил и детально описал пять ключевых компонентов того, что ныне называют «архитектурой фон Неймана» современного компьютера В нашей стране независимо от фон Неймана были сформулированы более детальные и полные принципы построения электронных цифровых вычислительных машин (Сергей Алексеевич Лебедев)
 Сергей Алексеевич Лебедев (1902-1974)	В 1946 г С А Лебедев становится директором института электротехники и организует в его составе свою лабораторию моделирования и регулирования В 1948 г С А Лебедев ориентировал свою лабораторию на создание МЭСМ (Малая электронная счетная машина) МЭСМ была вначале задумана как модель (первая буква в аббревиатуре МЭСМ) Большой электронной счетной машины (БЭСМ) Однако в процессе ее создания стала очевидной целесообразность превращения ее в малую ЭВМ Из за засекреченности работ, проводимых в области вычислительной техники, соответствующих публикаций в открытой печати не было Основы построения ЭВМ, разработанные С А Лебедевым независимо от Дж фон Неймана, заключаются в следующем 1) в состав ЭВМ должны входить устройства арифметики, памяти, ввода вывода информации, управления, 2) программа вычислений кодируется и хранится в памяти подобно числам 3) для кодирования чисел и команд следует использовать двоичную систему счисления, 4) вычисления должны осуществляться автоматически на основе хранимой в памяти программы и операций над командами,

Продолжение табл 1 1

МЭСМ

БЭСМ-6

5) помимо арифметических операций вводятся также логические - сравнения, условного и безусловного переходов, конъюнкция, дизъюнкция, отрицание,

6) память строится по иерархическому принципу,

7) для вычислений используются численные методы решения задач

25 декабря 1951 г МЭСМ была принята в эксплуатацию Это была первая в СССР быстродействующая электронная цифровая машина

В 1948 г создается Институт точной механики и вычислительной техники (ИТМ и ВТ) АН СССР, которому правительство поручило разработку новых средств вычислительной техники и С А Лебедев приглашается заведовать лабораторией № 1 (1951 г) Когда БЭСМ была готова (1953 г), она ничуть не уступала новейшим американским образцам

С 1953 г до конца своей жизни С А Лебедев был директором ИТМ и ВТ АН СССР, избран действительным членом АН СССР и возглавил работы по созданию нескольких поколений ЭВМ

В начале 60-х гг создается первая ЭВМ из серии больших электронных счетных машин (БЭСМ) - БЭСМ-1 При создании БЭСМ 1 были применены оригинальные научные и конструкторские решения Благодаря этому она была тогда самой производительной машиной в Европе (8—10 тысяч операций в секунду) и одной из лучших в мире Под руководством С А Лебедева были созданы и внедрены в производство еще две ламповые ЭВМ - БЭСМ 2 и М-20 В 60 х гг были созданы полупроводниковые варианты М-20 М-220 и М-222, а также БЭСМ-ЗМ и БЭСМ-4

При проектировании БЭСМ-6 впервые был применен метод предварительного имитационного моделирования (сдача в эксплуатацию была осуществлена в 1967 г)

С А Лебедев одним из первых понял огромное значение совместной работы математиков и инженеров в создании вычислительных систем По инициативе С А Лебедева все схемы БЭСМ-6 были записаны формулами булевой алгебры Это открыло широкие возможности для автоматизации проектирования и подготовки монтажной и производственной документации

IBM

IBM/360

Невозможно пропустить ключевой этап в развитии вычислительных средств и методов, связанных с деятельностью фирмы IBM Исторически первые ЭВМ классической структуры и состава - Computer Installation System/360 (фирменное наименование - «Вычислительная установка системы 360», в дальнейшем известная как просто IBM/360) были выпущены в 1964 г и с последующими модификациями (IBM/370, IBM/375) поставлялись вплоть до середины 80 х гг , когда под влиянием микроЭВМ (ПК) не начали постепенно сходить со сцены ЭВМ данной серии послужили основой для разработки в СССР и странах членах СЭВ так называемой Единой системы ЭВМ (ЕС ЭВМ), которые в течение нескольких десятилетий являлись основой отечественной компьютеризации

Продолжение табл 1 1

ЕС 1045

Машины включали следующие компоненты

• центральный процессор (32 разрядный) с двухадресной системой команд

• главную (оперативную) память (от 128 Кбайт до 2 Мбайт),

• накопители на магнитных дисках (НМД, МД) со сменными пакетами дисков (например, IBM-2314 - 7,25 Мбайт, IBM-2311 - 29 Мбайт, IBM 3330 - 100 Мбайт), аналогичные (иногда совместимые) устройства известны и для других из вышеупомянутых серий,

• накопители на магнитных лентах (НМЛ, МЛ) катушечного типа, ширина ленты 0,5 дюйма, длина от 2400 футов (720 м) и менее (обычно 360 и 180 м), плотность записи от 256 байт на дюйм (обычная) и большая в 2-8 раз (повышенная) Соответственно рабочая емкость накопителя определялась размером катушки и плотностью записи и достигала 160 Мбайт на бобину МЛ,

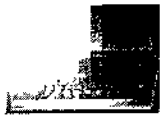
• устройства печати - построчные печатающие устройства барабанного типа, с фиксированным (обычно 64 или 128 знаков) набором символов, включающих прописную латиницу и кириллицу (либо прописную и строчную латиницу) и стандартное множество служебных символов, вывод информации осуществлялся на бумажную ленту шириной 42 или 21 см со скоростью до 20 строк/с,

• терминальные устройства (видеотерминалы, а первоначально - электрические пишущие машинки), предназначенные для интерактивного взаимодействия с пользователем (IBM 3270, DEC VT-100 и пр), подключаемые к системе для выполнения функций управления вычислительным процессом (консоль оператора - 1 -2 шт на ЭВМ) и интерактивной отладки программ и обработки данных (терминал пользователя - от 4 до 64 шт на ЭВМ)

Перечисленные стандартные наборы устройств ЭВМ 60-80 х гг и их характеристики приведены здесь как историческая справка для читателя, который может их самостоятельно оценить, сравнив с современными и известными ему данными

Фирмой IBM была предложена в качестве оболочки ЭВМ IBM/360 первая функционально полноценная ОС - OS/360 Разработка и внедрение ОС позволили разграничить функции операторов, администраторов, программистов, пользователей, а также существенно (в десятки и сотни раз) повысить производительность ЭВМ и степень загрузки технических средств Версии OS/360/370/375 - MFT (мультипрограммирование с фиксированным количеством задач), MVT (с переменным количеством задач), SVS (система с виртуальной памятью), SVM (система виртуальных машин) - последовательно сменяли друг друга и во многом определили современные представления о роли ОС

Окончание табл 1.1

 Пол Аллен  Альтаир	В 1974 г Фирма Intel разработала первый универсальный 8 разрядный микропроцессор 8080 с 4500 транзисторами Эдвард Роберте, молодой офицер ВВС США, инженер-электронщик построил на базе процессора 8080 микрокомпьютер Альтаир имевший огромный коммерческий успех продававшийся по почте и широко использовавшийся для домашнего применения В 1975 г молодой программист Пол Аллен и студент Гарвардского университета Билл Гейтс реализовали для Альтаира язык Бейсик Впоследствии они основали фирму Майкрософт (Microsoft)
 Стивен Джобе и Стивен Возняк  Apple-1  Lisa	В 1976 г студенты Стив Возняк и Стив Джобе, устроив мастерскую в гараже, реализовали компьютер Apple-1, положив начало корпорации Apple 1983 г - корпорация Apple Computers построила персональный компьютер Lisa - первый офисный компьютер, управляемый манипулятором «мышь»

1.2. Классы вычислительных машин

Электронная вычислительная машина (ЭВМ), компьютер — комплекс технических средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач.

ЭВМ можно классифицировать по ряду признаков, в частности:

- физическому представлению обрабатываемой информации;
- поколениям (этапам создания и элементной базе).
- сферам применения и методам использования (а также размерам и вычислительной мощности).

Физическое представление обрабатываемой информации

Здесь выделяют аналоговые (непрерывного действия); цифровые (дискретного действия); гибридные (на отдельных этапах обработки используются различные способы физического представления данных).

АВМ — аналоговые вычислительные машины, или вычислительные машины непрерывного действия, работают с информацией, представленной в непрерывной (аналоговой) форме, т. е. в виде непрерывного ряда значений какой-либо физической величины (чаше всего электрического напряжения):

ЦВМ — цифровые вычислительные машины, или вычислительные машины дискретного действия, работают с информацией, представленной в дискретной, а точнее, цифровой форме В силу универсальности цифровой формы представления информации ЭВМ является более универсальным средством обработки данных.

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

Поколения ЭВМ

Идея делить машины на поколения вызвана к жизни тем, что за время короткой истории своего развития компьютерная техника проделала большую эволюцию как в смысле элементной базы (лампы, транзисторы, микросхемы и др), так и в смысле изменения ее

структуры, появления новых возможностей, расширения областей применения и характера использования (табл 1.2).

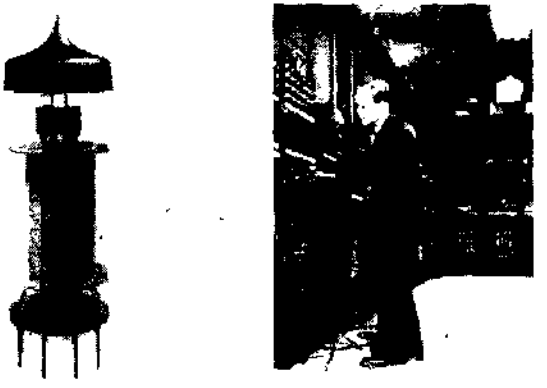
Таблица 1 2 **Этапы развития компьютерных информационных технологий**

Параметр	Период, годы				
	50-е	60-е	70-е	80-е	Настоящее время
Цель использования компьютера	Научно-технические расчеты	Технические и экономические расчеты	Управление и экономические расчеты	Управление, предоставление информации	Телекоммуникации, информационное обслуживание
Режим работы компьютера	Однопрограммный	Пакетная обработка	Разделение времени	Персональная работа	Сетевая обработка
Интеграция данных	Низкая	Средняя	Высокая	Очень высокая	Сверхвысокая
Расположение пользователя	Машинный зал	Отдельное помещение	Терминальный зал	Рабочий стол	Произвольное мобильное
Тип пользователя	Инженеры-программисты	Профессиональные программисты	Программисты	Пользователи с общей компьютерной подготавкой	Мало обученные пользователи
Тип диалога	Работа за пультом компьютера	Обмен перфоносителями и машинограммами	Интерактивный (через клавиатуру и экран)	Интерактивный с жестким меню	Интерактивный экранный типа «вопрос - ответ»

К первому поколению обычно относят машины, созданные на рубеже 50-х гг. и базирующиеся на электронных лампах Эти компьютеры были огромными, неудобными и слишком дорогими машинами, которые могли приобрести только крупные корпорации и правительства. Лампы потребляли значительное количество электроэнергии и выделяли много тепла (рис. 1.1).

Набор команд был ограничен, схемы арифметико-логического устройства и устройства управления достаточно просты, программное обеспечение практически отсутствовало Показатели объема оперативной памяти и быстродействия были низкими. Для ввода-вывода использовались перфоленты, перфокарты, магнитные ленты и печатающие устройства Быстродействие порядка 10—20 тыс. операций в секунду

Программы для этих машин писались на языке конкретной машины Математик, составивший программу, садился за пульт управ-



а *б*

Рис. 1.1. Электронная лампа (*а*), компьютер «Эниак» *(б).* Первое поколение

ления машины, вводил и отлаживал программы и производил по ним счет. Процесс отладки был весьма длительным по времени.

Несмотря на ограниченность возможностей эти машины позволили выполнить сложнейшие расчеты, необходимые для прогнозирования погоды, решения задач атомной энергетики и др.

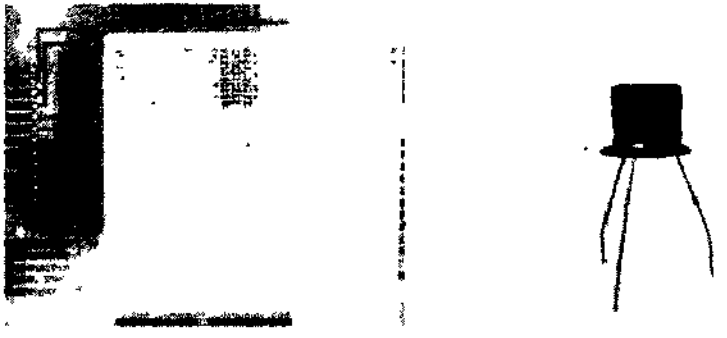
Опыт использования машин первого поколения показал, что существует огромный разрыв между временем, затрачиваемым на разработку программ, и временем счета. Эти проблемы начали преодолевать путем интенсивной разработки средств автоматизации программирования, создания систем обслуживающих программ, упрощающих работу на машине и увеличивающих эффективность ее использования. Это, в свою очередь, потребовало значительных изменений в структуре компьютеров, направленных на то, чтобы приблизить ее к требованиям, возникшим из опыта эксплуатации компьютеров.

Отечественные машины первого поколения: МЭСМ (малая электронная счетная машина), БЭСМ, Стрела, Урал, М-20.

Второе поколение компьютерной техники — машины, сконструированные в 1955—65 гг. Характеризуются использованием в них как электронных ламп, так и дискретных транзисторных логических элементов (рис. 1.2). Их оперативная память была построена на магнитных сердечниках. В это время стал расширяться диапазон применяемого оборудования ввода-вывода, появились высокопроизводительные устройства для работы с магнитными лентами (НМЛ), магнитные барабаны (НМБ) и первые магнитные диски (табл. 1.3).

Таблица 1.3. **Основные характеристики отечественных ЭВМ второго поколения**

Параметр	Первая очередь					Вторая очередь			
	Раздан-2	БЭСМ-4	М-220	Урал-11	Минск-22	Урал-16	Минск-32	М-222	БЭСМ-6
Адресность	2	3	3	1	2	1	1 и2	3	1
Форма представления данных	С плавающей запятой	С плавающей запятой	С плавающей запятой	С фиксированной запятой, символьная	С фиксированной запятой, символьная	С плавающей и фиксированной запятой, символьная	С плавающей и фиксированной запятой, символьная	С плавающей запятой, символьная	С плавающей запятой, символьная
Длина машинного слова (дв разр)	36	45	45	24	37	48	37	45	48
Быстродействие (оп /с)	5 тыс	20 тыс	20 тыс	14–15тыс	5 тыс	100тыс	До 65 тыс	27 тыс	1 млн
ОЗУ, тип, емкость (слов)	Ферритовый сердечник 2048	Ферритовый сердечник 8192	Ферритовый сердечник 4096- 16 384	Ферритовый сердечник 4096– 16 384	Ферритовый сердечник 8192	Ферритовый сердечник 8192– 65 536	Ферритовый сердечник 16 384– 65 636	Ферритовый сердечник 16 384– 32 768	Ферритовый сердечник 32768-131 071
ВЗУ, тип, емкость (слов)	НМЛ 120 тыс	НМЛ 8 млн	НМЛ 16 млн	НМЛ 8 млн	НМЛ до 5 млн	НМЛ 12 млн НМБ 130 тыс	НМЛ до 16 млн	НМЛ до 32 млн НМБ до 192 тыс	НМЛ 32 млн НМБ 512 тыс



а б

Рис. **1.2.** Память на магнитных сердечниках *(а),* транзистор *(б)*

Эти машины характеризуются быстродействием до сотен тысяч операций в секунду, емкостью памяти — до нескольких десятков тысяч слов

Появляются *языки высокого уровня,* средства которых допускают описание всей необходимой последовательности вычислительных действий в наглядном, легко воспринимаемом виде

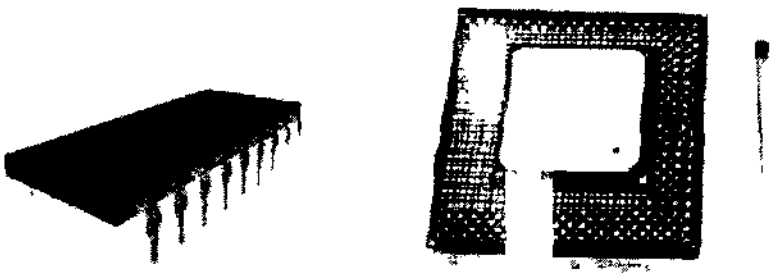
Программа, написанная на алгоритмическом языке, непонятна компьютеру, воспринимающему только язык своих собственных команд Поэтому специальные программы, которые называются *трансляторами,* переводят программу с языка высокого уровня на машинный язык.

Появился широкий набор библиотечных программ для решения разнообразных задач, а также мониторные системы, управляющие режимом трансляции и исполнения программ, из которых в дальнейшем выросли современные операционные системы

Операционная система — важнейшая часть программного обеспечения компьютера, предназначенная для автоматизации планирования и организации процесса обработки программ, ввода-вывода и управления данными, распределения ресурсов, подготовки и отладки программ, других вспомогательных операций обслуживания

Машинам второго поколения была свойственна программная несовместимость, которая затрудняла организацию крупных информационных систем Поэтому в середине 60-х гг наметился переход к созданию компьютеров, программно совместимых и построенных на микроэлектронной технологической базе

Машины третьего поколения — это семейства машин с единой архитектурой, т е программно совместимых В качестве элемент-



а б

Рис. **1.3.** Одна из первых интегральных схем (*а*) Микропроцессор Pentium 4 (б)

ной базы в них используются интегральные схемы, которые также называются микросхемами (рис 1 3, *а)*

Машины третьего поколения появились в 60-е гг Поскольку процесс создания компьютерной техники шел непрерывно, и в нем участвовало множество людей из разных стран, имеющих дело с решением различных проблем, трудно и бесполезно пытаться установить, когда «поколение» начиналось и заканчивалось Возможно, наиболее важным критерием различия машин второго и третьего поколений является критерий, основанный на понятии архитектуры.

Машины третьего поколения имеют развитые операционные системы Они обладают возможностями мультипрограммирования, т. е параллельного выполнения нескольких программ. Многие задачи управления памятью, устройствами и ресурсами стала брать на себя операционная система или же непосредственно сама машина

Примеры машин третьего поколения — семейства IBM-360, IBM-370, PDP-11, VAX, ЕС ЭВМ (Единая система ЭВМ), СМ ЭВМ (Семейство малых ЭВМ) и др

Быстродействие машин внутри семейства изменяется от нескольких десятков тысяч до миллионов операций в секунду Емкость оперативной памяти достигает нескольких сотен тысяч слов

Четвертое поколение — это основной контингент современной компьютерной техники, разработанной после 70-х гг

Наиболее важный в концептуальном отношении критерий, по которому эти компьютеры можно отделить от машин третьего поколения, состоит в том, что машины четвертого поколения проектировались в расчете на эффективное использование современных высокоуровневых языков и упрощение процесса программирования для конечного пользователя.

В аппаратурном отношении для них характерно широкое использование интегральных схем в качестве элементной базы, а также наличие быстродействующих запоминающих устройств с произвольной выборкой емкостью в десятки мегабайт (рис. 1.3, б).

С точки зрения структуры машины этого поколения представляют собой многопроцессорные и многомашинные комплексы, использующие общую память и общее поле внешних устройств. Быстродействие составляет до нескольких десятков миллионов операций в секунду, емкость оперативной памяти порядка 1—512 Мбайт.

Для них характерны:

- применение персональных компьютеров (ПК);
- телекоммуникационная обработка данных;
- компьютерные сети;
- широкое применение систем управления базами данных;
- элементы интеллектуального поведения систем обработки данных и устройств.

В компьютерах пятого поколения предположительно должен произойти качественный переход от обработки *данных* к обработке *знаний*.

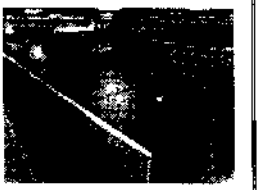
Архитектура компьютеров будущего поколения будет содержать два основных блока. Один из них — это традиционный компьютер, однако лишенный связи с пользователем. Эту связь осуществляет *интеллектуальный интерфейс*. Будет также решаться проблема децентрализации вычислений с помощью компьютерных сетей.

Сферы применения и методы использования

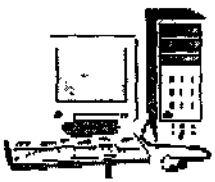
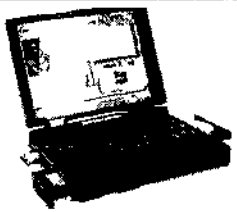
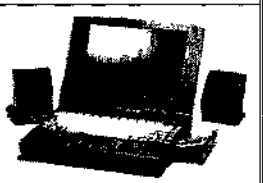
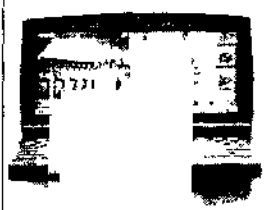
Здесь ЭВМ можно разделить на следующие группы (рис 1 4, табл. 1.4).

Суперкомпьютер (supercomputer) — предназначен для высокоскоростного выполнения прикладных процессов. В 1976 г. корпорация Cray Research изготовила первый сверхбыстродействующий компьютер, образовав новый класс компьютеров. Первоначально Cray Research предполагала, что потребность в таких компьютерах будет небольшой, однако она увеличивается, и особенно в последние годы. Кроме этого, производители суперкомпьютеров постоянно улучшали показатель стоимость/производительность. Появился и получил большую популярность новый класс — супермини-компьютеры. Это уменьшенные по габаритам и более экономичные варианты суперкомпьютеров, нередко настольного исполнения.

Таблица 1 4 **Классификация ЭВМ по производительности и габаритным характеристикам**

Класс ЭВМ	Основное назначение	Основные технические данные	Общий вид ЭВМ
СуперЭВМ, суперкомпьютер, вычислительная система (ВС)	Предназначен для высокоскоростного выполнения прикладных процессов	Имеет скалярные и векторные процессоры Совместная работа процессоров основывается на различных архитектурах	 Columbia
Супер-миниЭВМ	Многопультовые вычислительные системы	Мультипроцессорная архитектура, позволяющая подключение до нескольких сот терминалов (наличие наращиваемых дисковых запоминающих устройств)	 HP LD PRO
Большие ЭВМ (мэйнфреймы - mainframe)	Обработка больших объемов данных крупных предприятий и организаций	Мультипроцессорная архитектура, позволяющая подключение нескольких сот рабочих мест	 IBM-360
Мини-ЭВМ	Системы управления предприятиями	Однопроцессорная архитектура, разветвленная система периферийных устройств (ограниченные возможности, обработка слов меньшей длины и т д)	 VAX

Окончание табл 1 4

Класс ЭВМ	Основное назначение	Основные технические данные	Общий вид ЭВМ
Рабочие станции	Системы автоматизированного проектирования, системы автоматизации эксперимента, индустриальные процессы и др	Высокое быстродействие процессора, емкость оперативного запоминающего устройства 32-64 Мбайт, специализированная система периферийных устройств	
МикроЭВМ, настольный персональный компьютер (ПК)	Индивидуальное обслуживание пользователей	Центральный блок с одним или несколькими процессорами, монитор, акустическая система, клавиатура, электронное перо с планшетом, устройство ввода информации, принтеры, жесткие диски, гибкие диски, магнитные ленты, оптические диски и пр	
Переносной ПК «наколенник» (laptop)	То же	Малогабаритный книжного размера портативный вариант стационарного персонального компьютера	
Блокнотный ПК, ноутбук (notebook)	- » -	Модели могут иметь процессор Pentium, оперативную память до 96 Мбайт жесткий диск до 9 Гбайт, встроенные компакт-диск и факс-модем, дисплей жидкокристаллический, время работы от собственного источника питания от 2 до 8 ч	
Карманный компьютер «наладонник» (palmtop)	- » -	Оперативная память выполняет функцию долговременной памяти, размером в несколько Мбайт Жесткий диск отсутствует Работает под управлением Windows CE, имеет интерфейс с другими компьютерами, встроенные интегрированные системы, жидкокристаллический дисплей	

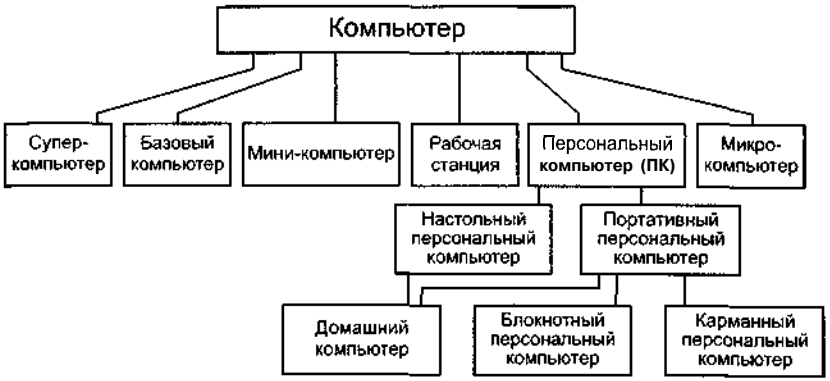


Рис. 1.4. Классификация по сферам применения и методам использования

Суперкомпьютер может иметь один процессор, и тогда в нем одна последовательность команд работает с одним потоком данных. Вместе с этим большие скорости обработки данных можно получить лишь в многопроцессорных системах. Поэтому во всех последующих архитектурах степень параллельной обработки возрастает. Растет соответственно и число входящих в суперкомпьютер процессоров В дополнение к обычным (скалярным) подключаются векторные процессоры. В первом случае обрабатываются скалярные величины, а во втором — векторные.

Внедрение суперкомпьютеров долго сдерживалось отсутствием развитого программного обеспечения. В настоящее время ситуация изменяется, появились языки, предназначенные для параллельной обработки, все больше предлагается эффективных операционных систем. Суперкомпьютеры выпускаются значительным числом фирм. Корпорация IBM создала суперкомпьютер в одном кристалле интегральной схемы (ИС).

Базовый (большой) компьютер — mainframe — основной тип компьютера, используемый в больших информационных сетях, работает с большой скоростью и по производительности уступает суперкомпьютеру, но охватывает более широкий круг решаемых задач. С другой стороны, он превосходит мини-компьютер по скорости работы и сложности выполняемых прикладных процессов. Мэйнфрейм обладает относительно большой оперативной памятью и предоставляет свои ресурсы через коммуникационную сеть большому числу пользователей. Вследствие сказанного, базовые компьютеры принимают на себя основные потоки обработки данных. Нередко под базовым компьютером понимают лишь центральную часть крупного

компьютера, включающую процессоры и оперативное запоминающее устройство.

В связи с развитием архитектуры клиент—сервер базовые компьютеры стали нередко использоваться в качестве серверов. Интеграция средств распределенной обработки данных с большими базовыми компьютерами обеспечивает эффективную обработку данных в корпоративных сетях. Базовые компьютеры не только функционируют как крупные серверы, но обеспечивают автоматизацию процессов, протекающих в сети.

Мини-компьютер — minicomputer — компьютер с ограниченными возможностями обработки данных. По сравнению с базовым компьютером мини-компьютер работает со словами меньшей длины, имеет ограниченную оперативную память и относительно небольшое быстродействие. Поэтому мини-компьютер используется для решения более простых задач, чем базовый. Но по сравнению с последним мини-компьютер имеет небольшую стоимость, размеры и проще в эксплуатации. Термин «мини-компьютер» появился тогда, когда не было персональных компьютеров. Теперь же существуют такие персональные компьютеры, которые превосходят даже базовые компьютеры восьмидесятых годов. Поэтому рассматриваемый термин применяется все реже, уступая понятиям *рабочая станция* и *персональный компьютер.*

Рабочая станция — workstation — абонентская система, специализированная на выполнение определенных задач пользователя.

Первая рабочая станция, названная Сетевым изделием Станфордского университета (SUN), была создана корпорацией SUN Microsystems, под девизом «сеть есть компьютер». Это связано с тем, что рабочая станция в своей основе предназначена для работы в информационной сети. Рабочая станция, нередко именуемая рабочим местом, создается на базе малого, но достаточно мощного настольного либо напольного компьютера. Для этого разрабатывается архитектура рабочей станции, подбираются необходимые устройства (процессоры, запоминающие устройства, графопостроители, принтеры и т. д.). Создается нужное программное обеспечение, станция включается в сеть. Она предназначается для любого специалиста (программиста, системотехника, менеджера, исследователя и др.). Рабочая станция занимает среднее место среди компьютеров и характеризуется *многозадачностью* — режимом, при котором пользователь может запускать несколько задач. Это позволяет выполнять группу прикладных процессов.

Важное значение в архитектуре рабочей станции имеет визуализация информации. Она заключается в создании условий для фор-

мирования и обработки изображений. Это позволяет пользователям не только удобно отображать на экране физические объекты, но также строить модели, манипулировать ими и наблюдать ход экспериментов в реальном масштабе времени.

Все большее распространение получают рабочие станции с комбинированным сервисом. Они имеют широкий набор устройств связи с внешней средой:

- измерительные приборы;
- устройства видеоввода и микрофоны;
- оптические диски;
- клавиатуры и сенсорные устройства.

В рабочих станциях часто используются графические акселераторы (платы, содержащие процессоры, специализирующиеся по обработке изображений). Акселераторы выполняют геометрические преобразования двухмерных изображений в трехмерные изображения, учитывая ряд сложных требований, таких, например, как движущиеся источники света на экранах, отображение особенностей структуры поверхностей объектов. Акселераторы повышают стоимость станций, но резко увеличивают скорость выполнения ими сложных прикладных процессов. Широкую известность получили рабочие станции корпораций SUN Microsystems и Silicon Graphics.

Микрокомпьютер (microcomputer) — устройство, созданное на основе одного либо нескольких микропроцессоров. Существует два подхода к определению микрокомпьютера. Первый из них заключается в том, что под микрокомпьютером понимается одна либо несколько сверхбольших интегральных схем. Для этого схемы должны содержать все логические элементы, необходимые для получения полноценного компьютера небольшой производительности. Во втором подходе микрокомпьютером называется любой компьютер, в котором основными компонентами являются микропроцессоры. В дальнейшем эти ЭВМ стали именовать персональными компьютерами (ПК). В этой связи под микрокомпьютером чаще всего понимают устройство, созданное на одной либо группе интегральных схем. Для этой цели нередко бескорпусные интегральные схемы группируются в одном корпусе. Что же касается высокопроизводительного персонального компьютера, то в нем может использоваться группа микрокомпьютеров. Микрокомпьютеры также широко используются в технологии производства и в разнообразной аппаратуре автоматического управления.

Персональный компьютер (ПК) — personal computer (PC) — недорогой компьютер, созданный на базе микропроцессора. ПК или персональные электронные вычислительные машины (ПЭВМ) в

ряду компьютеров характеризуются небольшими размерами и массовым производством. Это позволяет делать их широкодоступным товаром, обеспечивающим обработку различной информации. ПК предназначены для обработки текстов, звука и изображений.

Персональный компьютер для удовлетворения требований общедоступности и универсальности применения должен обладать такими качествами, как:

- малая стоимость, находящаяся в пределах доступности для индивидуального покупателя;
- автономность эксплуатации без специальных требований к условиям окружающей среды;
- гибкость архитектуры, обеспечивающая адаптируемость к разнообразным применениям в сфере управления, науки, образования и в быту;
- дружественность операционной системы и прочего программного обеспечения, обусловливающая возможность работы пользователя без специальной профессиональной подготовки;
- высокая надежность работы (более 5000 ч на отказ).

ПК делятся на несколько классов. Если за признак классификации взять «тип решаемых на ПК задач», то IBM-совместимые ПК могут быть разделены на: серверы; графические станции; портативные; ПК для корпоративных пользователей; ПК для дома и малого офиса (SOHO — Small Office, Home Office). По определенным характеристикам самих ПК, например, IBM-совместимые компьютеры можно поделить на классы: PC XT — extended Technology (8-разрядный ПК, у которого предусмотрен жесткий диск, шина ESA и т. д.); AT — Advanced Technology (16-, 32-, 64-разрядные ПК с шинами ISA (16), EISA (16/32), MCA (16/32), PCMCIA, VL-bus (16/32/64), PCI (32/64)).

Настольные персональные компьютеры являются стационарными и предоставляют наибольшие возможности их пользователям. Портативные персональные компьютеры имеют небольшие размеры. Особенно важно, что они транспортабельны, и с таким компьютером можно работать, находясь в самолете, поезде либо автомобиле. Управление ПК осуществляется с помощью клавиатуры, мыши или светового пера. Информация вводится с клавиатуры, сканера, микрофона или камеры. Вывод информации осуществляется на экран, динамик или принтер.

В последние годы использование высокоскоростных 32- и 64-разрядных микропроцессоров и версий операционной системы UNIX привело к слиянию ПК с рабочими станциями. С другой стороны, создаются устройства, в которых объединяются функции пер-

сонального компьютера с телевизором и телефонным аппаратом. Такое устройство, например, предложено корпорацией Microsoft. Это — простой интерактивный персональный компьютер (Simple Interactive Personal Computer — SIPC). Он несложен в эксплуатации, легко соединяется с телевизионной или телефонной сетью.

Теоретической основой организации и функционирования вычислительных машин и систем являются следующие дисциплины:

- элементы теории чисел и вычислительной математики — системы счисления, представление чисел в различных системах, операции над числами, точность представлений и результатов операций;
- элементы математической логики — логические выражения и переменные, операции над ними, эквивалентные преобразования выражений, схемные элементы, узлы и переключательные схемы ЭВМ, базирующиеся на подобных преобразованиях;
- элементы теории алгоритмов *(алгорифмов* «по-научному») — циклические, ветвящиеся, итерационные процессы, их свойства (эффективная вычислимость), зависимость точности вычислений от их длительности и пр.;
- элементы теории информации — измерение количества информации, пропускная способность каналов, «сигнал—шум», децибеллы, кодирование и сжатие—восстановление информации;
- другие разделы прикладной математики, в частности — теория графов, топологические преобразования конфигураций сетей и пр.

Очевидно, что мы не можем в ограниченных рамках данного учебного пособия подробно осветить указанные вопросы, однако далее мы приводим краткое изложение основных положений некоторых из этих дисциплин, избегая ненужную формализацию.

1.3. Информация, кодирование, обработка в ЭВМ

Понятие «информация» является таким же фундаментальным, как понятия «материя», «энергия» и другие философские категории. Это атрибут, свойство сложных систем, связанное с их развитием и самоорганизацией [24, 25]. Известно большое количество различных определений информации, отличие информации от данных, знаний и пр. Мы здесь ограничимся только рассмотрением некоторых практически важных понятий и определений.

Определение и классификация информации

В настоящее время наука пытается найти общие свойства и закономерности, присущие многогранному понятию информация, но пока это понятие во многом остается интуитивным и получает различные смысловые наполнения в различных отраслях человеческой деятельности

- в обиходе информацией называют *любые данные или факты,* которые кого-либо интересуют. Например, сообщение о каких-либо событиях, о чьей-либо деятельности и т. п «Информировать» в этом смысле означает «сообщить нечто, неизвестное раньше»;
- в технике под информацией понимают *сообщения,* передаваемые в форме знаков или сигналов;
- в кибернетике под информацией понимают ту часть *знаний,* которая используется для ориентирования, активного действия, управления, т. е. в целях сохранения, совершенствования, развития системы.

Приведем несколько определений информации:

- отрицание энтропии (Леон Бриллюэн);
- мера сложности структур (Моль);
- отраженное разнообразие (Урсул),
- содержание процесса отражения (Тузов);
- вероятность выбора (Яглом);
- снятая неопределенность наших знаний о чем-то (Клод Шеннон);
- обозначение содержания, полученного из внешнего мира в процессе нашего приспособления к нему и приспособления к нему наших чувств (Н Винер).

Информация может классифицироваться, например, по следующим основаниям.

а) признаки, отражающие структуру данных и форму представления информации (табл. 1 5);

б) содержание предметной области применения (табл 1.6)

Исторически первой технологической формой получения, передачи, хранения информации являлось *аналоговое* (непрерывное) представление звукового, оптического, электрического или другого сигнала (сообщения) Магнитная аудио- и видеозапись, фотографирование, запись на шеллачные или виниловые грампластинки, проводное и радиовещание — основные способы хранения и передачи информации в аналоговой форме (рис 1 5) Заметим, что с начала 50-х гг (а во многом и сейчас) под термином *теория*

Таблица ! **5 Некоторые классы информации (по структуре и форме)**

Основание для классификации	Классы информации			
По уровням сложности	Сигнал	Сообщение, документ	Информационный массив	Информационный ресурс
По типу сигнала	Аналоговая (непрерывная)	Цифровая (дискретная)		
По уровням доступа и организации	Данные в регистровой памяти	Данные в оперативной памяти	Файлы данных на внешних устройствах	Базы данных
По способам кодирования и представления (данные, файлы и БД)	Цифровая (вычислительные данные, двоичные)	Символьная (алфавитно-цифровая, строчная)	Графическая	
По организации данных (файлы и БД)	Табличная	Текстовая	Графическая	

Таблица 1 б **Классификация информации по содержанию**

Тип информации	Содержание	Поставщик содержания
Биржевая и финансовая	Индексы рынка, котировки, цены, обзоры	Биржи, банки, службы финансовой информации
Экономическая и демографическая статистика	Первичная и вторичная, национальная, региональная статистика	Переписи, опросы, аналитические исследования
Коммерческая	Данные о предприятиях, товарах, услугах	Аналитические службы
Деловые новости	Состояние рынка, события в области экономики	Службы фильтрации, агентства новостей
Научно-техническая	Фундаментальные, прикладные науки	Центры НТИ, издательства, библиотеки
Правовая	Нормативно-правовые акты	Законодательные органы, Минюст РФ
Медицинская	Медучреждения, болезни, лекарства, яды	Информационные центры, библиотеки, госпитали
Потребительская и развлекательная	Образование, музыка, музеи, библиотеки, кино	Справочные службы, учреждения
Бытовая	Погода, туризм, справочники	Информационные службы

информации подразумевались теоретические методы, связанные с обеспечением как можно более точного приема, передачи, записи, воспроизведения, преобразования непрерывных сигналов (основные понятия — *линейность, нелинейность, шум, спектр сигнала, полоса пропускания и пр)*

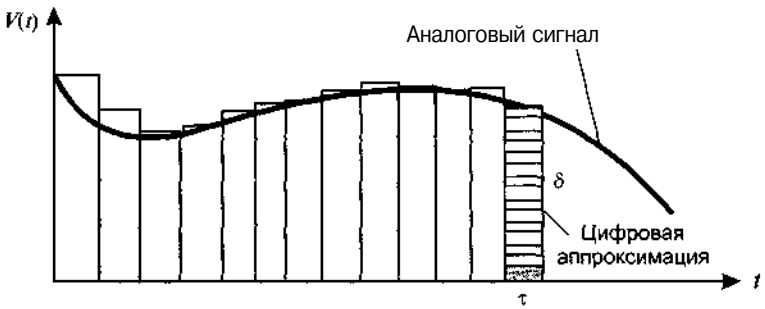


Рис. 1.5. Аналоговый сигнал и его дискретная (цифровая) аппроксимация (оцифровка)

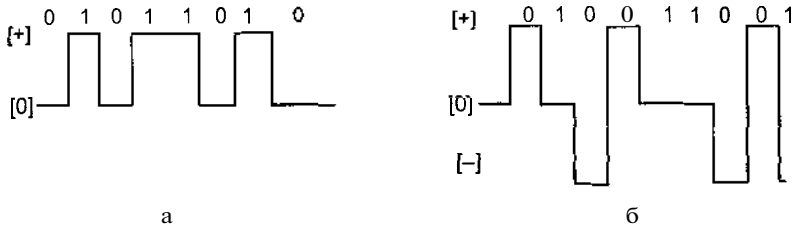
Аналого-цифровое (дискретное) преобразование — АЦП (analog-to-digital conversion) заключается в формировании последовательностей *n*-разрядных двоичных слов, представляющих с заданной точностью аналоговые сигналы. Для выполнения этого преобразования вначале осуществляется *квантование* аналогового сигнала. В результате преобразования получается дискретный сигнал. Наименьшее изменение аналогового сигнала, которое регистрируется устройством, осуществляющим преобразование, называется *разрешением*.

Аналого-дискретные преобразователи чаще всего изготавливаются в виде интегральных схем. В необходимых случаях осуществляется обратное — *дискретно-аналоговое (цифро-аналоговое преобразование — ЦАП).*

Дискретный сигнал — сигнал, имеющий конечное, обычно небольшое, число значений

Практически всегда дискретный сигнал имеет два либо три значения. Нередко его называют также *цифровым сигналом*

В цифровых системах используются двоичные сигналы (рис. 1.6, а), имеющие значения (+), (-). Вместе с тем при передаче данных в большинстве случаев применяются троичные сигналы (рис. 1 6, *б)* со значениями (+), (0), (–) Здесь «единица» представляется отсутствием потенциала в канале, тогда как «нуль» характеризуется положительным либо отрицательным импульсом. При этом полярность импульсов, представляющих «нули», должна чередоваться, т. е. за положительным (+) импульсом должен следовать отрицательный (-) и наоборот В форме троичного сигнала осуществляется не только кодирование передаваемых данных, но также обеспечивается синхронизация работы канала и проверка целостности данных



а б

Рис. **1.6.** Примеры дискретных сигналов двоичный (а) и троичный (б)

Дискретные сигналы по сравнению с аналоговыми имеют ряд важных преимуществ: помехоустойчивость, легкость восстановления формы, простота аппаратуры передачи.

Более чем тридцатилетнее развитие теории и практики ЭВМ приводит к вытеснению (в том числе и на бытовом уровне) аналоговых устройств и сигналов цифровыми. Наиболее популярным примером является несомненно аудиокомпакт-диск (digital audio CD).

В этом случае звуковой сигнал (рис. 1.5) сначала преобразуется в дискретную аппроксимацию («многоуровневый ступенчатый сигнал»), при этом происходит *квантование во времени*, которое заключается в измерении в дискретные промежутки времени необходимого параметра аналогового сигнала.

Кроме этого, осуществляется *квантование по амплитуде* сигнала. Элемент разбиения этого сигнала именуют квантом. Поэтому говорят, что квантование заключается в делении на кванты. При квантовании аналогового сигнала происходит округление его мгновенных значений до некоторой заданной фиксированной величины, называемой *уровнем.* Расстояние между соседними уровнями именуется шагом Из-за округления квантование всегда связано с определенным искажением сигнала. Уменьшение искажения требует увеличения числа уровней квантования и уменьшения шага квантования

При *квантовании по амплитуде* каждая ступенька представляется последовательностью бинарных двухуровневых цифровых сигналов. Принятый в настоящее время стандарт CD использует так называемый «16-разрядный звук с частотой сканирования 44 кГц». Для рис 1.5 в переводе на нормальный язык это означает, что «длина ступеньки» (т) равна 1/44 000 с, а «высота ступеньки» (§) составляет 1/65 536 от максимальной громкости сигнала (поскольку $2^{16} = 65\ 536$). При этом частотный диапазон воспроизведения составляет 0—22 кГц, а динамический диапазон — 96 децибел (что составляет совершенно недостижимую для магнитной или механической звукозаписи характеристику качества).

Измерение количества информации

Термин «информация» имеет корень «form» (форма), что разумно трактовать как «информирование — придание формы, вывод из состояния неопределенности, бесформенности», поэтому логично подходить к определению понятия «количество информации», исходя из того, что информацию, содержащуюся в сообщении, можно трактовать в смысле ее новизны или, иначе, уменьшения неопределенности знаний «приемника информации» об объекте.

Американский инженер Р. Хартли в 1928 г. рассматривал процесс получения информации как выбор одного сообщения из конечного заданного множества из N равновероятных сообщений, а количество информации I, содержащееся в выбранном сообщении, определял как двоичный логарифм N:

$$I = \log_2 N.$$

Допустим, нужно угадать одно число из набора чисел от единицы до ста. По формуле Хартли можно вычислить, какое количество информации для этого требуется:

$$I = \log_2 100 \approx 6{,}644.$$

Таким образом, сообщение о верно угаданном числе содержит количество информации, приблизительно равное 6,644 единицы информации.

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

Определим теперь, 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. Если это, например, станция метро, то вероятность выйти из дверей первым одинакова для мужчины и женщины, а если это военная казарма, то для мужчины эта вероятность значительно выше, чем для женщины.

Для задач такого рода американский ученый К. Шеннон предложил в 1948 г. другую формулу определения количества информации, учитывающую возможную неодинаковую вероятность сообщений в наборе.

Формула Шеннона:

$$I = -(p_1 \log_2 p_1 + p_2 \log_2 P_2 + \cdots + p_N \log_2 p_N) = -\sum_{i=1}^{N} p_i \log_2 p_i,$$

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

Очевидно, что если вероятности p_1, ..., p_N равны, то каждая из них равна $\frac{1}{N}$, и формула Шеннона превращается в формулу Хартли

Помимо двух рассмотренных подходов к определению количества информации, существуют и другие. Важно помнить, что любые теоретические результаты применимы лишь к определенному кругу случаев, очерченному первоначальными допущениями.

В качестве единицы информации Клод Шеннон предложил принять один *бит* (англ. bit — binary digit — двоичная цифра).

Бит в теории информации — количество информации, необходимое для различения двух равновероятных сообщений («орел—решка», «чет—нечет» и т. п.).

В вычислительной технике битом называют наименьшую «порцию» памяти компьютера, необходимую для хранения одного из двух знаков 0 и 1, используемых для машинного представления данных и команд.

За единицу информации можно было бы выбрать количество информации, необходимое для различения, например, десяти равновероятных сообщений. Это будет не двоичная (бит), а десятичная (дит) единица информации.

Поскольку бит — слишком мелкая единица измерения, на практике чаще применяется более крупная единица — байт, равная восьми битам. В частности, восемь бит требуется для того, чтобы закодировать любой из 256 символов основного компьютерного кода ASCII ($256 = 2^8$).

Используются также более крупные производные единицы информации.

Килобайт (Кбайт) = 1024 байт = 2^{10} байт;

Мегабайт (Мбайт) = 1024 Кбайт = 2^{20} байт;

Гигабайт (Гбайт) = 1024 Мбайт = 2^{30} байт.

В последнее время в связи с увеличением объемов обрабатываемой информации входят в употребление такие производные единицы, как:

Терабайт (Тбайт) = 1024 Гбайт = 2^{40} байт;

Петабайт (Пбайт) = 1024 Тбайт = 2^{50} байт;

Экзобайт = 10^{18} Мбайт и пр

Для описания скорости передачи данных можно использовать термин *бод* Число бод равно количеству значащих изменений сигнала (потенциала, фазы, частоты), происходящих в секунду. Первоначально бод использовался в телеграфии. Для двоичных сигналов нередко принимают, что *бод равен биту в секунду,* например

1200 бод = 1200 бит/с. Однако единого мнения о правильности использования этого термина нет, особенно при высоких скоростях, где число бит в секунду не совпадает с числом бод.

Кодирование символьной информации

Код (code) — совокупность знаков, символов и правил представления информации.

В частности, можно различать двоичный и троичный код. Алфавит первого ограничен двумя символами (О, 1), а второго — тремя символами (-1, 0, +1). Сигналы, реализующие коды, обладают одной из следующих характеристик:

- униполярный код (значения сигнала равны 0, +1, либо О, -1);
- полярный код (значения сигнала равны -1, +1);
- биполярный код (значения равны -1, 0, +1).

Биполярные коды часто используются в каналах передачи данных (рис. 1.7). Здесь единицы представляются чередующимися положительными и отрицательными импульсами. Отсутствие импульсов определяет состояние «нуль». Биполярное кодирование обеспечивает обнаружение одиночной ошибки. Так, если вместо нуля появится единица, либо единица ошибочно сменится на нуль, то это легко обнаруживается. В обоих случаях нарушается чередование полярности импульсов.

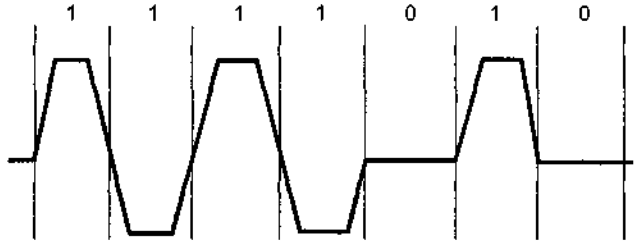


Рис. 1.7. Биполярное кодирование

Рассмотрим методы дискретного представления информации, или кодирования (которые, надо сказать, появились задолго до вычислительных машин). Первым широко известным примером является Азбука Морзе (табл. 1.7), в которой буквы латиницы (или кириллицы) и цифры кодируются сочетаниями из «точек» и «тире». Воспользуемся данным кодом для иллюстрации основных понятий, связанных с кодированием (не вдаваясь в теорию кодирования).

Таблица 1.7. **Фрагменты кода Морзе**

Символ входного алфавита	Мнемоническое обозначение по МСС*	Кодовая (знаковая) комбинация
A	alfa	.-
B	bravo	-...
C	Charlie	-.-.
D	delta	-..
E	echo	.
...	...	...
Y	yankee	-.--
Z	zulu	--..
1	one	.----
...	...	...
9	nine	----.

* Международный Свод Сигналов.

Кодируемые (обозначаемые) элементы входного алфавита обычно называют *символами.*

Символом (служит условным знаком какого-нибудь понятия, явления) как правило, является цифра, буква, знак пунктуации или иероглиф естественного языка, знак препинания, знак пробела, специальный знак, символ операции. Кроме этого, учитываются *управляющие («непечатные») символы.*

Кодирующие (обозначающие) элементы выходного алфавита называются *знаками;* количество различных знаков в выходном алфавите назовем *значностью* (*-арностью*, *-ичностью*); количество знаков в кодирующей последовательности для одного символа — *разрядностью кода;* последовательным кодом является такой, в котором знаки следуют один за другим во времени (например, радио- или оптические сигналы либо передача по двум проводам, 2-жильному кабелю), параллельным — тот, в котором знаки передаются одновременно (например, по четырем проводам, 4-жильному кабелю), образуя символ (т. е. символ передается в один прием, в один момент времени).

Применительно к азбуке Морзе (АМ):

- символами являются элементы языкового алфавита (буквы A—Z или А—Я) и цифровой алфавит (здесь — цифры 0—9);
- знаками — «точка» и «тире» (или «+» и «–» либо «1» и «О», короче — два любых разных знака);
- поскольку знаков два, АМ является *двузначным* (бинарным, двоичным) кодом, если бы их было три, то мы имели бы дело с троичным, тернарным, трехзначным кодом;

- поскольку число знаков в АМ колеблется от 1 (буквы Е, Т) до 5 (цифры), здесь имеет место код с *переменной разрядностью* (в АМ часто встречающиеся в тексте символы обозначены более короткими кодовыми комбинациями, нежели редкие сим волы)

Поскольку знаки передаются последовательно (электрические импульсы, звуковые или оптические сигналы разной длины, соответствующие «точкам» и «тире»), АМ есть *последовательный код* (можно представить себе некоторое табло, на котором одновременно вспыхивали бы сочетания лампочек, образующих точки и тире, представляющие передаваемый символ, но авторам не приходилось слышать о подобных абсурдных приспособлениях)

Первые опыты телеграфной и радиосвязи осуществлялись именно посредством АМ, причем приемное устройство записывало импульсы переменной длины в виде «точек» и «тире» на движущуюся телеграфную ленту, однако уже в начале XX в был осуществлен переход на 5-разрядный (5-битовый) телеграфный код

В табл 1 8 приводится перечень наиболее известных кодов, некоторые из них использовались первоначально для связи, кодирования данных, а затем для представления информации в ЭВМ

- *код Бодо* — 5-разрядный код, бывший в прошлом европейским стандартом для телеграфной связи (другое название — IA-1 — international alphabet #1),
- *М-2 (российское обозначение) или IA-2 (международное обозначение)* — телеграфный код, предложенный Международным Комитетом по телефонии и телеграфии (МККТТ) и заменивший код Бодо,
- *ASCII (American Standard Code for Information Interchange)* — стандартный 7-битовый код для передачи данных, поддерживает 128 символов, включающих заглавные и строчные симво-

Таблица 1 8 **Разрядность некоторых наиболее известных кодов**

Код	Разрядность
IA 2 (M2 MKKTT 2)	5
Baudot (Бодо)	5
ISO *1* (IA 5 ASCII 7 USASCII ANSI X3 4)	7
EBCDIC	8
ASCII 8	8
Hollerith (Перфокарты Холлерита)	12

лы латиницы, цифры, специальные значки и управляющие символы Этот код, к которому были добавлены некоторые национальные символы (10 бинарных комбинаций), был принят Международной организацией по стандартизации (ISO) как стандарт ISO 7,

- *EBCDIC (Expanded Binary Coded Decimal Interchange Code)* — 8-разрядный код, предложенный фирмой IBM для машин серий IBM/360-375 (внутреннее представление данных в памяти), а затем распространившийся и на системы других производителей,
- *ASC1I-8* — 8-разрядный код, принятый для внутреннего и внешнего представления данных в вычислительных системах Включает стандартную часть (128 символов) и национальную (128 символов) Соответственно в зависимости от национальной части, кодовые таблицы различаются (табл 1 9, 1 10, прил 4),
- *код Холлерита,* предложенный для ПК (1913 г), затем использовавшийся для кодирования информации перед вводом в ЭВМ с перфокарт (рис 1 8)

Одним из «последних слов» в процессе развития систем символьного кодирования является универсальный код UNICODE (UNIversal CODE) — стандарт 16-разрядного кодирования символов

Стандарт UNICODE разработан техническим комитетом, в который вошли представители ряда ведущих фирм Он определяет коды, обеспечивающие идентификацию различных символов букв, иероглифов, цифр и т д Код может использоваться вместо 7—8-битовых, в том числе и ASCII Поскольку в 16-разрядном UNICODE

Таблица 1 9 **Некоторые кодовые таблицы**

Наименование кодовой страницы (Code Page)	Интерпретация кодовой страницы
Latin-1	Международный стандарт (ISO 8859 1) для интерпретации 2 й половины (128–256) кода ASCII таблица предназначена для латиницы
Latin 8	Международный стандарт (ISO 8859 8) для иврита
Latin C	Международный стандарт (ISO 8859) для кириллицы
CP 437	Стандарт IBM для интерпретации 2 и половины (128–256) кода ASCII таблица предназначена для греческого алфавита
CP 850	Стандарт IBM для восточноевропейских алфавитов
CP 852	Стандарт IBM для греческого алфавита
CP 862	Стандарт IBM для иврита
CP 866	Стандарт IBM для русской кириллицы

Таблица 1.10. Фрагменты некоторых кодовых таблиц

Символ	IA-2	Бодо	ISO-7	EBCDIC	ASCII-8	Холлерит
A	03	10	41	C1	A1	900
B	19	06	42	C2	A2	880
C	0E	16	43	C3	A3	840
D	09	1E	44	C4	A4	820
a			61	81	E1	
b			62	82	E2	
c			63	83	E3	
d			64	84	E4	
. (точка)	1C	05	2E	4B	4E	842
, (запятая)	0C	09	2C	6B	4C	242
: (двоеточие)	1E		3B	5E	5B	40A
? (вопрос)	10	0D	3F	6F	5F	206

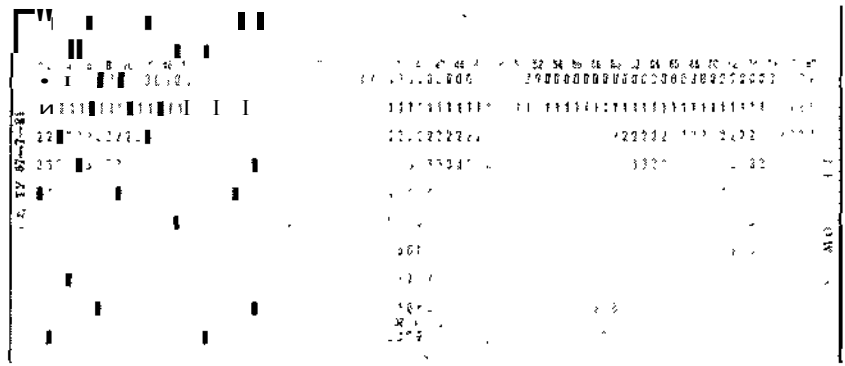


Рис. 1.8. Перфокарта Холлерита

можно закодировать 65 536 символов вместо 128 в ASCII, то отпадает необходимость в создании модификаций таблиц кодов. Это существенно упрощает обработку текстовых файлов, хотя и несколько увеличивает их размеры.

UNICODE охватывает 28 000 букв, знаков, слогов, иероглифов национальных языков мира, 30 000 мест в UNICODE зарезервировано. Использование этого резерва дает возможность пользователям вводить математические или технические символы, а также создавать свои собственные символы.

Единая стандартизация языковых форматов наводит порядок в международном кодировании алфавитов различных языков. Здесь учтено также то, что в таких языках, как иврит и арабский, текст пишется справа налево.

При передаче данных часто используются *избыточные коды,* т. е. такие, которые за счет усложнения структуры позволяют повысить надежность передачи данных. К ним, в первую очередь, относятся коды с обнаружением ошибок. Чаще всего это циклические избыточные коды. Простая разновидность такого кода — код с контролем по четности. Широко используется для обнаружения ошибок в блоках данных также код контроля циклической избыточности CRC. Он определяется на основе содержимого блока данных перед его передачей, включается в одно из полей блока, а затем повторно вычисляется после передачи. Несовпадение результатов свидетельствует об ошибке в передаваемом содержимом.

Важное значение имеют коды с исправлением ошибок. Использование этих кодов позволяет с большой вероятностью не только обнаруживать, но и исправлять возникшие при передаче ошибки (код Хемминга, позволяющий исправлять одиночные ошибки, появляющиеся в блоках данных).

Кодирование и обработка чисел

Кроме кодирования символов, в ЭВМ очевидное и важное значение имеют кодирование и представление чисел.

Системы счисления. Мы привыкли считать предметы десятками, сотнями: десять единиц образуют десяток, десять десятков — сотню, десять сотен — тысячу и т. д. Это — десятичная система счисления, которая не является единственно возможной. Существуют, например, двенадцатеричная система счисления (счет идет на дюжины) или римская система счисления.

Наиболее естественный способ представления числа в компьютерной системе заключается в использовании строки битов, называемой двоичным числом — числом в двоичной системе счисления (символ также может быть представлен строкой битов или символа).

Система счисления — способ именования и изображения чисел с помощью символов, имеющих определенные количественные значения. В зависимости от способа изображения чисел системы счисления делятся на:

- непозиционные;
- позиционные.

Непозиционные системы счисления В непозиционной системе цифры не меняют своего количественного значения при изменении их расположения в числе.

Самый простой и очевидный пример — система счисления, где количество обозначается I (палочкой / единицей).

1 = I;

2 = I I;

5 = I I I I I;

10 = I 1 I I I I I I I I I.

Пусть, далее следующие символы (цифры в гипотетической системе счисления) соответствуют числам (десятичной системе счисления):

♊ — 1;

♎ — 6;

♒ —12;

♐ - 24;

♍ - 60;

♌ - 365,

и пусть есть правило, по которому число можно записать любой комбинацией таких символов, чтобы сумма обозначаемых ими чисел была равна заданному числу.

Тогда 444 можно записать по крайней мере тремя способами:

♌♍♒♎♊ (365 + 60 + 12 + 6 + 1);

♎♊♌♍♒ (6 + 1 + 365 + 60 + 12),

т. е. ад^^п = ♎♊♌♍♒.

Такая система счисления является непозиционной, так как цифры не меняют своего количественного значения при изменении их расположения в числе.

Позиционные системы счисления. В позиционной системе счисления количественное значение каждой цифры зависит от ее места (позиции) в числе.

Десятичная система счисления является позиционной, так как значение каждой цифры зависит от ее места (позиции) в числе.

Например,

23 = 2 x 10 + 3;

32 = 3 x 10 + 2

и 23 ≠ 32

Римская система счисления является смешанной, так как значение каждой цифры частично зависит от ее места (позиции) в числе. Так, в числах:

VII;

VI;

IV

V обозначает 5, а I обозначает 1. Но, с другой стороны, важно, как цифры расположены относительно друг друга:

VII = 5 + 1 + 1 = 7;

VI = 5 + 1 = 6;

IV = 5 - 1 - 4.

Основание системы счисления — количество (P) различных цифр, используемых для изображения числа в позиционной системе счисления. Значения цифр лежат в пределах от 0 до $P - 1$.

В общем случае запись любого числа N в системе счисления с основанием Р будет представлять собой ряд (многочлен) вида:

$$N = a_{m-1} \times P^{m-1} + a_{m-2} \times P^{m-2} + \ldots + a_k \times P^k + \ldots$$

$$\ldots + a_1 \times P^1 + a_0 \times P^0 + \ldots + a_{-1} \times P^{-1} + a_{-2} \times P^{-2} + \ldots + a_{-s} \times P^{s}. \quad (1.1)$$

Нижние индексы определяют местоположение цифры в числе (разряд):

- положительные значения индексов — для целой части числа (m разрядов);
- отрицательные значения — для дробной (s разрядов).

Максимальное целое число, которое может быть представлено в m разрядах:

$$N_{max} = P^m - 1.$$

Минимальное значащее, не равное 0, число, которое можно записать в 5 разрядах дробной части:

$$N_{min} = P^{-s}.$$

Имея в целой части числа m разрядов, а в дробной — s, можно записать P^{m+s} разных чисел.

Двоичная система счисления имеет основание $P = 2$ и использует для представления информации две цифры: 0 и 1.

Существуют правила перевода чисел из одной системы счисления в другую, основанные, в том числе, и на выражении (1 1).

Например, двоичное число 101110,101 равно десятичному числу 46,625:

$$101110{,}101_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + \\ + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 46{,}625_{10}.$$

Практически перевод из двоичной системы в десятичную можно легко выполнить, надписав над каждым разрядом соответствующий ему вес и сложив затем произведения значений соответствующих цифр на их веса.

Например, двоичное число 01000001_2 равно 65_{10}. Действительно, $64 \cdot 1 + 1 \cdot 1 = 65$.

Вес	128	64	32	16	8	4	2	1
Цифра	0	1	0	0	0	0	0	1

Таким образом, для перевода числа из позиционной системы счисления с любым основанием в десятичную систему счисления можно воспользоваться выражением (1.1).

Обратный перевод из десятичной системы счисления в систему счисления с другим основанием непосредственно по (1.1) затруднителен, поскольку все арифметические действия, предусмотренные этой формулой, следует выполнять в той системе счисления, в которую число переводится. Обратный перевод выполняется значительно проще, если предварительно преобразовать отдельно целую и дробную части выражения (1.1) к виду

$$N_{цел} = (((...(a_{m-1} \times P + a_{m-2}) \times P + ... + a_2) \times P + a_1) \times P + a_0;$$

$$N_{др} = P^{-1} \times (a_{-1} + P^{-1} \times (a_{-2} + P^{-1} \times (a_{-3} + ... + P^{-1} \times (a_{-s+1} + P^{-1} \times a_{-s})...))).$$

Алгоритм перевода числа из десятичной системы счисления в систему счисления с основанием P, основанный на этих выражениях, позволяет оперировать числами в той системе счисления, из которой число переводится, и может быть сформулирован следующим образом.

При переводе смешанного числа следует переводить его целую и дробную части отдельно.

1. Для перевода целой части числа ее, а затем целые части получающихся частных от деления, следует последовательно делить на основание P до тех пор, пока очередная целая часть частного не окажется равной 0. Остатки от деления, записанные последовательно справа налево, образуют целую часть числа в системе счисления с основанием P.

2. Для перевода дробной части числа ее, а затем дробные части получающихся произведений, следует последовательно умножать на основание *P* до тех пор, пока очередная дробная часть произведения не окажется равной 0 или не будет достигнута нужная точность дроби. Целые части произведений, записанные после запятой последовательно слева направо, образуют дробную часть числа в системе счисления с основанием *P*.

Пусть требуется перевести смешанное число из десятичной в двоичную систему счисления на примере числа 46,625.

1. Переводим целую часть числа:

46 : 2 = 23 (остаток 0).

23 : 2 = 11 (остаток 1).

11 : 2 = 5 (остаток 1).

5 : 2 = 2 (остаток 1).

2 : 2 = 1 (остаток 0).

1 : 2 = 0 (остаток 1).

Записываем остатки последовательно справа налево — 101110, т. е. $46_{10} = 101110$,

2. Переводим дробную часть числа:

0,625 × 2 = 1,250.

0,250 × 2 = 0,500.

0,500 x 2 = 1,000 (дробная часть равна 0 ⇒ стоп).

Записываем целые части получающихся произведений после запятой последовательно слева направо — 0,101, т. е. $0{,}625_{10} = 0{,}101_2$.

Окончательно: $46{,}625_{10} = 101110{,}101_2$.

Кроме двоичной и десятичной при работе с компьютером часто используются также двоично-десятичная и шестнадцатеричная системы счисления (табл. 1.11)

Шестнадцатеричная система счисления часто используется при программировании. Перевод чисел из шестнадцатеричной системы счисления в двоичную систему весьма прост — он выполняется поразрядно.

Для изображения цифр, больших 9, в шестнадцатеричной системе счисления применяются буквы A = 10, B = 11, C = 12, D = 13, E= 14, F= 15.

Например, шестнадцатеричное число F17B в двоичной системе выглядит так: 1111000101111011, а в десятичной — 61 819.

Двоично-десятичная система счисления получила большое распространение в современных компьютерах ввиду легкости перевода в десятичную систему и обратно. Она используется там, где основ-

Таблица 1. П. **Перевод цифр из двоичной системы счисления в восьмеричную и десятичную и наоборот**

Триада	Восьмеричная цифра	Тетрада	Шестнадцатеричная цифра	Десятичное число	Двоично-десятичная запись
000	0	0000	0	0	0000-0000
001	1	0001	1	1	0000-0001
010	2	0010	2	2	0000-0010
011	3	ООП	3	3	0000-0011
100	4	0100	4	3	0000-0100
101	5	0101	5	5	0000-0101
110	6	0110	6	6	0000–0110
111	7	0111	7	7	0000-0111
		1000	8	8	0000-1000
		1001	9	9	0000-1001
		1010	A	10	0001-0000
		1011	B	11	0001-0001
		1100	C	12	0001-0010
		1101	D	13	0001-0011
		1110	E	14	0001-0100
		1111	F	15	0001-0101

ное внимание уделяется не простоте технического построения машины, а удобству работы пользователя. В этой системе счисления все десятичные цифры отдельно кодируются четырьмя двоичными цифрами и в таком виде записываются последовательно друг за другом.

Двоично-десятичная система не экономична с точки зрения реализации технического построения машины (примерно на 20 % увеличивается требуемое оборудование), но очень удобна при подготовке задач и при программировании. В двоично-десятичной системе счисления основанием системы счисления является число 10, но каждая десятичная цифра (О, 1, ..., 9) кодируется двоичными цифрами.

Представление чисел в ЭВМ

Как известно, в ЭВМ применяется двоичная система счисления. Может быть доказано, что при этом на построение ЭВМ тратится наименьшее количество базовых аппаратных элементов — «вентилей». Точнее, оптимальным основанием системы счисления по критерию «минимум аппаратных расходов» является *основание натурального логарифма* $e \approx 2{,}72$.

Однако по ряду очевидных причин для ЭВМ принято $P = 2$. Достаточно вспомнить, что одна из первых электронных ВМ ENIAC содержала 17 468 электронных ламп, имела размеры около 6 м в высоту и 30 м в длину. Обилие применяемых вакуумных ламп, габаритные размеры машины отчасти объяснялись тем, что она работала с десятичными числами.

В ЭВМ применяются две формы представления чисел:

- естественная форма, или форма с фиксированной запятой (точкой) — ФЗ (ФТ);
- нормальная форма, или форма с плавающей запятой (точкой) – ПЗ (ПТ).

Фиксированная запятая (точка). В форме представления с *фиксированной запятой (тонкой)* числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть от дробной.

Например, пусть числа представлены в десятичной системе счисления и имеют пять разрядов в целой части числа (до запятой) и пять в дробной части (после запятой). Числа, записанные в такую разрядную сетку, имеют вид:

+00721.35500.

+00000.00328.

-10301.20260.

Эта форма наиболее проста, естественна, но имеет небольшой диапазон представления чисел и поэтому чаще всего неприемлема при вычислениях.

Диапазон значащих чисел N в системе счисления с основанием P при наличии m разрядов в целой части и 5 разрядов в дробной части числа (без учета знака числа) будет таким:

$$P^{-s} \le N \le P^m - P^{-s}.$$

Например, при $P = 2$, $m = 10$ и $s = 6$ числа изменяются в диапазоне $0{,}015 < N < 1024$. Если в результате операции получится число, выходящее за допустимые пределы, произойдет переполнение разрядной сетки, и дальнейшие вычисления теряют смысл. В совре-

менных компьютерах естественная форма представления используется как вспомогательная и только для целых чисел.

В памяти ЭВМ числа с фиксированной точкой хранятся в трех форматах:

а) полуслово — это обычно 16 бит, или 2 байта;

б) слово — 32 бита, или 4 байта;

в) двойное слово — 64 бита, или 8 байтов.

Отрицательные числа с ФТ записываются в разрядную сетку в дополнительных кодах, которые образуются прибавлением единицы к младшему разряду обратного кода. Обратный код получается заменой единиц на нули, а нулей на единицы в прямом двоичном коде.

Плавающая запятая (точка). В форме представления с плавающей запятой (точкой) число изображается в виде двух групп цифр:

- мантисса;
- порядок.

При этом абсолютная величина мантиссы должна быть меньше 1, а порядок должен быть целым числом. В общем виде число в форме с плавающей запятой может быть представлено так:

$$N = \pm M \times P^{\pm r},$$

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

Например, приведенные ранее числа в нормальной форме запишутся следующим образом:

$+0{,}721355 \times 10^{3}$;

$+0{,}328 \times 10^{-3}$;

$-0{,}103012026 \times 10^{5}$.

Нормальная форма представления обеспечивает большой диапазон отображения чисел и является основной в современных компьютерах. Так, диапазон значащих чисел в системе счисления с основанием P при наличии m разрядов у мантиссы и s разрядов у порядка (без учета знаковых разрядов порядка и мантиссы) будет:

$$P^{-m} \times P^{(P^s-1)} \le N \le (1 - P^{-m}) \times P^{(P^s-1)}.$$

Например, при $P = 2$, $m = 22$ и $s = 10$ диапазон чисел простирается примерно от 10^{-300} до 10^{300}. Для сравнения: количество секунд, которые прошли с момента образования планет Солнечной системы, составляет около 10^{18}.

Следует заметить, что все числа с плавающей запятой хранятся в машине в так называемом *нормализованном* виде.

Нормализованным называют такое число, старший разряд мантиссы которого больше нуля. У нормализованных двоичных чисел, следовательно, $0{,}5 < |M| < 1$.

Нормализованные, т. е. приведенные к правильной дроби, числа:

$10{,}35_{10} = 0{,}1035_{10} \times 10^{+2}$;

$0{,}00007245_8 = 0{,}7245_8 \times 8^{-4}$;

$F5C{,}9B_{16} = 0{,}F5C9B_{16} \times 16^{+3}$;

В памяти ЭВМ числа с ПТ хранятся в двух форматах:

- слово — 32 бита, или 4 байта;
- двойное слово — 64 бита, или 8 байт.

Разрядная сетка для чисел с ПТ имеет следующую структуру:

- нулевой разряд — это знак числа (0 — «минус», 1 — «плюс»);
- с 1 по 7 разряд записывается порядок в прямом двоичном коде, пустые разряды заполняются нулями. В первом разряде указывается знак порядка (1 — «плюс» или 0 — «минус»);
- с 8 по 31 (63) указывается мантисса, слева направо без нуля целых в прямом двоичном коде и для отрицательных чисел и пустые разряды заполняются нулями.

Алгебраическое представление двоичных чисел

Знак числа обычно кодируется двоичной цифрой, при этом:

код 0 означает знак + (плюс);

код 1 — знак - (минус).

Для алгебраического представления чисел, т. е. для представления чисел с учетом их знака, в вычислительных машинах используются специальные коды:

- прямой код числа;
- обратный код;
- дополнительный код.

При этом два последних кода позволяют заменить неудобную для компьютера операцию вычитания на операцию сложения с отрицательным числом. Дополнительный код обеспечивает более быстрое выполнение операций, поэтому в компьютере чаще всего применяется именно он.

Прямой код числа N (обозначим $[N]_{пр}$).

Пусть $N = a_1, a_2, a_3, ..., a_m$, тогда:

при $N > 0$, $[N]_{пр} = 0, a_1, a_2, a_3, ..., a_m$;

при $N < 0$, $[N]_{пр} = 1, a_1, a_2, a_3, ..., a_m$;

при $N = 0$ имеет место неоднозначность $[0]_{пр} = 0, 0 ... = 1, 0 ....$

Если при сложении в ЭВМ оба слагаемых имеют одинаковый знак, то операция сложения выполняется обычным путем. Если при сложении слагаемые имеют разные знаки, то сначала необходимо выявить большее по абсолютной величине число, из него произвести вычитание меньшего по абсолютной величине числа и разности присвоить знак большего числа.

Выполнение операций умножения и деления в прямом коде выполняется обычным образом, но знак результата определяется по совпадению или несовпадению знаков участвовавших в операции чисел.

Операцию вычитания в этом коде нельзя заменить операцией сложения с отрицательным числом, поэтому возникают сложности, связанные с займом значений из старших разрядов уменьшаемого числа. В связи с этим прямой код в ЭВМ почти не применяется.

Обратный код числа N, обозначим $[N]_{обр}$.

Пусть $N = a_1, a_2, a_3, ..., a_m$ и a обозначает *инверсию a,* т. е. если $a = 1$, то $\bar{a} = 0$, и наоборот. Тогда:

при $N > 0$, $[N]_{обр} = 0, a_1, a_2, a_3, ..., a_m$;

при $N < 0$, $[N]_{обр} = 1, \bar{a}_1, \bar{a}_2, a_3, ..., \bar{a}_m$;

при $N = 0$ имеет место неоднозначность $[0]_{обр} = 0{,}00...0 = 1{,}11...1$.

Для того чтобы получить обратный код отрицательного числа, необходимо все цифры этого числа *инвертировать,* т. е. в знаковом разряде поставить 1, во всех значащих разрядах нули заменить единицами, а единицы нулями.

Например,

для $N = 1011$ $[N]_{обр} = 0{,}1011$;

для $N = -1011$ $[N]_{обр} = 1{,}0100$.

Дополнительный код числа N, обозначим $[N]_{доп}$.

Пусть, как и выше, $N = a_1, a_2, a_3, ..., a_m$ и $\bar{a}$ обозначает величину, обратную a (инверсию а), т. е. если $a = 1$, то $\bar{a} = 0$, и наоборот. Тогда:

при $N \geq 0$, $[N]_{доп} = 0, a_1, a_2, a_3, ..., a_m$;

при $N \leq 0$, $[N]_{доп} = 1, a_1, a_2, a_3, ..., a_m + 0{,}00...1$.

Для того чтобы получить дополнительный код отрицательного числа, необходимо все его цифры инвертировать (в знаковом разряде поставить единицу, во всех значащих разрядах нули заменить единицами, а единицы — нулями) и затем к младшему разряду прибавить единицу. В случае возникновения переноса из первого после запятой разряда в знаковый разряд к числу следует прибавить единицу в младший разряд.

Например,

для $N = 1011$, $[N]_{доп} = 0{,}1011$;

для $N = -1100$, $[N]_{доп} = 1{,}0100$;

для $N = -0000$, $[N]_{доп} = 10{,}0000 = 0{,}0000$ (1 исчезает). Неоднозначности в изображении 0 нет.

Эмпирическое правило: для получения дополнительного кода отрицательного числа необходимо все символы этого числа инвертировать, кроме последней (младшей) единицы и тех нулей, которые за ней следуют.

Элементы двоичной арифметики. Рассмотрим, как выполняются арифметические действия в двоичной системе. Для этого проведем анализ таблиц сложения и умножения в двоичной системе:

$0 + 0 = 0$, $0 \times 0 = 0$, $0 + 1 = 1$, $0 \times 1 = 0$, $1 + 1 = 10$.

Следует обратить внимание на аналогию в правилах выполнения арифметических действий в двоичной и десятичных системах счисления: если при сложении двух двоичных чисел (точнее, представленных в двоичной системе счисления) сумма цифр окажется больше единицы, то возникает перенос в старший разряд; если уменьшаемая цифра меньше вычитаемой, то нужно сделать «заем» единицы в старшем разряде.

Анализируя примеры умножения в двоичной системе счисления, необходимо обратить внимание на одну важную особенность выполнения этой операции в данной системе. Так как очередная цифра множителя может быть только 1 или 0, то промежуточное произведение равно либо множимому, либо 0. Таким образом, операция умножения в двоичной системе фактически не производится: в качестве промежуточного произведения записывается либо множимое, либо 0, а затем промежуточные произведения суммируются. Иначе говоря, операция умножения заменяется последовательным сложением.

Как уже известно, дополнительный код используется для вычитания чисел в компьютерах и позволяет эту операцию свести к сложению чисел.

Правила выполнения вычитания с дополнительным числом следующие. Чтобы вычесть число *A* из числа *B,* достаточно сложить *B* с дополнительным числом к *A* и отбросить перенос в соседний старший разряд. Например, чтобы вычесть 623 из 842, достаточно сложить 842 с 377; отбросив перенос, получим 219 (842 - 623 = 219).

Таким образом, важнейшее преимущество двоичной арифметики заключается в том, что она позволяет все арифметические дейст-

вия свести к одному — сложению, а это значительно упрощает устройство процессора ЭВМ.

Изложенные здесь основные принципы положены в основу функционирования элементов и узлов ЭВМ (см. далее — триггер, сумматор, полусумматор).

Типы и структуры данных

Типы данных (табл 1.12) Классификация информационных единиц, обрабатываемых на ЭВМ включает следующие аспекты:

- *типы данных,* или совокупность соглашений о программно-аппаратурной форме представления и обработки, а также ввода, контроля и вывода элементарных данных,
- *структуры данных* — способы композиции простых данных в агрегаты и операции над ними;
- *форматы файлов* — представление информации на уровне взаимодействия операционной системы с прикладными программами.

Ранние языки программирования (ЯП), а точнее — системы программирования (СП) — Фортран, Алгол, будучи ориентированы исключительно на вычисления, не содержали развитых систем типов и структур данных.

В ЯП Алгол символьные величины и переменные вообще не предусматривались, в некоторых реализациях строки (символы в апострофах) могли встречаться только в операторах печати данных.

Типы числовых данных Алгола: INTEGER (целое число), REAL (действительное) — различаются диапазонами изменения, внутренними представлениями и применяемыми командами процессора ЭВМ (соответственно арифметика с фиксированной и плавающей точкой). Нечисловые данные представлены типом BOOLEAN — логические, имеющие диапазон значений {*true*,*false*}.

Позже появившиеся ЯП (СП) Кобол, PL/1, Паскаль вводят новые типы данных·

- символьные (цифры, буквы, знаки препинания и пр);
- числовые символьные для вывода;
- числовые двоичные для вычислений,
- числовые десятичные (цифры 0—9) для вывода и вычислений

Разновидности числовых данных здесь соответствуют внутреннему представлению и машинным (или эмулируемым) командам обработки. Кроме того, вводятся числа двойного формата (два машинных слова), для обработки которых также необходимо наличие

Таблица 1 12 **Типы и структуры данных в некоторых системах программирования и управления данными**

	Данные	Система- язык программирования, СУБД, ИПС						
		Алгол	Кобол	PL/1	FoxBase/ Clipper	Adabas/ Natural	Oracle/ SQL	STAIRS, IRBIS, ISIS
Тип данных	Целое короткое (2 байта)	–	–	–	–	–	Small-int	–
	Целое норм (4 байта)	Integer	Compu-tational	Int	N(x)	N(x)	Int	–
	Целое длинн (8 байт)	–	–	Double	–	–	–	–
	Действительное норм (4 байта)	Real	Compu-tational	Roat	N(xy)	N(x y)	Float Real	–
	Действ двойное (8 байт)	–	–	–	–	–	Float Double	–
	Двоичное	–	–	Binary	–	B(x)	–	–
	Десятичное упакованное (2 цифры на байт)	–	PIC(9)	Decimal	–	P(x)	–	–
	Десятичное распакованное (1 цифра на байт)	–	PIC(X)	–	N(x)	U(x)	–	–
	Логическое	Boolean	–	+	Logical	–	–	–
	Символьное	–	PIC(A)	Char	CM	AM	Char	+
	Длинный текстовый или бинарный объект (BLOB)	–	–	–	Memo	–	VarGrafic VarChar	–
	Дата	–	–	–	Date	–	Date	–
	Время	–	–	–	–	–	Time	–
Структуры данных	Массивы	Array	–	Dim	Dimention	VAR(n)	–	–
	Записи (структуры)	–	+	+	+	+	+	–
	Множественные (векторные) поля записи	–	–	–	–	MU	–	+
	Групповые поля записи	–	+	+	–	GR	–	+
	Повторяющиеся группы в записи	–	–	–	–	PE	–	–
	Текстовые поля (параграфы, предложения, слова)	–	–	–	–	–	–	+

в процессоре (или эмуляция) команд обработки чисел двойной длины (точности).

Уместно привести пример представления числовой информации в различных перечисленных формах. Пусть задано число $135_{10} = 207_8 = 87_{16} = 100000111_2$, тогда:

- внутренняя стандартная форма представления (тип BINARY для обработки в двоичной арифметике) — сохраняется (100000111_2). Объем — 1 байт, или 8 двоичных разрядов;
- внутренняя форма двоично-десятичного представления (тип DECIMAL, каждый разряд десятичного числа представляется двоично-десятичной, в 4 бита, комбинацией). Представление 135 есть $001\ 011\ 101_2$. Объем — 2,5 байта, 12 двоичных разрядов;
- символьное представление (тип ALPHABETIC, для вывода) — каждый разряд представляется байтом в соответствии с кодом ASCII (табл. приложения 2). Представление 135 есть — $00110001\ 00110011\ 00110101_2$. Объем - 3 байта.

Некоторые системы программирования (Fortran IV, например) поддерживают операции над комплексными числами вида $Z = A + Bi$ (где A, B — действительные коэффициенты, а i — мнимая единица). Очевидно, для размещения таких чисел необходим как минимум двойной расход оперативной памяти (по одному слову для размещения действительной и мнимой частей при обычной точности и по два слова при двойной точности). Кроме того, очевидно, что процессоры обычных универсальных ЭВМ вряд ли поддерживают операции над такими числами, в связи с чем операции над ними требуют написания соответствующих подпрограмм *или эмуляции комплексной арифметики.*

Появление систем управления базами данных и систем программирования для разработки ИС приводит к появлению других типов данных:

- *дата и время;*
- *бинарные* (BLOB — Binary Large Object) и *текстовые объекты* без внутренней структуры (интерпретация возлагается на прикладные программы).

Понятие типа данных ассоциируется также с допустимыми значениями переменной и операциями над ними, например, данные типа время (ЧЧ:ММ:СС) или дата (ГГ/ММ/ДД) предполагают определенные диапазоны значений каждого из разрядов, а также машинные или эмулируемые операции (сложение/вычитание дат и/или моментов времени). Основной причиной «проблемы 2000 г.» являлось не столько двухразрядная запись года в базах данных,

сколько встроенные в огромное количество программ (часто не документированных) операции над данными типа DATE — ГГ/ММ/ДД.

Структуры данных. В Алголе были определены два типа структур: элементарные данные и массивы (векторы, матрицы, тензоры, состоящие из арифметических или логических переменных). Основным нововведением, появившимся первоначально в Коболе (затем PL/1, Паскаль и пр.), являются агрегаты данных (структуры, записи), представляющие собой именованные комплексы переменных разного типа, описывающих некоторый объект или образующих некоторый достаточно сложный документ.

Рассмотренные выше экзотические типы данных (комплексные числа), очевидно, занимают промежуточное положение между элементарными переменными и массивами (структурами).

Термин *запись* подразумевает наличие множества аналогичных по структуре агрегатов, образующих *файл* (картотеку), содержащих данные по совокупности однородных объектов, элементы данных образуют поля, среди которых выделяются элементарные и групповые (агрегатные).

Появление СУБД и АИПС приводит к появлению новых разновидностей структур:

- множественных полей данных;
- периодических групповых полей;
- текстовых объектов (документов), имеющих иерархическую структуру (документ, сегмент, предложение, слово).

Двоичное кодирование мультимедиа информации. С 80-х гг. бурно развивается технология обработки на компьютере графической информации. Компьютерная графика широко используется в компьютерном моделировании в научных исследованиях, компьютерных тренажерах, компьютерной анимации, деловой графике, играх и т. д.

В последнее время в связи с резким ростом аппаратных возможностей персональных компьютеров пользователи получили возможность обрабатывать видеоинформацию.

Графическая информация на экране дисплея представляется в виде изображения, которое формируется из точек (пикселей). В современных компьютерах разрешающая способность (количество точек на экране дисплея), а также количество цветов зависят от видеоадаптера и могут меняться программно.

Цветные изображения могут иметь различные режимы: 16 цветов, 256 цветов, 65 536 цветов (high color), 16 777 216 цветов (true

color) — табл. 1.13. Очевидно, что количество бит на точку (пиксель), например, режима true color, равно:

$$I = \log_2 65\,536 = 16 \text{ бит} = 2 \text{ байта}.$$

Таблица 1.13. **Характеристики различных стандартов представления графики**

Разрешение	16 цветов	256 цветов	65 536 цветов	16 777 216 цветов
640 × 480	150 Кбайт	300 Кбайт	600 Кбайт	900 Кбайт
800 × 600	234,4 Кбайт	468,8 Кбайт	937,5 Кбайт	1,4 Мбайт
1024 × 768	384 Кбайт	768 Кбайт	1,5 Мбайт	2,25 Мбайт
1280 × 1024	640 Кбайт	1,25 Мбайт	2,5 Мбайт	3,75 Мбайт

Наиболее распространенной разрешающей способностью экрана является разрешение 800 на 600 точек, т. е. 480 000 точек.

Рассчитаем необходимый для режима true color объем видеопамяти:

$$V = 2 \text{ байта} \times 480\,000 = 960\,000 \text{ байт} = 937{,}5 \text{ Кбайт}.$$

Аналогично рассчитывается объем видеопамяти, необходимый для хранения битовой карты изображений при других видеорежимах.

В видеопамяти компьютера хранится битовый план (bit map), являющийся двоичным кодом изображения, отсюда она считывается (не реже 50 раз в секунду) и отображается на экране.

Двоичное кодирование звуковой информации. С начала 90-х гг. персональные компьютеры получают широкие возможности для работы со звуковой информацией. Каждый компьютер, имеющий звуковую плату, может сохранять звук в виде файлов и воспроизводить его. С помощью специальных программных средств (редакторов аудиофайлов) открываются широкие возможности по созданию, редактированию и прослушиванию звуковых файлов. В дальнейшем создаются программы распознавания речи и появляется возможность голосового управления компьютером.

При двоичном кодировании аналогового звукового сигнала непрерывный сигнал дискретизируется (оцифровывается), т. е. заменяется серией отдельных выборок (см. рис. 1.6). Качество двоичного кодирования зависит от двух параметров: количества распознаваемых дискретных уровней сигнала и количества выборок в секунду.

Различные звуковые карты могут обеспечить как 8-, так и 16-битные выборки. При замене непрерывного звукового сигнала его дискретным представлением в виде ступенек 8-битные карты позволяют закодировать 256 различных уровней дискретизации звукового сигнала, соответственно 16-битные — 65 536 уровней.

Частота дискретизации аналогового звукового сигнала (количество выборок в секунду) также может принимать различные значения (5,5, 11, 22 и 44 кГц). Таким образом, качество звука в дискретной форме может быть очень плохим (качество радиотрансляции) при 8 битах и 5,5 кГц и весьма высоким (качество аудиоCD) при 16 битах и 44 кГц.

Можно оценить объем моноаудиофайла с длительностью звучания 1 с при среднем качестве звука (16 бит, 22 кГц). Для этого 16 бит на одну выборку необходимо умножить на 22 000 выборок в секунду, что дает в результате 43 Кбайта.

Сжатие информации. Объем обрабатываемой и передаваемой информации быстро растет. Это связано с выполнением все более сложных прикладных процессов, появлением новых информационных служб, использованием изображений и звука.

Сжатие данных (data compression) — процесс, обеспечивающий уменьшение объема данных. Сжатие позволяет резко уменьшить объем памяти, необходимый для хранения данных, сократить (до приемлемых размеров) время их передачи. Особенно эффективно сжатие изображений. Сжатие данных может осуществляться как программным, так и аппаратным или комбинированным методом.

Сжатие текстов связано с более компактным расположением *байтов,* кодирующих символы. Определенные результаты дает статистическое кодирование, в котором наиболее часто встречающиеся символы получают коды наименьшей длины. Здесь также используется счетчик повторений пробелов. Что же касается звука и изображений, то объем представляющей их информации зависит от выбранного шага квантования и числа разрядов аналого-дискретного преобразования. В принципе, здесь используются те же методы сжатия, что и при обработке текстов. Если сжатие текстов происходит без потери информации, то сжатие звука и изображения почти всегда приводит к ее некоторой потере. Сжатие широко используется при архивировании данных.

Сжатие изображений (images compression) — процесс минимизации данных, определяющих изображение. Минимизация количества информации, предоставляющей изображение или видеофильм прежде всего, осуществляется при выборе шага квантования и разрядности кодов. При этом, естественно, происходит определенная (до-

пустимая) потеря информации. Затем происходит сжатие изображения, представленного дискретным сигналом.

Сжатие изображения осуществляется в несколько этапов:

- изображение делится на блоки *пикселов,* каждый из которых подвергается обработке, устраняющей избыточность;
- осуществляется кодирование с переменной длиной кодов, что исключает длинные цепочки нулей и единиц в последовательностях *битов*;
- дополнительное сжатие движущегося изображения за счет сравнения каждого изображения с предыдущим, чтобы сохранять только изменившуюся его часть.

Допускается потеря той информации, которая в решении поставленной задачи считается несущественной. Например, можно при обработке изображений удалить из аналогового сигнала частоты, которые находятся вне спектра, воспринимаемого глазом человека (до 10 000 цветов, 250 оттенков серого цвета). Нередко допускается игнорирование цвета каждого второго пиксела либо группа пикселов заменяется одним со средним значением цвета. Осуществляется также групповое кодирование. Его сущность заключается в кодировании групп одинаковых пикселов (например, небо без облаков на картине).

Размер файла сжатого дискретного неподвижного изображения зависит от четырех параметров: площади изображения, квадрата разрешения, числа бит, необходимых для представления пиксела, и коэффициента сжатия. В видеофильме к этому еще добавляется число образующих его неподвижных изображений. Выбор коэффициентов сжатия — компромисс между пропускной способностью системы (скоростью переноса файлов) и качеством восстанавливаемого изображения. Чем выше коэффициент сжатия, тем ниже это качество. При этом следует иметь в виду, что при очень высокой разрешающей способности и большом коэффициенте сжатия можно получить изображение с низкой разрешающей способностью. Поэтому, выбор указанных параметров обосновывается технико-экономическим анализом и алгоритмом сжатия. Что касается качества изображения, то оно зависит от конкретной поставленной задачи. Например, в системах телеконференций основной объем необходимой информации содержится в речи, тогда как качество изображения может играть вторую роль.

В зависимости от скорости сжатия изображений выполняемые процессы подразделяются на два класса. К первому относится сжатие неподвижных изображений, которое может выполняться в фоновом режиме, с любой возможной скоростью Второй класс обра-

зуют алгоритмы сжатия движущихся изображений, которые должны выполняться в реальном времени по мере получения данных.

Существует немало технологий сжатия/восстановления изображений. Наиболее популярная из них предложена *Объединенной группой экспертов в области фотографии (JPEG)* и позволяет сократить размеры графического файла в 10—20 раз. Благодаря специальным процессорам и алгоритмам удается также сжимать видеосюжеты.

Кодирование видеоинформации. В связи с большим объемом информации, содержащейся в видеопотоке (до 6 Мбайт/с), для записи информации в ЭВМ обычно применяют сжатое кодирование потока данных на входе с использованием алгоритмов семейства MPEG/JPEG (табл. 1.14).

Таблица 1 14 **Характеристики представления видеоинформации в различных форматах**

№	Формат	Тип данных (размер изображения)	Длительность записи (CD/DVD), мин
1	VCD	288 x 384	63
2	S-VCD	480 x 576	32
3	DVD	576 x 720	59
4	VHS	288 x 384	–
5	S-VHS	540 x 720	–
6	Internet High Speed	193 x 144	–

Стандарт MPEG (Motion Picture Expert Group) включает несколько компонент: системного потока, описывающего структуру смешанного аудио- и видеопотока, а также MPEG-video и MPEG-audio.

В случае MPEG video сжатие достигается за счет четырех факторов.

1. Использование составляющих YUV вместо обычных RGB (красный, зеленый, синий).

Вместо элементарных цветов кодируется яркость (luminance, Y) и цветность (chrominance, U & V), причем цветность «прорежена» по вертикали и горизонтали в два раза по сравнению с яркостью (децимация). При этом вместо сильно коррелированных сигналов RGB получаются практически некоррелированные YUV, и за счет децимации достигается двукратное сжатие.

2. Дискретно-косинусное преобразование с последующим квантованием.

При этом квадраты пикселей (8 x 8) подвергаются двухмерному дискретно-косинусному преобразованию (DCT), которое родствен-

но преобразованию Фурье, различие заключается в наборе базисных функций (в преобразовании Фурье это синусы и косинусы, в DCT — косинусы). Это преобразование переводит пространственное представление сигнала в частотное. Результат преобразования подвергается квантованию, т. е. огрублению точности, при этом коэффициент квантования для более высоких пространственных частот выбирается более высоким, чем для низких, с учетом особенностей восприятия. При этом высокие пространственные частоты передаются с меньшей точностью, чем низкие частоты. При квантовании многие пространственные частоты не кодируются и не передаются.

3. Устранение временной избыточности с компенсацией движения.

Это означает, что для ликвидации избыточности, заключающейся в большой корреляции между соседними кадрами, передается разность между ними. Кадры видеопотока разбиваются на несколько типов — *Intra (I),* которые кодируются полностью, *Predicted* (*P*), для которых кодируется различие с предыдущим *I*- или *P*-кадром, и *Bidirectional (Б),* для которых в качестве *опорных (reference)* используются /- и/или *P*-кадры, между которыми он находится. Обычно *I*-кадры следуют 1 или 2 раза в секунду, и между двумя опорными кадрами лежит 2—4 В-кадра. Типичная последовательность кадров имеет вид: *IBBPBBPBBPBBIBBP.*В общем случае вид последовательности выбирается *кодеком* (кодирующее устройство или программа) и может зависеть или нет от содержания кадров. Поскольку изображение на соседних кадрах обычно сдвинуто, применяется компенсация движения, т. е. кодируется отклонение («разность») от некоторого сдвинутого опорного изображения Кодирование выполняется макроблоками (16 x 16 яркость, 8 × 8 цветность), для каждого макроблока определяется свой вектор движения.

4. Квазиоптимальное кодирование.

Коэффициенты, полученные после DCT, векторы движения и все остальное, кодируются кодами переменной длины. Это кодирование называют *квазиоптимальным*, поскольку кодовая таблица не строится заново для каждого конкретного случая, а выбрана при разработке стандарта на основе анализа типичных видеопоследовательностей.

MPEG-1 проектировался из расчета на поток 1,5 Мбит/с при 30 кадрах размером 352 x 240 в секунду, хотя он не ограничен этим и допускает существенно больший поток при произвольном размере кадра.

MPEG-2 проектировался с учетом опыта использования MPEG-1 и ориентируется на вещание, так как содержит средства для маскирования ошибок.

В случае MPEG audio исходный сигнал подвергается многоканальной фильтрации. Далее амплитуды сигналов в каждой полосе сравниваются для нахождения полос, подлежащих кодированию с учетом эффекта маскирования слабого сигнала сильным. Далее амплитуда сигнала в полосе квантуется и кодируется. При записи на Video CD скорость потока звука составляет 32 Кбайт/с.

1.4. Логические основы ЭВМ, элементы и узлы

Начало исследованиям в области формальной логики было положено работами Аристотеля в IV в. до нашей эры. Однако математические подходы к этим вопросам впервые были указаны Дж. Булем. В честь него алгебру высказывания называют *булевой алгеброй,* а логические значения — булевыми. Основу математической логики составляет алгебра высказываний. Это освобождает матлогику от неопределенности в толковании логических выражений, показывающих связь между отдельными суждениями и понятиями. Алгебра логики используется при построении основных узлов ЭВМ (дешифратор, сумматор, шифратор).

Алгебра логики оперирует с высказываниями. Под высказыванием понимают повествовательное предложение, относительно которого можно утверждать, истинно оно или ложно. Например, выражение «Расстояние от Москвы до Киева больше, чем от Москвы до Тулы» истинно, а выражение «5 < 2» — ложно.

Высказывания принято обозначать буквами латинского алфавита: *A, B, C, ..., X,* У и т. д. Если высказывание *C* истинно, то пишут *C* = 1 (C= *t, true),* а если оно ложно, то C= О *(C = f,false).*

Логические операции и базовые элементы компьютера

В алгебре высказываний над высказываниями можно производить определенные логические операции, в результате которых получаются новые высказывания. Истинность полученных высказываний зависит от истинности исходных высказываний и использованных для их преобразования логических операций.

Для образования новых высказываний наиболее часто используются логические операции, выражаемые словами «не», «и», «или».

Логический элемент компьютера — это часть электронной схемы, которая реализует элементарную логическую функцию.

Логическими элементами компьютеров являются электронные схемы И, ИЛИ, НЕ, И—НЕ, ИЛИ—НЕ и другие (называемые обычно *вентилями),* а также *триггер*

Может быть доказано, что с помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Обычно у вентилей бывает от двух до восьми входов и один или два выхода.

На структурных схемах ЭВМ каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию, но не указывает на то, какая именно электронная схема в нем реализована. Работу логических элементов описывают с помощью таблиц истинности.

Логические операции. Рассмотрим логические операции и соответствующие им элементы логических схем.

Конъюнкция. Соединение двух (или нескольких) высказываний в одно с помощью союза И (OR) называется операцией логического умножения, или конъюнкцией. Эту операцию принято обозначать знаками «л, &» или знаком умножения «×». Сложное высказывание *A & B* истинно только в том случае, когда истинны оба входящих в него высказывания. Истинность такого высказывания задается табл. 1.15.

Таблица 1 15 **Таблица истинности конъюнкции**

A	*B*	*A & B*
false	false	false
false	true	false
true	false	false
true	true	true

Логическая схема И реализует конъюнкцию двух или более логических значений. Условное обозначение на структурных диаграммах схемы И с двумя входами представлено на рис. 1.9, *а.*

Единица на выходе схемы И будет тогда и только тогда, когда на всех входах будут единицы. Когда хотя бы на одном входе будет нуль, на выходе также будет нуль.

Связь между выходом *z* этой схемы и входами *x* и *y* описывается соотношением: $z = x \& y$ (читается как «*x* И *y*»). Операция конъюнкции на структурных схемах обозначается знаком «&».

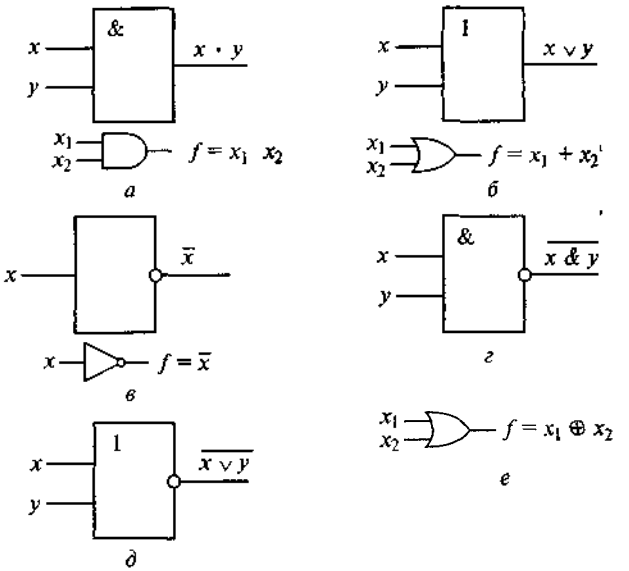


Рис. 1.9. Схемные логические элементы вычислительных машин

Дизъюнкция. Объединение двух (или нескольких) высказываний с помощью союза ИЛИ (OR) называется операцией логического сложения, или дизъюнкцией. Эту операцию обозначают знаками «', ∨» или знаком сложения «+». Сложное высказывание $A \vee B$ истинно, если истинно хотя бы одно из входящих в него высказываний (табл. 1.16).

Таблица 1 16 **Таблица истинности для логической суммы высказываний**

A	*B*	$A \vee B$	*A* XOR *B*
false	false	false	false
false	true	true	true
true	false	true	true
true	true	true	false

В последнем столбце табл. 1.16 размещены результаты модифицированной операции ИЛИ - ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR). Отличается от обычного ИЛИ последней строкой (см. также рис. 1.9, *в, г).*

Схема ИЛИ реализует дизъюнкцию двух или более логических значений. Когда хотя бы на одном входе схемы ИЛИ будет единица, на ее выходе также будет единица.

Условное обозначение на структурных схемах схемы ИЛИ с двумя входами представлено на рис. 1.9, *б*. Знак «1» на схеме — от классического обозначения дизъюнкции как «≥1» (т. е. значение дизъюнкции равно единице, если сумма значений операндов больше или равна 1). Связь между выходом z этой схемы и входами x и y описывается соотношением: $z = x \vee y$ (читается как «x ИЛИ y»).

Инверсия. Присоединение частицы НЕ (NOT) к некоторому высказыванию называется операцией отрицания (инверсии) и обозначается $\bar{A}$ (или $\neg A$). Если высказывание A истинно, то B ложно, и наоборот (табл. 1.17).

Таблица 1.17 **Таблица истинности отрицания**

A	$\bar{A}$
false	true
true	false

Схема НЕ (инвертор) реализует операцию отрицания. Связь между входом x этой схемы и выходом z можно записать соотношением $z = \bar{x}$, где x читается как «НЕ x» или «ИНВЕРСИЯ x»

Если на входе схемы «О», то на выходе «1», и наоборот. Условное обозначение на структурных схемах инвертора — на рис. 1.9, *в*.

Вентили. Кроме схемных элементов, соответствующих перечисленным логическим операторам, в состав логических схем входят комбинированные связки, именуемые *вентилями,* например следующие.

Схема И—НЕ состоит из элемента И и инвертора и осуществляет отрицание результата схемы И (табл. 1.18). Связь между выходом z и входами x и y схемы записывают как $\overline{x\&y}$, или «ИНВЕРСИЯ x И y». Условное обозначение на структурных схемах схемы И—НЕ с двумя входами представлено на рис. 1.9, *г*.

Таблица 1.18 **Таблица истинности схемы И—НЕ**

x	y	$\overline{x\&y}$
false	false	true
false	true	true
true	false	true
true	true	false

Схема ИЛИ—НЕ состоит из элемента ИЛИ и инвертора и осуществляет отрицание результата схемы ИЛИ (табл. 1.19). Связь между выходом z и входами x и y схемы записывают как $x \vee y$, или «ИНВЕРСИЯ x ИЛИ y». Условное обозначение на структурных схемах схемы ИЛИ—НЕ с двумя входами представлено на рис. 1 9, *д*.

Таблица } 19 **Таблица истинности схемы ИЛИ—НЕ**

x	y	$\overline{x \vee y}$
false	false	true
false	true	false
true	false	false
true	true	false

Схема «ИСКЛЮЧАЮЩЕЕ ИЛИ» (рис. 1.9, *е*) соответствует «сложению по модулю два» (см. также табл. 1.16).

Несколько слов о практической реализации схемных элементов в электронных цепях Может быть рассмотрен рис. 1.10, на котором достаточно упрощенно представлены транзисторные сборки И (последовательно включенные транзисторы) и ИЛИ (параллельное включение).

Входные и выходные сигналы «1» представляются высоким уровнем напряжения на коллекторе транзистора (практически равным напряжению питания). Сигналу «О», наоборот, соответствует низкий уровень выходного напряжения

Поскольку, например, в большинстве современных *персональных компьютеров* напряжение питания составляет 3,3 В (в более ранних

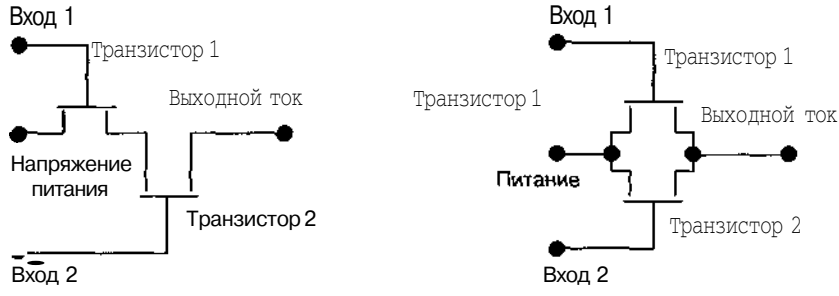


Рис. **1.10.** Пример реализации сборок И *(а)* и ИЛИ (б)

версиях, до Pentium, было 5 В), то выходная «1» задается напряжением 3,3 В.

Следует отметить, что помимо операций И, ИЛИ, НЕ в алгебре высказываний существует ряд других операций. Например, операция *эквиваленции* (эквивалентности) $A \sim B$ (или $A = B$, *A* EQV *B)* (табл. 1.20).

Таблица 1.20 Таблица истинности операции эквивалентности

A	*B*	$A \sim B$
false	false	true
false	true	false
true	false	false
true	true	true

Другим примером может служить логическая операция *импликации* или логического следования *(*$A \rightarrow B$, *A* IMP *Б),* иначе говоря, «ЕСЛИ *A,* то *B*» (табл. 1.21).

Таблица 1.21 **Таблица истинности импликации**

A	*B*	$A \rightarrow B$
false	false	true
false	true	true
true	false	false
true	true	true

Высказывания, образованные с помощью логических операций, называются сложными. Истинность сложных высказываний можно установить, используя таблицы истинности. Например, истинность сложного высказывания *A & B* определяется табл. 1.22.

Таблица 1 22 **Таблица истинности высказывания *A & B***

A	*B*	$\overline{A}$	*B*	$\overline{A}\&B$
false	false	true	true	true
false	true	true	false	false
true	false	false	true	false
true	true	false	false	false

Высказывания, у которых таблицы истинности совпадают, называются *равносильными.* Для обозначения равносильных высказываний используют знак «=» *(A = B).* Рассмотрим сложное высказывание *(A & B) \ (B & $\overline{B}$)* - табл. 1.23.

Таблица 1 23 **Таблица истинности выражения *(A & B) \ (B & $\overline{B}$)***

A	$\overline{A}$	*B*	$\overline{B}$	*A&B*	*B&* $\overline{B}$	*(B&* $\overline{B}$*) | (A & B)*
false	false	true	true	false	true	true
false	true	true	false	false	false	false
true	false	false	true	false	false	false
true	true	false	false	true	false	true

Если сравнить эту таблицу с таблицей истинности операции эквивалентности высказываний *A* и *B,* то можно увидеть, что высказывания *(A &* $\overline{B}$*) \| (B&* $\overline{B}$*)* и *A ~ B* тождественны, т. е. *A ~ B = = (A & B)* *(B &* $\overline{B}$*).*

В алгебре высказываний можно проводить тождественные преобразования, заменяя одни высказывания равносильными им другими высказываниями.

Свойства операций. Исходя из определений дизъюнкции, конъюнкции и отрицания, устанавливаются свойства этих операций и взаимные распределительные свойства. Приведем примеры некоторых из этих свойств.

Коммутативность (перестановочность)

A л $B = B \wedge A$

$A \vee B = B \vee A$

Закон идемпотентности

$A \& A = A,\ A \vee A = A.$

Двойное отрицание

$\overline{\overline{A}} = A$

Сочетательные (ассоциативные) законы

$A \vee (B \vee C) = (A \vee B) \vee C = A \vee B \vee C$

$A \wedge (B \wedge C) = (A \wedge B) \wedge C = A \wedge B \wedge C$

Распределительные (дистрибутивные) законы

$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$

A v $(B$ л $C) = (A$ v $B)$ л $(A$ v $C)$

Поглощение

x v (x л $y) = x$

x л (x v $y) = x$

Склеивание

$(x \text{ л } y) \text{ v } (\bar{x} \text{ л } y) = y$

$(x \vee y) \wedge (\bar{x} \vee y) = y$

Операция переменной с ее инверсией

$x \text{ v } \bar{x} = 1$

$x \text{ л } \bar{x} = 0$

Операция с константами

$x \text{ v } 0 = x, \quad x \vee 1 = 1$

$x \wedge 1 = x, \quad x \text{ л } 0 = 0$

Законы Де Моргана

1. $\overline{A \& B} = \bar{A} \text{ v } \bar{B}$ (условно его можно назвать 1-й);

2. $\overline{A \vee B} = \bar{A} \& \bar{B}$ (2-й) — описывает результаты отрицания переменных, связанных операциями И, ИЛИ.

Высказывания, образованные с помощью нескольких операций логического сложения, умножения и отрицания, называются сложными. Истинность всякого сложного высказывания устанавливается с помощью таблиц истинности. Сложные высказывания, истинные для любых значений истинности, входящих в них простых высказываний, называются *тождественно-истинными.* Наоборот, *тождественно-ложными* являются формулы, принимающие значение *(false)* для любых значений входящих в него простых высказываний.

В табл. 1.24 приведено доказательство истинности дистрибутивного закона. Аналогичным образом могут быть доказаны и другие тождества.

Таблица 1.24. **Доказательство истинности дистрибутивного** закона

A	B	C	$B \vee C$	$A \wedge (B \text{ v} C)$	$A \wedge B$	$A \wedge C$	$(A \wedge B) \vee (A \wedge C)$
false	false	false	false	false	false	false	false
false	false	true	true	false	false	false	false
false	true	false	true	false	false	false	false
false	true	true	true	false	false	false	false
true	false	false	false	false	false	false	false
true	false	true	true	true	false	true	true
true	true	false	true	true	true	false	true
true	true	true	true	true	true	true	true

На рис. 1.11, *а—ж* приведены иллюстрации к основным логическим операциям и их композициям (так называемые диаграммы Эйлера—Венна). В качестве высказывания A здесь принято утвер-

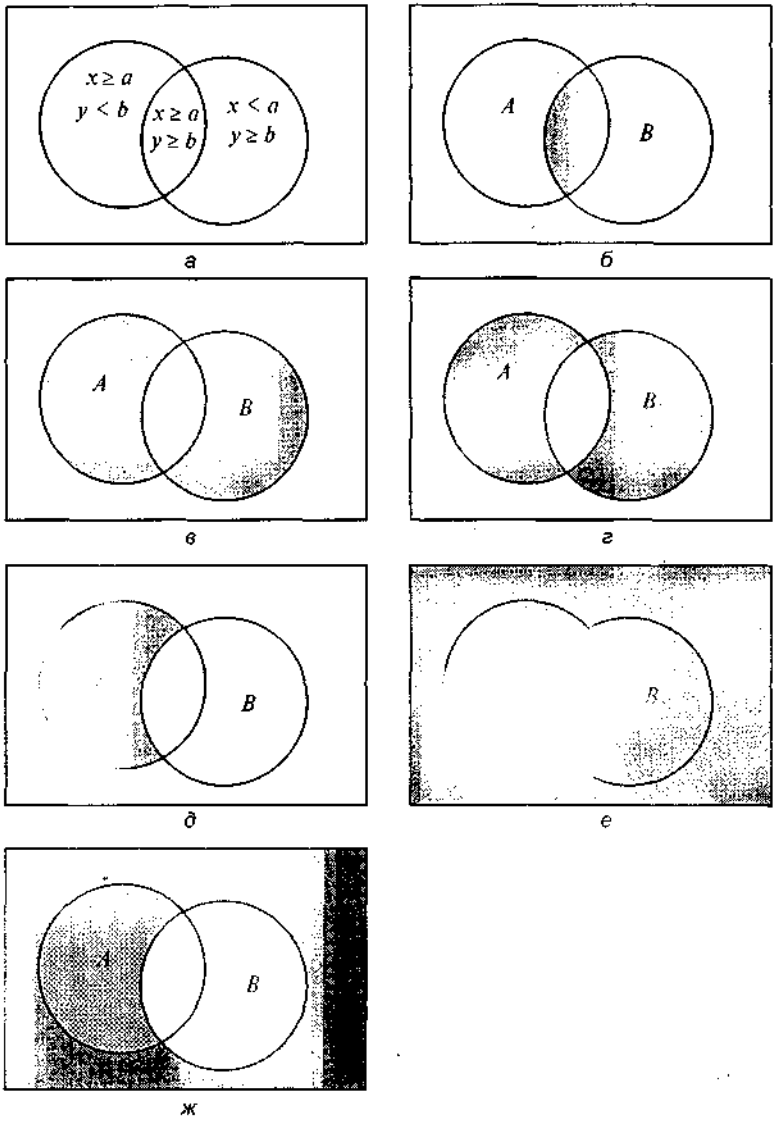


Рис. 1.11. Некоторые примеры диаграмм Эйлера—Венна:
а — диаграмма Эйлера—Венна, иллюстрирующая расположение областей истинности высказываний *A* и *B*; *б* — конъюнкция высказываний *A* и *B* (AND); в — дизъюнкция высказываний *A* и *B* (OR); г — исключающая дизъюнкция (XOR); *д* — разность высказываний *(A - B); е* — иллюстрация к законам де Моргана (дополнение пересечению высказываний); *ж* — иллюстрация к законам де Моргана (объединение дополнений)

ждение $x > a$, в качестве $B - y > b$. На рис. 1.11, *а* приведены области истинности каждого из высказываний, здесь же становится понятен смысл дополнения (отрицания), объединения (дизъюнкции), пересечения (конъюнкции) и других операций. Первый из законов де Моргана иллюстрируется рис. 1.11, *е, ж.*

Логическое значение null. В некоторых ЯП (Visual Basic и пр.) для расширения применимости логических выражений на те случаи, когда значения одного или нескольких логических аргументов неизвестны или не определены, вводится значение *null* (в дополнение к *false* и *true),* как правило, такое значение присваивается компилятором логической переменной по умолчанию.

С учетом значения *null* таблицы истинности основных логических операций приобретают следующий вид (табл. 1.25, 1.26).

Таблица 1.25. **Одноместная (унарная) операция отрицания с учетом значения *null***

A	$\overline{A}$
false	true
true	false
null	null

Таблица 1.26. **Некоторые двухместные (бинарные) операции с учетом значения *null***

A	B	A & B	A v B	$A \rightarrow B$	$A \sim B$	A XOR B
false	false	false	false	true	true	false
false	true	false	true	true	false	true
true	false	false	true	false	false	true
true	true	true	true	true	true	false
false	null	false	false	true	null	null
true	null	null	true	null	null	null
null	false	false	null	null	null	null
null	true	null	true	true	null	null
null	null	null	null	null	null	null

Побитовые операции. В некоторых современных ЯП включены операции побитового сравнения содержимого машинных слов (которые могут содержать числовые, строчные и др. данные), при этом каждый бит результата образуется в соответствии с табл. 1.27 (для бинарных операций). Унарная операция отрицания (NOT) в данном случае реализует очевидную замену «1» на «О» и наоборот.

Таблица 1.27 **Операнды и результаты некоторых операций побитового сравнения**

x	*y*	*x* & *y*	*x* v *y*	*x* IMP *y*	*x* EQV *y*	*x* XOR *y*
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	0	0	1
1	1	1	1	1	1	0

Другие схемные элементы ЭВМ

Триггер. Данное устройство — это электронная схема, широко применяемая в регистрах компьютера для запоминания одного разряда двоичного кода. Триггер имеет два устойчивых состояния, одно из которых соответствует двоичной единице, а другое — двоичному нулю.

Термин *триггер* происходит от английского слова *trigger* — защелка, спусковой крючок. Для обозначения этой схемы в английском языке чаще употребляется термин *flip-flop,* что в переводе означает «хлопанье». Это звукоподражательное название электронной схемы указывает на ее способность мгновенно переходить («перебрасываться») из одного состояния в другое и обратно.

Самый распространенный тип триггера — так называемый *RS*-триггер *(S* и *R* соответственно от английских *set* — установка и *reset* — сброс). Условное обозначение *RS*-триггера приводится на рис. 1.12, *а.*

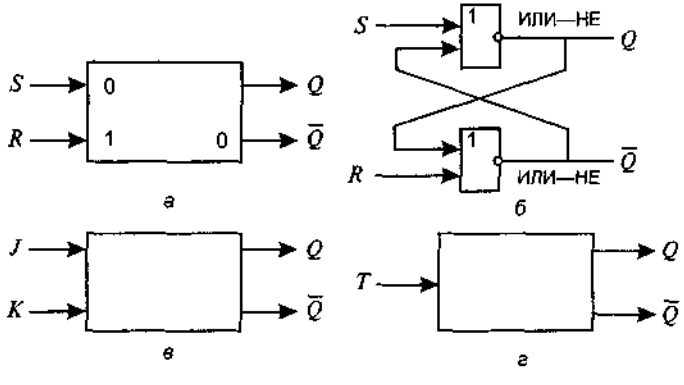


Рис. 1.12. Варианты триггерных цепей
а — *RS*-триггер, *б* — его реализация, *в* — *JK*-триггер; *г* — *T*-триггер

Он имеет два симметричных входа S и R и два симметричных выхода Q и $\overline{Q}$, причем выходной сигнал Q является логическим отрицанием сигнала $\overline{Q}$.

На каждый из двух входов S и R могут подаваться входные сигналы в виде кратковременных импульсов (_⎍_). Наличие импульса на входе считается единицей, а его отсутствие — нулем.

На рис. 1.12, *б* показана реализация триггера с помощью вентилей И—НЕ.

Перечислим возможные комбинации значений входов R и S триггера, используя его схему и таблицу истинности схемы ИЛИ-НЕ (табл. 1.28).

1. Если на входы триггера подать $S = 1$, $R = 0$, то (независимо от состояния) на выходе Q верхнего вентиля появится «О». После этого на входах нижнего вентиля окажется $R = $ О, $Q = 0$ и выход $\overline{Q}$ станет равным «1».

2. При подаче «О» на вход S и «1» на вход R на выходе $\overline{Q}$ появится «О», а на Q «1».

3. Если на входы R и S одновременно подан логический «О», то состояние Q и $\overline{Q}$ не меняется.

4. Подача на оба входа R и S логической «1» может привести к неоднозначному результату, поэтому эта комбинация входных сигналов запрещена.

Таблица 1.28. **Таблица истинности для *RS*-триггера**

S	R	Q	$\overline{Q}$
0	0	Без изменений	
0	1	1	0
1	0	0	1
1	1	Не определено	

Поскольку триггер может запомнить только один разряд двоичного кода, для запоминания байта нужно 8 триггеров, для запоминания килобайта соответственно $8 \times 2^{10} = 8192$ триггеров. Современные микросхемы памяти содержат миллионы триггеров.

Кроме *RS*-триггеров известны также *JK*- и *T*-триггеры. *JK*-триггер содержит схемные дополнения, которые снимают неопределенность состояния при подаче ⎍ на оба входа. Теперь при этом происходит «переброс» схемы в противоположное состояние ($\overline{Q}$ и Q меняются местами — «О» переходит в «1» и наоборот) — табл 1.29.

Таблица 1.29. Таблица истинности для ***JK*-триггера**

У	*K*	*Q*	$\overline{Q}$
0	0	Без изменений	
0	1	1	0
1	0	0	1
1	1	Переброс состояния	

T-триггер имеет единственный вход *(T),* при подаче _⎍_ на который осуществляется «переброс» схемы (см. табл. 1.30).

Таблица 1.30. **Иллюстрация к действию *T*-триггера**

T	*Q*	$\overline{Q}$
0	Без изменений	
1	Переброс состояния	

T-триггеры могут использоваться для создания двоичных *счетчиков* (например, счетчик адреса команд, обеспечивающий последовательную выборку слов из оперативной памяти).

Полусумматор. Вспомним, что при сложении двоичных чисел образуется сумма в данном разряде, при этом возможен перенос в старший разряд. Обозначим слагаемые *(A, B),* перенос (*P*), сумму (*S*) и рассмотрим соответствующую данной операции табл. 1.31.

Таблица 1.31. **Таблица сложения одноразрядных двоичных чисел с учетом переноса в старший разряд**

Слагаемые		Перенос	Сумма
A	*B*	*P*	*S*
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Из этой таблицы очевидно, что перенос можно реализовать с помощью операции логического умножения

$$P = A \,\&\, B.$$

Получим теперь формулу для вычисления суммы. Значения суммы более всего совпадают с результатом операции логического сложения (кроме случая, когда на вход подаются две единицы, а на выходе должен получиться нуль).

Нужный результат достигается, если результат логического сложения умножить на инвертированный перенос. Таким образом, для определения суммы можно применить выражение:

$$S = A \vee B \,\&\, \overline{A \,\&\, B}.$$

Теперь, на основе полученных логических выражений, можно построить из базовых логических элементов схему полусумматора.

Из логической формулы для суммы очевидно, что на выходе должен стоять элемент логического умножения И, который имеет два входа. На один из входов подается результат логического сложения исходных величин, т. е. на него должен подаваться сигнал с элемента логического сложения ИЛИ.

На второй вход требуется подать результат инвертированного логического умножения исходных сигналов $\overline{A \,\&\, B}$, т. е. на второй вход подается сигнал с элемента НЕ, на вход которого поступает сигнал с элемента логического умножения И.

Данная схема называется полусумматором, так как реализует суммирование одноразрядных двоичных чисел без учета переноса из младшего разряда (рис. 1.13).

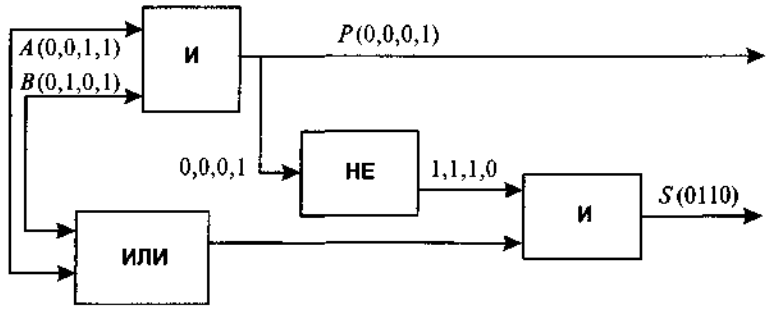


Рис. 1.13. Логическая схема полусумматора

Полный одноразрядный сумматор. Полный одноразрядный сумматор должен иметь три входа, a_i, b_i — слагаемые и p_{i-1} — перенос из предыдущего разряда и два выхода, сумма с, и перенос p_i. Порядок функционирования схемы иллюстрирует табл. 1.32

Таблица 1 32 Таблица сложения для полного одноразрядного сумматора

a_i	b_i	p_{i-1}	p_i	S,
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

Идея построения полного сумматора точно такая же, как и полусумматора. Перенос реализуется с помощью формулы для его получения

$$p_i = (a_i \& b_i) \vee (a_i \& p_{i-1}) \vee (b_i \& p_{i-1}).$$

Логическое выражение для вычисления суммы в полном сумматоре принимает следующий вид:

$$S, = (a_i \vee b, \vee p_{i-1}) \& p_{i-1} \vee (a_i \& b_i \& p_{i-1}).$$

Укрупненная схема, соответствующая полному сумматору, приведена на рис. 1.14. Составить соответствующую схему из вентилей мы рекомендуем читателю самостоятельно. Более сложные комбинации логических элементов (узлы) будут рассмотрены далее.

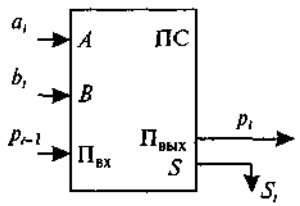


Рис. 1.14. Укрупненная схема одноразрядного сумматора

Преобразования логических формул

Равносильные преобразования логических формул имеют то же назначение, что и преобразования формул в обычной алгебре. Они служат для упрощения формул или приведения их к определенному виду путем использования основных законов алгебры логики.

Под *упрощением формулы,* не содержащей операции импликации и эквиваленции, понимают *равносильное преобразование,* приводящее к формуле, которая либо содержит по сравнению с исходной меньшее число операций конъюнкции и дизъюнкции и не содержит отрицаний неэлементарных формул, либо содержит меньшее число вхождений переменных.

Некоторые преобразования логических формул похожи на преобразования формул в обычной алгебре (вынесение общего множителя за скобки, использование переместительного и сочетательного законов и т. п.), тогда как другие преобразования основаны на свойствах, которыми не обладают операции обычной алгебры (использование распределительного закона для конъюнкции, свойств поглощения и склеивания, законы де Моргана и др.).

Приведем несколько примеров, иллюстрирующих методы, применяемые при упрощении логических формул:

1) $$\overline{(\bar{x} \vee y)} \wedge (x \wedge \bar{y}) = \bar{\bar{x}} \wedge \bar{y} \wedge (x \wedge \bar{y}) = \bar{\bar{x}} \wedge x \wedge \bar{y} = \bar{y}$$
$$= 0 \text{ л у л } y = 0 \wedge y = 0$$

(законы алгебры логики применяются в следующей последовательности: закон Де Моргана, сочетательный закон, правило конъюнкции переменной с ее инверсией и правило операций с константами);

2) $$\bar{x} \wedge y \vee \overline{x \vee y} \vee x = \bar{x} \wedge y \vee \bar{x} \wedge \bar{y} \vee x =$$
$$= \bar{x} \wedge (y \vee \bar{y}) \vee x = \bar{x} \vee x = 1$$

(применяется правило де Моргана, выносится за скобки общий множитель, используется правило операций переменной с ее инверсией);

3) $$(x \vee y) \text{л } (\bar{x} \vee y) \text{ л } (\bar{x} \vee \bar{y}) =$$
$$(x \vee y) \text{ л } (\bar{x} \vee y)(\bar{x} \vee y) \text{ л } (x \vee y) = y \text{ л } x$$

(повторяется второй сомножитель, что разрешено законом идемпотенции; затем комбинируются два первых и два последних сомножителя и используется закон склеивания);

4) $$x \wedge y \vee x \wedge y \wedge z \vee x \wedge \bar{z} =$$
$$= x \wedge \bar{y} \vee \bar{x} \wedge y \wedge z \vee x \wedge z \wedge (y \vee \bar{y}) =$$
$$x \wedge \bar{y} \vee \bar{x} \wedge y \wedge z \vee x \wedge y \wedge z \vee x \wedge \bar{y} \wedge z =$$
$$(x \wedge \bar{y} \vee x \wedge \bar{y} \wedge z) \vee (\bar{x} \wedge y \wedge z \vee x \wedge y \wedge z) = x \wedge \bar{y} \vee y \wedge z$$

(вводится вспомогательный логический сомножитель $(y \vee \bar{y})$; затем комбинируются два крайних и два средних логических слагаемых и используется закон поглощения);

5) $$\overline{x \wedge y \vee z} = \overline{x \wedge y} \wedge \bar{z} = (\bar{x} \vee \bar{y}) \wedge z$$

(сначала добиваемся, чтобы знак отрицания стоял только перед отдельными переменными, а не перед их комбинациями, для этого дважды применяется закон де Моргана; затем двойное отрицание);

6) $$x \wedge y \vee x \wedge y \wedge z \vee x \wedge z \wedge p =$$
$$= X \wedge (y \wedge (1 \vee z) \vee Z \wedge p) = X \wedge (y \vee z \wedge p)$$

(выносятся за скобки общие множители; применяется правило операций с константами);

7) $$x \vee \overline{y \wedge z} \vee \overline{x \vee y \vee z} = x \vee \bar{y} \vee \bar{z} \vee \bar{x} \wedge \bar{y} \wedge \bar{z} =$$
$$= x \vee \bar{y} \vee z \vee x \wedge \bar{y} \wedge z = x \vee \bar{z} \vee (\bar{y} \vee x \wedge \bar{y} \wedge z) = x \vee z \vee \bar{y}$$

(к отрицаниям неэлементарных формул применяется закон де Моргана; используются двойное отрицание и склеивание);

8) $$x \wedge \bar{y} \vee x \wedge y \wedge z \vee x \wedge \bar{y} \wedge z \vee x \wedge \overline{y \wedge z} =$$
$$= x \wedge (\bar{y} \vee y \wedge z \vee \bar{y} \wedge z \vee \overline{y \wedge z} =$$
$$= x \wedge ((\bar{y} \vee y \wedge z) \vee (y \wedge z \vee \bar{y} \wedge \bar{z})) =$$
$$= x \wedge (\bar{y} \vee \bar{y} \wedge z \vee 1) = x \wedge 1 = x$$

(общий множитель x выносится за скобки, комбинируются слагаемые в скобках — первое с третьим и второе с четвертым, к дизъюнкции применяется правило операции переменной с ее инверсией);

9) $$(x \wedge y \vee z) \wedge (\bar{x} \vee y) \vee Z =$$
$$= x \wedge \bar{y} \wedge \bar{x} \vee x \wedge \bar{y} \wedge y \vee z \wedge \bar{x} \vee z \wedge y \vee \bar{z} =$$
$$= 0 \vee 0 \vee z \wedge x \vee z \wedge y \vee \bar{z} = z \wedge \bar{x} \vee (z \vee \bar{z}) \wedge (y \vee \bar{z}) =$$
$$= z \wedge \bar{x} \vee 1 \wedge (y \vee \bar{z}) = z \wedge \bar{x} \vee y \vee \bar{z} = (z \wedge \bar{x} \vee \bar{z}) \vee y =$$
$$= (z \vee \bar{z}) \wedge (\bar{x} \vee \bar{z}) \vee y = 1 \wedge (\bar{x} \vee \bar{z}) \vee y = \bar{x} \vee \bar{z} \vee y$$

(используются распределительный закон для дизъюнкции, правило операции переменной с ее инверсией, правило операций с константами, переместительный закон и распределительный закон для конъюнкции);

10) $$x \wedge y \wedge (\bar{x} \wedge z \vee \overline{\overline{x \wedge y} \wedge z} \vee z \wedge t) =$$
$$= x \wedge y \wedge (\bar{x} \wedge z \vee \overline{\overline{x \wedge y}} z \vee z \wedge t) =$$
$$= x \wedge y \wedge (\bar{x} \wedge z \vee x \wedge y \vee \bar{z} \vee z \wedge t) =$$
$$= x \wedge y \vee x \wedge y \wedge \bar{z} \vee x \wedge y \wedge z \wedge t = x \wedge y$$

(используются закон де Моргана, двойное отрицание и поглощение).

Минимизация логического выражения

Целью минимизации логического выражения, представляющего заданную логическую функцию, является уменьшение стоимости ее реализации (количества используемых логических элементов). Общая схема процесса реализации логической функции такова. Для нее составляется сумма произведений, или *дизъюнктивная совершенная нормальная форма.* Затем полученное выражение минимизируют до эквивалентной *минимальной суммы произведений.* Чтобы определить критерий минимизации, вводится понятие *стоимости, или величины,* логического выражения.

Обычно при оценке стоимости выражения учитывается общее количество вентилей и их входных значений (входных линий), необходимых для реализации выражения в форме вида, приведенного на рис. 1.15.

Стоимость большей схемы (рис. 1.15, *а)* равна 21: 5 вентилей плюс 16 входных значений.

Инверсия входных значений при подсчете игнорируется. Стоимость более простого выражения (рис. 1.15, *б)* равна 9: 3 вентиля плюс 6 входных значений.

Теперь можно определить критерий минимизации: *сумма произведений считается минимальной, если не существует эквивалентного ей выражения меньшей стоимости.*

Стратегия упрощения заданного выражения заключается в следующем. Прежде всего термы-произведения разбиваются на пары, отличающиеся единой переменной, которая в одном терме стоит со знаком $\neg$ ($\neg x$), а во втором — без него (x). Затем в каждой паре общее произведение двух переменных выносится за скобки, а в скобках остается терм $\neg x + x$, всегда равный 1. Вот что мы получим, при-

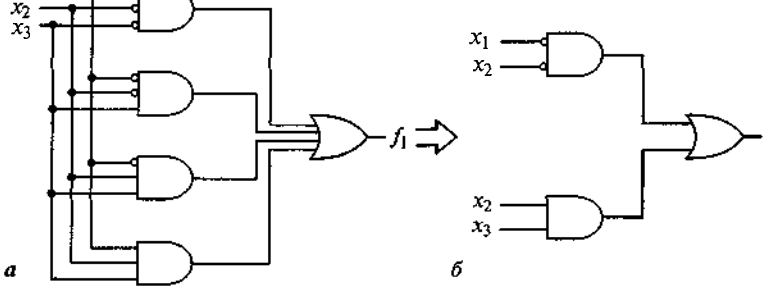


Рис. 1.15. Минимизация логических схем

менив эту процедуру к первому выражению для функции f_1 (на рис. 1.15, а)

$$f_1 = \neg x_1 \neg x_2 \neg x_3 + \neg x_1 \neg x_2 \neg x_3 + \neg x_1 \neg x_2 \neg x_3 + x_1 x_2 x_3 =$$
$$= \neg x_1 \neg x_2 (\neg x_3 + x_3) + (\neg x_1 + x_1) x_2 x_3 =$$
$$= \neg x_1 \neg x_2 \cdot 1 + 1 \cdot x_2 x_3 =$$
$$= \neg x_1 \neg x_2 + x_2 x_3.$$

Это выражение минимально. Соответствующая ему логическая схема приведена на рис 1.15, *б.*

Узлы ЭВМ

Узлами ЭВМ являются стандартизованные наборы логических элементов, из которых, как из «кирпичиков», набираются схемы, входящие в состав микропроцессоров, блоков памяти, контроллеров внешних устройств и пр.

Узлы ЭВМ разделяются на:

- *комбинационные,* или узлы, выходные сигналы которых определяются только сигналом на входе, действующим в настоящий момент времени (например, дешифратор). Выходной сигнал дешифратора зависит только от двоичного кода, поданного на вход в настоящий момент времени. Комбинационные узлы называют также *автоматами без памяти*;
- *последовательностные (автоматы с памятью)* — это узлы, выходной сигнал которых зависит не только от комбинации входных сигналов, действующих в настоящий момент времени, но и от предыдущего состояния узла (счетчик);
- *программируемые узлы* функционируют в зависимости от того, какая программа в них записана. Например, программируемая логическая матрица (ПЛМ), которая в зависимости от встроенной («прожженной») в ней программы может выполнять функции сумматора, дешифратора, ПЗУ.

Сумматоры. Многоразрядный сумматор процессора состоит из полных одноразрядных сумматоров (рис. 1.16). На каждый разряд ставится одноразрядный сумматор, причем выход (перенос) сумматора младшего разряда подключен ко входу сумматора старшего разряда.

Например, схема вычисления суммы $5 = (s_3 s_2 s_1 s_0)$ двух двоичных трехразрядных чисел $A = (a_2 a_1 a_0)$ и $B = (b_2 b_1 b_0)$ может иметь вид, приведенный на рис. 1.16.

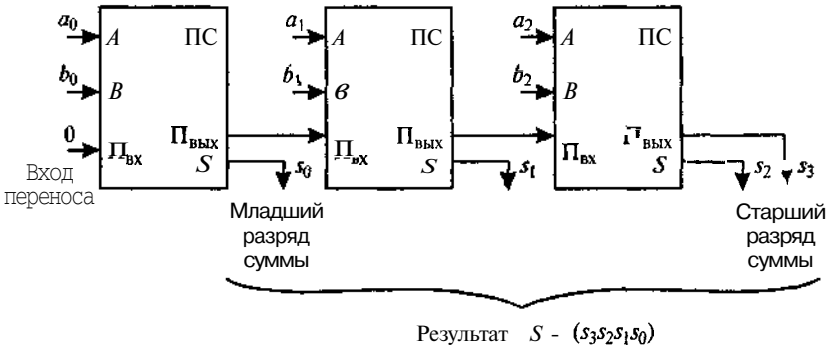


Рис. 1.16. Фрагмент (3 разряда) схемы многоразрядного сумматора

Сумматор может быть построен в двух вариантах.

- комбинационная схема (последовательный сумматор);
- последовательностная схема (накапливающий сумматор)

Последовательный сумматор (рис 1 17) осуществляет суммирование слагаемых и цифр переноса поразрядно начиная с младшего разряда Основой его схемы является одноразрядный сумматор Суммирование производится в одноразрядном сумматоре *SM* (рис 1 17)

Цифры *i*-го разряда слагаемого и цифра переноса из младшего разряда передаются на вход сумматора одновременно с приходом тактового импульса Регистры *1* и *2* используются для приема и хра-

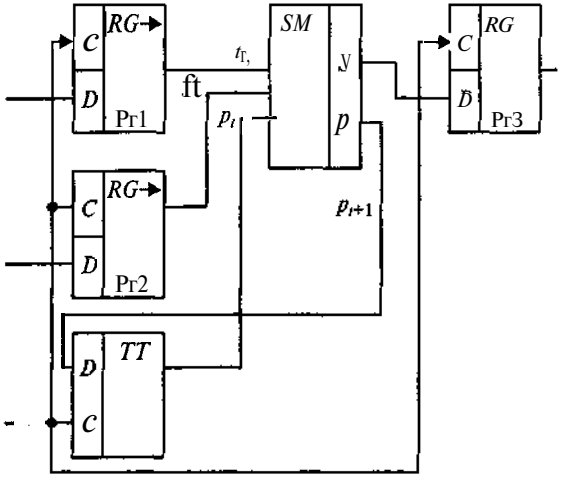
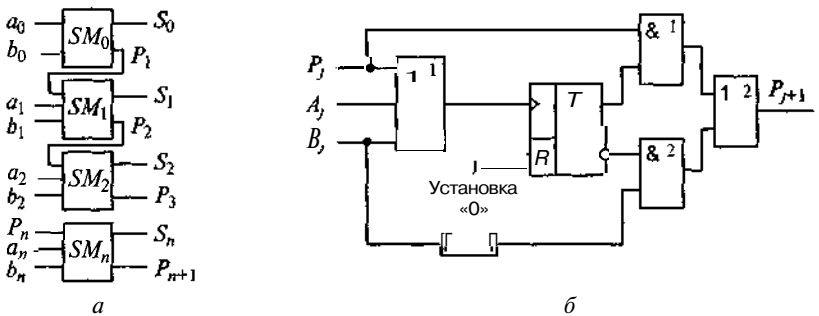


Рис. 1.17. Последовательный сумматор

нения цифр i-го разряда слагаемых В триггере *TT* хранится цифра переноса из младшего разряда Регистр *3* принимает и хранит г-ю цифру суммы С приходом тактового импульса из регистров *1, 2* и триггера *TT* разряды слагаемых и цифра переноса поступают на вход одноразрядного сумматора Одновременно регистр *3* освобождается для приема цифры суммы

В параллельном сумматоре все разряды операндов суммируются одновременно, но быстродействие снижается за счет времени передачи цифры переноса из младшего разряда (рис 1 18, *а)*



а *б*

Рис. **1.18.** Схемы параллельного (а) и накапливающего (б) сумматоров

Накапливающий сумматор является автоматом с памятью, т е слагаемые могут приходить поочередно в произвольные моменты времени и запоминаться в линиях задержки или триггерах Накапливающий сумматор применяется в асинхронных устройствах, в которых слагаемые не привязаны к тактам тактового генератора (рис 1 18, *б*)

С приходом слагаемого $a_i = 1$ элемент ИЛИ и триггер устанавливаются в «1» Если $b = 1$ и приходит через какое-то время после a_i, то оно запоминается в линии задержки и одновременно b_i «опрокидывает» триггер в «О» На инверсном выходе триггера устанавливается «1», следовательно, на вторую схему И подаются две единицы, следовательно, на выходе второй схемы ИЛИ формируется цифра переноса в старший разряд, равная «1» Если $P_i = 0$, то цифра суммы, которая снимается с прямого выхода триггера, равна «О» Если $P_i = 1$, то сумма $S_i = 1$

Дешифраторы. Схемы предназначаются для преобразования двоичного кода *(X)* на входе в управляющий сигнал (Z) на одном из выходов Если входов и, то выходных шин должно быть $N = 2^n$ (табл 1 33, $n = 3$, $N = 8$)

Таблица 1 33. **Пример таблицы состояний дешифратора**

X_1	X_2	X_3	Z_0	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Дешифраторы могут быть линейными и многокаскадными.

У линейных дешифраторов все переменные X_1, X_2, X_3 подаются на вход одновременно (рис. 1.19, *а).* Они обладают более высоким бы-

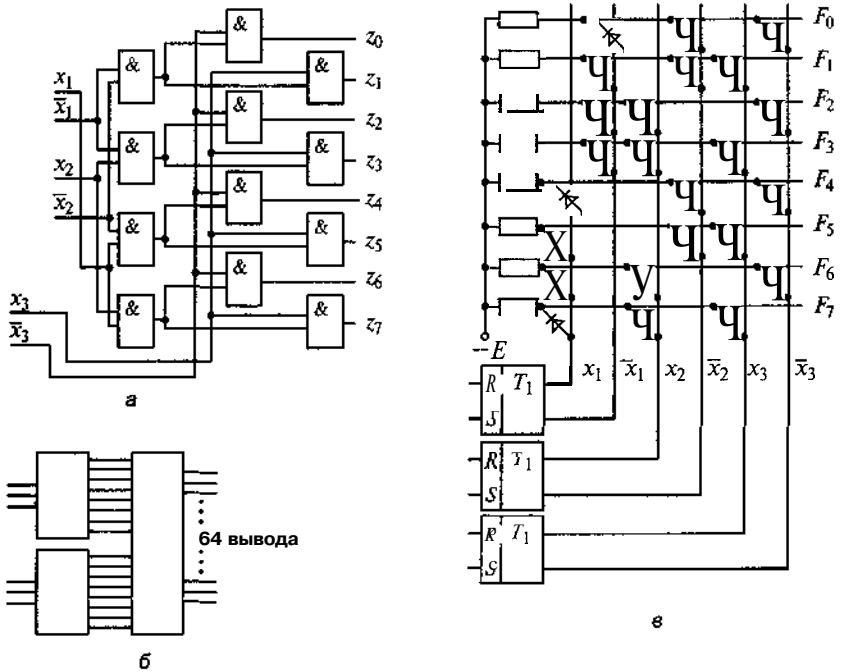


Рис. 1.19. Линейный дешифратор (а), диодная матрица (б), многокаскадный дешифратор (в)

стродействием, но более трех переменных одновременно подать нельзя, поэтому чаще применяются *многокаскадные* дешифраторы. Здесь количество элементов в каждом следующем разряде больше, чем в предыдущем. На вход первого каскада подается один слог, на вход следующего каскада — второй слог и результаты конъюнкций, произведенных в первом каскаде.

Простейший линейный дешифратор можно построить на *диодной матрице* (рис. 1.19, *б*). В этой схеме используется отрицательная логика. При подаче «1» на анод (коллектор) диода он закрывается. Если закрыты все три диода, подсоединенные к одной горизонтальной линии, то на этой линии появляется потенциал -Е, соответствующий уровню «1».

Многокаскадный дешифратор можно организовать так, как это изображено на рис. 1.19, *в*. Два линейных дешифратора обрабатывают по два слова. В последнем каскаде образуются конъюнкции выходного сигнала первого каскада. Многокаскадные дешифраторы обладают меньшим быстродействием.

1.5. Алгоритмы и программы

Понятие *алгоритма* является одним из основных в современной науке и практике. Еще на самых ранних ступенях развития математики (Древний Египет, Вавилон, Греция) в ней стали рассматриваться различные вычислительные процессы чисто механического характера. С их помощью искомые величины ряда задач вычислялись последовательно из исходных величин по определенным правилам и инструкциям. Со временем все такие процессы в математике получили название алгоритмов (алгорифмов).

Алгоритм есть совокупность четко определенных правил, процедур или команд, обеспечивающих решение поставленной задачи за конечное число шагов.

Термин *алгоритм* происходит от имени средневекового узбекского математика Аль-Хорезми, который еще в IX в. (825 г.) дал правила выполнения четырех арифметических действий в десятичной системе счисления. Процесс выполнения арифметических действий был назван *алгоризмом*.

С 1747 г. вместо слова *алгоризм* стали употреблять *алгорисмус,* смысл которого состоял в комбинировании четырех операций арифметического исчисления — сложения, вычитания, умножения, деления.

К 1950 г. *алгорисмус* стал *алгорифмом*. Смысл алгорифма чаще всего связывался с алгорифмами Евклида — процессами нахождения наибольшего общего делителя двух многочленов, наибольшей общей меры двух отрезков и т. п.

Важнейшими характеристиками алгоритма являются: выбранный язык описания, однозначность определения цели, расчлененность на отдельные элементарные акты и достижение искомого результата. Благодаря этому, алгоритм является точным описанием процесса обработки или передачи данных. Алгоритмы описываются в виде правил, формул, программ. Нахождение алгоритмов решения различных классов задач является одной из целей математики.

Способы записи алгоритмов

Алгоритм должен быть понятен (доступен) пользователю и/или машине. Доступность пользователю означает, что он обязан отображаться посредством конкретных формализованных изобразительных средств, понятных пользователю. В качестве таких изобразительных средств используются следующие способы их записи:

- словесный;
- формульный;
- табличный;
- операторный;
- графический;
- макроязык программирования.

При словесном способе записи содержание последовательных этапов алгоритма описывается в произвольной форме на естественном языке.

Формульный способ основан на строго формализованном аналитическом задании необходимых для исполнения действий.

Табличный способ подразумевает отображение алгоритма в виде таблиц, использующих аппарат реляционного исчисления и алгебру логики для задания подлежащих исполнению взаимных связей между данными, содержащимися в таблице.

Операторный способ базируется на использовании для отображения алгоритма условного набора специальных операторов: арифметических, логических, печати, ввода данных и т. д.; операторы снабжаются индексами и между ними указываются необходимые переходы, а сами индексированные операторы описываются чаще всего в табличной форме.

Графическое отображение алгоритмов в виде блок-схем — весьма наглядный и распространенный способ. Графические символы,

отображающие выполняемые процедуры, стандартизованы. Наряду с основными символами используются и вспомогательные, поясняющие процедуры и связи между ними.

Алгоритмы могут быть записаны и в виде команд какого-либо языка программирования. Если это макрокоманды, то алгоритм читаем и пользователем-программистом, и вычислительной машиной, имеющей транслятор с соответствующего языка.

Приведем пример словесного представления алгоритма на примере нахождения произведения n натуральных чисел $(c = n! = 1 \cdot 2 \ 3 \cdot 4 \cdot \ldots \cdot n)$.

Этот процесс может быть записан в виде следующей системы последовательных указаний (пунктов):

1. Полагаем c равным единице и переходим к следующему пункту.

2. Полагаем i равным единице и переходим к следующему пункту.

3. Полагаем c равным $c = i \cdot c$ и переходим к следующему указанию.

4. Проверяем, равно ли i числу n. Если $i = n$, то вычисления прекращаем. Если $i <$ я, то увеличиваем i на единицу и переходим к пункту 3.

Классификация и свойства алгоритмов

Алгоритмы, в соответствии с которыми решение поставленных задач сводится к арифметическим действиям, называются *численными алгоритмами.*

Алгоритмы, в соответствии с которыми решение поставленных задач сводится к логическим действиям, называются *логическими алгоритмами.* Примерами логических алгоритмов могут служить алгоритмы поиска минимального числа, поиска пути на графе, в лабиринте и др.

Алгоритмом является последовательность четких однозначных указаний, которые, будучи применены к определенным имеющимся данным, обеспечивают получение требуемого результата. *Данными* называют все величины, участвующие в решении задачи. Данные, известные перед выполнением алгоритма, являются начальными, *исходными данными.* Результат решения задачи — это конечные, *выходные данные.*

Каждое указание алгоритма предписывает исполнителю выполнить одно конкретное законченное действие. Исполнитель не может перейти к выполнению следующей операции, не закончив полно-

стью выполнения предыдущей. Предписания алгоритма надо выполнять последовательно одно за другим, в соответствии с указанным порядком их записи. Выполнение всех предписаний гарантирует правильное решение задачи.

Поочередное выполнение команд алгоритма за конечное число шагов приводит к решению задачи, к достижению цели. Разделение выполнения решения задачи на отдельные операции (выполняемые исполнителем по определенным командам) — важное свойство алгоритмов, называемое *дискретностью.*

Анализ примеров различных алгоритмов показывает, что запись алгоритма распадается на отдельные указания исполнителю выполнить некоторое законченное действие. Каждое такое указание называется *командой.* Команды алгоритма выполняются одна за другой. После каждого шага исполнения алгоритма точно известно, какая команда должна выполняться следующей. Алгоритм представляет собой последовательность команд (также — инструкций, директив), определяющих действия исполнителя (субъекта или управляемого объекта).

Таким образом, выполняя алгоритм, исполнитель может не вникать в смысл того, что он делает, и вместе с тем получать нужный результат. В таком случае говорят, что исполнитель действует формально, т. е. отвлекается от содержания поставленной задачи и только строго выполняет некоторые правила, инструкции.

Это очень важная особенность алгоритмов. Наличие алгоритма формализовало процесс, исключило рассуждения. Если обратиться к примерам других алгоритмов, то можно увидеть, что и они позволяют исполнителю действовать формально. Таким образом, создание алгоритма дает возможность решать задачу формально, механически исполняя команды алгоритма в указанной последовательности.

Построение алгоритма для решения задачи из какой-либо области требует от человека глубоких знаний в этой области, бывает связано с тщательным анализом поставленной задачи, сложными, иногда очень громоздкими рассуждениями. На поиски алгоритма решения некоторых задач могут быть потрачены многие годы. Но когда алгоритм создан, решение задачи по готовому алгоритму уже не требует каких-либо рассуждений и сводится только к строгому выполнению команд алгоритма.

Всякий алгоритм составляется в расчете на конкретного исполнителя с учетом его возможностей. Для того чтобы алгоритм мог быть выполнен, нельзя включать в него команды, которые исполнитель не в состоянии выполнить. У каждого исполнителя имеется свой перечень команд, которые он может исполнить. Совокупность

команд, которые могут быть выполнены исполнителем, называется *системой команд исполнителя.*

Каждый алгоритм строится в расчете на некоторого исполнителя. Для того чтобы исполнитель мог решить задачу по заданному алгоритму, необходимо, чтобы он был в состоянии понять и выполнить каждое действие, предписываемое командами алгоритма. Каждая команда алгоритма должна определять однозначное действие исполнителя. Такое свойство алгоритмов называется *определенностью (или точностью) алгоритма.*

Алгоритм, составленный для конкретного исполнителя, должен включать только те команды, которые входят в его систему команд. Это свойство алгоритма называется *понятностью.* Алгоритм не должен быть рассчитан на принятие каких-либо самостоятельных решений исполнителем, не предусмотренных составлением алгоритма.

Еще одно важное требование, предъявляемое к алгоритмам, — *результативность (или конечность)* алгоритма. Оно означает, что исполнение алгоритма должно закончиться за конечное число шагов.

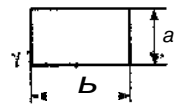
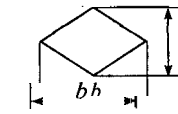
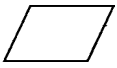
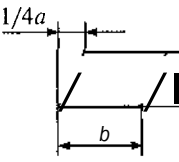
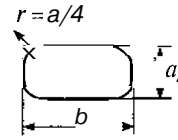
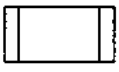
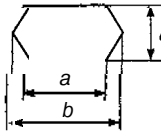
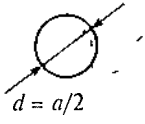
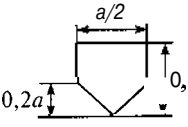
Поскольку разработка алгоритмов — процесс творческий, требующий умственных усилий и затрат времени, предпочтительно разрабатывать алгоритмы, обеспечивающие решения всего класса задач данного типа. Например, если составляется алгоритм решения кубического уравнения $ax^3 + bx^2 + cx + d = 0$, то он должен быть *вариативен,* т. е. обеспечивать возможность решения для любых допустимых исходных значений коэффициентов a, b, c, d. Про такой алгоритм говорят, что он удовлетворяет требованию *массовости.* Свойство массовости не является необходимым свойством алгоритма. Оно скорее определяет качество алгоритма; в то же время свойства точности, понятности и конечности являются необходимыми (иначе это не алгоритм).

Запись алгоритмов в виде блок-схем

Алгоритмы можно записывать по-разному. Форма записи, состав и количество операций алгоритма зависят от того, кто будет исполнителем этого алгоритма. Если задача решается с помощью ЭВМ, алгоритм решения задачи должен быть записан в понятной для машины форме, т. е. в виде программы.

Схема алгоритма — графическое представление алгоритма, дополняемое элементами словесной записи. Каждый пункт алгоритма отображается на схеме некоторой геометрической фигурой, или блоком. При этом правило выполнения схем алгоритмов регламентирует ГОСТ 19.002—80 «Единая система программной документации» (табл. 1.34).

Таблица 1.34 Основные элементы блок-схем

№	Символ	Наименование	Содержание	Размеры по ГОСТ 19 003-80 (ЕСПД) $a = 10, 15, 20$ мм, $b = 1{,}5a$
1	$a = b$ $b - c$	Блок вычислений	Вычислительные действия или последовательность действий	
2		Логический блок	Выбор направления выполнения алгоритма в зависимости от не которого условия	
3		Блоки ввода-вывода данных	1 Общие обозначения ввода (вывода) данных (вне зависимости от физического носителя) 2 Вывод данных, носителем которых является документ	
4		Начало (конец)	Начало или конец алгоритма, вход или выход в программу	
5		Процесс пользователя (подпрограмма)	Вычисление го стандартной программе или подпрограмме	
6		Блок модификации	Функция выполняет действия изменяющие пункты (например, заголовок цикла) алгоритма	
7		Соединитель	Указание связи прерванными линиями между потоками ин формации в пределах одного листа	
8		Межстраничные соединения	Указание связи между информацией на разных листах	

Блоки на схемах соединяются линиями потоков информации Основное направление потока информации идет сверху вниз и слева направо (стрелки могут не указываться), снизу вверх и справа налево — стрелка обязательна. Количество входящих линий для блока не ограничено Выходящих линий — одна, за исключением логического блока.

Базовые структуры алгоритмов

Это определенный набор блоков и стандартных способов их соединения для выполнения типичных последовательных действий К основным структурам относятся следующие — линейные (рис. 1.20, *а),* разветвляющиеся (рис. 1.20, *б),* циклические (рис. 1.20, в).

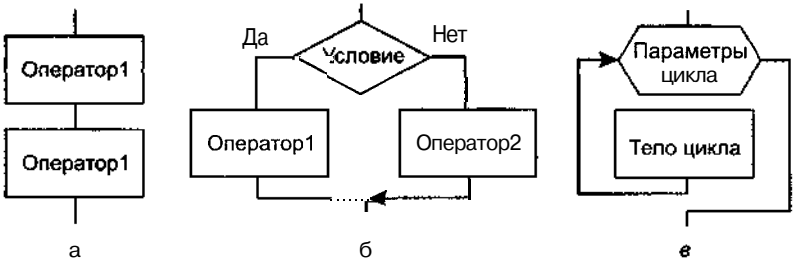


Рис. 1.20. Базовые структуры алгоритмов и программ

Линейными называются алгоритмы, в которых действия осуществляются последовательно друг за другом Стандартная блок-схема линейного алгоритма приводится на рис 1 20, *а* (вычисление суммы двух чисел — *А* и *В).*

Разветвляющимся называется алгоритм, в котором действие выполняется по одной из возможных ветвей решения задачи в зависимости от выполнения условий В отличие от линейных алгоритмов, в которых команды выполняются последовательно одна за другой, в разветвляющиеся алгоритмы входит условие, в зависимости от выполнения или невыполнения которого выполняется та или иная последовательность команд (серий).

В качестве условия в разветвляющемся алгоритме может быть использовано любое понятное исполнителю утверждение, которое может соблюдаться (быть истинно) или не соблюдаться (быть ложно) Такое утверждение может быть выражено как словами, так и

формулой. Таким образом, команда ветвления состоит из условия и двух последовательностей команд.

Примером может являться разветвляющийся алгоритм, изображенный в виде блок-схемы (рис. 1.20, *б)*. Аргументами этого алгоритма являются две переменные *A, B,* а результатом — переменная *X.* Если условие $A > B$ истинно, то выполняется операция $X := A \times B$, в противном случае выполняется $X := A + B$. В результате печатается то значение переменной *X,* которое она получает в результате выполнения одной из серий команд.

Циклическим называется алгоритм, в котором некоторая последовательность операций (тело цикла) выполняется многократно. Однако «многократно» не означает «до бесконечности». Организация циклов, никогда не приводящая к остановке в выполнении алгоритма, является нарушением требования его результативности — получения результата за конечное число шагов.

Перед операцией цикла осуществляется операция начального присвоения значений тем переменным, которые используются в теле цикла. В цикл входят в качестве базовых следующие структуры: блок проверки условия и тело цикла. Если тело цикла расположено после проверки условий *P* (цикл с предусловием), то может случиться, что при определенных условиях блок тело цикла не выполнится ни разу. Такой вариант организации цикла, управляемый предусловием, называется *цикл «пока»* (здесь условие — это условие на продолжение цикла).

Возможен другой случай, когда тело цикла выполняется по крайней мере 1 раз и будет повторятся до тех пор, пока не станет истинным условие. Такая организация цикла, когда его тело расположено перед проверкой условия, носит название цикла с постусловием, или *цикла «до».* Истинность условия в этом случае — условие окончания цикла. Отметим, что возможна ситуация с постусловием и при организации цикла «пока». Итак, цикл «до» завершается, когда условие становится истинным, а цикл «пока» — когда становился ложным. Современные языки программирования имеют достаточный набор операторов, реализующих как цикл «пока», так и цикл «до».

Рассмотрим пример алгоритма вычисления факториала, изображенный на рис. 1.21 (с циклом «пока»). Переменная *N* получает значение числа, факториал которого вычисляется. Переменной *N*!, которая в результате выполнения алгоритма должна получить значение факториала, присваивается первоначальное значение 1. Переменной *K* также присваивается значение 1. Цикл будет выполняться, пока справедливо условие $N > K$. Тело цикла состоит из двух операций: $N! = N! \times K$ и $K = K + 1$.

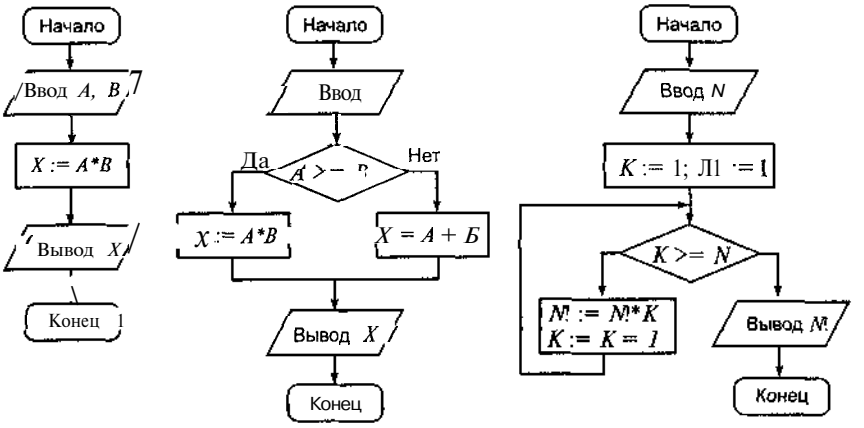


Рис. 1.21. Примеры структур алгоритмов:
а — линейный алгоритм; *б* — алгоритм с ветвлением; *в* — алгоритм с циклом

Циклические алгоритмы, в которых тело цикла выполняется заданное число раз, реализуются с помощью цикла со счетчиком. Цикл со счетчиком реализуется с помощью команды повторения.

Процесс решения сложной задачи довольно часто сводится к решению нескольких более простых подзадач. Соответственно при разработке сложного алгоритма он может разбиваться на отдельные алгоритмы, которые называются вспомогательными. Каждый такой вспомогательный алгоритм описывает решение какой-либо подзадачи.

Процесс построения алгоритма методом последовательной детализации состоит в следующем. Сначала алгоритм формулируется в «крупных» блоках (командах), которые могут быть непонятны исполнителю (не входят в его систему команд) и записываются как вызовы вспомогательных алгоритмов. Затем происходит детализация, и все вспомогательные алгоритмы подробно расписываются с использованием команд, понятных исполнителю.

Контрольные вопросы

1. Что такое поколения ЭВМ?
2. **Охарактеризуйте ЭВМ по областям применения.**
3. Дайте классификацию информации.
4. **Каковы преимущества цифровой информации по отношению к аналоговой?**

5. Перечислите методы кодирования символов.
6. Перечислите методы кодирования численной информации.
7. В чем заключаются особенности двоичной арифметики?
8. Перечислите логические элементы ЭВМ.
9. Что такое логические узлы ЭВМ?
10. Составьте таблицу истинности для (A & B) | (B & $\overline{B}$) с учетом значения *null.*
11. Составьте таблицы истинности для левого ($\neg(A \wedge B)$) и правого ($\neg A \vee \neg B$) выражений 1-го закона де Моргана. Проверьте их на соответствие.
12. Составьте таблицы истинности для левого ($\neg(A \vee B)$) и правого ($\neg A \wedge \neg B$) выражений 2-го закона де Моргана. Проверьте их на соответствие.
13. Проверьте выполнимость законов де Моргана с учетом значения *null.*
14. Последний столбец таблицы истинности для двухместных операций, очевидно, может содержать $16 = 2^4$ различных сочетаний «1» и «О». Следовательно, всего может быть определено 16 логических операций над двумя переменными, из которых нами рассмотрены только пять. Составьте таблицу истинности для одной из девяти оставшихся вне рассмотрения функций и попытайтесь построить логическое выражение для этой функции.
15. Перечислите базовые структуры алгоритмов и программ.

Глава 2
АРХИТЕКТУРА И СТРУКТУРА ВЫЧИСЛИТЕЛЬНЫХ МАШИН И СИСТЕМ

Прежде всего, следует определить основные объекты рассмотрения:

- обычные вычислительные машины;
- вычислительные комплексы (системы), в том числе многопроцессорные машины;
- суперкомпьютеры;
- вычислительные сети;

Три последних относятся к вычислительным системам.

Таким образом, в целом следует рассматривать все множество современных *вычислительных машин, систем, сетей.* Однако здесь мы ограничиваемся первыми тремя пунктами, поскольку *компьютерные сети* достаточно подробно рассмотрены, например в [26].

2.1. Базовые представления об архитектуре ЭВМ

Архитектурой компьютера считается его представление на некотором общем уровне, включающее описание пользовательских возможностей программирования, системы команд, системы адресации, организации памяти и т. д. Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: *процессора, оперативного запоминающего устройства (ОЗУ, ОП), внешних ЗУ и периферийных устройств.* Общность архитектуры разных компьютеров обеспечивает их совместимость с точки зрения пользователя.

Структура компьютера — это совокупность его функциональных элементов и связей между ними. Элементами могут быть самые различные устройства — от основных логических узлов компьютера до простейших схем. Структура компьютера графически представляется в виде структурных схем, с помощью которых можно дать описание компьютера на любом уровне детализации.

Принципы (архитектура) фон Неймана

В основу построения большинства компьютеров положены следующие общие принципы, сформулированные в 1945 г. американским ученым Джоном фон Нейманом.

1. Принцип программного управления. Из него следует, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

Выборка программы из памяти осуществляется с помощью *счетчика команд.* Этот регистр процессора последовательно увеличивает хранимый в нем адрес очередной команды на длину команды. Так как команды программы расположены в памяти друг за другом, то тем самым организуется выборка цепочки команд из последовательно расположенных ячеек памяти.

Если после выполнения команды следует перейти не к следующей, а к какой-то другой, используются команды *условного или безусловного переходов (ветвления),* которые заносят в счетчик команд номер ячейки памяти, содержащей следующую команду. Выборка команд из памяти прекращается после достижения и выполнения команды «стоп».

Таким образом, процессор исполняет программу автоматически, без вмешательства человека.

2. Принцип однородности памяти. Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными. Это открывает целый ряд возможностей. Например, программа в процессе своего выполнения также может подвергаться переработке, что позволяет задавать в самой программе правила получения некоторых ее частей (так в программе организуется выполнение циклов и подпрограмм). Более того, команды одной программы могут быть получены как результаты исполнения другой программы. На этом принципе основаны методы *трансляции* — перевода текста программы с языка программирования высокого уровня на язык конкретной машины.

3. Принцип адресности. Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена областям памяти, так, чтобы к запомненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен.

Компьютеры, построенные на этих принципах, относятся к типу фон-неймановских. Существуют и другие классы компьютеров, принципиально отличающиеся от фон-неймановских. Здесь, например, может не выполняться принцип программного управления, т. е. они могут работать без *счетчика (регистра адреса) команд,* указывающего на выполняемую команду программы. Для обращения к какой-либо переменной, хранящейся в памяти, этим компьютерам не обязательно давать ей имя. Такие компьютеры называются не-фон-неймановскими.

Логические узлы (агрегаты) ЭВМ, простейшие типы архитектур

Центральное устройство. ЦУ представляет основную компоненту ЭВМ и, в свою очередь, включает ЦП — центральный процессор (central processing unit — CPU) и ОП — оперативную (главную) память (main storage, core storage, random access memory — RAM).

Процессор непосредственно реализует операции обработки информации и управления вычислительным процессом, осуществляя выборку машинных команд и данных из оперативной памяти и запись в ОП, включение и отключение ВУ. Основными блоками процессора являются:

- устройство управления (УУ) с интерфейсом процессора (системой сопряжения и связи процессора с другими узлами машины);
- арифметико-логическое устройство (АЛУ);
- процессорная память (внутренний кэш).

Оперативная память предназначена для временного хранения данных и программ в процессе выполнения вычислительных и логических операций.

ЦУ описывается следующими характеристиками:

- длина машинного слова (разрядность, адресность);
- система команд;
- объем ОП;
- быстродействие (тактовая частота процессора, цикл записи/считывания ОП).

Внешние устройства (ВУ). ВУ обеспечивают эффективное взаимодействие компьютера с окружающей средой — пользователями, объектами управления, другими машинами. ВУ разделяются на следующие группы:

- интерактивные устройства (ввода/вывода);
- устройства хранения (массовые накопители);

- устройства массового ввода информации,
- устройства массового вывода информации.

В *специализированных* управляющих ЭВМ (технологические процессы, связь, ракеты и пр.) внешними устройствами ввода являются датчики (температуры, давления, расстояния и пр.), вывода — манипуляторы (гидро-, пневмо-, сервоприводы рулей, вентилей и др.).

В *универсальных* ЭВМ (человеко-машинная обработка информации) в качестве ВУ выступают терминалы, принтеры и др. устройства.

Каналы связи (внутримашинный интерфейс) служат для сопряжения центральных узлов машины с ее внешними устройствами.

Однотипные ЦУ и устройства хранения данных могут использоваться в различных типах машин. Известны примеры того, как фирмы, начавшие свою деятельность с производства управляющих машин, совершенствуя свою продукцию, перешли к выпуску систем, которые в зависимости от конфигурации ВУ могут исполнять как роль универсальных, так и управляющих машин (Hewlett-Packard и Digital Equipment Corporation).

Если абстрагироваться от подробностей, то основные классические типы архитектур можно определить как следующие: *«звезда», иерархическая, магистральная* (схематически — рис. 2.1, подробнее — рис. 2.2, 2.3, 2.4).

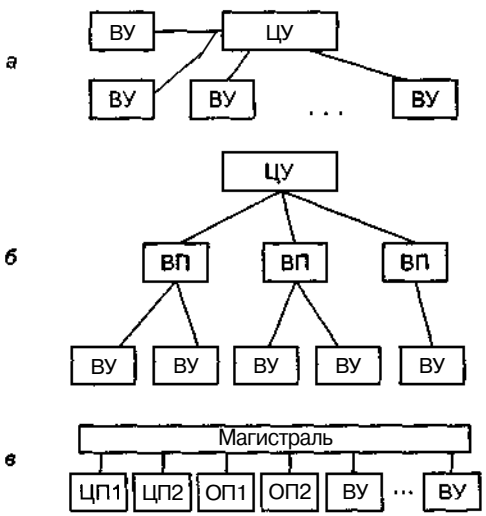


Рис. 2.1. Основные классы архитектур ЭВМ

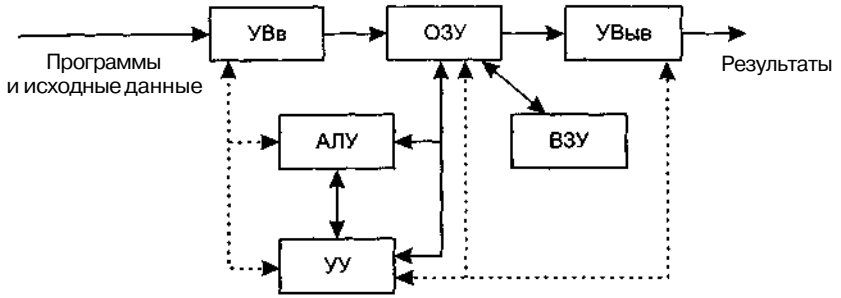


Рис. 2.2. Структурная схема ЭВМ 1-го и 2-го поколения (архитектура фон Неймана), «звезда»

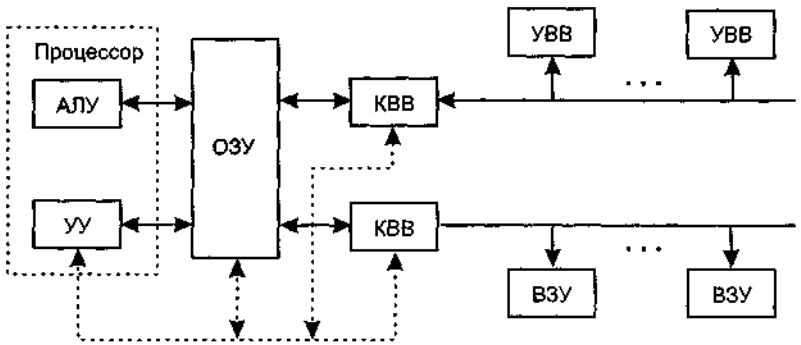


Рис. 2.3. Структурная схема ЭВМ 3-го поколения (иерархическая)

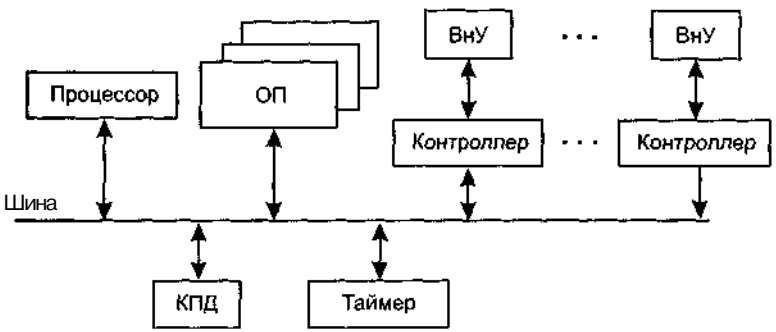


Рис. 2.4. Структурная схема ПЭВМ

Архитектура «звезда». Здесь ЦУ (рис. 2.1, *а)* соединено непосредственно с ВУ и управляет их работой (ранние модели машин).

Классическая архитектура (фон Неймана) — одно арифметико-логическое устройство (АЛУ), через которое проходит поток данных, и одно устройство управления (УУ), через которое проходит поток команд — программа (рис. 2.2). Это однопроцессорный компьютер.

Вычислительная машина включает пять базовых компонент и состоит из следующих типов устройств:

- центральный процессор (ЦП), включающийАЛУ и УУ;
- запоминающие устройства — память, в том числе оперативная (ОП) и внешние ЗУ;
- устройства ввода и устройства вывода информации — внешние (периферийные) устройства (ВУ).

Иерархическая архитектура (рис. 2.1, *б,* рис. 2.3) — ЦУ соединено с периферийными процессорами (вспомогательными процессорами, каналами и пр.), управляющими в свою очередь контроллерами, к которым подключены группы ВУ (системы IBM 360—375);

Магистральная структура (общая шина — unibas, рис. 2.1, *в,* рис. 2.4) — процессор (процессоры) и блоки памяти (ОП) взаимодействуют между собой и с ВУ (контроллерами ВУ) через внутренний канал, общий для всех устройств (машины DEC, ПЭВМ IBM PC-совместимые).

К этому типу архитектуры относится также архитектура персонального компьютера: функциональные блоки здесь связаны между собой общей шиной, называемой также системной магистралью.

Физически *магистраль* представляет собой многопроводную линию с гнездами для подключения электронных схем. Совокупность проводов магистрали разделяется на отдельные группы: шину адреса, шину данных и шину управления.

Периферийные устройства (принтер и др.) подключаются к аппаратуре компьютера через специальные *контроллеры* — устройства управления периферийными устройствами.

Контроллер — устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

Мы хотим обратить внимание читателя на тот факт, что все перечисленные архитектурные элементы ЭВМ базируются на следующих схемных элементах и базовых узлах (см. гл. 1):

- *память* обычно использует возможности и свойства триггера или его аналогов;

- *счетчик* (регистр) адреса команд, очевидно, есть схемный узел «счетчик»;
- *сумматор* — или полный сумматор, или полусумматор;
- *дешифратор* (например, команд) тоже здесь присутствует.

2.2. Процессор, структура и функционирование

В большинстве машин реализованы принципы фон Неймана в следующем виде:

- *оперативная память (ОП)* организована как совокупность *машинных слов (МС) фиксированной длины или разрядности* (имеется в виду количество двоичных единиц или бит, содержащихся в каждом МС). Например, ранние ПЭВМ имели разрядность 8, затем появились 16-разрядные, а затем — 32- и 64-разрядные машины. В свое время существовали также 45-разрядные (М-20, М-220), 35-разрядные (Минск-22, Минск-32) и др. машины;
- ОП образует единое адресное пространство, адреса МС возрастают от младших к старшим;
- в ОП размещаются как данные, так и программы, причем в области данных одно слово, как правило, соответствует одному числу, а в области программы — одной команде (машинной инструкции — минимальному и неделимому элементу программы);
- команды выполняются в *естественной последовательности* (по возрастанию адресов в ОП), пока не встретится *команда управления* (условного/безусловного перехода, или ветвления — branch), в результате которой естественная последовательность нарушится;
- ЦП может произвольно обращаться к любым адресам в ОП для выборки и/или записи в МС чисел или команд.

Абстрактное центральное устройство

Перечислим основные понятия и рассмотрим структуру и функции абстрактного центрального устройства ЭВМ (1—2-е поколения ЭВМ) (рис. 2.5), АЛУ которого предназначено для обработки целых чисел и битовых строк.

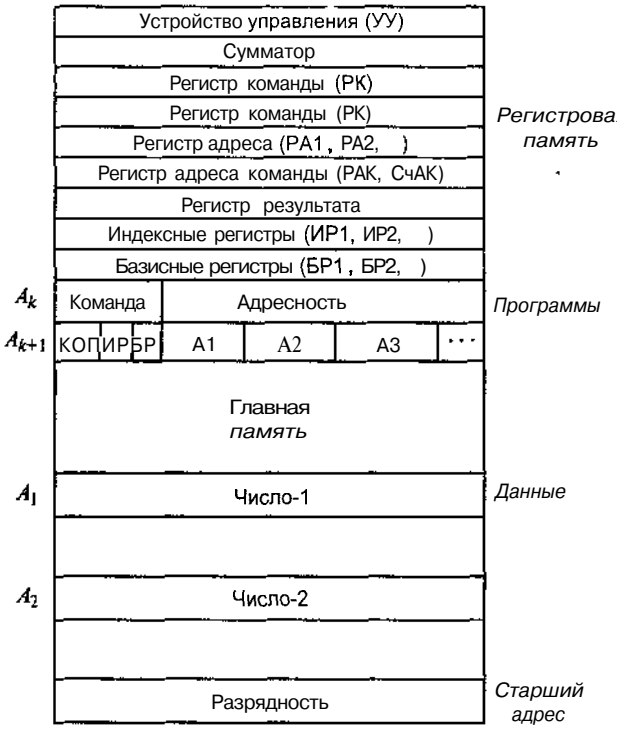


Рис. 2.5. Структура абстрактного центрального устройства ЭВМ

Команда (instruction) — описание операции, которую необходимо выполнить. Каждая команда начинается с *кода операции (КОП),* содержит необходимые *адреса,* характеризуется форматом, который определяет структуру команды, ее организацию, код, длину, метод расположения адресов. Длина различных команд может быть как одинаковой, так и разной.

Команды подразделяются на арифметические, логические, ввода/вывода, передачи данных. Каждая команда выполняется в компьютере за один либо несколько *тактов.*

Последовательность взаимосвязанных команд именуется *макрокомандой.* Использование макрокоманд упрощает программирование и обеспечивает механизм вставки добавлений в программы (см. далее *макроассемблер,* MASM).

Цикл процессора — период времени, за который осуществляется выполнение команды исходной программы в машинном виде; состоит из нескольких *тактов.*

Такт работы процессора — промежуток времени между соседними импульсами *генератора тактовых импульсов,* частота которых есть *тактовая частота процессора.* Эта частота является одной из основных характеристик компьютера и во многом определяет скорость его работы, поскольку каждая операция в вычислительной машине выполняется за определенное количество тактов. Выполнение *короткой команды* (арифметика с фиксированной точкой, логические операции), о которых речь здесь и пойдет, обычно занимает пять тактов:

- выборка команды;
- расшифровка кода операции (декодирование);
- генерация адреса и выборка данных из памяти;
- выполнение операции;
- запись результата в память.

Процедура, соответствующая такту, реализуется определенной логической цепью (схемой) процессора, обычно именуемой *микропрограммой.*

Регистры — устройства, предназначенные для временного хранения данных ограниченного размера. Важной характеристикой регистра является высокая скорость приема и выдачи данных. Регистр состоит из разрядов, в которые можно быстро записывать, запоминать и считывать слово, команду, двоичное число и т. д. Обычно регистр имеет ту же разрядность, что и машинное слово. Регистр, накапливающий данные, именуют *аккумулятором.*

Регистр, обладающий способностью перемещать содержимое своих разрядов, называют *сдвиговым.* В этих регистрах за один такт хранимое слово поразрядно сдвигается на одну позицию. Сдвиговые регистры используются при обработке данных, кодировании и декодировании.

Некоторые регистры служат *счетчиками.* Счетчик является устройством, которое на своих выходах выдает (в двоичной форме) сумму числа импульсов, подаваемых на его единственный вход. Максимальное число импульсов, которое счетчик может подсчитать, называется его *емкостью.*

Регистры общего назначения (РОН) — (General Purpose Registers) — общее название для регистров, которые временно содержат данные, передаваемые или принимаемые из памяти.

Регистр команды (РК),(Instruction Register — IR) служит для размещения текущей команды, которая находится в нем в течение текущего цикла процессора.

Регистр (РАК), счетчик (СчАК) адреса команды (Program Counter — PC) — регистр, содержащий адрес текущей команды.

Регистр адреса (числа) — РА(Ч) — содержит адрес одного из операндов выполняемой команды (регистров может быть несколько).

Регистр числа (РЧ) — содержит операнд выполняемой команды, РЧ также несколько.

Регистр результата (РР) — предназначается для хранения результата выполнения команды.

Сумматор — регистр, осуществляющий операции сложения (логического и арифметического двоичного) чисел или битовых строк, представленных в *прямом или обратном коде* (иногда РЧ и РР включают в состав сумматора).

Существуют и другие регистры, не отмеченные на рис. 2.5, например, регистр состояния — Status Register (SR).Типичным содержанием SR является информация о результатах завершения команды (ноль, переполнение, деление на ноль, перенос и пр.). УУ использует информацию из SR для исполнения условных переходов (например, «в случае переполнения перейти по адресу 4170»). Ниже (гл. 4) подробно будут рассмотрены *регистры процессора i8086.*

Цикл выполнения команды может выглядеть следующим образом.

1. В соответствии с содержимым СчАК (адрес очередной команды) УУ извлекает из ОП очередную команду и помещает ее в РК.

Некоторые команды УУ обрабатывает самостоятельно, без привлечения АЛУ (например, по команде «перейти по адресу 2478», величина 2478 сразу заносится в СЧАК и процессор переходит к выполнению следующей команды.

Типичная команда содержит:

- код операции (КОП) — характеризующий тип выполняемого действия (сложение, вычитание и пр. чисел; сравнение строк; передача управления, обращение к ВУ и пр.);
- номера индексного (ИР) и базисного (БР) регистров (в некоторых машинах — адреса слов, ячеек ОП, в которых размещена соответствующая информация);
- адреса операндовА1, А2 и т. д., участвующих в выполнении команды (чисел, строк, других команд программы).

2. Осуществляется расшифровка (декодирование) команды.

3. Адреса А1, А2 и пр. помещаются в регистры адреса.

4. Если в команде указаны ИР или БР, то их содержимое используется для модификации РА — фактически выбираются числа или команды, смещенные в ту или иную сторону по отношению к адресу, указанному в команде.

При этом ИР используются для текущего изменения адреса, связанного с работой программы (например, при обработке массива чисел). БР используется для глобального смещения программы или данных в ОП.

5. По значениям РА осуществляется чтение чисел (строк) и помещение их в РЧ.

6. Выполнение операции (арифметической, логической и пр.) и помещение результата в РР.

7. Запись результата по одному из адресов (если необходимо).

8. Увеличение содержимого СчАК на единицу (переход к следующей команде).

Очевидно, что за счет увеличения числа регистров возможно *распараллеливание, перекрытие* операций. Например, при считывании команды, СчАК можно автоматически увеличить на 1, подготовив выборку следующей команды. После расшифровки текущей команды РК освобождается и в него может быть помещена следующая команда программы. При выполнении операции возможна расшифровка следующей команды и т. д. Все это является предпосылкой построения так называемых *конвейерных структур (pipeline)*. Однако все это хорошо только при последовательном (естественном) порядке выполнения команд. Появление переходов (особенно по не определенному заранее условию) нарушает эту картину. Поэтому современные процессоры пытаются предсказывать переходы в программе *(branch prediction)*.

Системы команд и соответствующие классы процессоров

Основные команды ЭВМ классифицируются вкратце следующим образом (подробнее см. далее, описание команд для i8086): по функциям (выполняемым операциям), направлению приема-передачи информации, адресности.

Классы команд

1. Команды обработки данных, в том числе (О1 — первый операнд, О2 — второй).

1.1. Короткие операции (один такт).

1.1.1. Логические:

- логическое сложение (для каждого бита О1 и О2 осуществляется операция ИЛИ);

- логическое умножение (для каждого бита О1 и О2 осуществляется операция И);
- инверсия (вО1 все единицы заменяются на нули, и наоборот);
- сравнение логическое (если О1 = О2, то *некий* флаг или *регистр* устанавливается в «1», иначе в «0»);

1.1.2. Арифметические:

- сложение операндов;
- вычитание (сложение в обратном коде);
- сравнение арифметическое (если О1 > О2, или О1 = 02, или О1 < О2, то *некий* флаг или *регистр* устанавливается в 1, иначе — в 0);

1.2. Длинные операции (несколько тактов):

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

2. Операции управления:

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

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

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

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

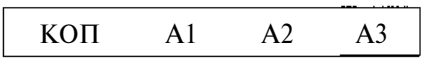
- регистр—регистр (01 и О2 размещаются в регистрахАЛУ);
- память—регистр (регистр—память) — один из операндов размещается в ОП;
- память—память (О1 и О2 размещены в ОП).

Далее, известны одно-, двух- и трехадресные машины (системы команд). Очевидна связь таких параметров ЦУ, как длина адресного пространства, адресность, разрядность. Увеличение разрядности позволяет увеличить адресность команды и длину адреса (т. е. объем памяти, доступной данной команде). Увеличение адресности, в свою очередь, приводит к повышению быстродействия обработки (за счет снижения числа требуемых команд).

В трехадресной машине, например, сложение двух чисел требует одной команды (извлечь число по А1, число по А2, сложить и записать результат по А3). В двухадресной необходимы две команды

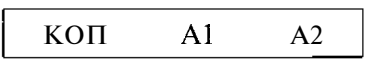
(первая — извлечь число по А1 и поместить в РЧ (или сумматор), вторая — извлечь число по А1, сложить с содержимым РЧ и результат записать по А2). Легко видеть, что одноадресная машина потребует три команды. Поэтому неудивительно, что основная тенденция в развитии ЦУ ЭВМ состоит в увеличении разрядности.

Типовая структура трехадресной команды:



где А2 и А3 — адреса ячеек (регистров), где расположены соответственно первое и второе числа, участвующие в операции; А1 — адрес ячейки (регистра), куда следует поместить число, полученное в результате выполнения операции.

Типовая структура двухадресной команды:



где А1 — это обычно адрес ячейки (регистра), где хранится первое из чисел, участвующих в операции, и куда после завершения операции должен быть записан результат операции; А2 — обычно адрес ячейки (регистра), где хранится второе участвующее в операции число.

Типовая структура одноадресной команды:

где А1 в зависимости от модификации команды может обозначать либо адрес ячейки (регистра), в которой хранится одно из чисел, участвующих в операции, либо адрес ячейки (регистра), куда следует поместить число — результат операции.

Безадресная команда содержит только код операции, а информация для нее должна быть заранее помещена в определенные регистры машины.

Наибольшее применение нашли *двухадресные системы* команд.

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

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

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

```
r1 = a + b;      — операция сложения;
r2 = r1*r1;      — операция умножения;
r3 = c/d;        — операция деления;
y = r2 - r3;     — операция вычитания;
Стоп.            — завершение обработки.
```

Здесь количество переменных, необходимых для хранения промежуточных результатов, связано с адресностью системы команд и с тем, разрешено или нет в процессе вычислений изменять значения исходных данных.

Классы процессоров. В зависимости от набора и порядка выполнения команд процессоры подразделяются на четыре класса, отражающих также последовательность развития ЭВМ. Ранее других появились процессоры CISC. Затем, с целью повышения быстродействия процессоров были разработаны процессоры RISC, которые характеризуются сокращенным набором быстро выполняемых команд. Ряд редко встречающихся команд процессора CISC выполняется последовательностями команд процессора RISC. Позже появилась концепция процессоров MISC, использующая минимальный набор длинных команд. Вслед за ними возникли процессоры VLIW, работающие со сверхдлинными командами.

CISC (complex instruction set computer) есть традиционная архитектура, в которой ЦП использует микропрограммы для выполнения исчерпывающего набора команд. Они могут иметь различную длину, методы адресации и требуют сложных электронных цепей для декодирования и исполнения. В течение долгих лет производители компьютеров разрабатывали и воплощали в изделиях все более сложные и полные системы команд. Однако анализ работы процессоров показал, что в течение примерно 80 % времени выполняется лишь 20 % большого набора команд. Поэтому была поставлена задача оптимизации выполнения небольшого по числу, но часто используемых команд.

В 1974 г. John Cocke (IBM Research) решил испробовать подход, который мог бы существенно уменьшить количество машинных команд в ЦП. В середине 70-х это привело многих производителей компьютеров к пересмотру своих позиций и к разработке ЦП с весьма ограниченным набором команд.

RISC (Redused Instuction Set Computer) — процессор, функционирующий с сокращенным набором команд. Так, в

процессоре CISC для выполнения одной команды необходимо в большинстве случаев 10 и более тактов. Что же касается процессоров RISC, то они близки к тому, чтобы выполнять по одной команде в каждом такте. Следует также иметь в виду, что благодаря своей простоте процессоры RISC не патентуются. Это также способствует их быстрой разработке и широкому производству. Между тем, в сокращенный набор RISC вошли только наиболее часто используемые команды.

Первый процессор RISC был создан корпорацией IBM в 1979 г. и имел шифр IBM 801. В настоящее время процессоры RISC получили широкое распространение. Современные процессоры RISC характеризуются следующим:

- упрощенный набор команд, имеющих одинаковую длину;
- большинство команд выполняются за один такт процессора;
- отсутствуют макрокоманды, усложняющие структуру процессора и уменьшающие скорость его работы;
- взаимодействие с оперативной памятью ограничивается операциями пересылки данных;
- резко уменьшено число способов адресации памяти (не используется косвенная адресация);
- используется *конвейер* команд, позволяющий обрабатывать несколько из них одновременно;
- применяется высокоскоростная память.

Новый подход к архитектуре процессора значительно сократил площадь, требуемую для него на кристалле интегральной схемы. Это позволило резко увеличить число регистров. В современном процессоре RISC уже используется более 100 регистров. В результате процессор на 20—30 % реже обращается к оперативной памяти, что также повысило скорость обработки данных. Упростилась топология процессора, выполняемого в виде одной интегральной схемы, сократились сроки ее разработки, она стала дешевле.

Начиная с процессора Pentium корпорация Intel начала внедрять элементы RISC-технологий в свои изделия.

Процессор MISC — MISC processor, работающий с минимальным набором длинных команд.

Увеличение разрядности процессоров привело к идее укладки нескольких команд в одно слово (связку, bound) размером 128 бит. Оперируя с одним словом, процессор получил возможность обрабатывать сразу несколько команд. Это позволило использовать возросшую производительность компьютера и его возможность обрабатывать одновременно несколько потоков данных.

Процессор MISC, как и процессор RISC, характеризуется небольшим набором чаще всего встречающихся команд. Вместе с этим принцип команд VLIW обеспечивает выполнение группы команд за один цикл работы процессора. Порядок выполнения команд распределяется таким образом, чтобы в максимальной степени загрузить маршруты, по которым проходят потоки данных. Таким образом, архитектура MISC объединила вместе суперскалярную (многопоточную) и VLIW концепции. Компоненты процессора просты и работают с высокими скоростями.

Процессор VLIW — процессор, работающий с системой команд сверхбольшой разрядности.

Идея технологии VLIW заключается в том, что создается специальный компилятор планирования, который перед выполнением прикладной программы проводит ее анализ, и по множеству ветвей последовательности операций определяет группу команд, которые могут выполняться параллельно. Каждая такая группа образует одну сверхдлинную команду. Это позволяет решать две важные задачи. Во-первых, в течение одного такта выполнять группу коротких («обычных») команд. И, во-вторых, упростить структуру процессора. Этим технология VLIW отличается от суперскалярности. В последнем случае отбор групп одновременно выполняемых команд происходит непосредственно в ходе выполнения прикладной программы (а не заранее). Из-за чего усложняется структура процессора и замедляется скорость его работы.

Технология VLIW появилась в результате работ, проведенных корпорациями HP и Intel.

Арифметико-логическое устройство (АЛУ)

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

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

АЛУ классифицируются следующим образом:

1. По способу действий над операндами:

- АЛУ последовательного действия;
- параллельного действия.

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

2. По виду обрабатываемых чисел АЛУ могут производить операции над двоичными числами с фиксированной или плавающей запятой и над двоично-десятичными числами. В последнем случае каждая десятичная цифра записывается четырьмя разрядами двоичного кода:

2003
0010 0000 0000 ООН

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

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

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

3. По организации действий над операндами:

- блочные;
- многофункциональные АЛУ.

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

В многофункциональных АЛУ одни и те же блоки обрабатывают числа с фиксированной запятой, плавающей запятой и двоично-десятичные числа (рис. 2.6).

Клапаны К1 и К2 объединяют сумматоры 1, 2 и 3 для действий над числами с фиксированной запятой.

Для действий над числами с плавающей запятой клапан К2 объединяет сумматоры 2 и 3 для обработки мантисс, а клапан К1 отсоединяет первый сумматор от второго. Сумматор 1 обрабатывает порядки.

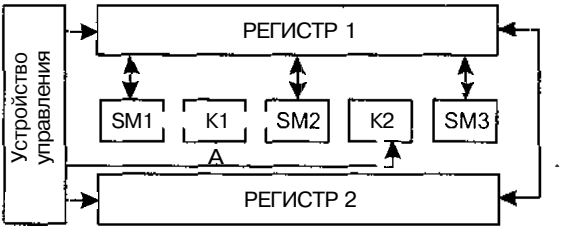


Рис. 2.6. Многофункциональное АЛУ

4. По структуре:

- АЛУ с непосредственными связями;
- многосвязные.

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

В АЛУ с непосредственной связью вход регистра приемника связан с выходом регистра источника операндов и регистра, в котором происходит обработка (рис. 2.7).

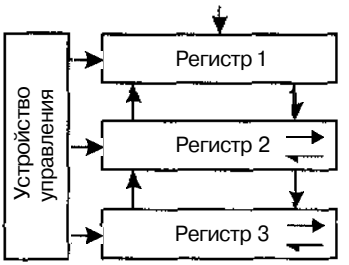


Рис. 2.7. АЛУ с непосредственной связью

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

Умножение в этом АЛУ происходит так: множимое помещают в регистр 4, множитель — в регистр 1. Регистры 2 и 3 являются, кроме того, сдвигающими регистрами. В зависимости от содержимого разряда множителя, множимое сдвигается на один разряд, если множитель содержит 1, и на два, если множитель содержит 0. Эти частные произведения суммируются в регистре 2.

2.3. Технологии повышения производительности процессоров

Конвейерная обработка команд (pipelining). Суперскаляризация

Рассмотрим процесс выполнения процессором команды для *коротких* (с фиксированной запятой или логические) операций. Как об этом говорилось выше, обработка команды, или цикл процессора, может быть разделена на несколько основных этапов, которые можно назвать *микрокомандами,* которых известно пять основных типов.

Каждая операция требует для своего выполнения времени, равного такту генератора процессора (tick of the internal clock). Отметим, что к длинным операциям (плавающая точка) это не имеет отношения — там другая арифметика. Очевидно, что при тактовой частоте в 100 МГц быстродействие составит 20 миллионов операций в секунду.

Все этапы команды задействуются только один раз и всегда в одном и том же порядке: одна за другой. Это, в частности, означает, что если первая микрокоманда выполнила свою работу и передала результаты второй, то для выполнения текущей команды она больше не понадобится, и, следовательно, может приступить к выполнению следующей команды.

Конвейеризация осуществляет многопоточную параллельную обработку команд, так что в каждый момент одна из команд считывается, другая декодируется и т. д., и всего в обработке одновременно находится пять команд. Таким образом, на выходе конвейера на каждом такте процессора появляется результат обработки одной команды (одна команда в один такт). Первая инструкция может считаться выполненной, когда завершат работу все пять микрокоманд.

Такая технология обработки команд носит название *конвейерной (pipeline)* обработки. Каждая часть устройства называется *ступенью конвейера,* а общее число ступеней — *длиной линии конвейера.*

С ростом числа линий конвейера и увеличением числа ступеней на линии (табл. 2.1) увеличивается пропускная способность процессора при неизменной тактовой частоте. Процессоры с несколькими линиями конвейера получили название *суперскалярных.* Pentium — первый суперскалярный процессор Intel. Здесь две линии, что позволяет ему при одинаковых частотах быть вдвое производительней i80486, выполняя сразу две инструкции за такт.

Во многих вычислительных системах, наряду с *конвейером команд,* используются *конвейеры данных.*

Таблица 2.1. Характеристики конвейеров процессоров Intel

Процессор	i80486	Pentium	Pentium Pro	Pentium MMX	Pentium II	Pentium III	Pentium IV
Число линий	1	2	3	2	3	3	3
Длина линии	5	5	14	6	14	20	31 (HyperPipeline)

Сочетание этих двух конвейеров дает возможность достичь очень высокой производительности на определенных классах задач, особенно если используется несколько различных конвейерных процессоров, способных работать одновременно и независимо друг от друга.

Одной из наиболее высокопроизводительных вычислительных конвейерных систем считается CRAY. В этой системе конвейерный принцип обработки используется в максимальной степени. Имеется как конвейер команд, так и конвейер арифметических и логических операций. В системе широко применяется совмещенная обработка информации несколькими устройствами. Максимальная пиковая производительность процессора может составлять 12 Гфлопс.

Матричные и векторные процессоры

В отличие от скалярных и даже суперскалярных процессоров данные устройства манипулируют массивами данных и предназначены для обработки изображений, матриц и массивов данных. Частным случаем векторного процессора является процессор изображений, который предназначен для обработки сигналов, поступающих от датчиков-формирователей изображения.

Матричный процессор имеет архитектуру, рассчитанную на обработку числовых массивов. Архитектура процессора включает в себя матрицу процессорных элементов, например 64 x 64, работающих одновременно. Постпроцессор предназначен для реализации некоторых специальных функций, например управления базой данных.

Векторный процессор обеспечивает параллельное выполнение операции над массивами данных, векторами. Он характеризуется специальной архитектурой, построенной на группе параллельно работающих процессорных элементов (рис. 2.8, 2.9).

Векторная обработка увеличивает производительность процессора за счет того, что обработка целого набора данных (вектора) производится одной командой. Векторные компьютеры манипулируют массивами сходных данных подобно тому, как скалярные машины обрабатывают отдельные элементы таких массивов. В этом случае каждый элемент вектора надо рассматривать как отдельный

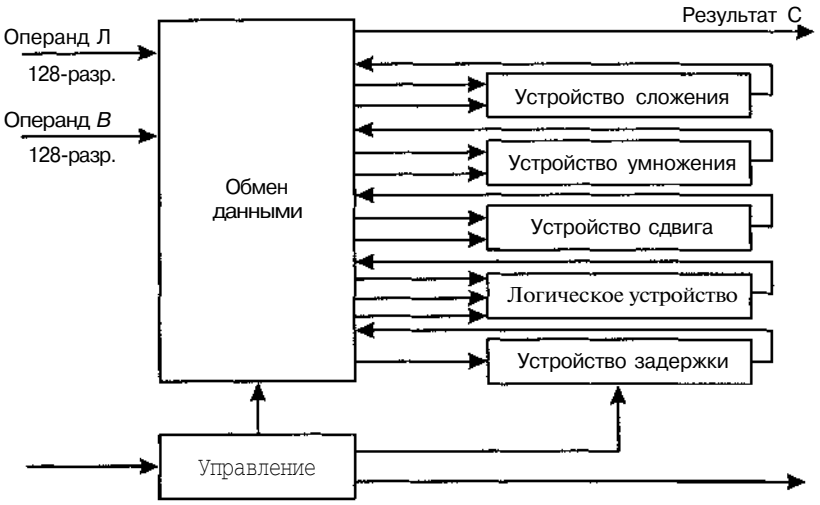


Рис. 2.8. Векторный процессор Cyber-205

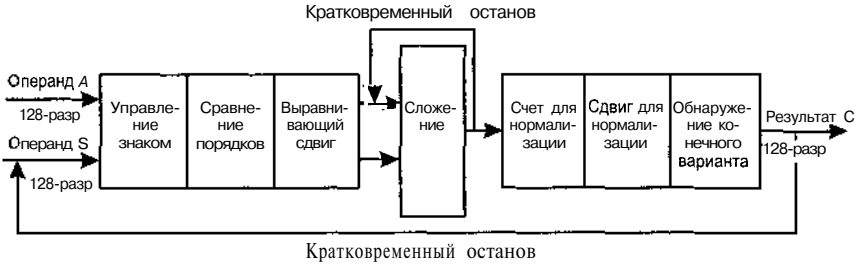


Рис. 2.9. Архитектура конвейера сложения векторного процессора Cyber-205

элемент потока данных. При работе в векторном режиме векторные процессоры обрабатывают данные практически параллельно, что делает их в несколько раз более быстрыми, чем при работе в скалярном режиме. Максимальная скорость передачи данных в векторном формате может составлять 64 Гбайт/с, что на два порядка быстрее, чем в скалярных машинах.

В настоящее время созданы однокристальные векторно-конвейерные процессоры, такие, как SX-6. Основными компонентами микропроцессора являются скалярный процессор и восемь идентичных векторных устройств, суммарная производительность которых составляет 64 Гфлопс. Примерами систем подобного типа являются, например, процессоры фирм NEC и Hitachi.

Динамическое исполнение (Dynamic execution technology)

Это совокупность технологий обработки данных в процессоре, обеспечивающая более эффективную работу процессора за счет манипулирования данными, а не простого исполнения списка инструкций. Динамическое исполнение представляет собой комбинацию трех методов обработки данных:

- множественное предсказание ветвлений;
- анализ потока данных;
- спекулятивное (по предположению) исполнение.

Впервые реализовано в процессоре Pentium Pro.

Множественное предсказание ветвлений. Предсказывается прохождение программы по нескольким ветвям: процессор может предвидеть разделение потока инструкций, используя алгоритм множественного предсказания ветвлений. С большой точностью (более 90 %) он предсказывает, в какой области памяти можно найти следующие инструкции. Это оказывается возможным, поскольку в процессе исполнения иструкции процессор просматривает программу на несколько шагов вперед. Этот метод позволяет увеличить загруженность процессора.

Хотя BTB (Branch Target Buffer — буфер предсказания переходов) и не может правильно предсказать абсолютно все переходы, но большинство предсказаний оказывается точными, что обеспечивает значительное повышение производительности. Например, программный цикл, состоящий из пересылки, сравнения, сложения и перехода в 80486 DX выполняется за 6 тактов синхронизации, а в Pentium за 2 (команды пересылки и сложения, а также сравнения и перехода сочетаются и предсказывается переход).

Анализ потока данных. Анализирует и составляет график исполнения инструкций в оптимальной последовательности, независимо от порядка их следования в тексте программы. Используя анализ потока данных, процессор просматривает декодированные инструкции и определяет, готовы ли они к непосредственному исполнению или зависят от результата других инструкций. Далее процессор определяет оптимальную последовательность выполнения и исполняет инструкции наиболее эффективным образом.

Спекулятивное выполнение. Повышает скорость выполнения, просматривая программу вперед и исполняя те инструкции, которые необходимы. Процессор выполняет инструкции (до пяти инструкций одновременно) по мере их поступления в оптимизированной последовательности (спекулятивно). Поскольку выполнение инструкций происходит на основе предсказания ветвлений,

результаты сохраняются как «спекулятивные». На конечном этапе порядок инструкций восстанавливается и переводится в обычное машинное состояние.

Процессоры уровня IA-64 имеют мощные вычислительные ресурсы, включая 128 регистров целых чисел, 128 регистров действительных чисел, 64 *предикационных* регистра, а также ряд специальных регистров.

Набор команд оптимизирован для решения задач криптографии, обработки видеосигналов и других процессов, встречающихся в серверах и рабочих станциях.

На рис. 2.10 и 2.11 представлены иллюстрации к возможностям архитектуры IA-64:

- *предикация (predication)* — одновременное исполнение двух ветвей программы, вместо предсказания переходов (выполнения наиболее вероятного);

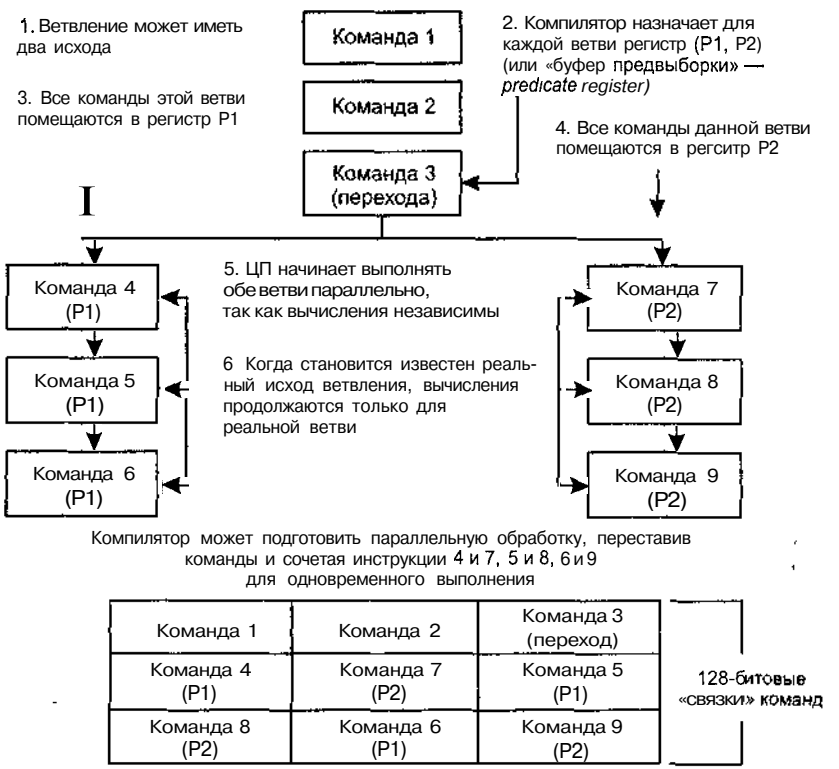


Рис. 2.10. Предикация или одновременное предварительное выполнение всех ветвей оператора условного перехода

Рис. 2.11. Опережающее считывание данных в регистр из памяти (speculative loading)

- *опережающее чтение данных (speculative loading),* т. е. загрузка данных в регистры с опережением, до того, как определилось реальное ветвление программы (переход управления).

Эти возможности осуществляются комбинированно — при компиляции и выполнении программы.

Предикация — центральный метод планирования параллельной обработки. Компилятор транслирует операторы исходного кода, содержащие ветвление (условный переход), в совокупность блоков машинных команд, идущих друг за другом. Обычный процессор, в зависимости от исхода условия, исполняет один из этих базовых блоков, пропуская все другие. Более развитые процессоры пытаются прогнозировать исход операции перехода и заранее *(спекулятивно, по предположению)* выполняют один из блоков, теряя время при ошибке прогнозирования.

Базовые блоки обычно малы (2—3 команды) и ветвление встречается в среднем через каждые шесть операторов языка программирования. Поэтому выигрыш оказывается небольшим.

Когда компилятор уровня IA-64 обнаруживает оператор ветвления в исходной программе, он анализирует все возможные ветви (блоки) и помечает их метками или *предикатами (predicate).* После этого он определяет, какие из них могут быть выполнены параллельно (из соседних, независимых, ветвей).

Затем компилятор группирует машинные коды в 128-битовые *связки (bundles),* по 3 команды в каждой. В *описания связок (template)* заносится информация о том, какие команды могут исполняться параллельно (независимо). Например, если компилятор находит 16 команд, которые не имеют взаимной связи, он укладывает их в 6 независимых связок (по 3 в первых 5 и одна в 6-й) и помечает их в описании.

В процессе выполнения программы ЦП просматривает описания связок, выбирает команды, которые взаимно независимы и распределяет их на параллельную обработку. Если ЦП обнаруживает оператор ветвления, он не пытается предсказать переход, а начинает выполнять все возможные ветви программы.

Таким образом, могут быть обработаны все ветви программы, но без записи полученного результата. В определенный момент процессор наконец «узнает» о реальном исходе условного оператора, записывает в память результат «правильной ветви» и отменяет остальные результаты.

Опережающее чтение (по предположению) разделяет загрузку данных в регистры и их реальное использование, избегая ситуации, когда процессору приходится ожидать прихода данных, чтобы начать их обработку.

Прежде всего, компилятор анализирует программу, определяя команды, которые требуют приема данных из оперативной памяти. Там, где это возможно, он вставляет команды опережающего чтения и парную команду контроля опережающего чтения (speculative check). В то же время компилятор переставляет команды таким образом, чтобы ЦП мог их обрабатывать параллельно.

В процессе работы ЦП встречает команду опережающего чтения и пытается выбрать данные из памяти. Может оказаться, что они еще не готовы (результат работы блока команд, который еще не выполнился). Обычный процессор в этой ситуации выдает сообщение об ошибке, однако система уровня IA-64 откладывает «сигнал тревоги» до момента прихода процесса в точку «команда проверки опережающего чтения». Если к этому моменту все предшествующие подпроцессы завершены и данные считаны, то обработка продолжается, в противном случае вырабатывается сигнал прерывания.

Технология Hyper-Threading (HT)

Здесь реализуется разделение времени на аппаратном уровне физически процессор разбивается на два логических процессора, каждый из которых использует ресурсы чипа — ядро, кэш-память, шины, исполнительное устройство (рис 2 12) Ядро процессора выполняет два процесса одновременно

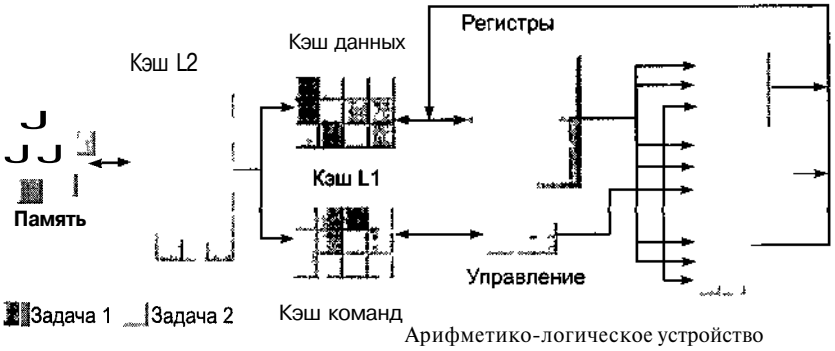


Рис. 2 12 Технология Hyper Threading (HT)

Специалисты Intel оценивают повышение эффективности в 30 % при использовании на HT-процессорах многопрограммных ОС и обычных прикладных программ

Процессор Pentium

Что же представляют собой современные процессоры, в которых реализованы все эти «штучки», воочию? Рассмотрим рис 2 13

Процессор Pentium состоит из следующих блоков

Ядро (Core) Основное исполнительное устройство

Производительность МП при тактовой частоте 66 МГц составляет около 112 млн инструкций в секунду (Mips) Пятикратное повышение (по сравнению с 80486 DX) достигалось благодаря двум конвейерам, позволяющим выполнить одновременно несколько инструкций Это два параллельных 5-ступенчатых конвейера обработки целых чисел, которые позволяют читать, интерпретировать, исполнять две команды одновременно

Целочисленные команды могут выполняться за один такт синхронизации Эти конвейеры неодинаковы U-конвейер выполняет любую команду системы команд семейства 86, V-конвейер выпол-

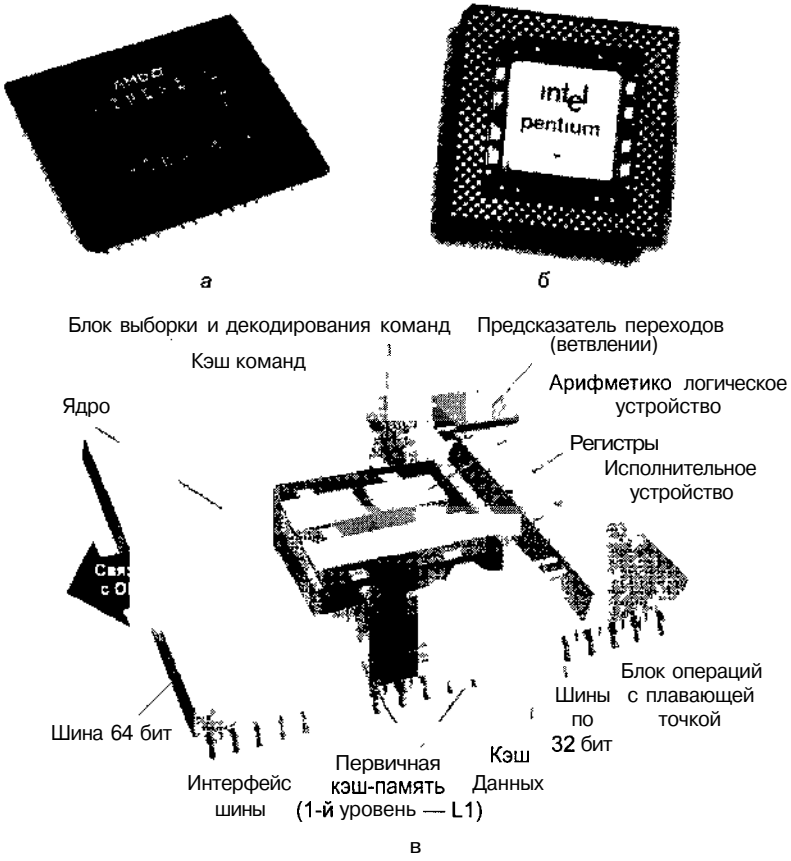


Рис 2 13. Микропроцессоры AMD *(а),* Intel Pentium MMX *(б),* основные компоненты процессора Pentium (в)

няет только «простые» команды, т е команды, которые полностью встроены в схемы МП и не требуют микропрограммного управления (microcode) при выполнении (это команды, допускающие спаривание с другими командами регистр—регистр, память—регистр, регистр—память, переходы, вызовы, арифметико-логические операции)

Предсказатель переходов (Branch Predictor) Пытается угадать направление ветвления программы и заранее загрузить информацию в блоки предвыборки и декодирования команд

Буфер адреса переходов (Branch Target Buffer, BTB) Обеспечивает динамическое предсказание переходов Он улучшает выполнение

команд путем запоминания состоявшихся переходов (256 последних переходов) и с опережением выполняет наиболее вероятный переход при выборке команды ветвления. Если предсказание верно, то эффективность увеличивается. Если нет, то конвейер приходится сбрасывать полностью. Согласно данным Intel, вероятность правильного предсказания переходов в процессорах Pentium, Pentium MMX составляет 75—80 %, а для Pentuim Pro, Pentium II — 90 %.

Статические методы предсказания упрощены — они предписывают всегда выполнять или нет определенные виды переходов. При динамических методах исследуется поведение команд перехода за предшествующий период.

Блок плавающей точки (Floating Point Unit). Выполняет обработку чисел с плавающей точкой.

Кэш-память 1-го уровня (Level 1 cache, L1). Процессор имеет два банка памяти по 8 Кбайт: 1-й — для команд, 2-й — для данных, которые обладают большим быстродействием, чем более емкая внешняя кэш-память (L2 cache).

Интерфейс шины (Bus Interface). Передает в ЦП поток команд и данных, а также передает данные из ЦП.

2.4. Организация оперативной памяти

Запоминающие устройства (ЗУ), именуемые также *устройствами памяти,* предназначены для хранения данных. Они, в свою очередь, включают процессоры, схемы логики, матрицы памяти, схемы контроля данных, дешифраторы, буферы, регистры, электрические и механические компоненты.

Основными характеристиками ЗУ являются:

- емкость памяти, измеряемая в *битах* либо *байтах*;
- методы доступа к данным;
- быстродействие (время обращения к устройству);
- надежность работы, характеризуемая зависимостью от окружающей среды и колебаний напряжения питания;
- стоимость единицы памяти.

ЗУ делятся на электронные и электронно-механические. Первые базируются на интегральных схемах, характеризуются высокой стоимостью, обладают большим быстродействием, надежностью в работе. Электронно-механические устройства используют механические средства, но более экономичны и имеют большую емкость памяти. В этой связи в каждой системе создается и используется иерархия ЗУ. Последние делятся как минимум на два класса: опера-

тивные запоминающие устройства (ОЗУ) и внешние запоминающие устройства (ВЗУ).

Внешние запоминающие устройства подробно рассмотрены далее. Здесь же речь пойдет об устройствах оперативной памяти.

В адресном ОЗУ каждый элемент памяти имеет адрес, соответствующий его пространственному расположению в запоминающей среде. Поэтому, обращение к определенному элементу производится в соответствии с кодом его адреса. В ЗУ после приема кода осуществляется его дешифрация, после чего следует выборка из элемента конкретной группы битов или слов.

В ассоциативном ОЗУ поиск данных происходит по конкретному содержимому, независимо от его адреса. Такой поиск информации идет с использованием определенных признаков, например, ключевых слов, которые связаны с искомыми данными. Ассоциативные устройства, хотя и являются более сложными, обеспечивают более быстрый поиск и выбор хранимых данных.

Необходимо отметить, что все распространенные операционные системы, если для работы нужно больше памяти, чем физически присутствует в компьютере, не прекращают работу, а сбрасывают не используемое в данный момент содержимое памяти в дисковый файл (называемый свопом — swap) и затем по мере необходимости «перегоняют» данные между ОЗУ и свопом. Это гораздо медленнее, чем доступ системы к самой ОЗУ. Поэтому от количества оперативной памяти напрямую зависит быстродействие системы в целом.

Основные принципы

Рассмотрим адресные ЗУ. Команды, исполняемые ЭВМ при выполнении программы, равно как и числовые и символьные операнды, хранятся в памяти компьютера. Память состоит из многих миллионов ячеек, в каждой из которых содержится один бит информации (значения 0 или 1). Биты редко обрабатываются поодиночке, а как правило, группами фиксированного размера. Для этого память организуется таким образом, что группы по *n* бит могут записываться и считываться за одну операцию. Группа *n* бит называется *словом,* а значение *n* — длиной слова. Схематически память компьютера можно представить в виде массива слов (рис. 2.14).

Обычно длина машинного слова компьютеров составляет от 16 до 64 бит. Если длина слова равна 32 битам, в одном слове может храниться 32-разрядное число в дополнительном коде или четыре символа ASCII, занимающих 8 бит каждый.

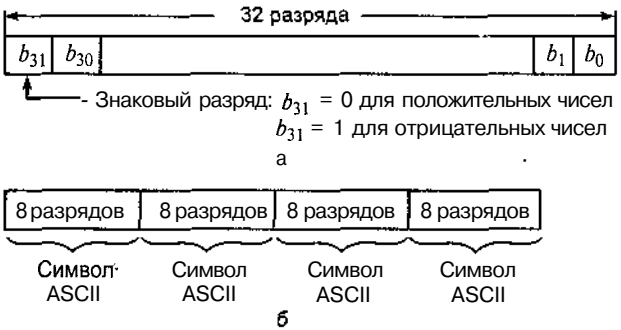


Рис. 2.14. Размещение числовой *(а)* и символьной *(б)* информации в слове

Восемь идущих подряд битов являются байтом. Для представления машинной команды требуется одно или более слов.

Для доступа к памяти с целью записи или чтения отдельных элементов информации, будь то слова или байты, необходимы имена или адреса, определяющие их расположение в памяти. В качестве адресов традиционно используются числа из диапазона от 0 до 2^k - 1 со значением *k,* достаточным для адресации всей памяти компьютера. Все 2^k адресов составляют адресное пространство компьютера. Следовательно, память состоит из 2^k адресуемых элементов. Например, использование 24-разрядных (как в процессоре 80286) адресов позволяет адресовать 2^{24} (16 777 216) элементов памяти. Обычно это количество адресуемых элементов обозначается как 16 Мбайт (1 Мбайт = 2^{20} = 1 048 576 байт, адресное пространство 8086 и 80186). Поскольку у процессоров 80386, 80486, Pentium и их аналогов 32-разрядные адреса, им соответствует адресное пространство из 2^{32} байт, или 4 Гбайт элементов.

Байтовая адресация. Итак, отдельные биты, как правило, не адресуются и чаще всего адреса назначаются байтам памяти. Именно так адресуется память большинства современных компьютеров. Память, в которой каждый байт имеет отдельный адрес, называется памятью с байтовой адресацией. Последовательные байты имеют адреса О, 1, 2 и т. д. Таким образом, при использовании слов длиной 32 бита последовательные слова имеют адреса 1, 4, 8,..., и каждое слово состоит из 4 байт.

Прямой и обратный порядок байтов. Существует два способа адресации байтов в словах:

- в прямом порядке (рис. 2.15, а);
- в обратном порядке (рис. 2.15, *б).*

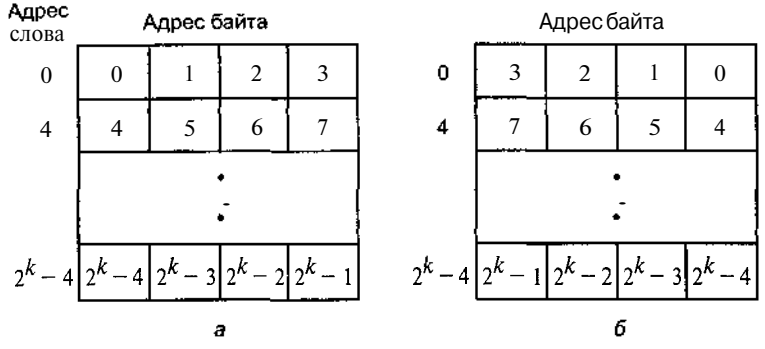


Рис. 2.15. Способы адресации байтов в ОЗУ

Обратным порядком байтов *(big-endian)* называется система адресации, при которой байты адресуются слева направо, так что самый старший байт слова (расположенный с левого края) имеет наименьший адрес.

Прямым порядком байтов *(little-endian)* называется противоположная система адресации, при которой байты адресуются справа налево, так что наименьший адрес имеет самый младший байт слова (расположенный с правого края). Слова «старший» и «младший» определяют вес бита, т. е. степень двойки, соответствующей данному биту, когда слово представляет число. В ПЭВМ на основе 80x86 используется прямой порядок, а в ПЭВМ на основе Motorola 68000 — обратный. В обеих системах адреса байтов 0, 4, 8 и т. д. применяются в качестве адресов последовательных слов памяти в операциях чтения и записи слов.

Наряду с порядком байтов в слове важно также определить порядок битов в байте. Наиболее естественный порядок битов для кодирования числовых данных (непосредственно соответствующий их разрядам) «слева направо»: b_{32}, ..., b1, b_0. Однако существуют компьютеры, для которых характерен обратный порядок битов.

Расположение слов в памяти. В случае 32-разрядных слов их естественные границы располагаются по адресам 0, 4, 8 и т. д. При этом считается, что слова *выровнены по адресам* в памяти. Если говорить в общем, слова считаются выровненными в памяти в том случае, если адрес начала каждого слова кратен количеству байтов в нем. По практическим причинам, связанным с манипулированием двоично-кодированными адресами, количество байтов в слове обычно является степенью двойки. Поэтому, если длина слова равна 16 бит (2 байтам), выровненные слова начинаются по байтовым

адресам 0, 2, 4, ..., а если она равна 64 бит (2^3, т. е. 8 байтам), то выровненные слова начинаются по байтовым адресам 0, 8, 16, ...

Не существует причины, по которой слова не могли бы начинаться с произвольных адресов. Такие слова называются невыровненными. Как правило, слова выравниваются по адресам памяти, но иногда этот принцип нарушается.

Доступ к числам, символам и символьным строкам. Обычно число занимает целое слово, поэтому, для того чтобы обратиться к нему, нужно указать адрес слова, по которому оно хранится. Точно так же доступ к отдельно хранящемуся в памяти символу осуществляется по адресу содержащего его байта.

Во многих приложениях необходимо обрабатывать строки символов переменной длины. Для доступа к такой строке нужно указать адрес байта, в котором хранится ее первый символ. Последовательные символы строки содержатся в последовательных байтах. Существует два способа определения длины строки. Первый из них заключается в использовании специальною управляющего символа, обозначающего конец строки и являющегося ее последним символом. Второй способ состоит в использовании отдельного слова памяти или регистра процессора, содержащего число, которое определяет длину строки в байтах.

Операции с памятью. Как команды программ, так и данные, являющиеся операндами этих команд, хранятся в памяти. Для выполнения команды управляющие схемы процессора должны инициировать пересылку содержащего ее слова или слов из памяти в процессор.

Операнды и результаты также должны пересылаться между памятью и процессором. Таким образом, для выполнения команды программы необходимо произвести две операции с памятью:

Load (также Read, или Fetch) — загрузка (или чтение, или выборка соответственно);

Store (или Write) — сохранение (или запись).

Операция загрузки пересылает в процессор копию содержимого памяти по заданному адресу. При этом содержимое памяти остается неизменным. Для того чтобы начать операцию загрузки, процессор отсылает в память адрес и запрашивает содержимое памяти по этому адресу. Из памяти считываются соответствующие данные и пересылаются в процессор.

Операция сохранения пересылает элемент информации из процессора в память по заданному адресу, уничтожая предыдущие данные, хранившиеся по этому адресу. Для выполнения такой операции процессор отсылает в память данные и адрес, по которому они должны быть записаны.

Информацию из одного слова или одного байта можно переслать между процессором и памятью за одну операцию. Процессор содержит небольшое количество регистров, вмещающих по одному слову. Эти регистры служат либо источниками, либо приемниками данных, пересылаемых в память и из памяти. Пересылаемый байт обычно располагается в младшей (крайней справа) позиции в регистре.

Конкретные системы оперативной памяти. Память, хранящая обрабатываемые в текущее время данные и выполняемые команды (программу), называется основной памятью — RAM (Random Access Memory), т. е. память с произвольным доступом. Она составляет основу системной памяти. В ПК в большинстве случаев основная оперативная память строится на микросхемах динамического типа (DRAM — Dynamic Random Access Memory), где в качестве запоминающего элемента (ЗЭ) используется простейшая сборка, состоящая из одного транзистора и одного конденсатора. Основными причинами широкого применения этой памяти является высокая плотность интеграции (увеличение числа ЗЭ на чип и сокращение числа чипов, необходимых для одного модуля), малое потребление энергии (тратится минимум энергии на хранение одного бита, уменьшается потребляемая системой мощность, снижается стоимость) и т. д. Но имеются и недостатки: каждый ЗЭ представляет, по сути дела, разряжаемый со временем конденсатор, поэтому чтобы предотвратить потерю хранящейся в конденсаторах информации, микросхема RAM постоянно должна регенерироваться.

Имеется другой вид памяти, который лишен этого недостатка. Эта память называется статической (Static RAM — SRAM), где в качестве ЗЭ используется так называемый статический триггер (состоящий из 4—6 транзисторов). Из-за сложности ЗЭ плотность упаковки микросхем SRAM меньше, чем для DRAM. Следовательно, если бы SRAM устанавливалась в качестве оперативной памяти, то это привело бы к увеличению быстродействия ПК, однако при этом существенно изменилась бы его стоимость, поскольку стоимость микросхемы SRAM значительно выше стоимости DRAM. Для повышения быстродействия в настоящее время применяются различные архитектурно-логические решения. Сейчас имеется множество различных типов памяти, отличающихся друг от друга своими основными характеристиками.

Основная память соединяется с процессором посредством *адресной шины* и *шины данных*. Каждая шина состоит из множества электрических цепей (линий или бит). Ширина (разрядность) адресной шины определяет, сколько адресов может быть в ОЗУ (адресное пространство), а шины данных — сколько данных может быть передано

за один цикл. Например, в 1985 г. процессор Intel 386 имел 32-разрядную адресную шину, что дало возможность поддерживать адресное пространство в 4 Гбайт. В процессоре Pentium (1993 г.) ширина шины данных была увеличена до 64 бит, что позволяет передавать 8 байт информации одновременно.

Каждая передача данных между процессором и памятью называется *циклом шины*. Количество бит, которое процессор может передать за один цикл шины, влияет на производительность компьютера и определяет, какой тип памяти требуется.

Для описания характеристик быстродействия оперативной памяти применяются так называемые *циклы чтения/записи* (или временные схемы пакета). Дело в том, что при обращении к памяти на считывание или запись первого машинного слова расходуется больше тактов, чем на обращение к трем последующим словам. Так, для асинхронной SRAM чтение одного слова выполняется за 3 такта, запись — за 4 такта, чтение нескольких слов определяется последовательностью 3—2—2—2 такта, (что означает, что чтение 1-го элемента данных занимает 3 такта ЦП, включая 2 такта ожидания, а чтение последующих — по 2 временных такта), а запись — 4—3—3—3.

Динамическая память

Динамическая память (DRAM) в современных ПК используется обычно в качестве оперативной памяти общего назначения, а также как память для видеоадаптера. Из применяемых в современных и перспективных ПК типов динамической памяти наиболее известны DRAM и FPM DRAM, EDO DRAM и BEDO DRAM, EDRAM и CDRAM, Synchronous DRAM, DDR SDRAM и SLDRAM, видеопамять MDRAM, VRAM, WRAM и SGRAM, RDRAM и некоторые другие (табл. 2.2).

Микросхема памяти этого типа представляет собой прямоугольный массив ячеек со вспомогательными логическими схемами, которые используются для чтения или записи данных, а также цепей регенерации, поддерживающих целостность данных. Массивы памяти организованы в *строки (raw)* и *столбцы (column)* ячеек памяти, именуемые соответственно линиями *слов (wordlines)* и линиями *бит (bitlines)*. Каждая ячейка памяти имеет уникальное размещение, задаваемое пересечением строки и столбца. Цепи, поддерживающие работу памяти, включают:

- усилители, считывающие сигнал, обнаруженный в ячейке памяти;
- схемы адресации для выбора строк и столбцов;

Таблица 2 2 **Некоторые характеристики различных типов динамической памяти**

Характеристика	PC100	PC133	DDR SDRAM	SLDRAM	Base Rambus	Concurrent Rambus	Direct Rambus
Частота, МГц	100	133	200/266	800	700	700	600/800
Максимальная скорость, Гбайт/с	0,80	1,00	1,6/2,1	1,60	0,70	0,70	1,2/1,6
Ожидаемая скорость, Гбайт/с	0,50	0,60	0,9/1,2	–	0,40	0,50	1,1/1,5
Эффективность, %	65	60	60	–	60	80	97
Число бит в слове	64	64	64	16	8/9	8/9	16/18

- схемы выбора адреса строки (Row address select — /RAS) и столбца (Column address select — /CAS), чтобы открывать и закрывать адреса строк и столбцов, а также начинать и заканчивать операции чтения и записи;
- цепи записи и чтения информации;
- внутренние счетчики или регистры, следящие за циклами регенерации данных;
- схемы разрешения вывода (Output enable — OE).

Каждый бит такой памяти представляется в виде наличия (или отсутствия) заряда на конденсаторе, образованном в структуре полупроводникового кристалла. Конденсатор управляет транзистором. Если транзистор открыт и ток идет, это означает «1», если закрыт — «О». С течением времени конденсатор разряжается, и его заряд нужно периодически восстанавливать. Между периодами доступа к памяти посылается электрический ток, обновляющий заряд на конденсаторах для поддержания целостности данных (вот почему данный тип памяти называется динамическим ОЗУ). Этот процесс называется регенерацией памяти. Интервал регенерации измеряется в наносекундах (нс) и это число отражает «скорость» ОЗУ. Большинство ПК на основе процессоров Pentium используют скорость 60 или 70 нс. Процесс регенерации снижает скорость доступа к данным, поэтому доступ к DRAM обычно осуществляется через кэш-память. Однако когда быстродействие процессоров превысило 200 МГц, кэширование перестало существенно влиять на присущую DRAM низкую скорость и возникла необходимость использования других технологий ОЗУ.

Цикл чтения включает следующие события (рис. 2.16, для EDO DRAM):

- выбор строки. Активизация цепи /RAS используется для связывания со строкой памяти и инициации цикла памяти. Это

требуется при начале каждой операции с памятью. Активное состояние /RAS задается низким уровнем напряжения на линии, т. е. сигнал /RAS соответствует переходу от высокого напряжения в цепи к низкому. Сигнал /RAS может также использоваться для запуска цикла регенерации.

- выбор столбца. Сигнал /CAS используется для связывания со столбцом памяти и инициации операции записи-чтения. Активное состояние /CAS также задается низким напряжением на линии.
- разрешение записи (Write enable/WE). Сигнал/WE задает тип операции; высокий уровень напряжения определяет операцию записи, низкий — чтения информации.
- разрешение вывода (Output enable /OE). Во время операций чтения из памяти этот сигнал предотвращает появление данных прежде времени. Если уровень напряжения в цепи низкий, то данные передаются на выходные линии, как только возможно. При записи в память эта линия игнорируется.
- ввод/вывод данных. Выводы DQ (также именуемые входо-выходными или I/Os) на чипе памяти предназначены для ввода и вывода. Во время операции записи высокое («1») или низкое («О») напряжение подается на DQ. При чтении данные считываются из выбранной ячейки и передаются на DQ, если доступ осуществлен и /OE открыт. Все остальное время DQ находятся в закрытом состоянии (высокое входное сопротивление) — они не потребляют электрический ток и не выдают сигналов.

Рассмотрим модификации систем динамической оперативной памяти.

FPM DRAM (Fast page mode DRAM) — представляет собой стандартный тип памяти, быстродействие которой составляет 60 или 70 нс. Система управления памятью в процессе считывания активирует адреса строк, столбцов, осуществляет проверку данных и передачу информации в систему. Столбцы после этого деактивируются, что приводит к нежелательному состоянию ожидания процессора в некоторых сочетаниях операций с памятью. В наилучшем случае данный режим реализует временную схему пакета вида 5-3-3-3.

EDO RAM (RAM с расширенным выходом). Обращение на чтение осуществляется таким же образом, как и в FPM, за исключением того, что высокий уровень /CAS не сбрасывает выходные данные, а использование триггера позволяет сохранять данные до тех пор, пока уровень CAS снова не станет низким. Тем самым не про-

исходит сброса адреса столбцов перед началом следующей операции с памятью.

Упрощенная схема работы EDO показана на рис. 2.16. Выходная величина поддерживается последовательностью стробирующих импульсов до тех пор, пока она не будет считана ЦП, что особенно важно для быстрых процессоров наподобие Pentium; эта память обеспечивает лучшие параметры для серии быстрых последовательных считываний, чем FPM RAM. Теоретически быстродействие памяти на 27 % выше, чем для FMP DRAM.

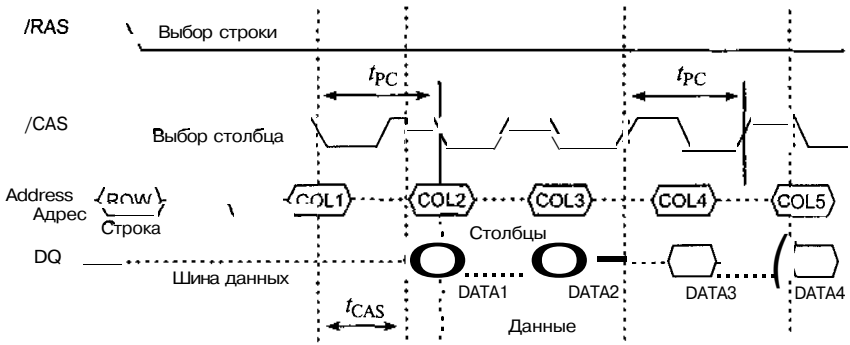


Рис. 2.16. Временная диаграмма EDO DRAM

Данный вид памяти является модификацией типовой FPM RAM с небольшими отличиями во временной последовательности /CAS и выходных данных. EDO DRAM обеспечивает более частую выдачу выходных данных, чем стандартная DRAM. Наибольшая скорость EDO RAM в циклах процессора — это 5—2—2—2 для пакета чтения из четырех величин (байт/слово/двойное слово). Память выпускается в трех вариантах — 70, 60 и 50 нс. EDO RAM не может работать при частоте шины, превышающей 66 МГц, а этот предел уже достигнут.

BEDO RAM (Burst extended data out DRAM — пакетная с расширенным выходом), как это видно из названия, читает данные в виде пакета, что означает, что после получения адреса каждая из следующих трех единиц информации читается за один цикл таймера, а процессор считывает данные в виде пакета 5—1—1—1. Быстродействие системы на 100 % превосходит FPM и на 50 % — EDO DRAM.

Обращение к BEDO на чтение имеет два отличия от доступа к EDO. Первое из них — это то, что в первом цикле /CAS данные не попадают на выходы. Преимущество такого внутреннего конвейер-

ного звена состоит в том, что во втором цикле время появления данных после выдачи переднего фронта /CAS (т. е. t_{CAS}) будет меньше. Другое отличие состоит в том, что системы BEDO содержат внутренний счетчик адреса, т. е. они получают извне только первый из четырех последовательных адресов. Первый цикл /CAS, загружающий внутреннее конвейерное звено, не приводит" к задержке при получении первого элемента данных.

Основным недостатком BEDO RAM является также невозможность работы на частоте шины, превышающей 66 МГц.

SDRAM (Synchronous DRAM — синхронная динамическая память). Этот тип памяти существенно отличается от других тем, что использует тот факт, что большинство обращений к памяти являются последовательными и спроектирован так, чтобы передать все биты пакета данных как можно быстрее (когда начинается передача пакета, все последующие биты поступают с интервалом 10 нс). SDRAM содержит в своем составе счетчик пакетов, который автоматически увеличивает адреса и обеспечивает быструю последовательную выборку. Контроллер памяти обеспечивает локализацию требуемого блока памяти с максимальной скоростью (рис. 2.17).

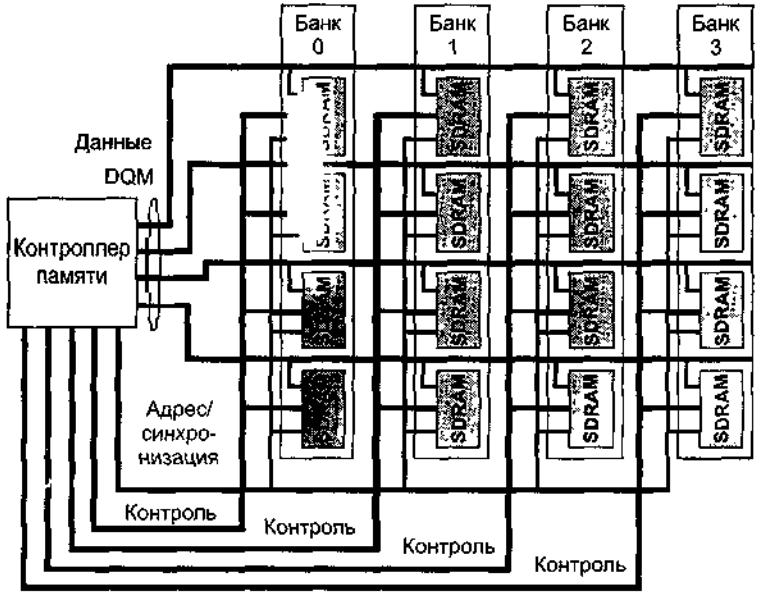


Рис. 2.17. Организация SDRAM

Данная система памяти может превосходить по быстродействию EDO RAM на 18 %.

Как видно из названия, этот тип памяти обеспечивает синхронизацию всех входных и выходных сигналов с системным таймером. Наибольшая скорость SDRAM в циклах процессора — это 5—1—1—1 для пакета чтения четырех единиц информации (байт/слово/двойное слово), что делает ее такой же быстродействующей, как и BEDO RAM; однако самое большое достоинство SDRAM — то, что она легко поддерживает частоту шины до 100 МГц.

SDRAM PC100. Для материнских плат, поддерживающих внешние частоты в 100 МГц и выше, необходима память (SDRAM), которая сможет нормально и без сбоев работать с такими частотами, обеспечивая оптимальную скорость. Такие модули памяти должны иметь время доступа не более 8 нс, но самого быстродействия как такового недостаточно. Память, способная устойчиво работать на внешних частотах 100 МГц и выше, должна удовлетворять специальному стандарту — PC100.

SDRAM PC133 — память, соответствующая стандарту PC133. Спецификация PC133 SDRAM DIMM разработана группой компаний VIA Technologies, IBM Microelectronics, Micron Semiconductor Products, NEC Electronics, Samsung Semiconductor (Revision 0.4, 7 июня 1999 г.). Было установлено, что память будет совместима с более ранними технологиями, стоить дешевле, хотя и не сможет работать на частотах выше 133 МГц. Память PC133 — это лучшие образцы памяти стандарта PC100, ускоренные до 133 МГц.

Спецификация PC133 почти ничем не отличается от PC100 (табл. 2.2а). Пиковая пропускная способность PC133 SDRAM приблизительно равна 1 Гбайт/с и средняя пропускная способность около 250 Мбайт/с, что соответствует пропускной способности AGP 4-х (1 Гбайт/с — пиковая и 200 Мбайт/с — средняя). Пиковая пропускная способность PC100 SDRAM приблизительно 800 Мбайт/с, что меньше, чем у порта AGP 4х; т. е. память PC133 может использоваться в графических станциях и других аналогичных системах.

Таблица 2.2а. **Сравнительные характеристики стандартов PC100 и PC133**

Параметр, нс	PC100		PC133	
	Минимально допустимое	Наилучшее	Минимально допустимое	Наилучшее
Время доступа	10	8	7,5	6
Время удержания данных на выходе	3		2,7	
Время установки	2		1,5	

Требования к конструктивному исполнению были заложены в спецификацию PC100 с большим запасом, поэтому в их изменении не было необходимости. Практически аналогичные требования к памяти PC133 предлагаются и фирмой VIA Technologies в своей спецификации на PC13.

Пиковая пропускная способность PC133 SDRAM приблизительно равна 1 Гбайт/с и средняя пропускная способность около 250 Мбайт/с, что соответствует пропускной способности AGP 4-х (1 Гбайт/с — пиковая и 200 Мбайт/с — средняя). Пиковая пропускная способность PC100 SDRAM приблизительно 800 Мбайт/с, что меньше, чем у порта AGP 4-х.

DDR SDRAM (SDRAM II). Это еще один тип конкурирующих технологий. Традиционно, по логике устройств с синхронизацией, данные передаются по фронту импульса синхронизации (clock tick). Так как сигнал генератора импульсов изменяется между «1» и «О», данные могут передаваться как по переднему фронту импульса (изменение с «О» на «1»), так и по заднему (с «1» на «0»).

Следующим шагом в развитии Synchronous DRAM (SDRAM) может стать предложенная компанией Samsung DDR (Double Data Rate) SDRAM или SDRAM II, в которой передача данных осуществляется по обоим фронтам тактовых импульсов одновременно, чем достигается удвоение скорости передачи при той же тактовой частоте. То есть DDR позволяет выполнить две операции доступа к данным из двух разных модулей, находящихся в одном банке памяти, за время одного обращения стандартной SDRAM благодаря более точной внутренней синхронизации. Это есть дальнейшее развитие принципа чередования данных для увеличения скорости доступа к ним.

Кроме того, DDR использует DLL (delay-locked loop — цикл с фиксированной задержкой) для выдачи сигнала DataStrobe, означающего доступность данных на выходных контактах. Используя один сигнал DataStrobe на каждые 16 выводов, контроллер может осуществлять доступ к данным более точно и синхронизировать входящие данные, поступающие из разных модулей, находящихся в одном банке. DDR фактически увеличивает скорость доступа вдвое по сравнению с SDRAM, используя при этом ту же частоту.

В случае с 64-битовой шиной — это два 8-байтных пакета, 16 байт за такт. Или в случае с шиной на 133 МГц — не 1064, а 2128 Мбайт/с.

Цена, энергопотребление и площадь микросхемы DDR отличаются не более чем на несколько процентов от соответствующих DRAM. Следует отметить, что первые чипы DDR использовали производители видеокарт GeForce 256. Производительность карт на системах с мощным центральным процессором при использовании

приложений, оказывающих заметную нагрузку именно на шину памяти (например, 32-бит цвет), возрастает в 1,5 раза.

DDR-II SDRAM. К числу основных отличий технологии DDR-II от предыдущего варианта (DDR-I) относится то, что в ней размер выборки данных увеличен вдвое — с 2 до 4 бит, а значит, во столько же раз возрастает и скорость передачи данных. Например, при 100 МГц она составит 400 Мбайт/с. Кроме этого, в DDR-II планируется использовать новую схему синхронизации.

Также память DDR-II отличается от DDR-I более низким напряжением питания — 1,8 вместо 2,5 В. Изменена схема компоновки, как на уровне отдельных микросхем, так и на уровне модулей, в частности, предполагается, что модули DDR-II DIMM будут иметь не 184 контакта, как DDR-I DIMM, а 230 контактов.

SLDRAM (Synchronous linked DRAM). Этот тип устройств разработан консорциумом крупнейших производителей модулей памяти — SLDRAM Consortium. Считается, что применение SLDRAM экономически выгодно при объеме ОЗУ не менее 256 Мбайт. Этот тип памяти включает основные прогрессивные технологии, заложенные в его предшественниках — SDRAM и DDR RAM. Повышение производительности достигается за счет распространения пакетного протокола передачи данных на сигналы управления (отчего и пошло название этого типа памяти — Linked SDRAM). В SLDRAM адреса, команды, а также сигналы управления передаются в пакетном режиме по однонаправленной шине Command Link.

Одновременно с ними по другой, двунаправленной шине Data Link, и тоже в пакетном режиме передаются данные, причем передача происходит на обоих фронтах тактовых импульсов, как и в случае с DDR SDRAM. Величина всего пакета данных может равняться целой странице (строке памяти). Поскольку пропускная способность обеих шин (команд и данных) одинакова, можно переключаться на любую страницу памяти без потери производительности.

По сравнению со SDRAM набор команд у SLDRAM значительно увеличен, что очень облегчает работу контроллера. Команда представляет собой четыре 10-битных пакета и содержит всю информацию для проведения следующей операции. Таким образом, возрастает эффективность управления памятью — всего за четыре такта передается вся информация, описывающая массив данных.

Максимальная достижимая нынешним поколением SLDRAM скорость передачи превышает 1 Гбайт/с на каждый разряд при частоте 400 МГц. Надо заметить, что при такой частоте очень важно, чтобы все сигналы точно синхронизировались с тактовыми импульсами системной шины, и чтобы все микросхемы памяти в пределах

одного модуля имели близкие временные задержки. Для этого контроллер программирует все чипы модуля памяти так, чтобы они выдавали данные на шину одновременно, независимо от разброса их параметров и степени удаленности микросхем от контроллера. В результате самая удаленная микросхема выдает данные без задержки, а самая близкая — через промежуток времени, нужный, чтобы сигнал распространился от самой удаленной до самой близкой. Эти значения определяются в момент подачи питания на ИС и постоянно корректируются во время работы.

ESDRAM (Enhanced SDRAM — улучшенная SDRAM) — более быстрая версия SDRAM, сделанная в соответствии со стандартом JEDEC компанией Enhanced Memory Systems (EMS). С точки зрения времени доступа производительность ESDRAM в 2 раза выше по сравнению со стандартной SDRAM. В большинстве приложений ESDRAM, благодаря более быстрому времени доступа к массиву SDRAM и наличию кэша, обеспечивает даже большую производительность, чем DDR SDRAM.

Сначала появился EDRAM (с асинхронным интерфейсом), а затем с появлением SDRAM был разработан ESDRAM (с синхронным). Основные его отличия от SDRAM:

- более быстрое время доступа (27 нс вместо стандартных 60 нс);
- производительность, повышенная почти до уровня статического ОЗУ, по цене динамического;
- кэш-память, связанная с каждым банком памяти;
- скрытая регенерация;
- гибкое использование кэш-памяти для обеспечения максимальной производительности при различных типах обращений.

Принцип работы ESDRAM в том, что из динамической в кэш-память целиком переносится вся строка, в которой находится считываемая ячейка. После этого считывание производится уже из кэш-памяти, а в основной памяти в это время можно выбирать нужную строку или производить регенерацию. Перенос почти не сказывается на быстродействии, поскольку длится один такт.

ESDRAM может работать в режиме «упреждающего обращения» к массиву SDRAM, в результате следующий цикл записи или чтения может начаться в момент, когда выполнение текущего цикла не завершено. Возможность использовать такой режим напрямую зависит от центрального процессора, управляющего работой конвейера адресации.

Операция записи, в отличие от чтения, происходит в обход кэш-памяти, что увеличивает производительность ESDRAM при возобновлении чтения из ранее уже загруженной в кэш строки. При этом скорость работы ячеек ESDRAM составляет 22 нс в отличие от стандартной скорости работы ячеек SDRAM, имеющей значения 50—60 нс.

Недостаток ESDRAM — усложнение контроллера: он должен учитывать возможность подготовки к чтению новой строки памяти. Кроме того, при произвольных адресах чтения кэш-память используется крайне неэффективно, поскольку чтение строки памяти целиком происходит очень редко. Этого недостатка нет у другого типа памяти — CDRAM.

CDRAM (Cached DRAM — DRAM с кэш-памятью). Этот тип ОЗУ разработан в корпорации Mitsubishi и представляет собой улучшенный вариант ESDRAM. Изменения коснулись кэш-памяти — ее объема, принципа размещения данных, средств доступа. Cached DRAM имеет раздельные адресные линии для статического кэша и динамического ядра памяти. Необходимость управлять разнородными типами памяти усложняет контроллер, однако эффективность кэш-памяти, размещенной «внутри» микросхемы, выше, чем при традиционной архитектуре ПК, так как перенос в кэш осуществляется блоками, в восемь раз большими, чем при выдаче «наружу» из микросхемы обычной DRAM. В CDRAM объем одного блока данных, помещаемого в кэш, уменьшен до 128 битов. Это позволяет использовать кэш-память гораздо эффективнее, чем в ESDRAM, поскольку в 16-килобитной кэш-памяти могут одновременно храниться данные из 128 различных участков памяти. Затирание первого помещенного в кэш участка памяти начнется лишь при обращении к 129-му. Поскольку перенос из DRAM в SRAM совмещен с выдачей данных на шину, то частые, но короткие пересылки не снижают производительности всей микросхемы при перекачке больших объемов информации и уравнивают CDRAM с EDRAM, а при выборочном чтении преимущество остается за CDRAM.

Direct Rambus (DRDRAM). Обычная архитектура DRAM достигает своего практического потолка при частоте ЦП в 300 МГц. Появление концепции Direct Rambus DRAM (1999 г.) дает долговременное решение этой проблемы.

DRDRAM — высокоскоростная динамическая память с произвольным доступом, разработанная Rambus Inc. Она обеспечивает высокую пропускную способность по сравнению с большинством других DRAM. Direct Rambus DRAMs представляет интегрированную на системном уровне технологию.

Подсистема памяти Rambus состоит из следующих компонентов:

- основного контроллера (RMC — Rambus Memory Controller);
- канала (RC — Rambus Channel);
- разъема для модулей (RRC — Rambus RIMM Connector);
- модуля памяти (RIMM — Rambus In-line Memory Module);
- генератора дифференциальных импульсов (DRCG — Direct Rambus Clock Generator);
- микросхемы памяти (RDRAM — Rambus DRAM).

Рассмотрим основные компоненты данной системы (рис. 2.18).

Direct Rambus Controller — главная шина подсистемы памяти, помещается на чипе, таком, как PC-чипсет, микропроцессор, графический контроллер и пр. Физически можно поместить до четырех контроллеров Direct Rambus на одном чипе. Контроллер представляет собой интерфейс между чипом логики и каналом Direct Rambus. В его обязанности входит генерирование запросов, управление потоком данных и еще ряд функций. Direct Rambus-контроллер состоит из двух отдельных функциональных блоков: Rambus Application Specific Integrated Circuit Cell (RAC) и контроллера Rambus. Роль RAC — обеспечить физический и электрический интерфейс контроллера с внешней шиной, а 8-разрядный контроллер Rambus служит ядром 16-разрядного контроллера Direct Rambus. Шина Direct Rambus Channel соединяет чипы памяти друг с другом и модули RIMM с контроллером. Она состоит из трех независимых шин: 16-разрядной шины данных и двух шин адреса (отдельно для строк и для столбцов) общей «шириной» 8 битов.

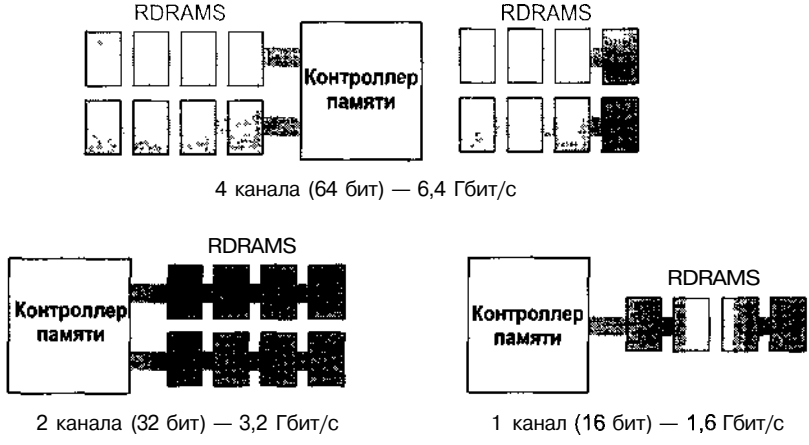


Рис. 2.18. Варианты построения RDRAM

Direct Rambus Channel создает электрическое соединение между Rambus-контроллером и чипами Direct RDRAM. Работа канала основана на 30 сигналах, составляющих высокоскоростную шину. Эта шина работает на тактовой частоте 400 МГц и позволяет передавать данные на 800 МГц (данные передаются по обоим фронтам такта). Такая высокая частота достигается за счет использования некоторых технических приемов. Два канала данных (шириной в байт каждый) позволяют получить пиковую пропускную способность в 1,6 Гбайт/с. Канал может быть выполнен на обычных системных платах и соответствует форм-фактору SDRAM.

*RIMM*содержит 16 банков памяти (до 32 — в реализации NEC). К каждому из банков подключаются два усилителя, объединенных двумя внутренними шинами. Впервые RDRAM стали устанавливать в высокопроизводительных графических станциях Silicon Graphics. Чтобы обеспечить нормальное функционирование системы с сигналами, измеряемыми долями наносекунд (1,25 нс), была разработана специальная топология шины модуля памяти. Этот модуль назвали RIMM — Rambus In-line Memory Module (по аналогии с SIMM и DIMM). Его особенность — одинаковая длина всех сигнальных дорожек; благодаря этому ко всем чипам RDRAM модуля сигналы приходят одновременно. Дело в том, что при работе на частоте 800 МГц размеры RIMM становятся соизмеримыми с длиной волны (около 37 см), что ужесточает требования к расположению элементов на модуле. Работа на восьмикратной скорости достигается благодаря целому комплексу мер:

- малой амплитуде сигналов (Rambus Signaling Logic с 0,8 В);
- хорошо продуманной топологии шины;
- плотной установке микросхем на модуле;
- установке заглушек на концах дорожек, чтобы гасить отражение сигнала.

Применение нескольких распространяющихся в противофазе тактовых импульсов сводит к минимуму влияние помех. Специальная ИС тактирования микросхем (RCG — Direct Rambus Clock Generator) дает возможность согласовать направление распространения данных по шине с направлением распространения тактовых импульсов. Это позволяет надежно считывать информацию из чипа RDRAM независимо от того, насколько они удалены от контроллера.

Memory Expansion. Один канал Direct Rambus максимум может поддерживать 32 чипа Direct RDRAM. В материнской плате может использоваться до трех RIMM-модулей. Используя 64-Мбит, 128-Мбит и 256-Мбит устройства, максимальная емкость памяти на

канал достигает 256 Мбайт, 512 Мбайт и 1 Гбайт соответственно. Для поддержки целостности канала все свободные RIMM-слоты должны заполняться continuity-модулями.

Чтобы расширить канал сверх 32 устройств, могут использоваться два чипа-повторителя. С одним повторителем канал может поддерживать 64 устройства на 6 RIMM-модулях, а с двумя — 128 устройств на 12 модулях.

Direct Rambus DRAM. Чипы Direct Rambus DRAM составляют часть подсистемы Rambus, запоминающую данные, это непосредственно носители информации. Все устройства в системе электрически расположены на канале между контроллером и терминатором. Устройства Direct Rambus могут только отвечать на запросы контроллера, который делает их шину подчиненной или отвечающей.

VCM (Virtual Channel Memory) — разработанная NEC и Siemens технология, позволяющая оптимизировать доступ к оперативной памяти нескольких процессов (запись данных центральным процессором, перенос содержимого оперативной памяти на жесткий диск, обращения графического процессора и т. п.) таким образом, что переключение между процессами не приводит к падению производительности. В отличие от традиционной схемы, когда все процессы делят одну и ту же шину ввода-вывода, в технологии VCM каждый из них использует «виртуальную» шину. Организованное на уровне чипа взаимодействие «виртуальных» и реальной шины позволяет достичь прироста производительности системы до 25 %. Схема VCM может быть реализована в рамках уже существующей технологии.

Фирма определила новый протокол и схемные решения, разрешающие подсистемам, обращающимся к памяти, управлять виртуальными каналами (VC) — независимыми интерфейсными блоками DRAM. Любой прибор, скажем, L2-контроллер или графический процессор, должен иметь свой виртуальный канал. Каждый канал содержит статический буфер страниц. Прибор может читать или писать в буферы, копировать их или загружать из накопителя DRAM. Операционная система, распознающая архитектуру VCM, может назначить собственный виртуальный канал.

В современных системах доступ к памяти разных контроллеров, установленных на шине PCI, обычно чередуется в трудно предсказуемом порядке. Доступ к разным участкам основной памяти инициируется как программными приложениями, разработанными с помощью модульных языков (например, C++), так и многозадачными операционными системами. Поэтому микросхемы DRAM вынуждены работать с разными страницами. Возникает задержка, тре-

буемая для предзаряда битовых линий накопителя и передачи информации от ячеек памяти через усилители считывания к блоку ввода/вывода, что часто приводит к значительному замедлению работы всей системы.

При разработке VCM основными целями являлись снижение длительности задержки, а также снижение энергопотребления модулей памяти. Добиться выполнения этой задачи, невыполнимой, если судить по опыту Direct RDRAM, где механизм управления питанием является одним из основных источников задержек, удалось следующим образом.

При работе обычной памяти Memory Master (любое активное системное устройство, которому по какой-то причине понадобился доступ к системной памяти, — контроллер PCI или AGP, кэш процессора L2, видеокарта, и т. п.) выдает запрос, обладающий уникальными характеристиками, — адресом, размером блока данных, и т. д.

При наличии нескольких устройств, одновременно выполняющих запросы в разные области памяти (причем доступ в один момент времени может иметь только одно из них), о большой эффективности работы говорить не приходится (рис. 2.19, *а)*.

В соответствии с технологией VCM в VC SDRAM активное устройство (memory master) может сделать запрос, обладающий уникальными характеристиками — адресом, размером блока данных к памяти, посредством виртуальных каналов. Системный контроллер памяти ассоциирует каналы с процессами, что ускоряет работу системы, как если бы каждому процессу выделялся отдельный ресурс — доступ к памяти. Каждый канал может выполнить обмен данными с любой строкой любого банка памяти.

По этой технологии при записи данные не сразу заносятся в память, а помещаются в буфер — виртуальный канал — и хранятся там до тех пор, пока память не будет готова их принять (она, например, может быть занята регенерацией или обменом с другим устройством) (рис. 2.19, *б)*.

Запись данных в VC SDRAM выполняет следующим образом: вначале данные записываются в виртуальный канал, а потом по мере освобождения контроллера DRAM происходит запись непосредственно в ячейки памяти SDRAM.

Чтение данных осуществляется путем запроса данных у виртуального канала, который в соответствии с запросом напрямую считает содержимое блока ячеек SDRAM, содержащих необходимые данные, и дополнительно выполнит упреждающее чтение.

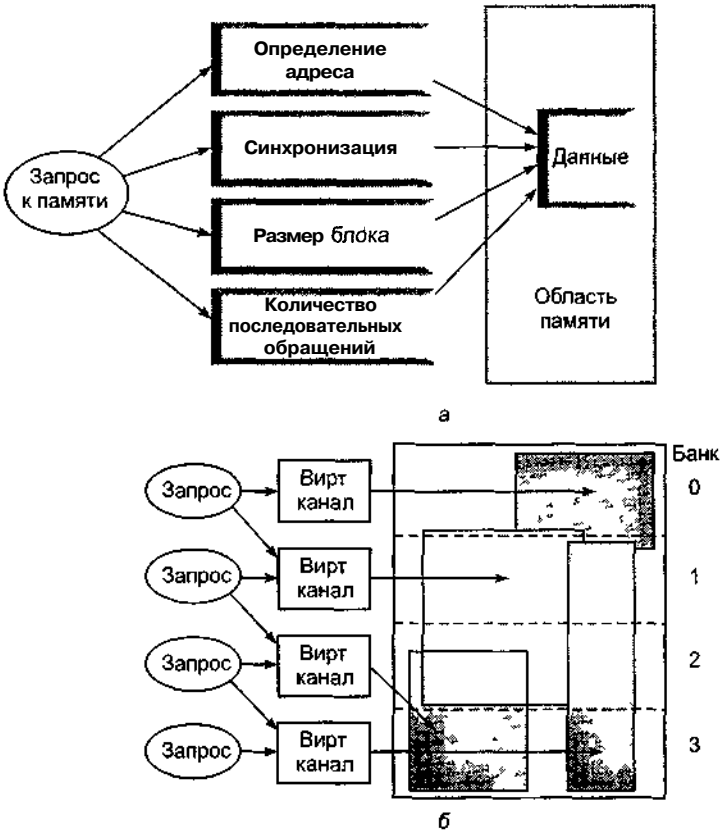


Рис. 2 19. Организация доступа к обычной DRAM (а) и VCM *(б)*

Чтобы при одновременном обращении к памяти нескольких процессов не снизилась производительность, число каналов доведено до 16 по 1024 бита каждый (в модулях по 256 Мбайт), причем каждый канал может передавать до 2048 бит Работает VC SDRAM при частоте до 143 МГц Тип корпуса — стандартный, совместимый по контактам и набору команд с SDRAM

В итоге эффективность доступа к памяти значительно повышается, особенно если учесть, что ничто не мешает, например, той же видеокарте открыть три таких канала — один для загрузки вершин треугольников, второй для загрузки текстур, третий для системного обмена с памятью

По данным NEC увеличение эффективности может достичь 90 %, а по тестам VCM133 SDRAM превосходит PC133 на 10—30 %

Это включает уменьшившиеся задержки, и более высокую пропускную способность, и уменьшение энергопотребления (примерно на те же 30 %) за счет того, что в тот момент, когда происходит передача результатов запроса системному устройству, вся фоновая активность по другую сторону виртуального канала может быть приостановлена

Active Link — разработка NEC, которая использует в DRAM архивацию (сжатие информации) Чтобы не загружать этой работой процессор, функции компрессии/декомпрессии данных возлагаются на микросхемы DRAM В результате несколько расширилось обрамление кристалла, но получен двойной выигрыш — нужна меньшая по количеству ячеек микросхема DRAM, и доступ к информации происходит быстрее, чем обычно Поскольку все больше информации в компьютерах имеет мультимедийную природу, то может быть выбран соответствующий алгоритм компрессии Процессор будет иметь возможность управлять DRAM (например, для выбора алгоритма компрессии) не только обычным образом (через контроллер памяти), но и непосредственно По сведениям NEC, видеоданные сжимаются в чипе ActiveLink в 4 раза

IRAM (Intellectual Random Access Memory) Главная идея технологии IRAM заключается в размещении процессора и DRAM в одном чипе Это дает возможность считывания и записи данных длинными словами (в пределах 128—16 384 бит), обеспечивая высокую пропускную способность памяти Раньше это было невозможно — все упиралось в неприемлемо большое число выводов микросхемы Средняя скорость RAC/CAS равна приблизительно 10—30 нс для модулей емкостью 64—256 Мбайт IRAM При этом снижается энергопотребление и уменьшается место, занимаемое микросхемами памяти

Магнитная оперативная память. Следует отметить, что первые образцы ОЗУ были построены на магнитных ферритовых сердечниках, которые пронизывали адресные и информационные шины (провода) Емкость таких ЗУ обычно не превосходила 64 Кбайт В последующем длительный период времени устройства ОЗУ выполнялись на кремниевых полупроводниковых элементах

В 2000 г IBM и немецкая фирма по производству полупроводников Infineon Technologies AG объявили программу разработки MRAM (Magnetic Random Access Memory) Принцип организации элементов памяти — магнитная среда, заключенная между металлическими пленками, образующими линии записи и чтения данных (рис 2 20)

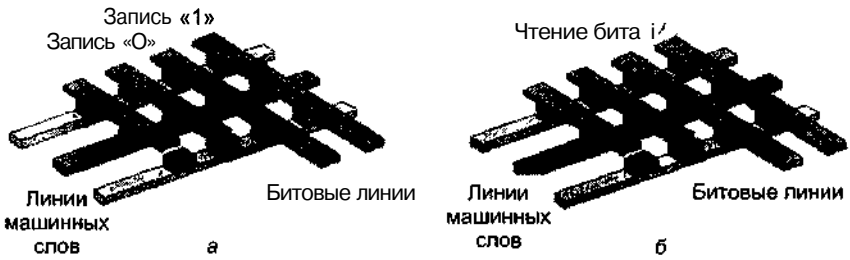


Рис. 2.20. Принцип функционирования MRAM
а — запись данных, *б* — считывание

Преимущества технологии — высокая емкость и скорость, низкая стоимость, возможность применения как в форме статической, так и динамической памяти, более низкое энергопотребление.

Статическая память

Статическая память (SRAM) обычно применяется в качестве кэш-памяти второго уровня (L2) для кэширования основного объема ОЗУ. Статическая память выполняется обычно на основе ТТЛ-, КМОП- или БиКМОП-микросхем и по способу доступа к данным может быть как *асинхронной,* так и *синхронной.* Асинхронным называется доступ к данным, который можно осуществлять в произвольный момент времени. Асинхронная SRAM применялась на материнских платах для третьего — пятого поколений процессоров. Время доступа к ячейкам такой памяти составляло от 15 нс (33 МГц) до 8 нс (66 МГц).

Синхронная память обеспечивает доступ к данным не в произвольные моменты времени, а одновременно (синхронно) с тактовыми импульсами. В промежутках между ними память может готовить для доступа следующую порцию данных. В большинстве материнских плат пятого поколения используется разновидность синхронной памяти — синхронно-конвейерная SRAM (Pipelined Burst SRAM), для которой типичное время одиночной операции чтения/записи составляет 3 такта, а групповая операция занимает 3—1—1—1 такта при первом обращении и 1—1—1—1 при последующих обращениях, что обеспечивает ускорение доступа более чем на 25 %.

SRAM в качестве элементарной ячейки использует так называемый *статический триггер* (схема которого состоит из нескольких транзисторов). Статический тип памяти обладает более высоким быстродействием и используется, например, для организации кэш-памяти. Рассмотрим разновидности статической памяти.

Async SRAM (Асинхронная статическая память) Это кэш-память, которая используется в течение многих лет с тех пор, как появился первый 386-й компьютер с кэш-памятью второго уровня. Обращение к ней осуществляется быстрее, чем к DRAM, и может, в зависимости от скорости процессора, использовать варианты с 20-, 15- или 10-нс доступом (чем меньше время обращения к данным, тем быстрее память и тем короче может быть пакетный доступ к ней). Тем не менее, как видно из названия, эта память является недостаточно быстрой для синхронного доступа, что означает, что при обращении процессора все-таки требуется ожидание, хотя и меньшее, чем при использовании DRAM.

SyncBurst SRAM (Синхронная пакетная статическая память). При частотах шины, не превышающих 66 МГц, синхронная пакетная SRAM является наиболее быстрой из существующих видов памяти. Причина этого в том, что, если процессор работает на не слишком большой частоте, синхронная пакетная SRAM может обеспечить полностью синхронную выдачу данных, что означает отсутствие задержки при пакетном чтении процессором 2—1—1—1, т. е. синхронная пакетная SRAM выдает данные в пакетном цикле 2—1—1—1. Когда частота процессора становится больше 66 МГц, синхронная пакетная SRAM не справляется с нагрузкой и выдает данные пакетами по 3—2—2—2, что существенно медленнее, чем при использовании конвейерной пакетной SRAM. К недостаткам относится и то, что синхронная пакетная SRAM производится меньшим числом компаний и поэтому стоит дороже. Синхронная пакетная SRAM имеет время адрес/данные от 8,5 до 12 нс.

Существует несколько основных конструктивных особенностей синхронной пакетной SRAM, которые делают ее существенно превосходящей асинхронную SRAM при использовании в качестве высокоскоростной кэш-памяти:

- синхронизация с системным таймером. В простейшем смысле это означает, что все сигналы запускаются от фронта сигнала таймера Получение сигналов по фронту тактового импульса таймера существенно упрощает создание быстродействующей системы;
- пакетная обработка. Синхронные пакетные SRAM обеспечивают высокое быстродействие при небольшом количестве логических схем, организующих циклическую работу памяти с последовательными адресами. Четырехадресная пакетная последовательность может быть перемежающейся для совместимости с Intel или линейной для PowerPC и остальных систем.

Указанные особенности дают микропроцессору возможность более быстрого доступа к последовательным адресам, чем это можно сделать при других способах использования технологии SRAM. Хотя у некоторых поставщиков и имеется асинхронная SRAM 3 3V со временем таймер-данные, равным 15 нс, конвейерная синхронная пакетная SRAM, выполненная по такой же технологии, может обеспечить время таймер-данные менее 6 нс.

PB SRAM (Конвейерная пакетная статическая память). Конвейер — это распараллеливание операций SRAM с использованием входных и выходных регистров. Заполнение регистров требует дополнительного начального цикла, но, будучи однажды заполненными, регистры обеспечивают быстрый переход к следующему адресу за то время, пока по текущему адресу считываются данные.

Благодаря этому такая память является наиболее быстрой кэш-памятью для систем с производительностью шины более 75 МГц. PB SRAM может работать при частоте шины до 133 МГц. Она, кроме того, работает не намного медленнее, чем синхронная пакетная SRAM при использовании в медленных системах: она выдает данные все время пакетами по 3—1—1—1. Насколько высока производительность этой памяти, можно видеть по времени адрес/данные, которое составляет от 4,5 до 8 нс.

1-T SRAM. Как уже отмечалось ранее, традиционные конструкции SRAM используют статический триггер для запоминания одного разряда (ячейки). Для реализации одной такой схемы на плате должно быть размещено от 4 до 6 транзисторов (4-T, 6-T SRAM). Фирма Monolithic System Technology (MoSys) объявила о создании нового типа памяти, в которой каждый разряд реализован на одном транзисторе (1-T SRAM). Фактически здесь применяется технология DRAM, поскольку приходится осуществлять периодическую регенерацию памяти. Однако интерфейс с памятью выполнен в стандарте SRAM, при этом циклы регенерации скрыты от контроллера памяти. Схемы 1-T позволяют снизить размер кремниевого кристалла на 50—80 % по сравнению с аналогичными для традиционных SRAM, а потребление электроэнергии — на 75 %.

Табл. 2.3 показывает возможности технологий SRAM при различных значениях быстродействия шины. Цифры в таблице обозначают типичные параметры доступа к памяти, выраженные в терминах количества циклов ожидания при первом и последующих обращениях. Следует отметить, что при увеличении быстродействия шины наиболее экономически эффективной технологией сначала становится асинхронная, затем сквозная синхронная и, наконец, конвейерная синхронная SRAM. Однако, в настоящее время все

Таблица 2 3 **Сравнительные характеристик различных типов SRAM**

Тип памяти	Быстродействие шины, МГц							
	33	50	60	66	75	83	100	125
Асинхронная SRAM	2-1-1-1	3-2-2-2	3-2-2-2	3-2-2-2	3-2-2-2	3-2-2-2	3-2-2-2	3-2-2-2
Синхронная пакетная SRAM	2-1-1-1	2-1-1-1	2-1-1-1	2-1-1-1	3-2-2-2	3-2-2-2	3-2-2-2	3-2-2-2
Конвейерная пакетная SRAM	3-1-1-1	3-1-1-1	3-1-1-1	3-1-1-1	3-1-1-1	3-1-1-1	3-1-1-1	3-1-1-1

меньше поставщиков выпускают модели высокоскоростных сквозных синхронных SRAM. В результате для систем, для которых производительность не особенно важна, на частотах от 50 до 66 МГц конструкторы используют конвейерную синхронную память, поскольку этот вид памяти выпускается многими производителями.

2.5. Интерфейсы

Связь устройств автоматизированных систем друг с другом осуществляется с помощью средств сопряжения, которые называются *интерфейсами.* Интерфейс представляет собой совокупность линий и шин, сигналов, электронных схем и алгоритмов (протоколов), предназначенную для осуществления обмена информацией между устройствами.

Другими словами, интерфейс это не просто набор проводников для связи между устройствами, а целый комплекс технических средств. На практике используются в основном унифицированные интерфейсы — унифицированный по составу и назначению набор линий и шин, унифицированные сигналы и алгоритмы (протоколы) обмена, унифицированные конструктивные характеристики.

Более строгое определение стандартного интерфейса — совокупность унифицированных технических, программных и конструктивных средств, необходимых для реализации взаимодействия различных функциональных элементов в автоматических системах обработки информации при условиях, предписанных стандартом и направленных на обеспечение информационной, электрической и конструктивной совместимости указанных элементов.

Классификация интерфейсов

Выделяют следующие основные классификационные признаки:

- способ соединения компонентов (магистральный, радиальный, цепочный, комбинированный);
- способ передачи информации (параллельный, последовательный, параллельно-последовательный);
- принцип обмена информацией (синхронный, асинхронный);
- режим передачи информации (односторонняя, двухсторонняя, двухсторонняя поочередная).

В соответствии с функциональным назначением интерфейсы можно поделить на следующие основные классы:

- системные интерфейсы ЭВМ;
- интерфейсы периферийного оборудования (общего назначения и специализированные);
- программно-управляемых модульных систем и приборов;
- интерфейсы сетей передачи данных и др.

Рассмотрим подробнее первых два типа интерфейсов.

Современные системы включают два типа шин:

- *системная шина,* соединяющая процессор с ОЗУ и кэш-памятью 2-го уровня;
- множество *шин ввода-вывода,* соединяющие процессор с различными периферийными устройствами. Последние соединяются с системной шиной посредством *моста,* встроенного в набор микросхем (chipset), который поддерживает функционирование процессора (рис. 2.21).

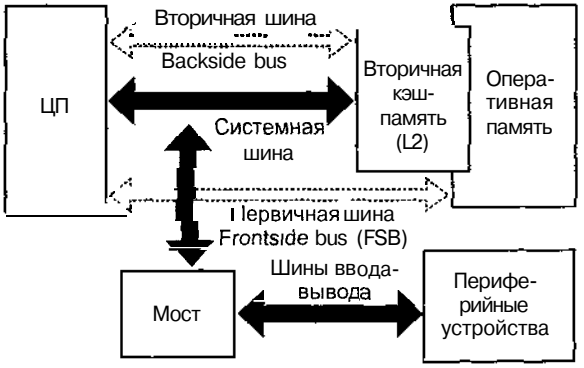


Рис. 2.21. Системные и интерфейсы ввода-вывода

Системная шина при архитектуре DIB (Dual independent bus) физически разделена на две:

- первичная шина (FSB, Frontside bus), связывающая процессор с ОЗУ и ОЗУ с периферийными устройствами;
- вторичная шина (BSB, Backside bus) для связи с кэш-памятью.

Использование двойной независимой шины повышает производительность за счет возможности для процессора параллельно обращаться к различным уровням памяти. Обычно термины «FSB» и «системная шина» используют как синонимы.

Следует отметить, что используемая в настоящее время для описания интерфейсов терминология не является вполне однозначной и ясной. Системная шина часто упоминается как «главная шина», «шина процессора», или «локальная шина». Для шин ввода-вывода используются термины «шина расширения», «внешняя шина», «хост-шина» и опять же — «локальная шина».

Внутренние интерфейсы

Интерфейсы, характеристики которых приводятся в табл. 2.4, а конфигурация разъемов — на рис. 2.22, относятся к внутренним интерфейсам.

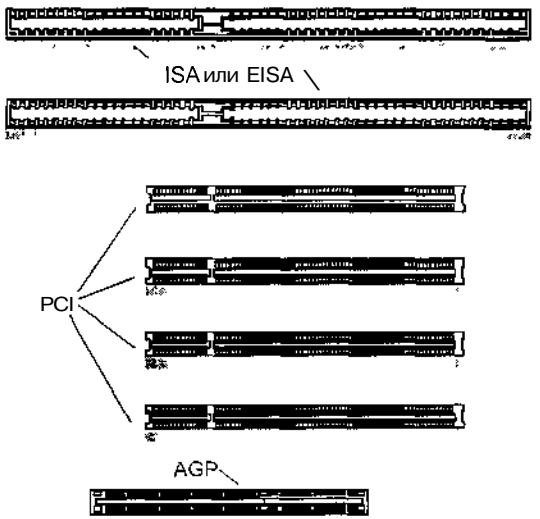


Рис. 2.22. Разъемы шин ISA, PCI, AGP

Таблица 2.4 **Основные характеристики внутренних интерфейсов**

Стандарт	Типичное применение	Пиковая пропускная способность	Примечания
ISA	Звуковые карты, модемы	2 Мбит/с до 8,33 Мбит/с	Практически не используется начиная с 1999 г.
EISA	Сети, адаптеры SCSI	33 Мбит/с	Практически не используется, замещается PCI
PCI	Графические карты, адаптеры SCSI, звуковые карты новых поколений	133 Мбит/с (32-битовая шина с частотой 33 МГц)	Стандарт для периферийных устройств
PCI-X	-,.	1 Гбит/с (64-битовая шина с частотой 133 МГц)	Расширение PCI, предложенное IBM, HP, Compaq Увеличена скорость и количество устройств
PCI Express	-»-	До 16 Гбит/с	Разработка «интерфейса 3-го поколения» (Third generation Input/Output - 3GIO), может заменить AGP. Последовательная шина
AGP	Графические карты	528 Мбит/с, 2х-графика (2х-графические карты)	Стандарт для Intel-PC начиная с Pentium II, сосуществует с PCI
AGP PRO	3D-графика	800 Мбит/с (4х-графика)	Поддерживает видеокарты, потребляющие мощность до 100 Вт (AGP – до 25 Вт)

Шина ISA (Industry Standard Architecture) — стандартные шины XT (8 бит) и AT (16 бит).

Шина XT имеет:

- 8-битовую шину данных;
- 20-битовую шину адреса, что позволяет адресоваться к $2^{20} = 1$ Мбайт памяти;
- 3 канала прямого доступа к памяти (DMA);
- тактовую частоту 8 МГц;
- пропускную способность 4 Мбайт/с;
- 62-контактный разъем.

Из архитектурных особенностей шины укажем, что шина XT поддерживает централизованный метод арбитража, для чего в ней имеются общие линии запроса и ответа. Для обеспечения арбитража всем устройствам присваивается фиксированный уровень приоритета. Шина снабжена пятью линиями запроса на прерывания от раз-

личных устройств ПЭВМ к центральному процессору для привлечения его внимания. Каждая линия имеет фиксированный приоритет. Запуск процедуры прерывания производится по фронту сигнала.

В настоящее время XT практически не применяется.

В компьютерах AT шина расширена до 16 бит данных и 24 бит адреса. В таком виде она существует и поныне как самая распространенная шина для периферийных адаптеров. Шина AT имеет:

- 16-битовую шину данных:
- 24-битовую шину адреса, что позволяет адресоваться к 16 Мбайт памяти:
- 8 каналов прямого доступа (DMA):
- тактовую частоту 8/16 МГц;
- пропускную способность 8 (16) Мбайт/с

Максимальная скорость передачи данных составляет (8 МГц х х 16 бит = 128 Мбит/с, 128 Мбит/с / 2 (передача данных требует от 2 до 8 тактов) = 64 Мбит/с = 8 Мбайт/с).

Все перечисленные ресурсы системной шины должны быть бесконфликтно распределены между абонентами. Это подразумевает, что каждый абонент должен при операциях чтения управлять шиной данных (выдавать информацию) только по своим адресам или по обращению к используемому им каналу DMA. Области адресов для чтения не должны пересекаться. «Подсматривать» не ему адресованные операции записи не возбраняется.

Для шины ISA выпускаются два типа плат расширения — 16- и 8-разрядные платы. Шина ISA до сих пор используется в компьютерах i286, i386 и в части i486, а также в более современных в сочетании с шиной PCI.

Шина EISA (Extended Industry Standard Architecture). Появилась в 1988 г. в качестве альтернативного варианта шины MCA и представляет собой дальнейшее развитие шины ISA. Главное преимущество по сравнению с MCA — совместимость с ISA и соответственно возможность использования многочисленных плат адаптеров, разработанных для ISA.

Основные характеристики:

- 32-разрядная шина данных;
- 32-разрядная шина адреса, что позволяет адресоваться к 4 Гбайт памяти;
- восемь каналов прямого доступа (DMA);
- частота работы 8,33 МГц;
- пропускная способность — 33,3 Мбит/с.

Поддерживает режим автоматического конфигурирования Plug&Play. Шина позволяет однозначно определять устройства, подключенные к каждому из разъемных соединителей расширения. Специальные коды плат расширения считываются ПЭВМ во время ее загрузки. Используется «географический» принцип адресации устройств, когда магистральные линии системной шины дополнены индивидуальными (радиальными) линиями связи от каждого разъема расширения. Режим автоконфигурации устанавливается автоматически для плат EISA во время инициации.

При создании EISA использован ряд технических новшеств, позволивших существенно улучшить характеристики шины, в частности шина поддерживает программно-конфигурируемые адаптеры, позволяющие автоматически разрешать конфликты использования системных ресурсов программным путем.

В шине применен режим Burst Mode — скоростной режим пересылки пакетов данных, что существенно повышает производительность шины. Активное устройство, подсоединенное к шине (bus master), может использовать два режима передачи данных — стандартный и пакетный. Стандартный режим передачи требует, чтобы управляющее и управляемое устройства обменивались соответствующими сигналами управления для записи или чтения 16- или 32-разрядных чисел. В пакетном режиме данный обмен выполняется лишь в начале передачи. Для обеспечения пакетного режима в шине EISA использованы специальные символы управления.

EISA — дорогая, но оправдывающая себя архитектура, применяющаяся в многозадачных системах, на файл-серверах и везде, где требуется высокоэффективное расширение шины ввода-вывода. Перед шиной PCI у нее есть некоторое преимущество в количестве слотов, которое для одной шины PCI не превышает четырех, а у EISA может достигать восьми.

Шина MCA (MicroChannel Architecture). MCA — микроканальная архитектура — была введена в пику конкурентам фирмой IBM для своих компьютеров PS/2 начиная с модели 50. Шина MCA несовместима с ISA/EISA и другими адаптерами.

Состав управляющих сигналов, протокол и архитектура ориентированы на асинхронное функционирование шины и процессора, что снимает проблемы согласования скоростей процессора и периферийных устройств.

Существует два варианта шины — 16- и 32-разрядный (имеются в виду биты данных). В обоих вариантах поддерживается технология Plug&Play (программное конфигурирование).

Основные характеристики:

- для 16-разрядной MCA:
  - 6-разрядная шина данных;
  - 24-разрядная шина адреса, что позволяет адресоваться к 16 Мбайт памяти;
  - частота работы — 10 МГц (возможны и другие частоты — 16 МГц и более);
  - пропускная способность — 16 Мбайт/с;
- для 32-разрядной MCA:
  - 32-разрядная шина данных;
  - 32-разрядная шина адреса, что позволяет адресоваться к 4 Гбайт памяти;
  - частота работы — 10 МГц;
  - пропускная способность — 20 Мбайт/с (в литературе говорится о возможности достижения 40 Мбайт/с и выше.)

Основные преимущества шины MCA заключаются в следующем.

1. Микроканал позволяет эффективно и автоматически конфигурировать все устройства программным путем (в MCA PS/2 нет ни одного переключателя).

2. Шина MCA дает возможность реализовать многопользовательский и многозадачный режимы работы микропроцессора. Этому способствует схема арбитража CACP (Central Arbitration Control Point), с помощью которой различные устройства получают доступ к системной шине данных; если два и более устройства одновременно пытаются получить управление MCA, аппаратные средства MCA разрешают этот конфликт. Схема преодоления конфликтов децентрализована, с защитой от монополизации. Аппаратные средства дают возможность поддерживать до 16 «интеллектуальных» устройств, взаимодействующих по шине без всяких помех: до 8 микропроцессоров и 8 других устройств, например контроллеров ПДП.

3. Повышенная по сравнению с шиной AT частота и разрядность обеспечивают сравнительно высокую пропускную способность (до 40 Мбайт/с).

4. Такие секции шины, как видеорасширения отображаемой памяти, а также аудиоканал, позволяют адаптерным платам соответственно: получать доступ к контроллеру VGA на системной плате; реализовывать специальный режим передачи данных Matched Memory; обмениваться аудиосигналами с системной платой.

5. Использование новых технологий монтажа и формата адаптерных плат, повышенная плотность размещения выводов микросхем и разъемов, существенно уменьшающих генерацию ПЭВМ

электромагнитного излучения. Снижению помех прежде всего способствует размещение выводов в разъемах расширения и адаптерных платах: через каждые четыре вывода имеется вывод «земля» с обеих сторон разъема. Размещение выводов сдвинуто на противоположных сторонах разъема на два вывода, в результате чего каждый сигнальный вывод становится смежным с «землей». Более короткие проводники также обеспечивают снижение генерации помех.

При всей прогрессивности архитектуры (относительно ISA) шина MCA не пользуется популярностью из-за узости круга производителей MCA-устройств и полной их несовместимости с массовыми ISA-системами. Однако MCA еще находит применение в мощных файл-серверах, где требуется обеспечение высоконадежного производительного ввода-вывода.

Локальные шины VLB и PCI. Попытки улучшить системные шины за счет создания шин MCA и EISA имели ограниченный успех и не решали кардинальным образом проблемы, связанной с необходимостью быстрой «прокачки» по шине больших объемов информации, что особенно важно для высококачественных графических задач.

При появлении новых высокопроизводительных процессоров сдерживание производительности всей системы из-за недостаточной пропускной способности шины стало особенно очевидным.

Для решения проблемы были разработаны так называемые локальные шины (рис. 2.23), при этом исторически первой появилась шина VLB (VESA Local Bus), а приблизительно год спустя — шина PCI.

Основные предпосылки, заложенные при создании локальной шины сводятся к следующему. Локальные шины обеспечивают подключение периферии к скоростной шине собственно процессора (аналогично внешнему кэшу). При этом шина работает с частотой,

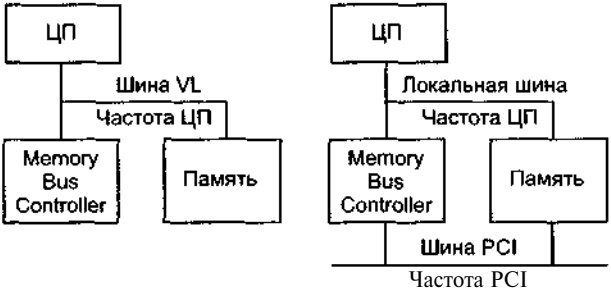


Рис. 2.23. Локальные шины

соответствующей тактовой частоте процессора. Передачей данных управляет не центральный процессор, а плата расширения (мост), который высвобождает микропроцессор для выполнения других работ. Локальная шина обслуживает наиболее быстрые устройства: память, дисплей, дисковые накопители, при этом обслуживание сравнительно медленных устройств — мышь, модем, принтер и др. — производится системной шиной типа ISA (EISA).

Локальная шина VL-Bus (VESA). Исторически появилась первой и была создана специально для лучшего микропроцессора того времени 480DX/2. В зависимости от используемого центрального процессора тактовая частота шины может составлять от 20 до 66 МГц.

Стандарт шины VL 1.0 поддерживает 32-разрядный тракт данных, но его могут использовать и 16-разрядные устройства. Стандарт 2.0 рассчитан на 64-битовую шину в соответствии с новыми процессорами. Спецификация 1.0 ограничена частотой 40 МГц, а 2.0 — 50 МГц. В спецификации 2.0 шина поддерживает до 10 устройств, 1.0 — только три. Устойчивая скорость передачи составляет до 106 Мбайт/с (для 64-разрядной шины — до 260 Мбайт/с). Несмотря на то, что шина была оптимизирована на процессоры семейства 86, она может работать и с другими процессорами, что делает ее потенциальным кандидатом для разработки смешанных платформ.

Интересная возможность шины заключается в том, что 64-битовое устройство в 32-битовом слоте может работать как 32-битовое, и наоборот, 32-битовое устройство может работать в 64-битовом слоте как 32-битовое.

Спецификация определяет два вида устройства VL-шины — целевые устройства (Local Bus Target — LBT) и активные устройства шины (Local Bus master — LBM). Активное устройство может инициировать передачу данных по шине и иметь собственный процессор. С другой стороны, целевые устройства отвечают на запросы, инициированные активными устройствами шины. Слоты шины VL располагаются в ряд с имеющейся шиной ISA, EISA или MCA.

Шину VLB обычно использовали для подключения графического адаптера и контроллера дисков. Адаптеры локальных сетей для VLB практически не встречаются. Иногда встречаются системные платы, в описании которых указано, что они имеют встроенный графический и дисковый адаптер с шиной VLB, но самих слотов VLB нет. Это означает, что на плате установлены микросхемы указанных адаптеров, предназначенные для подключения к шине VLB. Такая *неявная шина* по производительности естественно не уступает шине с явными слотами. С точки зрения надежности и совместимо-

сти это даже лучше, поскольку проблемы совместимости карт и системных плат для шины VLB стоят особенно остро.

Локальная шина PCI. PCI (Peripheral Component Interconnect bus) — шина соединения периферийных компонентов. Эта шина занимает особое место в современной архитектуре ПК (mezzanine bus), являясь мостом между локальной шиной процессора и шиной ввода-вывода ISA/EISA или MCA. Эта шина разрабатывалась в расчете на Pentium-системы, но хорошо сочетается и с 486 процессорами, а также с не-Intel'овскими процессорами. Шина PCI является четко стандартизованной высокопроизводительной шиной расширения ввода-вывода. Частота шины 20—33 МГц, PCI 2.1 допускает частоту 66 МГц. Теоретическая максимальная скорость — 132/264 Мбайт/с для 32/64 бит при 33 МГц. Слот PCI достаточен для подключения адаптера (в отличие от VLB), на системной плате он может сосуществовать с любой из шин ввода-вывода и даже с VLB (хотя в этом и нет необходимости). Шина PCI является самой высокоскоростной шиной расширения современных ПК (не считая специализированного порта AGP).

На одной шине PCI может быть не более четырех устройств (слотов). Мост шины PCI (PCI Bridge) — это аппаратные средства подключения шины PCI к другим шинам. Host Bridge — главный мост — используется для подключения PCI к системной шине (шине процессора или процессоров). Peer-to-Peer Bridge (одноранговый мост) используется для соединения двух шин PCI. Две и более шины PCI применяются в мощных серверных платформах, дополнительные шины PCI позволяют увеличить количество подключаемых устройств.

Автоконфигурирование устройств (выбор адресов, запросов прерывания) поддерживается средствами BIOS и ориентировано на технологию Plug and Play. Стандарт PCI определяет для каждого слота конфигурационное пространство размером до 256 восьмибитных регистров, не приписанных ни к пространству памяти, ни к пространству ввода-вывода. Доступ к ним осуществляется по специальным циклам шины Configuration Read и Configuration Write, вырабатываемым контроллером при обращении процессора к регистрам контроллера шины PCI, расположенным в его пространстве ввода-вывода.

В состав команд шины PCI введены сигналы для тестирования адаптеров по интерфейсу JTAG. На системной плате эти сигналы не всегда задействованы, но могут и организовывать логическую цепочку тестируемых адаптеров.

Шина PCI все обмены трактует как пакетные: каждый кадр начинается фазой адреса, за которой может следовать одна или несколько фаз данных. Количество фаз данных в пакете не определено, но ограничено таймером, определяющим максимальное время, в течение которого устройство может пользоваться шиной. Каждое устройство имеет собственный таймер, параметры которого задаются при конфигурировании устройств шины.

В каждом обмене участвуют два устройства — инициатор обмена (Initiator) и целевое устройство (Target). Арбитражем запросов на использование шины занимается специальный функциональный узел, входящий в состав чипсета системной платы. Для согласования быстродействия устройств-участников обмена предусмотрены два сигнала готовности IRDY и TRDY. Для адреса и данных на шине используются общие мультиплексированные линии AD. Четыре мультиплексированные линии C/BE[3:0] используются для кодирования команд в фазе адреса и разрешения байт в фазе данных.

Известны также более поздние разновидности — PCI-X и PCI-Express, кроме того, к данному типу относится и PCMCIA — стандарт на шину блокнотных компьютеров. Она позволяет подключать расширители памяти, модемы, контроллеры дисков и стримеров, SCSI-адаптеры, сетевые адаптеры и др.

Базовый вариант адресует до 64 Мбайт памяти, разрядность 16 бит, частота до 33 МГц. Прямой доступ DMA не поддерживается, конфигурирование адаптеров — только программное.

Различают четыре типа PCMCIA. Электрически идентичные, они отличаются по габаритам и совместимы снизу вверх (меньшие адаптеры вставляются в большие гнезда). Толщина адаптера типа 1 не более 3,3 мм, типа 2 — 5 мм, типа 3 — 10,5 мм (НЖМД типа 3 —имеет толщину 13 мм), типа 4 — больше (нет стандарта).

Все PCMCIA-устройства имеют минимальное энергопотребление. Ведется работа по улучшению и расширению упомянутого базового стандарта, так в литературе говорится о 32-разрядной реализации, в которой производится мультиплексирование имеющихся 16 аппаратных линий данных.

AGP (Accelerated graphics port). Несмотря на разрядность и скорость шины PCI, оставалась проблема, которая превышала ее возможности, — передачу графической информации. Повышение разрешающей способности и цветности мониторов привело к необходимости повышения скорости передачи до 250 Мбит/с, в то время как пиковая пропускная способность PCI составляла до 132 Мбит/с.

Фирмой Intel было предложено решение в виде AGP — Accelerated graphics port (порт ускоренного графического вывода).

Схемы AGP взаимодействуют непосредственно с четырьмя источниками информации (Quadro port acceleration, рис 2 24)

- процессор (кэш-память 2-го уровня),
- оперативная память,
- графическая карта AGP,
- шина PCI

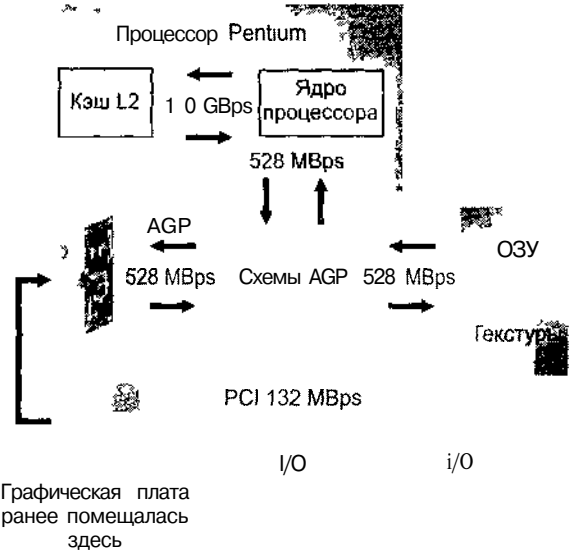


Рис. 2.24. Схема взаимодействия элементов с использованием AGP

AGP функционирует на скорости процессорной шины (FSB) При тактовой частоте 66 МГц, например, это в 2 раза выше, чем скорость PCI, и позволяет достичь пиковой пропускной способности в 264 Мбит/с В графических картах, специально спроектированных для AGP, передача происходит как по переднему, так и по заднему фронту тактовых импульсов ЦП, что позволяет при частоте 133 МГц осуществлять передачу со скоростью до 528 Мбит/с (это называется «2-х графика») В дальнейшем была выпущена версия AGP 2 0, которая поддерживала «4-х графику», или четырехкратную передачу данных за один такт ЦП

Интерфейсы периферийных устройств. Рассмотрим наиболее распространенные интерфейсы периферийных устройств

IDE (Integrated Device Electronics) — интерфейс устройств со встроенным контроллером (рис 2 25) При создании

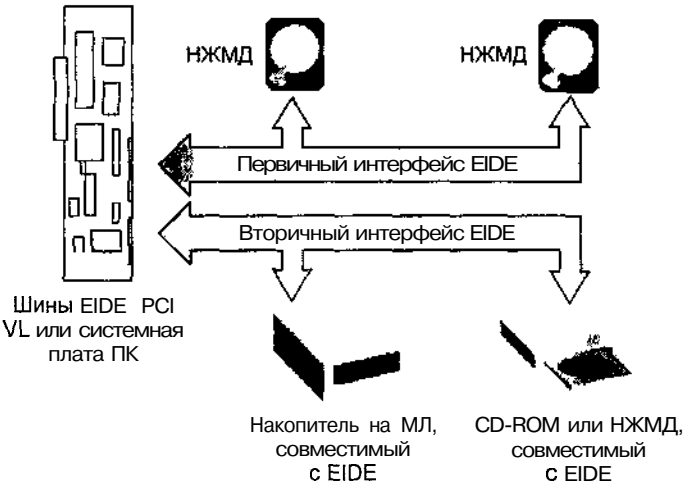


Рис. 2 25. Интерфейс EIDE

этого интерфейса разработчики ориентировались на подключение дискового накопителя За счет минимального удаления контроллера от диска существенно повышается быстродействие Сигналы интерфейса — сокращенный набор буферизированных сигналов

Проблема накопитель-компьютер состоит из трех частей Компьютер должен взаимодействовать с контроллером (и наоборот), контроллер должен оперировать данными, и контроллер должен взаимодействовать с дисковым накопителем (и наоборот)

До недавнего времени проблема рассматривалась со всех трех сторон, что заставляло производителей накопителей выполнять всю работу Большая часть «интеллекта» для управления передачей данных между компьютером и дисковым накопителем была сосредоточена на плате контроллера и компьютера

В результате при установке нового или замене старого накопителя требовалось обеспечить полную совместимость контроллера с новым жестким диском Контроллеры IDE существенно изменили ситуацию, так как в этом стандарте значительно большую роль стал играть контроллер на плате дисковода, поэтому фактический интерфейс между накопителем и компьютером стал относительно простым Благодаря этому упрощается электроника внутри компьютера и предоставляется большая гибкость в проектировании электроники для дисковых накопителей

Интерфейс имеет скорость передачи порядка 1,5—3 Мбайт/с Имеются определенные проблемы при работе с дисками емкости

Схемы AGP взаимодействуют непосредственно с четырьмя источниками информации (Quadro port acceleration, рис 2 24)

- процессор (кэш-память 2-го уровня),
- оперативная память,
- графическая карта AGP,
- шина PCI

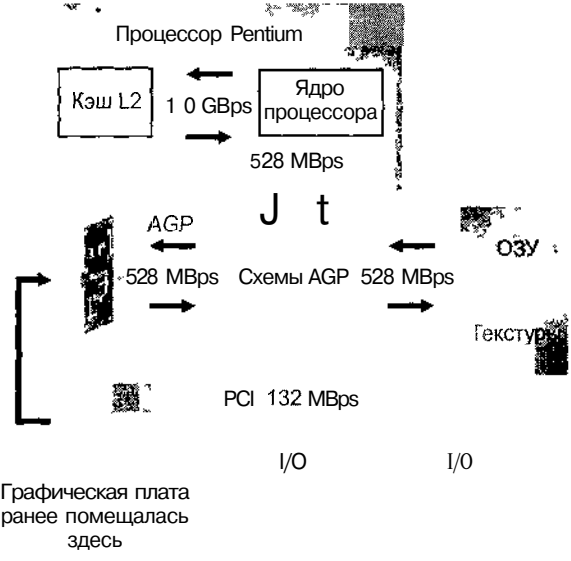


Рис. 2.24. Схема взаимодействия элементов с использованием AGP

AGP функционирует на скорости процессорной шины (FSB) При тактовой частоте 66 МГц, например, это в 2 раза выше, чем скорость PCI, и позволяет достичь пиковой пропускной способности в 264 Мбит/с В графических картах, специально спроектированных для AGP, передача происходит как по переднему, так и по заднему фронту тактовых импульсов ЦП, что позволяет при частоте 133 МГц осуществлять передачу со скоростью до 528 Мбит/с (это называется «2-х графика») В дальнейшем была выпущена версия AGP 2 0, которая поддерживала «4-х графику», или четырехкратную передачу данных за один такт ЦП

Интерфейсы периферийных устройств. Рассмотрим наиболее распространенные интерфейсы периферийных устройств

IDE (Integrated Device Electronics) — интерфейс устройств со встроенным контроллером (рис 2 25) При создании

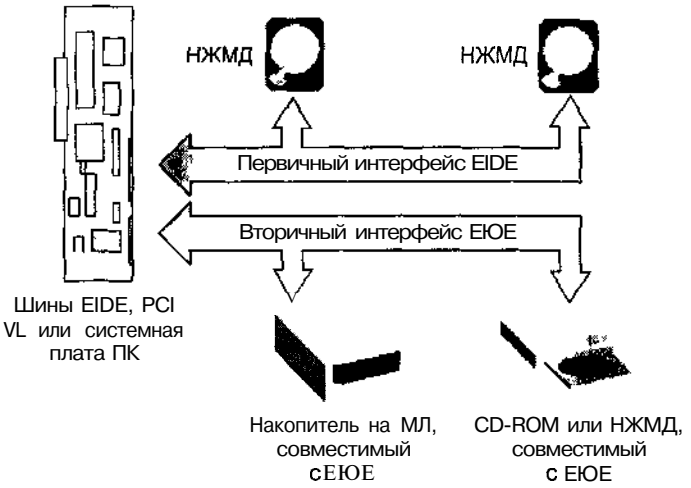


Рис. 2.25. Интерфейс EIDE

этого интерфейса разработчики ориентировались на подключение дискового накопителя За счет минимального удаления контроллера от диска существенно повышается быстродействие Сигналы интерфейса — сокращенный набор буферизированных сигналов

Проблема накопитель-компьютер состоит из трех частей Компьютер должен взаимодействовать с контроллером (и наоборот), контроллер должен оперировать данными, и контроллер должен взаимодействовать с дисковым накопителем (и наоборот)

До недавнего времени проблема рассматривалась со всех трех сторон, что заставляло производителей накопителей выполнять всю работу Большая часть «интеллекта» для управления передачей данных между компьютером и дисковым накопителем была сосредоточена на плате контроллера и компьютера

В результате при установке нового или замене старого накопителя требовалось обеспечить полную совместимость контроллера с новым жестким диском Контроллеры IDE существенно изменили ситуацию, так как в этом стандарте значительно большую роль стал играть контроллер на плате дисковода, поэтому фактический интерфейс между накопителем и компьютером стал относительно простым Благодаря этому упрощается электроника внутри компьютера и предоставляется большая гибкость в проектировании электроники для дисковых накопителей

Интерфейс имеет скорость передачи порядка 1,5—3 Мбайт/с Имеются определенные проблемы при работе с дисками емкости

более 528 Мбайт, которые решаются с помощью специальных драйверов

Наиболее распространенной модификаций интерфейса IDE является AT Attachment (ATA) IDE — 40-проводный интерфейс (рис 2 26)

а б

Рис. 2.26. Параллельный разъем (шлейф) ATA/IDE (а), последовательный разъем ATA *(б)*

IDE-адаптер часто встраивается в системную плату К настоящему времени разработаны усовершенствованные адаптеры с улучшенными характеристиками

ATAPI (Package Interface), ATA-2 — расширение, позволяющее подключать до четырех устройств, включая CD-ROM и стример

EIDE (Enhanced IDE) — расширенный адаптер для шин PCI и VLB, позволяющий подключать до четырех устройств (рис 2 25), в том числе CD-ROM и стримеры с максимальной скоростью 11,1 Мбайт/с, объем диска до 8 Гбайт

FAST ATA и FAST-ATA-2 — скоростные интерфейсы, обеспечивающие скорость передачи соответственного 13,3 и 16,6 Мбайт/с и доступ к 8-Гбайтному диску (табл 2 5)

Таблица 2 5 **Характеристики IDE/ATA интерфейсов**

Спецификация	ATA	ATA-2	ATA 3	ATA 4	ATA-5	ATA 6
Максимальная пропускная способность Мбайт/с	4	16	16	33	66	100
Количество соединений	2	2	2	2 на 1 кабель	2 на 1 кабель	2 на 1 кабель
Характеристики кабеля	40 контактов	40 контактов	40 контактов	40 контактов	40 контактов, 80 жильный	40 контактов 80 жильный
Контроль по CRC	Нет	Нет	Нет	Есть	Есть	Есть
Дата выпуска г	1981	1994	1996	1997	1999	2000

SCSI (Small Computer System Interface), произносится «скази» — интерфейс системного уровня, стандартизованный ANSI, в отличие от интерфейсных портов (COM, LPT, IR, MIDI) представляет собой шину сигнальные выводы множества устройств-абонентов соединяются друг с другом

Интерфейс позволяет подключать до семи устройств (рис 2 27) с контроллерами дисковые накопители, CD-ROM и т п Любое устройство может инициировать обмен с другим целевым (Target) устройством Режим обмена может быть асинхронным или более производительным синхронным с согласованием скорости (Synchronous Negotiation), данные контролируются по паритету

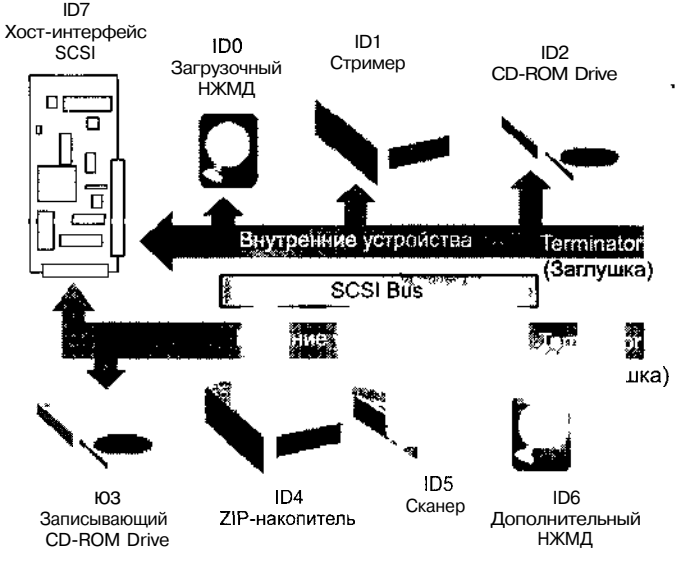


Рис. 2.27. Архитектура использования интерфейса SCSI

Основным назначением SCSI-шины, как это было сформулировано в первой спецификации, разработанной в 1985 г , было «обеспечение аппаратной независимости подключаемых к компьютеру устройств определенного класса»

В отличие от жестких шин расширения SCSI-шина реализуется в виде отдельного кабельного шлейфа, который допускает соединение до восьми устройств (спецификация SCSI-1) внутреннего и внешнего исполнения Одно из них — хост-адаптер (host adapter), связывает шину SCSI с системной шиной компьютера, семь других свободны для периферии

К шине могут подключаться:

- дисковые внутренние и внешние накопители (CD-ROM, винчестеры, сменные винчестеры, магнитооптические диски и др.);
- стримеры;
- сканеры;
- фото- и видеокамеры;
- другое оборудование, применяемое не только для IBM PC.

Каждое устройство, подключенное к шине, имеет свой идентификатор (SCSI ID), который передается позиционным кодом по 8-битной шине данных (отсюда и ограничение на количество устройств на шине). Устройство (ID) может иметь до восьми подустройств со своими LUN (Logical Unit Number — логический номер устройства).

Любое устройство может инициировать обмен с другим целевым устройством (Target).

Перечислим основные модификации интерфейса (табл. 2.6):

- SCSI-1 — строго определяет физические и электрические параметры интерфейса и минимум команд;

Таблица 2.6. **Модификации интерфейса SCSI**

Версия	Максимальная скорость, Мбайт/с	Ширина шины, бит	Максимальная длина, м			Максимальное количество устройств
			Single-ended	LVD	HVD	
SCSI-1	5	8 (узкая)	6	–	25	8
Fast SCSI	10	8	3	–	25	8
Fast Wide SCSI	20	16 (широкая)	3	–	25	16
Ultra SCSI	20	8	1,5	–	25	8
Ultra SCSI	20	8	3	–	–	4
Wide Ultra SCSI	40	16	–	–	25	16
Wide Ultra SCSI	40	16	1,5	–	–	8
Wide Ultra SCSI	40	16	3	–	–	4
Ultra 2 SCSI	40	8	Не определено для скоростей выше Ultra	12	25	8
Wide Ultra 2 SCSI	80	16	–	12	25	16
Ultra 3 SCSI или Ultra160 SCSI	160	16	–	12	Не определено для скоростей выше Ultra 2	16
Ultra 320 SCSI	320	16	–	12	–	16

- SCSI-2 — определяет 18 базовых SCSI команд, обязательных для всех периферийных устройств, и дополнительные команды для CD-ROM и другой периферии. Устройства поддерживают очереди до 256 команд и выполняют их в предварительно оптимизированном порядке автономно. Устройства на одной SCSI-шине могут обмениваться данными без участия процессора.

Дополнительные расширения SCSI-2:

- Fast SCSI-2 — удвоение скорости синхронной передачи;
- Wide SCSI-2 — 16- и 32-битные расширения SCSI-2;
- Ultra SCSI — сверхскоростной интерфейс.

SCSI-3 — дальнейшее развитие: 32 устройства, дополнительные команды, автоконфигурирование (Plug and Play) и т. п. Физический интерфейс допускает более длинные кабели. Возможно применение оптических кабелей.

Ультрапорт. Альтернативой шинным архитектурам является коммутационное устройство *архитектуры ультрапорта UPA,* предложенной корпорацией SUN Microsystems. Эта архитектура используется для тех же целей, что и шина. Однако, если через последнюю одновременно могут взаимодействовать только два объекта, UPA определяет одновременную передачу друг другу данных между несколькими парами объектов. Например (рис. 2.28), один из процессоров может взаимодействовать с оперативной памятью (пунктирная линия), а другой в то же время — с принтером (штрих-пунктирная линия). UPA обеспечивает пропускную способность, равную 1,3 Гбайт/с.

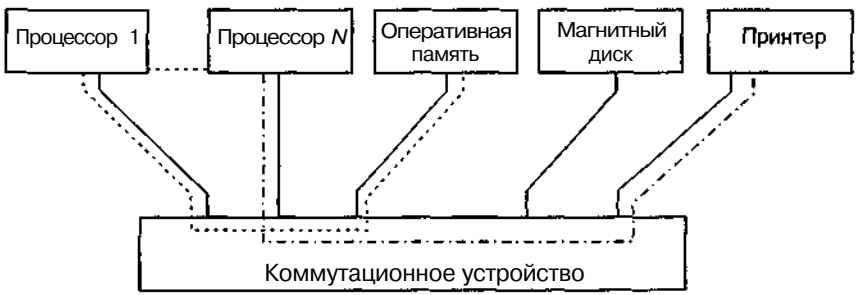


Рис. 2.28. Архитектура ультрапорта

Внешние интерфейсы

Принтеры, модемы и другое периферийное оборудование подключаются к компьютеру через стандартизированные интерфейсы, иногда называемые *портами.* В зависимости от способа передачи

информации (параллельного или последовательного) между сопрягаемыми устройствами различают параллельные и последовательные интерфейсы.

Распространенными являются интерфейс Centronics (аналог в СНГ ИРПР-М ГОСТ 27942—88), обеспечивающий подключение широкого круга устройств с параллельной передачей информации, и RS-232S (аналог СНГ «Стык-2», ГОСТ 18125-81, ГОСТ 23675-79), наиболее широко применяемый для синхронной и асинхронной связи при соединении устройств с последовательной передачей информации. Данные традиционные интерфейсы в настоящее время вытесняются более быстродействующими — USB и Fire Wire (табл. 2.7).

Таблица 2.7. **Характеристики основных внешних интерфейсов**

Стандарт	Год выпуска	Первоначальная скорость, Мбит/с
Последовательный порт (RS 232)	1960	0,02
Параллельный порт (Lpt)	1981	1,1
USB	1995	12
FireWire	1995	400
USB 20	2000	480
FireWire 800	2001	850

Последовательный порт стандарта RS-232-C. Интерфейс RS-232-C разработан EIA (Electronic Industries Association — Ассоциация производителей электроники) и является стандартом для соединения ЭВМ с различными последовательными внешними устройствами, в качестве которых первоначально выступали в основном терминалы и печатающие устройства.

IBM-PC совместимый ПК поддерживает интерфейс RS-232-C не в полном объеме (разъем, являющийся последовательным портом передает/принимает некоторые из сигналов, входящих в состав RS-232-C и имеющих соответствующие этому стандарту уровни напряжения). В операционных системах компьютеров IBM PC каждому порту RS-232-C присваивается логическое имя COM1:—COM4:.

Последовательная передача данных состоит в побитовой передаче каждого байта цифровой информации, в форме *кадра данных,* содержащего сигнал начала передачи (Start), сигнал окончания передачи (Stop) и информационные биты (рис. 2.29).

Бит ST сигнализирует о начале передачи данных, затем передаются информационные биты — вначале младшие, потом старшие

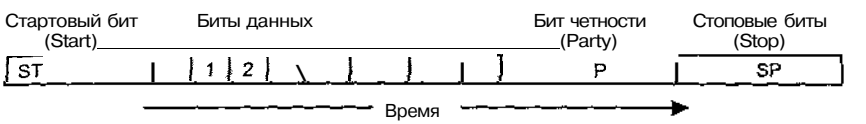


Рис. 2.29. Структура кадра данных при передаче байта информации в стандарте RS-232-C

Иногда используется контрольный бит P, которому присваивается такое значение, чтобы общее число единиц или нулей было четным или нечетным. Это используется для контроля правильности передачи кадра. Приемное устройство проверяет кадр на четность и при несовпадении с ожидаемым значением передает запрос о повторе передачи кадра. Бит (или биты) SP сигнализирует об окончании передачи байта.

Использование (или нет) битов P, ST, SP задает *формат передачи данных (кадра)* на уровне RS-232. Принимающее и передающее устройства должны применять одинаковые форматы.

Основу последовательного порта составляет микросхема UART (Universal Asyncronous Receiver-Transmitter — универсальный асинхронный приемопередатчик — Intel 16450/16550/16550A).

UART содержит регистры (буферные, сдвиговые и др.) приемника и передатчика данных. При передаче байта информации он вначале записывается в буферный регистр передатчика, затем в сдвиговый регистр, откуда выдвигается по битам для последовательной передачи по линии связи. Обратный процесс происходит при приеме данных.

Разъем для подключения последовательного порта может содержать 25 или 9 выводов (соответственные обозначения — *D25* и *D9*) (табл. 2.8). Только два провода этих разъемов используются для передачи и приема данных. Остальные отведены для вспомогательных и управляющих сигналов, причем для соединения различных типов устройств может потребоваться различное количество выводов разъемов.

Стандарт RS-232-C определяет взаимодействие между устройствами двух типов:

* DTE (Data terminal equipment — оконечное/терминальное устройство);
* DCE (Data communication equipment — устройство связи).

В большинстве случаев компьютер, терминал являются DTE, модемы, принтеры, графопостроители — DCE.

Если опустить ненужные подробности, то можно сказать, что для связи DTE—DCE (например, компьютер — внешний модем)

Таблица 2.8 Структура разъемов интерфейса RS-232-C

Номер контакта D25	Принятое сокращение по RS-232	Номер контакта D9	Содержание информации	Вход или выход
1	AA	–	FG (Frame ground - защитное (силовое) заземление)	–
2	BA	3	TD (Transmitted data - передаваемые данные)	Выход
3	BB	2	RD (Received data - принимаемые данные)	Вход
4	CA	7	RTS (Request to send - запрос для передачи)	Выход
5	CB	8	CTS (Clear to send - сброс для передачи)	Вход
6	CC	6	DSR (Data set ready - готовность данных)	Вход
7	AB	5	SG (Signal ground - сигнальное заземление)	–
8	CF	1	DCD (Data earner detect - обнаружение передачи данных)	Вход
20	CD	4	DTR (Data terminal ready - данные готовы к передаче)	Выход
22	CE	9	RI (Ring indicator - индикатор вызова)	Вход

Примечания.

\. Контакты 9—19, 21, 23—25 не используются.

2. Уровни напряжения на контактах составляют:

для логического нуля –15...–13 В;

для логической единицы +3...+15 В (промежуток –3...+3 В соответствует отсутствию сигнала).

необходимо в разъемах осуществить соединение проводов по принципу «вход—вход» и «выход—выход» (рис. 2.30, *а),* для связи же DTE—DTE (например, *компьютер—компьютер)* принцип соедине-

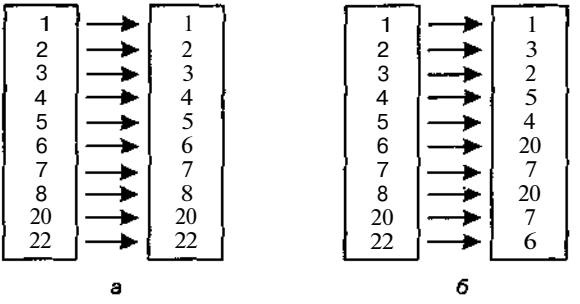


Рис. 2.30. Схемы связи проводов в кабелях для соединения DTE—DCE *(а)* и DTE-DTE *(б)*

ния другой — «выход—вход» и «вход—выход» (рис. 2.30, б). Структура, приведенная на рис. 2.30, *б,* используется для соединения двух ПК и в обиходе получила название «нуль-модем».

Параллельный порт. Параллельный порт (Centronics) используется для одновременной передачи 8 битов информации. В компьютерах этот порт используется главным образом для подключения принтера, хотя это не исключает возможности подсоединения к нему других устройств, например графопостроителей или других ПК. Конструктивно он обычно оформлен в виде 25-контактного разъема типа D (DB25). Параллельные порты компьютера обозначаются LPT1—LPT4.

Имеется восемь шин (линий) данных, восемь шин заземления (для каждой шины данных), кроме того, имеются следующие управляющие сигналы:

- сигнал STROBE (строб) на контакте *1* сообщает принтеру, что текущая передача данных окончена и принтер может печатать символ;
- *линия* ACK (подтверждение готовности) на контакте *10.* До тех пор, пока на этой линии высокий потенциал, компьютер не посылает данных;
- линия BUSY (занятость) сигнализирует компьютеру о том, что принтер занят;
- линия SELECT (выбор) показывает, что принтер выбран (т. е. режим online);
- линия FDXT — автоматический перевод строки;
- линия Error (ошибки) — принтер сообщает об ошибке (например, кончилась бумага);
- линия Ink — компьютер переводит принтер в то состояние, в котором он находился после включения питания (т. е. начальное состояние);
- линия Slctin — по этой линии компьютеру сообщается, готов ли принтер принимать данные — при низком уровне сигнала принтер готов принимать данные, при высоком — нет.

Хотя чаще всего параллельный порт применялся для передачи данных из компьютера в принтер, его можно использовать и для приема данных от внешнего устройства. Сейчас имеются программно-аппаратные средства, которые позволяют осуществить ввод.

Параллельное соединение применяется на расстояниях не более 5 м, некоторые источники ограничивают расстояние *1—2* м; при увеличении длины параллельных проводов возрастает межпроводная емкость, что приводит к перекрестным помехам, кроме того, растут материальные затраты на реализацию линии.

В принципе, современные параллельные порты должны быть двунаправленными и соответствовать требованиям стандарта EPP, поскольку EPP позволяет передавать данные в 10 раз быстрее, чем стандартные параллельные порты (2 Мбит/с против 200 Кбит/с).

USB (Universal system bus) — стандарт, разработанный совместно фирмами Compaq, DEC, Microsoft, IBM, Intel, NEC и Northern Telecom, предназначен для организации соединения многочисленных и разнотипных внешних устройств с помощью единого интерфейса (рис. 2.31).

Стандарт USB позволяет подсоединить до 127 устройств последовательно или использовать концентратор USB *(hub),* к которому подсоединяются семь устройств. Разъемы содержат четыре контак-

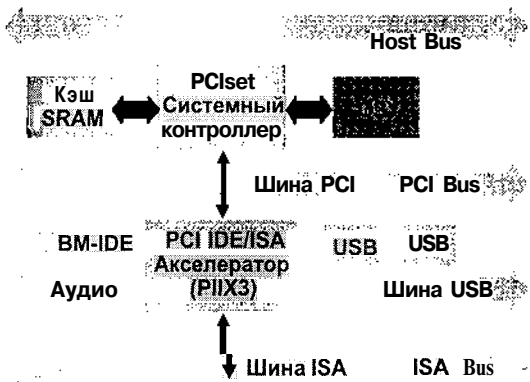


Рис. 2.31. Архитектура взаимодействия элементов с участием USB

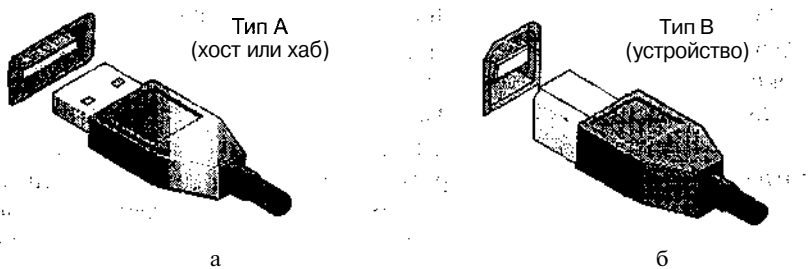


Рис. 2.32. Разъемы интерфейса USB типа А (а) и В (б)

та, включая провода питания (5 В) для таких небольших устройств, как ручной сканер или звуковая колонка (рис. 2.32).

Шина USB позволяет многоуровневое каскадирование — архитектурную особенность шины USB, ее логическая топология — многоуровневая звезда.

Переломным моментом в истории USB стал выход Windows 98, когда появилась поддержка, начался выпуск устройств, и технология начала свое существование не только у разработчиков, но и у пользователей, кстати, многие из которых относят ее появление именно к этому моменту.

Сегодня USB — это очень популярная универсальная последовательная шина. Предназначена для легкого подключения различного вида устройств — клавиатуры, мыши, джойстики, колонки, модемы, мобильные телефоны, ленточные, дисковые, оптические и магнитооптические накопители, флэш-диски, сканеры и принтеры, дигитайзеры, — словом все, что подключается к ПК. Также с ожиданием большого роста в области интеграции компьютеров и телефонии шина USB может выступать в качестве интерфейса для подключения устройств цифровой сети с интегрированными услугами (ISDN) и цифровых устройств Private Branch eXchange (PBX).

Пропускной способности в 480 Мбит/с (версия 2.0) достаточно для удовлетворения потребностей всех этих применений в полной мере. Добавление устройств больше не сопряжено с установкой дополнительных адаптеров, выполнением сложного конфигурирования, ручным инсталлированием дополнительного программного обеспечения: система автоматически определяет, какой ресурс, включая программный драйвер и пропускную способность, нужен каждому периферийному устройству и делает этот ресурс доступным без вмешательства пользователя. Популярная периферия сегодня доступна в вариантах с USB гораздо чаще, чем с другими.

IEEE 1394 (FireWire). История IEEE 1394, теперь известного также как FireWire или i-Link, началась в 1986 г., когда члены Microcomputer Standards Committee (Комитет Стандартов для Микрокомпьютеров) захотели объединить существовавшие в то время различные варианты последовательной шины (Serial Bus).

Задачей разработчиков стало создание универсального внешнего интерфейса ввода-вывода, пригодного как для работы с мультимедиа, так и с накопителями данных (Mass Storage Device), не говоря уже о таких устройствах, как принтеры, сканеры, и т. п. Результатом труда разработчиков стал окончательно утвержденный 12 декабря 1995 г. документ, который описывал IEEE 1394.

Ведущую роль в разработке стандарта играла Apple, которая дала ему имя FireWire, и сразу же сделала ставку на использование этого стандарта в своих компьютерах.

При разработке любительских цифровых видеокамер (DV) стало ясно, что наиболее подходящим внешним интерфейсом для них является IEEE 1394. Поэтому, Digital VCR Conference (DVC) приняла решение использовать IEEE 1394 как стандартный интерфейс для цифровых камер.

Первой стала Sony с цифровыми камерами DCR-VX1000 и DCR-VX700, которые впервые имели выход IEEE 1394. Сегодня любая DV-камера оснащается интерфейсом IEEE 1394. Компания Texas Instruments, организовав массовое производство дешевых микросхем для реализации интерфейса IEEE 1394, сыграла огромную роль в росте количества контроллеров IEEE 1394 в персональных компьютерах.

Из главных особенностей IEEE 1394 можно отметить:

- последовательная шина вместо параллельного интерфейса позволила использовать кабели малого диаметра и разъемы малого размера (рис. 2.33);
- поддержка «горячего подключения» и отключения;
- питание внешних устройств через IEEE 1394 кабель;
- высокая скорость;
- возможность строить сети из различных устройств и самой различной конфигурации (рис. 2.34);
- простота конфигурации и широта возможностей;
- поддержка асинхронной и синхронной передачи данных.

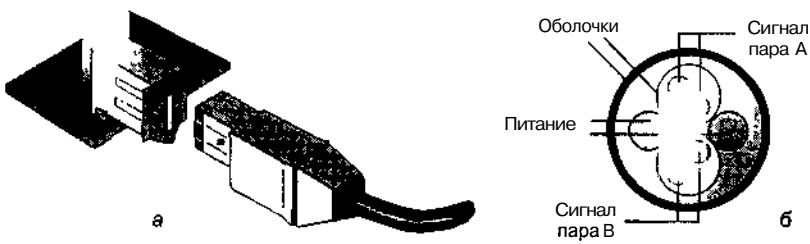


Рис. 2.33. Разъем *(а)* и кабель *(б)* интерфейса IEEE 1394.
Примечание: конструкция разъема аналогична игровому порту приставки Nintendo Gameboy

Интерфейс во многом подобен USB 1.0, но является более быстродействующим. В различных спецификациях устанавливается быстродействие от 12,5 Мбит/с до 1,6 Гбит/с и выше.

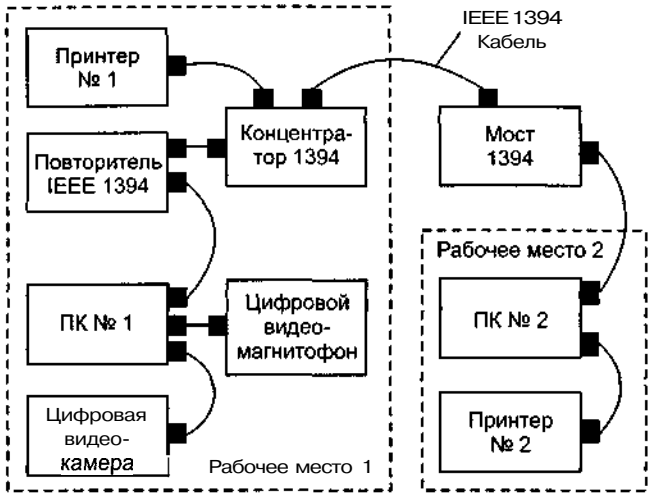


Рис. 2.34. Соединение разнотипных устройств в сеть с использованием IEEE 1394

Это создает возможность для соединения интерфейсом FireWire с ЭВМ таких устройств, как аналоговые и цифровые видеокамеры, телевизоры, принтеры, сетевые карты и накопители информации.

2.6. Внешние устройства

Устройства ЭВМ перечислены на рис. 2.35. Рассмотрим внешнее устройство — ВУ или external device, — это устройство, как правило, конструктивно отделенное от основного блока системы.

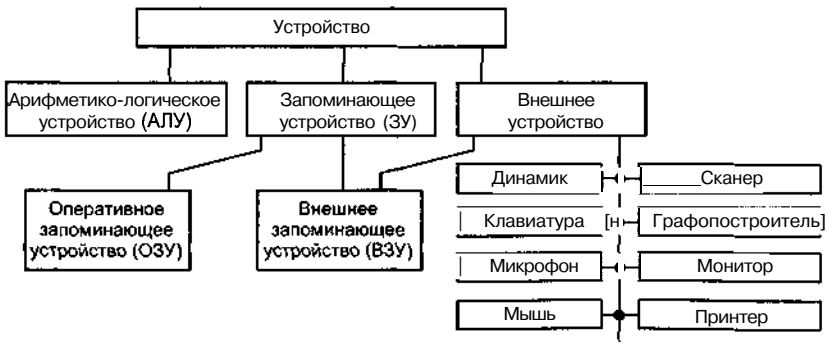


Рис. 2.35. Устройства ЭВМ

Внешние устройства предназначены для ввода данных в ЭВМ, пересылки данных, их хранения. В список внешних устройств, именуемых также периферийными устройствами, входит большое число аппаратов К ним, в первую очередь, относятся устройства ввода/вывода Характерной особенностью внешних устрбйств является их независимость от ЦП. Они имеют собственную систему управления и функционируют по ее командам. Многие внешние устройства имеют прямой доступ к памяти.

Накопители массивов информации (ВЗУ)

Массовая память, или mass storage, — внешнее ЗУ большой емкости. Выделяют три группы устройств массовой памяти. Первая из них служит для хранения наиболее часто используемых программ и данных. Обычно это осуществляется на комплексах резервирования матрицей независимых дисков и отдельных жестких дисков. Вторая группа предназначена для хранения регулярно, но не часто используемых программ и данных. К третьей группе относятся редко применяемые программы и данные, например изображения, особенно видеофильмы. В этих группах широко используются магнитные ленты (МЛ) и оптические диски Наряду с обычными здесь все чаще используются *съемные накопители.* Последние подключаются к компьютерам через разъемы, стандарты на которые определяют фирмы либо международные организации.

Внешние запоминающие устройства (ВЗУ) можно разделить на следующие классы:

- по типу доступа:
  - с произвольным доступом (диски, флэш-карты);
  - с последовательным доступом (ленты);
- по используемой технологии записи/считывания информации:
  - с магнитными носителями (НЖМД, НГМД),
  - с оптическими носителями (CD, DVD);
  - с магнитооптическими носителями (Fujitsu DynaMO);
  - использующие флэш-память,
- по типу носителя:
  - с постоянным носителем (жесткие диски);
  - со сменными носителями (гибкие диски, картриджи стримеров), сменные пакеты жестких дисков

Накопители на магнитных лентах. Эти накопители относятся к классу внешних запоминающих устройств последовательного доступа В них доступ к требуемому набору данных происходит только после завершения перемотки всей предшествующей части магнит-

ной ленты (МЛ). Такие накопители, благодаря низкой стоимости, простоте эксплуатации и хранения, компактности и долговременности использования, обладают несомненными преимуществами в тех случаях, когда порции данных обрабатываются последовательно друг за другом (рис. 2.36).

Магнитные ленты для цифровой записи данных размещаются на бобинах или кассетах (подобно лентам для бытовой аудио- или видеозаписи). Однако принципы размещения информации на МЛ в данном случае существенно другие (рис. 2.36, *в*):

- информация размещается на носителе в виде блоков (массивов данных фиксированной или переменной длины);
- информационные блоки разделены пустыми промежутками *(gap),* позволяющими считывающему устройству распознать начало (окончание) блока. Размер промежутка между записями выбирается достаточным для разгона ленты до установленной скорости и остановки ее точно на следующем промежутке. Недостаток промежутков между записями — уменьшение полезного объема МЛ, так как области, отведенные под про-

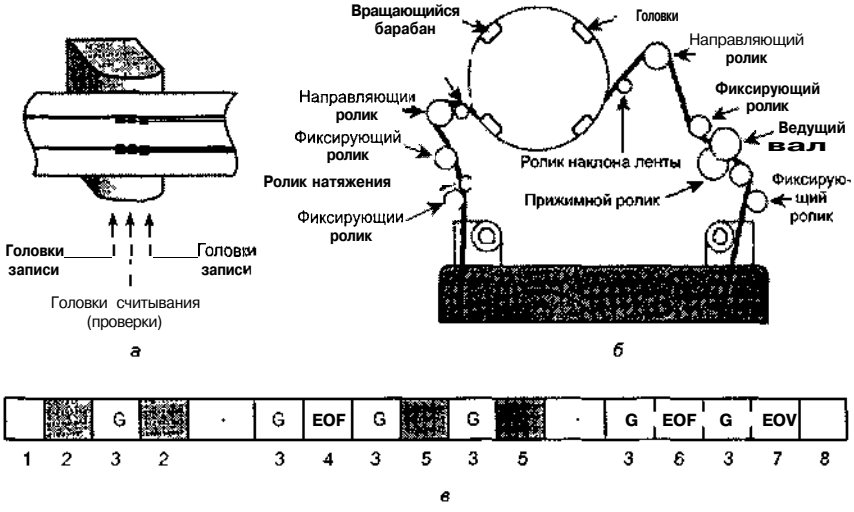


Рис. 2.36. Различные конструкции ЛПМ (лентопротяжных механизмов) *(а, б),* структура данных на магнитных лентах (в) 1 — физическое начало ленты (начальный ракорд), 2 — информационные блоки (ИБ) 1-го файла, 3 — GAP, промежуток между блоками, 4 — EOF — end-of-file, служебный блок, задающий конец 1-го файла, 5 — информационные блоки (ИБ) 2-го файла, 6 — конец 2-го файла, 7 — EOV — end-of-volume, служебный блок, задающий логический конец ленты, 8 — физический конец ленты (ракорд)

межутки, нельзя использовать для хранения данных. Частично указанный недостаток устраняет метод *блокирования,* суть которого состоит в объединении нескольких записей в блоки;

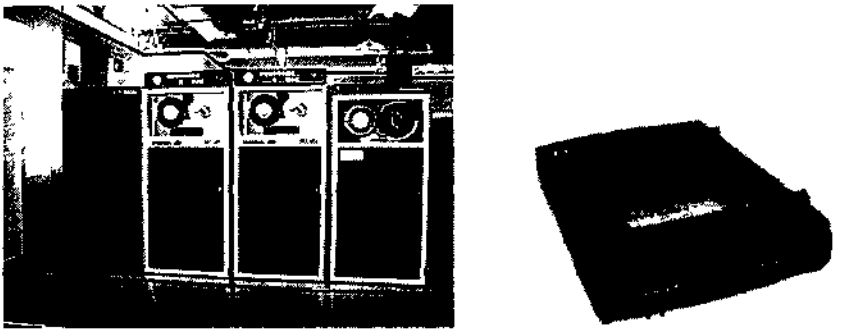
- блоки разделяются на *информационные* (ИБ — распознаются программами) и *служебные* (распознаются устройством — конец файла и конец тома);
- физическое начало и физический конец ленты обычно определяются оптическим или механическим образом (независимо от содержания ленты).

В ЭВМ обычно применяется девятидорожечная запись. Информация записывается одновременно девятью магнитными головками. Из девяти одновременно записываемых битов информации восемь являются информационными (один байт) и один — контрольным битом четности. Начало области магнитной ленты, в которую записывается информация, называется точкой загрузки и помечается специальным физическим маркером. Физический маркер представляет собой кусочек алюминиевой фольги, наклеиваемый на расстоянии 1—2 м от начала магнитной ленты. Конец информационной области МЛ помечается таким же физическим маркером, наклеиваемым на расстоянии от конца МЛ. Наличие указанных специальных маркеров, распознавание которых производится фотоэлектронным способом, позволяет осуществить перемотку МЛ к началу информационной области и автоматический останов по достижении ее конца.

Максимальное ограничение на размер блока зависит от размера доступной оперативной памяти (возможность размещения буфера считывания файла). Блокирование увеличивает полезный объем магнитной ленты за счет сокращения числа промежутков между записями. Кроме того, уменьшается количество операций ввода-вывода, так как за одну операцию производится пересылка не одной записи, а сразу нескольких. Преимущества блокирования, заключающиеся в увеличении полезного объема МЛ и уменьшении общего времени работы программы на ввод-вывод данных, значительно превосходят возникающие при этом недостатки, связанные с увеличением объемов данных в программе пользователя и необходимостью выполнения процедур по формированию блоков и разделению принятых блоков на записи.

Устройства записи-считывания информации с МЛ (рис. 2.37) («магнитофон») позволяет осуществлять следующие операции:

- пропустить (вперед или назад) несколько ИБ;
- пропустить (вперед или назад) несколько файлов;
- прочитать (записать) блок;



а б

Рис. 2.37. Накопители информации на магнитных лентах (НМЛ ЕС5017) *(а)* и стример ПЭВМ (б)

- прочитать (записать) файл;
- позиционировать на конец файла (для записи дополнительных ИБ в этот файл; очевидно, что данные последующих файлов будут затерты);
- позиционировать на начало ленты;
- позиционировать на конец ленты (для записи дополнительного файла).

Очевидно, если блок EOV будет записан в начале ленты, то все файлы становятся недоступными, и лента приобретает статус *инициализированной.*

Значение контрольного бита четности выбирается в зависимости от значений восьми информационных битов. Если число единиц в восьми информационных битах нечетное, то в контрольный бит четности заносится единица, а если четное — нуль. Таким образом, общее число единиц в девяти записываемых битах всегда должно быть четным, это контролируется в процессе чтения данных.

Предусмотрена возможность пропуска выявленных дефектных участков на МЛ. Помимо посимвольного контроля, производимого с помощью контрольного бита четности, существует поблочный контроль данных. Его суть заключается в том, что в конце каждого блока записывается контрольная комбинация. В случае возникновения в блоке единичной ошибки посимвольный и поблочный контроль позволяет определить ее местонахождение и выполнить автоматическое исправление. Для этого перед байтом поблочного контроля записывается байт циклического контроля. После обнаружения ошибки делается предположение о наличии временного дефекта МЛ и осуществляется повторная попытка записи информа-

ции на то же место. Если и последующие попытки заканчиваются неудачей, то дефектный участок пропускается. В целях контроля правильности выполнения операций записи-чтения помимо основного набора магнитных головок используется дополнительный.

С помощью дополнительного набора магнитных головок считываются только что записанные на МЛ биты информации, в случае их несовпадения идентифицируется состояние ошибки. Оба набора магнитных головок считывают данные с МЛ и при несовпадении какой-либо пары битов также будет выработан сигнал об ошибке.

Магнитные ленты в силу ряда своих положительных достоинств играют важную роль при организации больших информационных архивов и фондов пакетов программ

Картриджи с магнитными лентами. В одной из таких систем, получившей довольно широкое распространение, используются 8-миллиметровые ленты видеоформата, заключенные внутрь кассеты (рис. 2.38). Такая кассета называется *картриджем.* Емкость картриджа составляет от 2 до 5 Гбайт, а скорость считывания с него данных — несколько сотен килобайтов в секунду. Чтение и запись выполняются системой спиральной развертки, подобной той, что применяется в видеокассетах (рис. 2.36, *б).* Плотность записи данных составляет десятки миллионов битов на квадратный дюйм. Существуют системы, позволяющие автоматизировать загрузку и выгрузку картриджей таким образом, чтобы десятки гигабайт данных можно было скопировать с диска без вмешательства оператора.

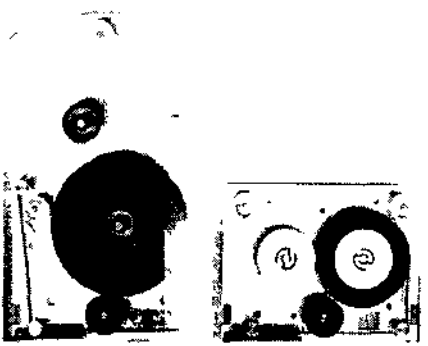


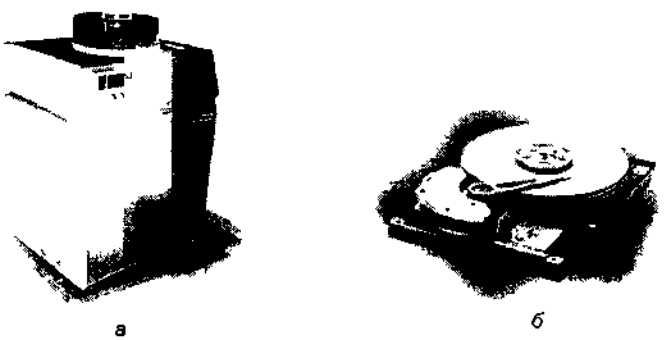
Рис. 2.38. Разновидности кассет (картриджей) для магнитной записи

Стример (англ. tape streamer) — устройство для резервного копирования больших объемов информации В качестве носителя здесь применяются кассеты с магнитной лентой емкостью 1—2 Гбайт и более Встроенные в стример средства аппаратного сжатия позволя-

ют автоматически уплотнять информацию перед ее записью и восстанавливать после считывания, что увеличивает объем сохраняемой информации. Недостатком стримеров является их сравнительно низкая скорость записи, поиска и считывания информации.

Размещение информации на МЛ связано со следующими проблемами. Для уверенного распознавания промежутка *(gap),* он должен иметь значительную длину (особенно при высоких скоростях перемотки/чтения). При скорости движения ленты 2—3 м/с длина промежутка должна составлять не менее 1—2 см. Очевидно, что для того, чтобы эффективность использования МЛ была достаточно высокой, длина ИБ должна как минимум в 2—3 раза превышать длину промежутка (при этом коэффициент полезного использования МЛ будет составлять 60—75 %). При этом также увеличивается скорость обмена между ОП и ВУ, поскольку за одно обращение к МЛ считывается как минимум один ИБ. Однако увеличение длины ИБ требует увеличения объема ОП для размещения буфера, связанного с данным файлом (буфер выделяется операционной системой при открытии файла), в связи с чем одновременное открытие большого числа файлов может оказаться невозможным при ограниченном размере ОП

Накопители на магнитных дисках (МД) получили наибольшее распространение. В них каждая запись данных имеет свой собственный уникальный адрес, обеспечивающий непосредственный (минуя все остальные записи) доступ к ней (рис. 2.39). В НМД предусмотрена аналогичная НМЛ возможность последовательного доступа к информации. Накопитель на магнитных дисках сочетает в себе не-



а б

Рис. 2.39. Накопители на жестких МД
а — НМД ЕС 5061 (прототип IBM 2311 — сменные и съемные пакеты дисков — 29 Мбайт), *б* — «винчестер» — несменные и несъемные пакеты дисков (до 80 Гбайт)

сколько устройств последовательного доступа, причем сокращение времени поиска данных обеспечивается за счет независимости доступа к записи от ее расположения относительно других записей. Конструкция НМД сложнее, чем у НМЛ а, следовательно, выше их стоимость. В НМД в качестве носителей данных используется *пакет магнитных дисков,* закрепленных на одном стержне, вокруг которого они вращаются с постоянной скоростью. Поверхность магнитного диска, покрытая ферромагнитным слоем, называется *рабочей.*

Первые подобные устройства (рис. 2.39, *а)* были оборудованы *сменными пакетами* МД. Вставленные в кожух с герметически закрывающимся поддоном, они образовывали компактные единицы хранения, именуемые томами. Наиболее распространенными емкостями томов были 7,25 Мбайт, 29 Мбайт (рис. 2.39, *а),* 100 Мбайт. Оператор ставил пакет на шпиндель устройства, снимал кожух (при этом пакет автоматически фиксировался на шпинделе) и включал двигатели раскрутки пакета. После достижения определенной скорости вращения осуществляется ввод в пространство между дисками пакета блока магнитных головок («гребенки»). Принцип размещения головок — *плавающий,* поскольку они удерживаются на необходимом расстоянии от поверхности диска расходящимися потоками воздуха, возникающими при вращении пакета. В дальнейшем в основном применялись или *полноконтактные* головки (гибкие диски) или *механически фиксируемые* в вакууме на определенном расстоянии от поверхности («винчестер»). Попытки использовать жидкие среды (различные масла) для обеспечения необходимого размещения головок успеха не имели.

Каждый магнитный диск пакета, кроме верхнего и нижнего, имеет две рабочие поверхности. Верхний и нижний магнитные диски обладают по одной *рабочей поверхности,* расположенной соответственно на нижней и верхней частях указанных дисков. Каждая рабочая поверхность магнитного диска разбита на N окружностей *(дорожек),* пронумерованные от 0 до $N - 1$ от края к центру. На каждой из дорожек начало области данных механически идентифицировано с помощью маркера начала оборота. Дорожки, расположенные одна под другой на разных магнитных дисках, образуют соответственно N *цилиндров.*

Из N цилиндров M являются резервными и $N - M$ — основными. Дорожки резервных цилиндров пользователям недоступны. Системные средства обеспечивают замену дорожки основного цилиндра, ставшей дефектной, на дорожку запасного цилиндра. Запись и считывание информации в НМД производит механизм дос-

тупа, состоящий из держателей магнитных головок (блок магнитных головок).

Количество магнитных головок равно числу рабочих поверхностей на одном пакете дисков. Если пакет состоит из 11 дисков, то механизм доступа состоит из 10 держателей с двумя магнитными головками на каждом из них. Держатели магнитных головок объединены в единый блок таким образом, чтобы обеспечить их синхронное перемещение вдоль всех цилиндров. Фиксируя блок механизма доступа на каком-либо из цилиндров с помощью только электронного переключения головок, можно сделать переход с одной дорожки на другую данного цилиндра. При фиксированном положении блока механизма доступа возможно обращение к любой из записей, находящихся на дорожках текущего цилиндра. Дорожки в цилиндре нумеруются, начиная с верхних. Как правило, обращение к дорожкам происходит с нулевой по последнюю одного цилиндра, потом с нулевой дорожки следующего цилиндра и т. д.

Любая операция чтения (записи) информации с (на) магнитного диска состоит из трех этапов. На первом этапе происходит механический подвод магнитной головки к дорожке, содержащей требуемые данные. На втором этапе обеспечивается ожидание момента, пока требуемая запись не окажется в зоне магнитной головки. На третьем этапе осуществляется собственно процесс обмена информацией между вычислительной машиной и магнитным диском. Таким образом, общее время, затрачиваемое на операцию записи-считывания, состоит из суммы времен поиска соответствующей дорожки, ожидания подвода записи (так называемое время ротационного запаздывания) и обмена с ЭВМ. Максимальное значение времени ротационного запаздывания равно времени, за которое совершается полный оборот магнитного диска.

В идейном плане размещение информации на МД аналогично МЛ (дорожка МД эквивалентна отрезку МЛ). Адрес блока на МД состоит из номера дорожки и номера блока на дорожке. Начало и конец блока распознаются по промежуткам, начало и конец дорожки — оптическим (для сменных МД) или электромагнитным (для постоянных МД) датчиком угла поворота оси пакета МД.

Размер блока, очевидно, не может быть больше длины дорожки МД. Считывающее устройство в данном случае ориентировано на выполнение единственной операции — прочитать (или записать) информационный блок, который задан своим адресом. За считывание файла несет ответственность операционная система, поддерживающая файловые структуры на МД.

Соображения по поводу длины блоков, отмеченные выше по поводу МЛ, сохраняют свою силу и для МД, однако здесь возникают и некоторые дополнительные сложности. Использование блоков фиксированной длины на МЛ не дает никаких преимуществ, в то время как для НМД использование блоков фиксированной длины позволяет использовать датчик угла поворота как дополнительный идентификатор конца блока, что приводит к увеличению КПД использования поверхности диска.

Очевидно, что дорожки внешних и внутренних цилиндров по плотности записи различаются, так как на всех дорожках находится одинаковое фиксированное количество секторов (для внутренних дорожек плотность записи больше, чем для внешних).

Теоретически внешние цилиндры могут содержать больше данных, так как имеют большую длину окружности. Однако в накопителях, не использующих метод зонной записи, все цилиндры содержат одинаковое количество данных, несмотря на то, что длина окружности внешних цилиндров может быть вдвое больше, чем внутренних. В результате теряется пространство внешних дорожек, так как оно используется крайне неэффективно (рис. 2.40, а).

Процесс управления плотностью записи называется прекомпенсацией. Для компенсации различной плотности записи используют метод зонно-секторной записи (Zone Bit Recording), где все рабочее пространство диска делится на зоны (8 и более). В зоне, расположенной на внешнем радиусе (младшая зона), записывается большее количество секторов на дорожку (120—96). К центру диска количество секторов уменьшается и в самой старшей зоне достигает 64—56. Так как скорость вращения диска постоянная величина, то от внешних зон при одном обороте диска поступает больше информации, чем от зон внутренних. Эта неравномерность поступления информации компенсируется путем увеличения скорости работы канала считывания/преобразования данных и использования спе-

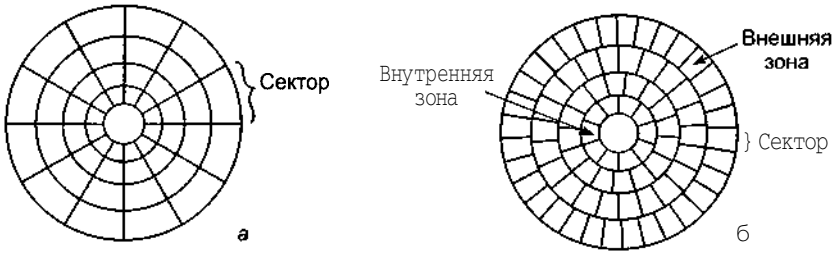


Рис. 2.40. Запись с постоянной угловой скоростью (*а*); зонная запись, компенсирующая различие линейных скоростей на различных цилиндрах НЖМД *(б)*

циальных перестраиваемых фильтров для частотной коррекции по зонам.

Чаще всего для этой цели применяются специальные однокристальные контроллеры. При этом емкость жестких дисков можно увеличить приблизительно на 30 %. Однако при конфигурировании системы такие винчестеры создают определенные трудности, которые приходится преодолевать.

С увеличением плотности записи на диск возникают трудности при детектировании пиков аналоговых сигналов, поступающих от магнитных головок. В последнее время для устранения этого недостатка стали применять PRLM-метод (Partial Response Maximum Likelihood — максимальная вероятность правильной реакции), в котором используется специальный алгоритм цифровой фильтрации входного сигнала.

Рассмотрим некоторые характеристика НЖМД. Для пользователя ПК весьма важно иметь большую емкость НЖМД, высокую производительность, а также обеспечить сохранность данных.

Рассмотренные ниже технические параметры определяют затраты времени на позиционирование магнитных головок (МГ) и передачу больших объемов информации, а также оказывают наибольшее влияние на работу прикладных программ, которые часто обращаются к диску для чтения и записи.

Среднее время поиска — время, необходимое для позиционирования МГ на нужную дорожку (для НЖМД емкостью 540 Мбайт — 10—12 мс; для НЖМД емкостью 720 Мбайт — 8—10 мс).

Средняя латентность (запаздывание) — время ожидания, в течение которого диски поворачиваются, пока нужный сектор не окажется под МГ (для НЖМД емкостью от 540 Мбайт до 1 Гбайт — около 5,6 мс; для НЖМД емкостью более 1 Гбайт — 4,2 мс).

Среднее время доступа = (Среднее время поиска) + (Среднее запаздывание). Среднее время доступа составляет от 12,2 до 18 мс.

Скорость передачи данных (пропускная способность) определяет скорость, с которой данные считываются или записываются на диск после позиционирования МГ.

Скорость передачи данных в групповом (burst) режиме (скорость внешнего обмена) — это фактически скорость считывания данных из буфера НЖМД (внутреннего оперативного запоминающего устройства — ОЗУ НЖМД). Буфер служит для промежуточного хранения считываемых с диска и записываемых на диск данных и ускорения доступа. Для EIDE и Fast ATA максимальная скорость передачи данных — 11,1—1,6 Мбайт/с, в SCSI 2 - 10—40 Мбайт/с.

Скорость внутреннего обмена (долговременная максимальная или минимальная скорость передачи данных) характеризует производительность НЖМД, когда буфер НЖМД не используется. Эта характеристика сильно зависит от скорости вращения цилиндра (5400 об/мин для НЖМД емкостью от 540 Мбайт до 1 Гбайта, 7200 об/мин для НЖМД емкостью более 1 Гбайта).

Накопители на компакт-дисках. Здесь носителем информации является CD-ROM (Compact Disk Read-Only Memory — компакт-диск, предназначенный только для чтения).

CD-ROM представляет собой прозрачный полимерный диск диаметром 12 см и толщиной 1,2 мм, на одну сторону которого напылен светоотражающий слой алюминия, защищенный от повреждений слоем прозрачного лака. Толщина напыления составляет несколько десятитысячных долей миллиметра.

Информация на диске представляется в виде последовательности впадин (углублений в диске) и выступов (их уровень соответствует поверхности диска), расположенных на спиральной дорожке, выходящей из области вблизи оси диска (рис. 2.41). На каждом дюйме (2,54 см) по радиусу диска размещается 16 тысяч витков спиральной дорожки. Для сравнения — на поверхности жесткого диска на дюйме по радиусу помещается лишь несколько сотен дорожек. Емкость CD достигает 780 Мбайт. Информация наносится на диск при его изготовлении и не может быть изменена.

Используемая в CD-системах оптическая технология основана на применении лазерного луча. Лазерный луч направляется на поверхность вращающегося диска, вдоль дорожек которого располагаются впадины, отражающие сфокусированный луч в направлении фотодетектора, фиксирующего записанные на диске двоичные данные. Лазер излучает когерентный свет, состоящий из синхронизированных волн одинаковой длины. Если объединить два одинаковых луча в одной фазе, получится более яркий луч, а если сдвинуть лучи на полфазы, они погасят друг друга. Но если два таких луча будут направлены на фотодетектор, то в первом случае он зафиксирует яркое пятно, а во втором — темное.

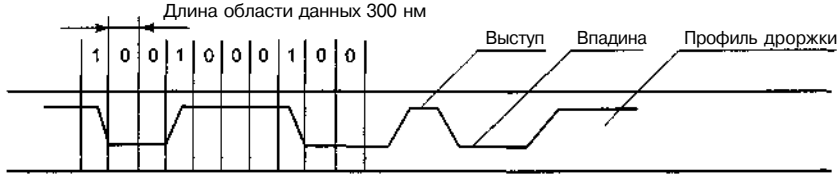


Рис. 2.41. Профиль дорожки CD-ROM

При вращении диска возможны три разных позиции источника луча и детектора относительно впадин и площадок на его поверхности. Когда свет отражается только от впадины или только от площадки, детектор фиксирует яркое пятно Однако на границе впадины, глубина которой равна четверти длины волны лазера, ситуация меняется. Волна, отраженная от впадины, смещается на 180° по фазе относительно волны, отраженной от площадки, и они гасят друг друга. Таким образом, на границах впадина—площадка и площадка—впадина детектор не видит отраженного луча и фиксирует темное пятно. На рис. 2.41 показано несколько переходов между площадками и впадинами. Если каждый переход, фиксируемый как темное пятно, обозначается двоичным значением «1», а поверхность впадины или площадки — двоичным значением «О», результирующая двоичная последовательность будет такой, как на этом рисунке. Она не является непосредственным представлением хранящихся на диске данных. Для CD применяется сложная система кодирования информации. Каждый байт представлен 14-разрядным кодом, позволяющим выявлять и исправлять ошибки.

CD-ROM обладают высокой удельной информационной емкостью, что позволяет создавать на их основе справочные системы и учебные комплексы с большой иллюстративной базой. Один CD по информационной емкости равен почти 500 дискетам. Считывание информации с CD-ROM происходит с достаточно высокой скоростью, хотя и заметно меньшей, чем скорость работы накопителей на жестком диске. CD-ROM просты и удобны в работе, имеют низкую удельную стоимость хранения данных, практически не изнашиваются, не могут быть поражены вирусами, на них невозможно случайно стереть информацию.

Сегодня почти все персональные компьютеры имеют накопитель CD-ROM (рис. 2.42, *а).* Накопитель включает двигатель для враще-

а б

Рис. 2.42. Накопитель CD-ROM *(а),* накопитель CD-MO *(б)*

ния диска, генератор лазерного луча и преобразователь отраженного лазерного луча в электрические сигналы, соответствующие «О» и «1».

В настоящее время весьма распространены перезаписываемые компакт-диски CD-R (CD-Recordable). Носители типа CD-R могут быть записаны самим пользователем на специальном CD-R-приводе. В основном здесь применяются технологии, основанные на изменении отражающих свойств вещества подложки компакт-диска под действием луча лазера. Перезаписываемые компакт-диски дороже обычных, так как в качестве светоотражающего слоя в них используется не алюминий, а золото. Подобные компакт-диски могут служить в качестве матриц (мастер-диск) для дальнейшего тиражирования.

Между этим слоем и поликарбонатной основой расположен регистрирующий слой из органического материала, темнеющего при нагревании. В процессе записи лазерный луч нагревает выбранные точки слоя, которые темнеют и перестают пропускать свет к отражающему слою, образуя участки, аналогичные впадинам. Накопители CD-R благодаря сильному удешевлению, приобретают все большее распространение.

Магнитооптические накопители. МО-диск представляет собой поликарбонатную подложку (часто ее также называют слоем) толщиной 1,2 мм, на которую нанесено несколько тонкопленочных слоев: защитный слой, предохраняющий поверхность диска от повреждений; отражающий слой, необходимый для корректной работы лазера; диэлектрические слои, которые теплоизолируют магнитный слой (чтобы эффективнее использовать энергию лазера при записи) и увеличивают эффект поляризации при чтении; магнитный слой — собственно хранитель информации.

Технология записи данных такова: лазерный луч нагревает точку на диске, а электромагнит изменяет магнитную ориентацию этой точки в зависимости от того, что необходимо записать — «О» или «1».

Считывание производится лазерным лучом меньшей (чем при записи) мощности, который, отражаясь от этой точки, меняет свою полярность.

МО-диски (и соответственно дисководы) выпускаются двух размеров:

- 3,5 дюйма, содержит 500 Мбайт данных;
- 5,25 дюйма, содержит 2,3 Гбайт данных.

Время доступа к данным составляет около 50 мс.

Магнитооптические накопители выпускаются двух типов: перезаписываемые и типа WORM (Write-Once, Read-Many — «однократная запись — многократное чтение»).

Дальнейшее развитие фирма Sony связывает с новой технологией оптической записи UDO (Ultra Density Optical). Технология UDO базируется на новом коротковолновом лазере с длиной волны 405 нм, применение которого позволяет существенно увеличить плотность размещения дорожек записи и плотность записи в дорожке. Процесс записи основан не на магнитооптической технологии, а на технологии изменения фазы. Формат UDO предполагает начальную емкость 5,25 диска в 40 Гб (по 20 Гб на сторону). В дальнейшем емкость диска может быть доведена до 60 и даже до 120 Гб. Устройства нового формата могут появиться примерно к 2005 г.

FMD ROM-накопители (fluorescent multilayer disk). Относительно недавно компанией C3D было объявлено о создании нового типа носителей информации под общим названием FMD ROM (fluorescent multilayer disk), т. е. флюоресцентный многослойный диск.

В отличие от обычного CD ROM, в котором отражающий алюминиевый слой нанесен на выдавленную подложку из полимера, из-за чего он собственно и непрозрачен, диск FMD ROM монолитен и при этом разделен по вертикали на некоторые условные области, названные разработчиками «слоями» (layer). Это не слои в привычном смысле, а скорее параметр форматирования диска, ближайший аналог — это сектора и дорожки для магнитных носителей. Толщина этих слоев строго фиксирована. Чтобы понять, почему разработчики выбрали именно эту толщину каждого из слоев, надо рассмотреть принципы записи/считывания информации на FMD ROM.

В оптических носителях (CD, DVD, магнитооптика) во время чтения луч полупроводникового лазера отражается от слоя с записанной информацией. Отраженный луч затем фиксируется детектором — (приемником).

Максимальная удельная емкость диска определяется размером светового пятна от лазера, которое в свою очередь зависит от длины волны (у красных лазеров — 650 нм). Можно использовать два слоя, причем сделать один из слоев прозрачным для излучения с определенной длиной волны, как это реализовано в DVD. Но два слоя — это предел, больше сделать очень сложно, так как нужны очень точные фокусирующие системы, которые будут работать только в лабораторных условиях. Массовое производство таких систем окажется дорогим и нерентабельным. Технологии отражающих слоев, вероятно, подошли к своему пределу развития.

Разработчиками FMD было предложено следующее решение: материал, содержащий записанную информацию, не отражает, как подложка в DVD или CD, а излучает световые волны. Использовано явление флюоресценции, т. е. при освещении активирующим

излучением (в данном случае полупроводниковым лазером с определенной длиной волны) вещество начинает излучать, сдвигая спектр падающего на него излучения в сторону красного цвета на определенную величину, причем величина сдвига зависит от толщины слоя.

Таким образом, выбрав такую толщину слоя, чтобы спектр отраженного света получался смещенным относительно длины волны излучающего лазера на строго определенную величину, например на 30 или 50 нм, можно с высокой достоверностью записывать информацию вглубь диска и впоследствии считывать ее без потери данных. Для FMD ROM разработчиками так же предложено название «трехмерный диск», и в данном случае это вполне оправданно. Таким образом, плотность записи будет зависеть и от чувствительности регистрирующего детектора. Чем меньше то дополнительное излучение флюоресцирующего вещества, добавляющееся к частоте рабочего лазера, который удастся зафиксировать, тем большее число слоев можно вместить в один диск. Излученный свет от флюоресцентного слоя некогерентен и хорошо контрастирует с отраженным светом лазера, что является дополнительной гарантией надежности считывания, ведь без отражений все равно не обойтись, они будут происходить от поверхности диска и других записанных слоев. Качественное ухудшение сигнала в обычных (отражающих) многослойных дисках нарастает с увеличением числа слоев, но вот в случае с флюоресцентными дисками это ухудшение происходит гораздо медленнее. По заявлению разработчиков FMD ROM, даже при количестве слоев больше сотни не будет происходить сильного искажения полезного сигнала.

Используя синий лазер (480 нм) можно увеличить плотность записи до десятков терабайт на один FM-диск. Возможно создание диска с 1000 слоями — это уже субмолекулярные размеры и теоретически возможно создание пятна размером в несколько молекул, проблема лишь в том, как зафиксировать столь малое флюоресцентное излучение. Одна из главных особенностей этой разработки — возможность параллельного чтения слоев (т. е. последовательность бит будет записана не по «дорожкам», а по слоям); скорость выборки данных в этом случае должна возрастать.

Принцип записи на FMD ROM основан на явлении фотохромизма — это свойство некоторых веществ под действием активирующего излучения обратимо переходить из одного состояния в другое, при этом изменяя свои физические свойства (например, такие как цвет, появление/исчезновение флюоресценции и т. д.). Материал, из которого состоит FMD ROM содержит специальную фотохромную

субстанцию, которая циклизуется под воздействием лазерного луча определенной длины волны, превращаясь в необходимый устойчивый флюоресцент.

Обратная реакция рециклизации, приводящая к исчезновению флюоресцентных свойств (операция стирания), происходит под действием лазера с другой длиной волны. Стирающая частота лазера выбирается с таким расчетом, чтобы она не встречалась в повседневной жизни во избежание потери данных.

Флэш-накопитель — портативный носитель информации с интерфейсом USB. Одним из первых на отечественном рынке появился накопитель MAXIMUS Flash USB Drive (корейской фирмы Jung MyungTelecom). Строго говоря, слово Drive в названии корейского флэш-накопителя — это маркетинговое преувеличение — никакого привода там нет, как нет и движущихся частей. По сути, разработчики просто отразили в названии процедуру работы с MAXIMUS Flash USB Drive, как с любым внешним дисководом (CD-RW, Zip, жестким диском). На самом же деле «псевдодиск» состоит из микросхемы флэш-ПЗУ, спецконтроллера и интерфейса USB.

У этого типа памяти есть много преимуществ:

- быстрое время доступа;
- высокая надежность (в силу отсутствия движущихся частей);
- компактность;
- долговечность.

Устройства поддерживаются операционными системами Windows 2000 и XP без необходимости установки каких-либо специальных драйверов.

При включении устройства в разъем оно автоматически распознается системой и регистрируется. При завершении работы необходимо выполнить отключение устройства, после чего оно будет удалено из системы и может быть снято.

Рассмотрим далее характеристики некоторых представителей данного типа устройств.

До недавнего времени карты Flash-памяти использовались в основном, только в карманных компьютерах и цифровых камерах. И вот перед нами соединение двух прогрессивных технологий: шины USB и Flash-памяти — USB Drive компании J.M.Tek (рис. 2.43). Устройство небольшого размера (с зажигалку), USB-разъем закрывается защитной заглушкой с защелкой для закрепления в кармане. С торца имеется микропереключатель для защиты диска от случайной записи и контрольный индикатор режима работы. В режиме записи он светится желтым светом, в режиме чтения — зеленым. Характеристики устройства: емкость диска — 32 Мбайт; интерфейс — USB 1.1; ско-

Рис. 2.43. Флэш-накопитель USB Drive

рость чтения — 800 Кбайт/с; скорость записи — 500 Кбайт/с; рабочая температура -0. .+45 °С, влажность — 5—95 %, срок службы — 10 лет; размеры — 54 х 20 х 10 мм; вес — 15 г.

Периферийные устройства: ввод-вывод текстовой, графической, мультимедиа информации

Рассматриваются устройства, предназначенные для ввода-вывода *массивов данных* в пакетном режиме (листами, страницами, бланками и другими крупными блоками информации).

В основном речь пойдет о вводе-выводе графической информации. За редким исключением, большинство современных устройств пакетного ввода-вывода предназначаются именно для работы с этим типом информации.

При этом можно утверждать, что:

- *принтеры и сканеры* — это, как правило (хотя есть и исключения), устройства для работы с *растровой* информацией;
- *плоттеры и дигитайзеры* — предназначены для обработки векторной графической информации.

Принтеры. Принтер — устройство для вывода текстовой или графической информации на различные твердые носители. Существует несколько типов принтеров: матричные, струйные, лазерные, твердочернильные, термосублимационные и т. д. Каждую группу принтеров характеризуют свои отличительные черты, присущие только этому типу устройств вывода информации. Рассмотрим каждую из групп более подробно.

Принтеры ударного типа (impact printer) Принтеры ударного действия, или impact-принтеры, создают изображение путем механического давления на бумагу через ленту с красителем. В качестве ударного механизма применяются либо шаблоны символов (механизм печатающей машинки) или иголки, конструктивно объединенные в матрицы В остальных типах принтеров (*non-impact*-принтерах) вывод изображения осуществляется с помо-

шью чернил, сухого электростатического переноса изображения или с применением тепла.

Первые модели печатающих устройств для вывода информации конструктивно представляли собой модернизированные варианты электрических пишущих машинок и применялись в 60-х и 70-х гг. в основном для диалогового ввода-вывода небольшого количества данных.

Основным типом устройств вывода массовой информации в те времена были *построчные печатающие устройства барабанного типа,* использующие механизм, состоящий из символьного барабана, красящей ленты, системы продвижения перфорированной бумажной ленты (обычно, рулонной либо сфальцованной в стопу) и ударных пуассонов.

В матричных принтерах (dot matrix printer) изображение формируется несколькими иголками, расположенными в головке принтера (рис. 2.44). Иголки, расположенные внутри головки, обычно активизируются электромагнитным методом. Каждая ударная иголка приводится в движение независимым электромеханическим преобразователем на основе соленоида Головка двигается по горизонтальной направляющей и управляется шаговым двигателем. Обычно печать выполняется как при прямом, так и при обратном проходе печатающей головки. Бумага продвигается с помощью вала, а между бумагой и головкой принтера располагается красящая лента У большинства моделей принтеров красящая лента заключена в специальный пластмассовый корпус, называемый картриджем. Картриджи различаются по величине и форме для различных моде-

а б

Рис. 2.44. Печатающая головка *(а)* и картридж *(б)* матричного принтера

лей. Красящая лента находится внутри корпуса картриджа в виде бесконечной ленты Мебиуса.

Струйные принтеры. Главным элементом струйного принтера является печатающая головка, состоящая из сопел, к которым подводятся чернила. Число сопел находится в диапазоне от 16 до 64, а иногда доходит до нескольких сотен.

Чернила подаются к соплам за счет капиллярных свойств и удерживаются от вытекания за счет сил поверхностного натяжения жидкости. В головку встроен специальный механизм, позволяющий выбрасывать из сопла микроскопическую капельку чернил. Печатающая головка при печати перемещается поступательно слева направо, отпечатав строку, передвигается вниз по листу. Работают эти принтеры практически бесшумно. В зависимости от устройства этого механизма различают принадлежность принтера к тому или иному классу.

Струйные принтеры подразделяются на устройства непрерывного (continuous drop) и дискретного (drop on demand) действия. Ввиду менее высокой цены более распространенными являются принтеры второго типа, в свою очередь, подразделяющиеся на следующие основные:

- пьезоэлектрические (piezo-ink) — Epson, Brother;
- пузырьковые (bubble-jet) — Hewlett-Packard, Canon, Lexmark.

Каждый из этих двух способов по-своему привлекателен, однако каждый из них не свободен от недостатков.

Пьезоэлектрическая технология дешева, отличается надежностью (так как не используется высокая температура). Этот способ управления менее инерционен, чем нагрев, что позволяет повысить скорость печати.

Для реализации пьезоэлектрического метода в каждое сопло установлен пьезокристалл, связанный с диафрагмой. Под воздействием электрического заряда происходит деформация пьезоэлемента. При печати находящийся в трубке пьезоэлемент, сжимая и разжимая трубку, наполняет капиллярную систему чернилами. Чернила, которые отжимаются назад, перетекают обратно в резервуар, а чернила, которые выдавились наружу, образуют на бумаге точки.

Пузырьковая технология связана с высокой температурой. При высокой температуре нагреватель со временем покрывается слоем нагара, поэтому в принтерах, использующих эту технологию, печатающая головка довольно часто выходит из строя. В таких случаях она вместе с резервуаром для чернил образует конструктивный единый узел. Достоинством этого типа принтеров является высокая долговечность (исключая печатающие головки, которые изнашива-

ются очень быстро и заменяются вместе со сменой чернильного картриджа), а недостатком — недостаточная резкость получаемых отпечатков.

Метод газовых пузырей базируется на термической технологии. Каждое сопло оборудовано нагревательным элементом, который при пропускании через него тока, за несколько микросекунд нагревается до температуры около 500 ° С. Возникающие при резком нагревании газовые пузыри стараются вытолкнуть через выходное отверстие сопла порцию (каплю) жидких чернил, которые переносятся на бумагу. При отключении тока нагревательный элемент остывает, паровой пузырь уменьшается, и через входное отверстие поступает новая порция чернил.

Цветные струйные принтеры имеют более высокое качество печати по сравнению с игольчатыми цветными принтерами и невысокую стоимость по сравнению с лазерными. Цветное изображение получается за счет использования (наложения друг на друга) четырех основных цветов. Уровень шума струйных принтеров значительно ниже, чем у игольчатых, поскольку его источником является только двигатель, управляющий перемещением печатающей головки. При черновой печати скорость струйного принтера значительно выше, чем у игольчатого. При печати с качеством LQ скорость составляет 3—4 (до 10) страницы в минуту. Качество печати зависит от количества сопел в печатающей головке — чем их больше, тем выше качество. Большое значение имеет качество и толщина бумаги. Выпускается специальная бумага для струйных принтеров, но можно печатать на обычной бумаге плотностью от 60 до 135 г/кв. м. В некоторых моделях для быстрого высыхания чернил применяется подогрев бумаги. Разрешение струйных принтеров при печати графики составляет от 300 x 300 до 720 x 720 dpi (точек на дюйм).

Печать цветных изображений на струйных принтерах происходит путем смешения четырех основных цветов — голубого, пурпурного, желтого и черного. Эти цвета часто называют базовыми триадными цветами, а в полиграфии это называется цветовой моделью CMYK (от английских названий цветов Cyan, Magenta, Yellow, blacK). В дорогих моделях принтеров используются дополнительные два цвета — либо светло-голубой и светло-пурпурный, либо оранжевый и зеленый (такие модели называют также фотопринтерами: они отличаются повышенным качеством цветопередачи. Хороший струйный фотопринтер на сегодняшний день обеспечивает вполне приемлемую альтернативу дорогим цветным лазерным устройствам.

Отметим, что струйные принтеры очень критичны к качеству бумаги, поэтому здесь следует придерживаться рекомендаций про-

изводителя принтера. Наивысшее качество струйной печати достигается на специальной фотобумаге, отличающейся достаточно высокой ценой.

Фотоэлектронные печатающие устройства. Фотоэлектронные способы печати основаны на освещении заряженной светочувствительной поверхности промежуточного носителя и формировании на ней изображения в виде электростатического рельефа, притягивающего частицы красителя, которые далее переносятся на бумагу.

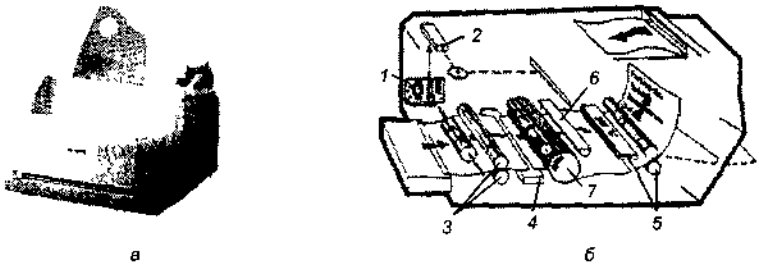
Для освещения поверхности промежуточного носителя используют:

- в лазерных принтерах — полупроводниковый лазер;
- в светодиодных — светодиодную матрицу;
- в принтерах с жидкокристаллическим затвором — люминесцентную лампу.

Лазерные принтеры обеспечивают более высокое качество, чем струйные принтеры. Наиболее известными фирмами-разработчиками лазерных принтеров являются Hewlett-Packard, Lexmark, Epson, Canon, Toshiba, Ricoh (рис. 2.45).

Принцип действия лазерного принтера основан на методе сухого электростатического переноса изображения, изобретенном Ч. Ф. Карлсоном в 1939 г. и реализуемом также в копировальных аппаратах.

Функциональная схема лазерного принтера приведена на рис. 2.45, *б*. Основным элементом конструкции лазерного принтера является вращающийся барабан, служащий промежуточным носителем, с помощью которого производится перенос изображения на бумагу. Барабан представляет собой цилиндр, покрытый тонкой пленкой светопроводящего полупроводника. Обычно в качестве та-



а *б*

Рис. 2.45. Лазерный принтер:
а — общий вид; *б* — схема принтера: *1* — процессор; *2* — лазер; *3* — механизм «зарядки» бумаги; *4* — барабан-девелопер; *5* — механизм нагрева бумаги; *6* — лезвие; *7* — фотобарабан

кого полупроводника используется оксид цинка или селен. По поверхности барабана равномерно распределяется статический заряд. Это обеспечивается с помощью тонкой проволоки или сетки, называемой коронирующим проводом. На этот провод подается высокое напряжение, вызывающее возникновение вокруг него светящейся ионизированной области, называемой короной.

Лазер, управляемый микроконтроллером, генерирует тонкий световой луч, отражающийся от вращающегося зеркала. Развертка изображения происходит так же, как и в телевизионном кинескопе: движение луча по строке и кадру. С помощью вращающегося зеркала луч скользит вдоль цилиндра, причем его яркость меняется скачком: от полного света до полной темноты, и также скачкообразно (по точкам) заряжается цилиндр. Этот луч, приходя на барабан, изменяет его электрический заряд в точке прикосновения. Размер заряженной площади точки зависит от фокусировки луча лазера. Фокусируется луч с помощью объектива. Признаком хорошей фокусировки считают наличие четких кромок и углов на изображении. Для некоторых типов принтеров в процессе подзарядки потенциал поверхности барабана уменьшается от 900 до 200 вольт. Таким образом, на барабане, промежуточном носителе, возникает скрытая копия изображения в виде электростатического рельефа.

На следующем этапе на фотонаборный барабан наносится тонер — краска, представляющая собой мельчайшие частицы. Под действием статического заряда эти частицы легко притягиваются к поверхности барабана в точках, подвергшихся экспозиции, и формируют изображение уже в виде рельефа красителя.

Бумага втягивается из подающего лотка и с помощью системы валиков перемещается к барабану. Перед самым барабаном бумаге сообщается статический заряд. Затем бумага соприкасается с барабаном и притягивает благодаря своему заряду частички тонера, нанесенные ранее на барабан.

Для фиксации тонера бумага вновь заряжается и пропускается между двумя роликами с температурой около 180 °С. После окончания процесса печати барабан полностью разряжается, очищается от прилипших лишних частиц, тем самым готовясь для нового процесса печати. Лазерный принтер является постраничным, т. е. формирует для печати полную страницу.

Светодиодные принтеры, или LED-принтеры (Light Emitting Diode), основаны на том же принципе действия, что и лазерные. Конструктивным различием является то, что барабан освещается не лучом лазера, развертка которого обеспечивается с помощью механически управляемых зеркал, а неподвижной диодной строкой, состоящей

из 2500 светодиодов, которая описывает не каждую точку, а целую строку. На основе этой технологии работают принтеры фирмы OKI

В принтерах с жидкокристаллическим затвором в качестве источника света служит люминесцентная лампа Свет лампы экспонируется через жидкокристаллический затвор, своеобразный прерыватель света, управляемый от компьютера Скорость печати такого принтера ограничена скоростью срабатывания жидкокристаллического затвора и не превышает 9 листов в секунду

Сканеры. Сканер — устройство для ввода графической растровой информации в ЭВМ. Сканеры классифицируются по типу самого устройства и его активных элементов, интерфейсу, поддерживаемой глубине цвета и разрешению (рис. 2 46).

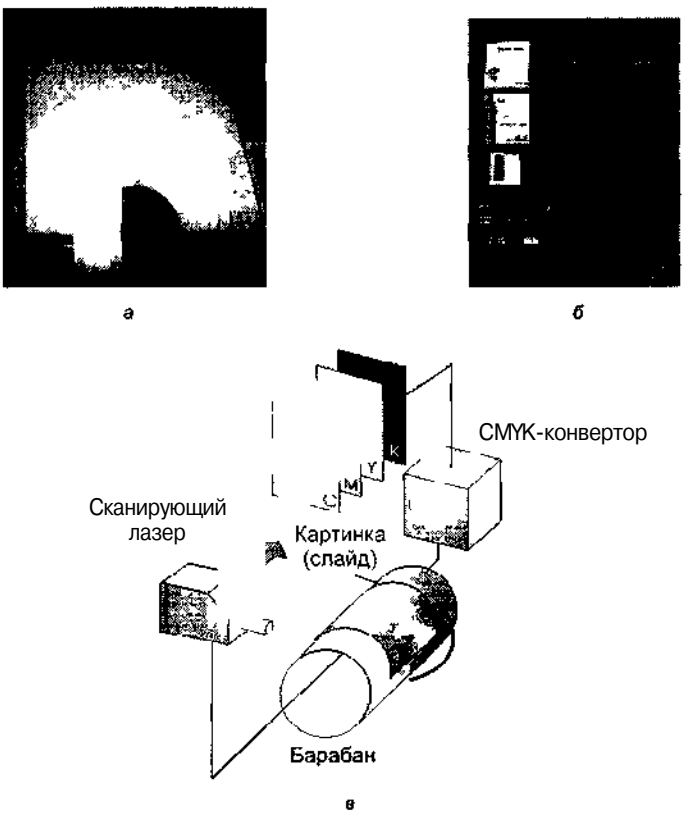


Рис. 2.46. Планшетный *(а),* ручной *(б)* персональные сканеры и устройство барабанного сканера (в)

По типу активных элементов различают сканеры

- на основе приборов с зарядовой связью (ПЗС, CCD),
- на основе фотоэлектронных умножителей (ФЭУ).

П З С основаны на явлении увеличения проводимости полупроводникового *p-n*-перехода под действием света Эти приборы состоят из большого количества датчиков, преобразующих световую энергию в аналогичную по интенсивности электрическую.

ПЗС — это полупроводниковый прибор со структурой полевого МОП-транзистора с очень длинными каналами и большим количеством затворов, расположенных близко друг от друга между электродами истока и стока. Каждый затвор и пластина образуют МОП-конденсатор. Так как конденсатор способен долго хранить заряд, то ПЗС могут использоваться в качестве запоминающего устройства. По существу ПЗС действует как длинный сдвигающий регистр, имеющий высокую плотность расположения разрядов. Манипулируя напряжением на затворах, можно перемещать заряд с одного МОП-конденсатора на соседний по цепочке канала. ПЗС особенно пригодны в тех ситуациях, когда запись и считывание данных производится последовательно, что характерно для процесса сканирования.

Как правило, датчики располагаются в линию, называемую CCD-линейкой, или матрицей. В цветных сканерах таких линеек три — по одной на каждый из основных цветов (красный, зеленый, синий — RGB). Обычно в документации на сканер приводится число элементов CCD-линейки на единицу длины Другими важными параметрами матрицы являются уровень ее собственного шума (флуктуации напряжения), девиация (разброс) электрических параметров разных ячеек в линейке, а также уровень наводок одной ячейки на другую.

Фотоэлектронные умножители по свойствам похожи на электронные лампы и также обладают двумя электродами — анодом и катодом Сканируемый с помощью ФЭУ оригинал освещается мощной галогенной лампой; отраженный от него свет попадает на катод ФЭУ, выбивая из последнего электроны, вызывая слабый электрический ток Затем внутри ФЭУ этот электрический ток усиливается и снимается с анода. Числовые значения снятого с каждого анода напряжения квантуются, с помощью АЦП преобразуются в цифровой вид и выдаются как результат сканирования. Фотоэлектронные умножители, как правило, обеспечивают лучшее качество сканирования, тогда как ПЗС обычно плохо «различают» детали в тенях (темных областях изображения)

По технической реализации различают ручные, планшетные и проекционные устройства.

Барабанный сканер — устройство, основанное на ФЭУ. Сканируемый оригинал закрепляется на специальном прозрачном барабане, вращающемся с большой скоростью, датчики считывают интенсивность света сквозь небольшое окошечко.

Барабанные сканеры обеспечивают наивысшее качество. Сканер вводит изображение в ЭВМ как множество точек, указав для каждой координаты и номер цвета, и по этим данным на монитор выводится копия изображения. Если с помощью сканера вводится текст, компьютер воспринимает его как картинку, а не как последовательность символов. Для преобразования такого графического текста в обычный символьный формат используют программы оптического распознавания образов.

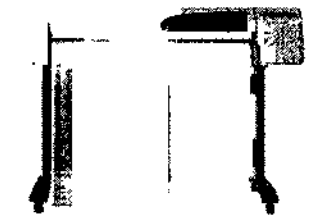
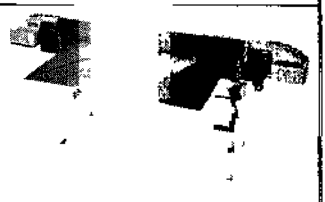
Проекционный сканер — это практически фотоаппарат, только очень медленный. Сканируемый предмет закрепляется перед сканирующей головкой, подсвечивается и построчно сканируется. В считывающих датчиках проекционных сканеров также используются ПЗС.

Ротационные сканеры (этот тип механизма используют, в частности, Bell+Howell и Kodak) менее других склонны к замятию листов и обычно допускают настройку на толщину бумаги, однако сканирование сильно разнородного материала (широких и узких документов, грубой и тонкой бумаги) в одном пакете не рекомендуется. К достоинствам ротационных сканеров принято относить их неприхотливость в работе и возврат отсканированного материала на операторский стол, а к недостаткам — несколько повышенное число документов, отсканированных с перекосом.

Плоттеры. Задача вывода информации, представленной в графической форме, возникла одновременно с появлением вычислительных машин, и ее решение — одна из основных целей вычислительных средств, применяемых для автоматизации проектирования. Устройства, выполняющие функции вывода графической информации на бумажные и некоторые другие типы носителей, называются графопостроителями или плоттерами (от англ. plotter) — термин, который, как и многие другие транслитерированные англоязычные термины, уже вытеснил свой русскоязычный аналог.

Большинство рассмотренных ниже технологий используется в плоттерах не только большого (A0, A1) формата, но и в плоттерах меньшего формата и даже в принтерах (табл. 2.9).

Таблица 2.9. **Примеры промышленных плоттеров**

Марка	Характеристики	Общий вид устройства
HP DesignJet 500 Printer (24 inch) 16 MB memory	Ширина носителя 1070 мм. Ширина 1057 мм Разрешение 600 x 1200 dpi Объем памяти (макс) 16 Мбайт. Интерфейс LPT / USB. Сетевой интерфейс JetDire. Скорость 90 с/ 3 мин	
Encad NovaJet 880	Формат 1067/1520 мм. Ширина печати 1055/1510 мм. Разрешение 600 x 600 dpi. Подача чернил - двойная система непрерывной подачи из емкостей по 500 мл. Объем памяти 64 (256). Язык PostScript. Опция (программ./апп. RIP). Интерфейсы Centronics, RS 232C/RS 422C, AppleTalk. Сетевой интерфейс 10/100 Base TX. Производительность, 5,8/12,4 $м^2$/час. Драйверы ADI (AutoCAD R12-14),WINDOWS 95/98, NT	
OCE 9300 P2R с двумя рулонами	OCE 9300 - плоттер для рабочих групп с производительностью 3 м/мин и до 100 отпечатков в день OCE 9300 построен по технологии печати OCE Instant Fusing с бесконтактной низкотемпературной системой закрепления тонера, позволяющей выполнять печать даже сложных чертежей в ярких отчетливых линиях. Программное обеспечение OCE Plot Director предоставляет функции для управления печатью с удаленной станции, в том числе организацию очередей, сохранение настроек плоттера, эмуляцию перьевого графопостроителя, выбор подающего рола, задание числа копий, масштабирование, поворот оригинала	

Перьевые плоттеры (ПП, pen plotter). Перьевые плоттеры — это электромеханические устройства векторного типа, и на ПП традиционно выводят графические изображения различные векторные программные системы типа AutoCAD. Они создают изображение с помощью пишущих элементов, обобщенно называемых *перьями,* хотя имеется несколько видов таких элементов, отличающихся друг от друга используемым видом жидкого красителя. Пишущие элементы бывают *одноразовые* и *многоразовые* (допускающие перезарядку). Перо крепится в держателе пишущего узла, который имеет одну или две степени свободы перемещения.

Существует два типа ПП: *планшетные,* в которых бумага неподвижна, а перо перемещается по всей плоскости изображения, и *барабанные* (или рулонные), в которых перо перемещается вдоль одной оси координат, а бумага — вдоль другой за счет захвата транспортным валом, обычно фрикционным. Перемещения выполняются с помощью шаговых (в подавляющем большинстве плоттеров) или линейных электродвигателей, создающих довольно большой шум. Хотя точность вывода информации барабанными плоттерами несколько ниже, чем. планшетными, она удовлетворяет требованиям большинства задач. Эти плоттеры более компактны и могут отрезать от рулона лист необходимого размера автоматически, что определило их доминирование на рынке больших ПП (ПП формата А3 обычно планшетные).

Отличительными особенностями перьевых плоттеров являются высокое качество получаемого изображения и хорошая цветопередача при использовании цветных пишущих элементов. К сожалению, скорость вывода информации в ПП невысока, несмотря на все более быструю механику и попытки оптимизации процедуры рисования; существует и проблема подбора пары носитель — чернила.

Карандашно-перьевые плоттеры (КПП, pen/pencil) — разновидность перьевых — отличаются возможностью установки специализированного пишущего узла с цанговым механизмом для использования обычных карандашных грифелей, который обеспечивает постоянное усилие нажима грифеля на бумагу и его автоподачу при истачиваний. В результате не требуется постоянно следить за процессом вывода информации, как при эксплуатации ПП, в которых может засоряться канал истечения красителя.

Дополнительные преимущества карандашной технологии:

- карандашный грифель не высыхает, и карандаш пишет на любой скорости (при использовании жидких красителей необходимо учитывать время их вытекания из пера и время высыхания);
- карандаш позволяет рисовать на любых бумажных носителях, в том числе и не очень высокого качества; при этом изображения качественны, дают хорошие оттиски при копировании и в то же время их можно корректировать ластиком;
- грифели просто купить, значительно экономя на расходных материалах.

ПП и КПП особенно привлекательны для тех, кому важнее качество, нежели количество изображений, и кто имеет скромный бюджет.

Все остальные типы плоттеров образуют изображения на носителе информации, используя различные физические процессы, в частности прибегая к дискретному (растровому) способу его создания.

Струйные плоттеры (СП, ink-jet plotter). Струйная технология создания изображения известна с 70-х гг., но ее массовый выход на рынок стал возможен только с разработкой фирмой Canon технологии создания реактивного пузырька (Bubblejet) — направленного распыления чернил на бумагу с помощью сотен мельчайших форсунок одноразовой печатающей головки. Каждой форсунке соответствует свой микроскопический нагревательный элемент (терморезистор), который мгновенно (за 7—10 мкс) нагревается под воздействием электрического импульса. Чернила закипают, и пары создают пузырек, который выталкивает из форсунки каплю чернил. Когда импульс кончается, терморезистор столь же быстро остывает, а пузырек исчезает.

Печатающие головки могут быть «цветными» и иметь соответствующее число групп форсунок. Для создания полноценного изображения используется стандартная для полиграфии цветовая схема CMYK, использующая четыре цвета: Cyan — голубой, Magenta — пурпурный, Yellow — желтый и Black — черный. Сложные цвета образуются смешением основных, причем получение оттенков различных цветов достигается путем сгущения или разрежения точек соответствующего цвета в фрагменте изображения (аналогичный способ используется при получении различных оттенков «серого» при выводе монохромных изображений).

Струйная технология имеет ряд достоинств. Сюда можно отнести простоту реализации, высокое разрешение, низкую потребляемую мощность и относительно высокую скорость печати. Приемлемая цена, высокое качество и большие возможности делают СП серьезным конкурентом перьевых устройств. Спрос на СП со стороны работающих с настольными издательскими системами и пользователей систем автоматизированного проектирования, выпускающих сложные чертежи формата А0, растет, однако невысокая скорость вывода графической информации и выцветание со временем полученного цветного изображения без принятия специальных мер (использования ламинирования или специальной «самоламинирующейся» бумаги) ограничивает их применение.

Электростатические плоттеры (ЭП, electrostatic plotter). Электростатическая технология основывается на создании скрытого электрического изображения (потенциального рельефа) на поверхности носителя — специальной электростатической бумаги, рабочая поверхность которой покрыта тонким слоем диэлектрика, а

основа пропитана гидрофильными солями для обеспечения требуемых влажности и электропроводности. Потенциальный рельеф формируется при осаждении на поверхность диэлектрика свободных зарядов, образующихся при возбуждении тончайших электродов записывающей головки высоковольтными импульсами напряжения.

Когда бумага проходит через проявляющий узел с жидким намагниченным тонером, частицы тонера оседают на заряженных участках бумаги. Полная цветовая гамма получается за четыре цикла создания скрытого изображения и прохода носителя через четыре проявляющих узла с соответствующими тонерами.

Электростатические плоттеры можно было бы считать идеальными устройствами, если бы не необходимость поддержания стабильных температуры и влажности в помещении, необходимость тщательного обслуживания и их высокая стоимость, в связи с чем ЭП приобретают пользователи, имеющие оправданно высокие требования к производительности и качеству. Для достижения максимальной эффективности ЭП обычно работают как сетевые устройства, для чего снабжены адаптерами сетевого интерфейса. Немаловажны также высокая устойчивость изображения к воздействию ультрафиолетовых лучей и невысокая (на уровне стоимости высококачественной типографской) стоимость электростатической бумаги. ЭП применяют при высокой степени автоматизации проектных работ в солидных организациях и в геоинформационных системах (ГИС).

Плоттеры прямого вывода изображения (ППВИ, direct imaging plotter). Изображение в ППВИ создается на специальной термобумаге (бумаге, пропитанной теплочувствительным веществом) длинной (на всю ширину плоттера) «гребенкой» миниатюрных нагревателей. Термобумага, которая обычно подается с рулона, движется вдоль «гребенки» и меняет цвет в местах нагрева. Изображение получается высококачественным (разрешение до 800 dpi), но только монохромным.

Сейчас цены на термобумагу снизились, недостатки, когда-то присущие ей (чувствительность к изменениям температуры окружающей среды и низкая контрастность изображения), устранены, а типы термоносителей включают в себя стандартную белую бумагу, кальку и даже полиэфирную пленку. Качество этих носителей удовлетворяет самым строгим архивным требованиям.

Учитывая их высокую надежность, производительность (может достигать 50 листов формата А0 в день) и низкие эксплуатационные затраты, плоттеры ПВИ применяют в крупных проектных организациях для вывода проверочных копий. В связи с этим в их стандартную конфигурацию входит сетевой адаптер. Технические характери-

стики ППВИ соответствуют требованиям прикладных задач инженерного проектирования, архитектуры, строительства, городского планирования и электросхемотехники.

Плоттеры на основе термопередачи (ПТП, thermal transferplotter). Отличие этих плоттеров от ППВИ состоит в том, что в них между термонагревателями и бумагой (или прозрачной пленкой) размещается «донорный цветоноситель» — тонкая, толщиной 5—10 мкм, лента (например, лавсановая), обращенная к бумаге красящим слоем, выполненным на восковой основе с низкой (менее 100 °С) температурой плавления.

На донорнои ленте последовательно нанесены области каждого из основных цветов размером, соответствующим листу используемого формата. В процессе вывода информации бумажный лист с наложенной на него донорнои лентой проходит под печатающей головкой, которая состоит из тысяч мельчайших нагревательных элементов. Воск в местах нагрева расплавляется, и пигмент остается на листе. За один проход наносится один цвет. Все изображение получается за четыре прохода. Таким образом, на каждый лист цветного изображения затрачивается в 4 раза больше красящей ленты, чем на лист монохромного.

Ввиду дороговизны каждого отпечатка эти плоттеры используются в составе средств автоматизированного проектирования для высококачественного вывода объектов трехмерного моделирования, в системах картографии, где требуется высокое качество воспроизведения цветов, и рекламными агентствами для вывода цветопроб плакатов и транспарантов для красочных презентаций.

Лазерные (светодиодные) плоттеры (ЛП, laser/led plotter). Эти плоттеры базируются на электрографической технологии, в основу которой положены физические процессы внутреннего фотоэффекта в светочувствительных полупроводниковых слоях селеносодержащих материалов и силовое воздействие электростатического поля. Промежуточный носитель изображения (вращающийся селеновый барабан) в темноте может быть заряжен до потенциала в сотни вольт. Луч света снимает этот заряд, создавая скрытое электростатическое изображение, которое притягивает намагниченный мелкодисперсный тонер, переносимый затем механическим путем на бумагу. После этого бумага с нанесенным тонером проходит через нагреватель, в результате чего частицы тонера запекаются, создавая изображение.

Некоторое время назад создание скрытого изображения на барабане осуществлялось исключительно с помощью лазера. Для управления перемещением лазерного луча служила сложная система

вращающихся зеркальных многогранников — призм либо линз. Вследствие этого плоттеры, использующие лазеры, боятся тряски и ударов, которые могут сбить настройку.

Избежать сложностей с оптикой и сделать систему проще, легче и надежнее позволило применение линеек точечных полупроводниковых светодиодов (light-emitting diode — LED).

Лазерные и LED-плоттеры ввиду высокого быстродействия (лист формата A1 выводится менее чем за полминуты) удобно использовать как сетевые устройства, и они имеют в стандартной комплектации адаптер сетевого интерфейса. Не менее важно и то, что эти плоттеры могут работать на обычной бумаге, что сокращает эксплуатационные затраты.

LED-плоттеры становятся все более популярными, хотя по стоимости сравнимы с монохромными электростатическими.

Дигитайзеры. Дигитайзер — это абсолютное устройство. Для выбора некоторой позиции на экране необходимо указателем дигитайзера выбрать соответствующую точную точку на планшете. Дигитайзер состоит из двух частей — планшета и наводчика (puck), или пера. Планшет — это плоский прямоугольник позиционирования, а наводчик или перо — устройство управления позицией. Наводчик похож на мышь и имеет перекрестную мишень выбора необходимой позиции на планшете и набор кнопок (табл. 2.10).

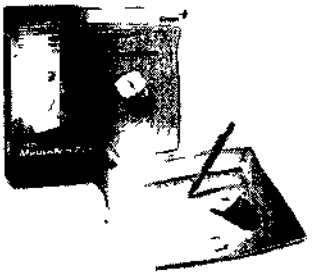
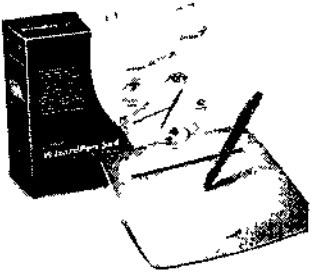
За каждой кнопкой как наводчика, так и пера можно закрепить определенные действия (COPY, SAVE и др.). Гибкость выполнения такого назначения зависит от программируемости драйвера устройства.

Планшет подключается к последовательному порту, а наводчик или перо — к планшету. Планшеты могут иметь различные размеры (форматы A2 или A3 и т. д.). Удобнее всего дигитайзеры применять в системах с программами CAD (Computer-Aided Design — система автоматизированного проектирования).

Часто с дигитайзером связывают командное управление в AutoCAD и аналогичных системах с помощью накладных меню. Команды меню расположены в разных местах на поверхности дигитайзера. При выборе курсором одной из них специальный программный драйвер интерпретирует координаты указанного места, посылая соответствующую команду на выполнение.

Не последнюю роль играет применение планшета в создании на компьютере рисунков и набросков. Художник рисует на экране, но его рука водит пером по планшету. Наконец, дигитайзер можно использовать просто как аналог мыши. Особый случай — это чувствительные к нажиму дигитайзеры.

Таблица 2 10. **Характеристики некоторых моделей дигитайзеров**

Марка	Характеристика	Общий вид устройства
Genius MousePen 8x6	Интерфейс USB Поддерживает Win 2003/XP/Me/2000/98 Планшет рабочая поверхность 8" x 6 , разрешение 1000 lpi (строк на дюйм) по умолчанию Перо беспроводное, количество кнопок - 2, чувствительность нажатия - 1024 уровней Мышь количество кнопок - 3 колеса прокрутки, беспроводная	
WizardPen 5x4	Устройство с самым обыкновенным пером, которым легко рисовать, чертить и подписывать документы Перо без провода с 512 уровнями чувствительности к нажиму предоставляет неограниченную свободу движений. Разрешение 4064 линий/дюйм и рабочее поле 5,5" × 4" Настраиваемая кнопка пера для быстрого просмотра вправо и влево, вверх и вниз WizardPen позволяет делать заметки от руки и чертежи в Internet и в любой другой прикладной программе	
CalComp DrawingBoard III	Серия широкоформатных дигитайзеров для САПР и ГИС Имеют модификации· со стандартной точностью (±0,2 мм), повышенной (±0,1 мм), высокой (±0,05 мм) Модели с высокой точностью комплектуются курсором с подсветкой рабочей зоны Типы указателей беспроводной, 4- или 16-кнопочный курсор Дигитайзеры по дополнительному заказу можно укомплектовать напольными подставками различных типов Модели формата А4-А3 Точность ± 0,25, разрешающая способность 100 строк/мм	

Принцип действия. Принцип действия дигитайзера основан на фиксации местоположения курсора с помощью встроенной в планшет сетки, состоящей из проволочных или печатных проводников с довольно большим расстоянием между ними (от 3 до 6 мм). Но механизм регистрации положения курсора позволяет получить шаг считывания информации намного меньше шага сетки (до 100 линий на миллиметр). Шаг считывания информации называется разрешением дигитайзера.

По технологии изготовления дигитайзеры делятся на два типа:

- электростатические (ЭС);
- электромагнитные (ЭМ).

В первом случае регистрируется локальное изменение электрического потенциала сетки под курсором. Во втором — курсор излучает электромагнитные волны, а сетка служит приемником. Фирма Wacom создала технологию на основе электромагнитного резонанса, когда сетка излучает, а курсор отражает сигнал. Но в обоих случаях приемником является сетка. Следует отметить, что при работе ЭМ-планшетов возможны помехи со стороны излучающих устройств, в частности мониторов.

Независимо от принципа регистрации существует погрешность в определении координат курсора, называемая точностью дигитайзера. Эта величина зависит от типа дигитайзера и от конструкции его компонент. На нее влияет неидеальность регистрирующей сетки планшета, способность воспроизводить координаты неподвижного курсора (повторяемость), устойчивость к разным температурным условиям (стабильность), качество курсора, помехозащищенность и прочие факторы. Точность существующих планшетов колеблется в пределах от 0,005 до 0,03 дюйма. В среднем точность электромагнитных дигитайзеров выше, чем у электростатических.

Шаг считывания регистрирующей сетки является физическим пределом разрешения дигитайзера. Мы говорим о пределе разрешения, потому что следует различать разрешение как характеристику прибора и как программно-задаваемое разрешение, а это переменная величина в настройке дигитайзера. В спецификации на изделие всегда указываются обе характеристики — предел разрешения и точность.

На результат работы также влияет точность действий оператора. В среднем хороший оператор вносит погрешность не более 0,004 дюйма. Требования к нему достаточно высокие.

Технологии чувствительных к нажиму дигитайзеров. В настоящее время есть две технологии, применяемые в чувствительных к нажиму дигитайзерах: первая — это *электромагнитный резонанс,* на основе которого работают дигитайзеры фирмы Wacom, позволяющий применять пассивное стило, а вторая — метод *активного курсора.*

При использовании электромагнитного резонанса излучающим (активным) устройством является сам дигитайзер. Перо отражает волны, а дигитайзер анализирует это отражение, для того чтобы установить координаты пера в данный момент. Поэтому перо или курсор не имеют ни батарей, ни шнура, подающего напряжение на

микросхемы внутри курсора, их там просто нет. При использовании же активного курсора именно он излучает волны, сообщая таким образом дигитайзеру о своем местоположении. В этом случае либо батареи, либо провод являются его неотъемлемым атрибутом. Но независимо от системы в обоих случаях информация о положении курсора относительно сетки, встроенной в поверхность дигитайзера, преобразуется в компьютере так, что мы получаем данные о точном положении курсора.

Для подключения планшета обычно используется последовательный порт. Распространенными параметрами являются разрешение порядка 2400 dpi и высокая чувствительность к уровням нажатия (256 уровней). Эта особенность позволяет моделировать нажатие на кисть или перо при работе с соответствующими графическими программами. Графические планшеты и дигитайзеры производят компании CalComp, Mutoh, Wacom и др.

Для устройств рукописного ввода информации характерна такая же схема работы, только введенные образы букв дополнительно преобразуются в буквы с помощью специальной программы распознавания, а размер площадки для ввода меньше. Устройства перьевого ввода информации чаще используются в сверхминиатюрных компьютерах PDA (Personal Digital Assistant) или HPC (Handheld PC), в которых нет полноценной клавиатуры.

Указующее устройство. Ранее при упоминании указующего устройства мы называли его *курсором,* хотя существует еще *перо* (или *стило).* Курсоры больше популярны в среде пользователей САПР. Перья в виде ручки производятся с одной, двумя и тремя кнопками. Кроме того, есть простые перья и перья, чувствительные к нажиму, которые особенно интересны для художников и аниматоров.

Выбирайте указующее устройство тщательно. Если курсор неудобен, то связанные с его использованием затраты составят гораздо большую сумму, чем разница в стоимости дорогих и дешевых дигитайзеров.

Курсоры бывают четырех-, восьми-, двенадцати- и шестнадцатикнопочными. Желая выделиться, некоторые фирмы стараются стать исключением из правила. Так, Oce Graphics добавляет на большом курсоре семнадцатую, «самую главную» кнопку. Форма курсора, легкость нажатия и расположение кнопок — вот в чем отличия. Во всем мире одними из лучших признаны четырехкнопочные курсоры фирмы CalComp. Их чаще прочих фотографируют и помещают в журналах. На них вторая и третья кнопки расположены рядом, а первая и четвертая L-образной формы обрамляют средние.

Традиционным же считается ромбовидное расположение кнопок, которому продолжают следовать другие известные производители. Однако для двенадцати- и шестнадцатикнопочных курсоров стандарт один — «табличное» расположение кнопок, как на телефонном аппарате.

При выборе курсора надо принимать во внимание; кроме удобства пользования, еще и количество клавиш на нем. Тот, кто работал в AutoCAD'e для DOS, знает, чтр чем больше на курсоре клавиш, тем лучше, потому что дополнительным кнопкам можно назначить одношаговые функции в файле AutoCAD MNU.

Однако для AutoCAD for Windows это не совсем так. Дело в том, что использование дополнительных, числом более трех, клавиш при работе в mole-режиме — непростая задача. Чтобы избежать проблем, лучше использовать специальные программы управления дигитайзером, которые часто входят в его комплект поставки. Но проще отказаться от курсора с большим количеством кнопок в пользу четырехкнопочного и задействовать только три его кнопки. В курсоре немаловажно также качество изготовления визира.

Перья. Как уже говорилось, перья производятся с одной, двумя и тремя кнопками. Кроме того, есть среди них чувствительные к нажиму, которые могут воспринимать до 256 градаций усилия нажима. Степени нажима ставят в соответствие или толщину линии, или цвет в палитре, или его оттенок. В результате можно имитировать на компьютере процесс рисования масляными красками, темперой или акварелью на специально подобранной «фактуре». Для реализации этих возможностей необходимо иметь специальное программное обеспечение. Среди подобных программ для персональных компьютеров можно упомянуть Adobe PhotoShop, Aldus PhotoStyler, Fauve Matisse, Fractal Design Painter, Autodesk Animator Pro, CorelDraw. Чувствительные к нажиму перья могут пригодиться и пользователям AutoCAD'a для последующей трехмерной визуализации спроектированных объектов. Данный вид указывающих устройств применяют только с ЭМ-дигитайзерами.

Удобство пера — характеристика сугубо субъективная, как и при выборе авторучки. Некоторым нравятся легкие перья фирмы Wacom, в то время как другие предпочитают более тяжелые, но хорошо сбалансированные перья от Kurta. И курсоры, и перья бывают как с проводом, так и без него. Беспроводный указатель удобнее, но он должен иметь батарейку, что утяжелит его и потребует дополнительного обслуживания.

Исключение составляют пассивные неизлучающие перья Wacom, которые воспринимают вдвое меньше градаций нажима. Не

так давно на рынке дигитайзеров появились предложения с модифицируемыми курсорами, которые могут работать и с проводом, и с батарейкой.

Мультимедиа оборудование. Мультимедиа — собирательное понятие для различных компьютерных технологий, при которых используется несколько информационных сред, таких, как графика, текст, видео, фотография, движущиеся образы (анимация), звуковые эффекты, высококачественное звуковое сопровождение.

Области применения мультимедиа:

- обучение с использованием компьютерных технологий (Специальными исследованиями установлено, что из услышанного в памяти остается только четверть, из увиденного — треть, при комбинированном воздействии зрения и слуха — 50 %, а если вовлечь учащегося в активные действия в процессе изучения с помощью мультимедийных приложений — 75 %);
- информационная и рекламная служба;
- развлечения, игры, системы виртуальной реальности.

Мультимедийные средства. Технологию мультимедиа составляют две основные компоненты — аппаратная и программная.

Аппаратные средства мультимедиа:

- основные — компьютер с высокопроизводительным процессором, оперативной памятью 256—512 Мбайт, НЖМД емкостью 40—100 Гбайт и выше, накопителем на гибких магнитных дисках, манипуляторами, мультимедиа-монитором со встроенными стереодинамиками и видеоадаптером SVGA;
- специальные — приводы CD-ROM; TV-тюнеры и фрейм-грабберы (frame-grabber — карта видеозахвата); графические акселераторы (в том числе для поддержки трехмерной графики); платы видеовоспроизведения, устройства для ввода видеопоследовательностей; звуковые платы с установленными микшерами и музыкальными синтезаторами, воспроизводящими звучание реальных музыкальных инструментов; акустические системы с наушниками или динамиками и др.;
- средства создания мультимедийных приложений — редакторы видеоизображений; профессиональные графические редакторы; средства для записи, создания и редактирования звуковой информации, позволяющие подготавливать звуковые файлы для включения в программы, изменять амплитуду сигнала, наложить или убрать фон, вырезать или вставить блоки данных на каком-то временном отрезке; программы для манипуляции с сегментами изображений, изменения цвета, палитры; программы для реализации гипертекстов и др.

Рассмотрим характеристики некоторых технических и программных средств.

Аудиоадаптер (Sound Blaster или звуковая плата) это специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

Аудиоадаптер содержит в себе два преобразователя информации:

- аналого-цифровой, который преобразует непрерывные (т. е. аналоговые) звуковые сигналы (речь, музыку, шум) в цифровой двоичный код и записывает его на магнитный носитель;
- цифроаналоговый, выполняющий обратное преобразование сохраненного в цифровом виде звука в аналоговый сигнал, который затем воспроизводится с помощью акустической системы, синтезатора звука или наушников.

Для записи звука к звуковой плате может быть подключен микрофон или устройство воспроизведения звука (магнитофон, CD-плейер). Для воспроизведения звука к ее выходу могут быть подключены акустические колонки или наушники, а также любая акустическая система (магнитофон, музыкальный центр и т. д.).

Профессиональные звуковые платы позволяют выполнять сложную обработку звука, обеспечивают стереозвучание, имеют собственное ПЗУ с хранящимися в нем сотнями тембров звучаний различных музыкальных инструментов. Звуковые файлы обычно имеют очень большие размеры. Так, трехминутный звуковой файл со стереозвучанием занимает примерно 30 Мбайт памяти. Поэтому платы Sound Blaster, помимо своих основных функций, обеспечивают автоматическое сжатие файлов.

Область применения звуковых плат — компьютерные игры, обучающие программные системы, рекламные презентации, «голосовая почта» (voice mail) между компьютерами, озвучивание различных событий, происходящих в компьютерном оборудовании, таких, например, как отсутствие бумаги в принтере и т. п.

Программный продукт Magix 2004 MP3 Maker может реализовывать следующие преобразования звуковых данных:

- дорожка аудио-CD — файл MP3;
- микрофонный или линейный вход — файл MP3;
- файл MP3 — аудио-CD и пр.

Видеоадаптер — это электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Содержит видеопамять, регистры ввода вывода и модуль BIOS. Посы-

лает в дисплей сигналы управления яркостью лучей и сигналы развертки изображения.

Наиболее распространенный видеоадаптер на сегодняшний день — адаптер SVGA (Super Video Graphics Array — супервидеографический массив), который может отображать на экране дисплея 1280 x 1024 пикселей при 256 цветах и 1024 x 768 пикселей при 16 миллионах цветов.

С увеличением числа приложений, использующих сложную графику и видео, наряду с традиционными видеоадаптерами широко используются разнообразные устройства компьютерной обработки видеосигналов:

Графические акселераторы (ускорители) — специализированные графические сопроцессоры, увеличивающие эффективность видеосистемы. Их применение освобождает центральный процессор от большого объема операций с видеоданными, так как акселераторы самостоятельно вычисляют, какие пиксели отображать на экране и каковы их цвета.

Фрейм-грабберы — устройства, которые позволяют отображать на экране компьютера видеосигнал от видеомагнитофона, камеры, лазерного проигрывателя и т. п., с тем, чтобы захватить нужный кадр в память и впоследствии сохранить его в составе видеофайла.

TV-тюнеры — видеоплаты, превращающие компьютер в телевизор. TV-тюнер позволяет выбрать любую нужную телевизионную программу и отображать ее на экране в масштабируемом окне. Таким образом можно следить за ходом передачи, не прекращая работу.

Технологии мультимедиа. Перечислим основные технологические приемы работы с мультимедийными данными:

- телевизионный прием — вывод телевизионных сигналов на монитор компьютера на фоне работы других программ;
- видеозахват — «захват» и «заморозка» в цифровом виде отдельных видеокадров;
- анимация — воспроизведение последовательности картинок, создающее впечатление движущегося изображения;
- звуковые эффекты — сохранение в цифровом виде звучания музыкальных инструментов, звуков природы или музыкальных фрагментов, созданных на компьютере, либо записанных и оцифрованных;
- трехмерная (3D) графика — графика, создаваемая с помощью изображений, имеющих не только длину и ширину, но и глубину;
- музыка MIDI (Musical Instrument Digital Interface — цифровой интерфейс музыкальных инструментов) — стандарт, позво-

ляющий подсоединять к компьютеру цифровые музыкальные инструменты, используемые при сочинении и записи музыки;

- виртуальная реальность (Virtual Reality, VR). Слово «виртуальный» означает «действующий и проявляющий себя как настоящий». Виртуальная реальность — это высокоразвитая форма компьютерного моделирования, которая позволяет пользователю погрузиться в модельный мир и непосредственно действовать в нем. Зрительные, слуховые, осязательные и моторные ощущения пользователя при этом заменяются их имитацией, генерируемой компьютером.

Признаки устройств виртуальной реальности: моделирование в реальном масштабе времени; имитация окружающей обстановки с высокой степенью реализма; возможность воздействовать на окружающую обстановку и иметь при этом обратную связь.

Средства интерактивного взаимодействия (ввод-вывод данных и управление компьютером)

Понятие *терминала* (DTE — оконечное оборудование данных) в соответствии с телекоммуникационными стандартами относится к сочетанию устройств ввода и вывода информации (например, сканер и принтер и т. п.), однако чаще всего под терминалом понимается оконечное устройство ЭВМ, предназначенное для диалога «человек—машина». Узко специализированные устройства — банкоматы, кассовые аппараты со сканерами штрих-кода — здесь не рассматриваются.

Терминалы, или *терминальные устройства ЭВМ,* являются важнейшей компонентой систем, основанных на человеко-машинном взаимодействии. Это *диалоговые* или *интерактивные* устройства, предназначенные для ввода/вывода небольших количеств информации, первоначально с целью *управления* вычислительным процессом и наблюдения за его ходом, а в дальнейшем также для ввода-вывода *исходных* данных и *результатов* работы программ.

Механические терминалы. Первоначально в ЭВМ использовались в качестве терминалов *механические* устройства, заимствованные из смежных технологий — связь и оргтехника — телетайпы (типа ТА-67), телеграфные аппараты (СТА-2М), электрические пишущие машинки (ПМ типа CONSUL). Это был довольно длительный период, в течение которого сложились определенные стандарты, приемы работы оператора и протоколы ввода/вывода и интерпретации данных. Строка информации, вводимая оператором,

являлась, как правило, *командой,* требующей выполнения определенных действий от ЭВМ (ОС). Конечная ширина листа (или бумажной ленты) ПМ (80 знаков) ограничивала длину возможных команд. Признаком окончания ввода команды являлось нажатие клавиши <ВК> *(возврат каретки,* она же <CR> — <Carriage Return>, <Return>, <Enter> и пр.). Реакция системы (ответ на запрос, сообщение об ошибке, небольшая порция выходных данных) также выводилась строками по 80 символов, образуя вместе с копиями команд *протокол диалогового сеанса* (или журнал — log) в бумажной форме.

Низкие скорость обмена информацией с ЭВМ и надежность механических терминалов, а также трудности с исправлением информации (редактированием) ограничивали их применимость и, в частности, делали бессмысленным их использование пользователями-программистами для отладки программ и прочих манипуляций.

Электронные терминалы. Физически электронный или *видеотерминал* — CRT-device (Catode Ray Tube — устройство с электронно-лучевой трубкой), VDU (Video Display Unit — устройство отображения информации), первоначально получивший в отечественной практике наименование *дисплей,* представляет собой клавиатуру (keyboard), сопряженную с экранным устройством (screen). Ранние модели видеотерминалов (ВТ) не были избавлены от наследия ПМ — состав клавиатуры, построчный ввод и исправление ошибок, прокручивание экрана наподобие бумажной ленты (scrolling) и, самое главное, — *символьный* (алфавитно-цифровой) характер выводимой информации, хотя, как это хорошо известно из опыта телевидения, никаких технических ограничений экран (в отличие от каретки ПМ) не вносит. Более совершенные ВТ, разработанные в 80-е гг. (IBM 3270, VT-100), во многом определили современное состояние устройств:

- появились возможности прямого доступа к информации на экране (для ввода и корректировки);
- на клавиатуре добавились *функциональные* клавиши, реакция на которые определялась программой, работающей с ВТ;
- *клавиши редактирования* — <Del>, <Ins>;
- *клавиши управления курсором* (для выбора места на экране);
- *управляющая клавиша* <Control> (<Ctrl>), модифицирующая вводимый код, при одновременном нажатии с символьной клавишей и т. п.

Однако это все еще были алфавитно-цифровые устройства, отображающие на экране массив символьной информации размером в 80 столбцов на 17 строк (т. е. до 1600 символов).

Типовая конфигурация машины (до появления ПЭВМ) включала в себя 8 (или 16, или 32) *терминалов пользователя,* размещенных в специальных помещениях *(дисплейные классы)* и одну или более *дисплей-консоль* (терминал оператора), размещенную поближе к месту основных событий (в машинном зале).

Клавиатура терминала — устройство для ввода информации в компьютер и подачи управляющих сигналов Содержит стандартный набор клавиш печатной машинки и некоторые дополнительные клавиши — управляющие и функциональные клавиши, клавиши управления курсором и малую цифровую клавиатуру.

Все символы, набираемые на клавиатуре, немедленно отображаются на мониторе в позиции курсора *(курсор* — светящийся символ на экране монитора, указывающий позицию, на которой будет отображаться следующий вводимый с клавиатуры знак).

Клавиатура содержит встроенный *микроконтроллер* (местное устройство управления), который выполняет следующие функции:

- последовательно опрашивает клавиши, считывая введенный сигнал и вырабатывая двоичный скан-код клавиши;
- управляет световыми индикаторами клавиатуры;
- проводит внутреннюю диагностику неисправностей;
- осуществляет взаимодействие с центральным процессором через порт ввода-вывода клавиатуры.

Клавиатура имеет встроенный буфер — промежуточную память малого размера, куда помещаются введенные символы. В случае переполнения буфера нажатие клавиши будет сопровождаться звуковым сигналом — это означает, что символ не введен (отвергнут). Работу клавиатуры поддерживают специальные программы, «зашитые» в BIOS, а также драйвер клавиатуры, который обеспечивает возможность ввода русских букв, управление скоростью работы клавиатуры и др.

Клавиатуры с дополнительными функциональными возможностями. Существуют клавиатуры, отличающиеся от стандартных дополнительными функциональными возможностями. Они могут быть как простыми (со встроенными калькулятором и часами), так и сложными (со встроенными устройствами позиционирования (манипуляторами), особой раскладкой или формой и возможностью перепрограммирования клавиш).

Примером может являться Elegance 5000 — мультимедийная модель, содержащая дополнительно четырнадцать кнопок в ряду над стандартными функциональными клавишами (рис. 2.47). Самая крупная отвечает за включение и выключение ATX-компьютера. Мультимедийные кнопки выполняют типичные операции: умень-

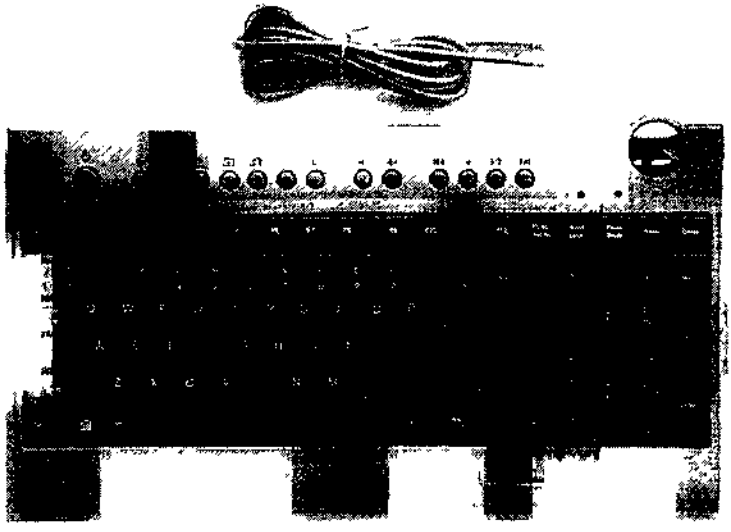


Рис. 2.47. Комбинированная мультимедийная клавиатура Elegance 5000

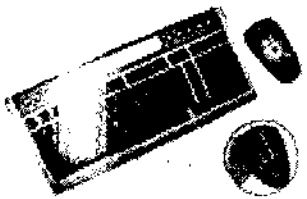
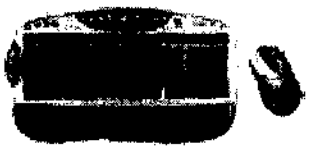
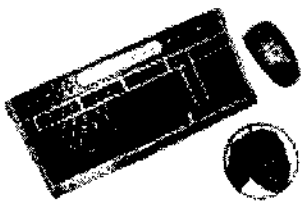
шение и увеличение громкости звука (в микшере Windows), пуск/пауза воспроизведения звука, стоп, трек вперед и трек назад для CD/DVD/MP3 программных плейеров. Интернет/офисные кнопки выполняют быстрый вызов браузера, почтовой программы, поиска файлов, избранного калькулятора, а также переход на предыдущий и последующий просмотренные сайты. Все четырнадцать дополнительных кнопок работают только в Windows 2000/XP.

Беспроводные клавиатуры В последнее время большинством производителей выпускается новый тип клавиатур — беспроводные. Такая клавиатура содержит инфракрасный или радиопередатчик, а приемник с помощью кабеля подключается к стандартному разъему клавиатуры системной платы. Естественно такая клавиатура существенно дороже стандартной и чаще всего используется в домашних системах (см., например, табл. 2 11).

Монитор (устройство отображения). Монитор — устройство визуального отображения информации (в виде текста, таблиц, рисунков, чертежей и др.)

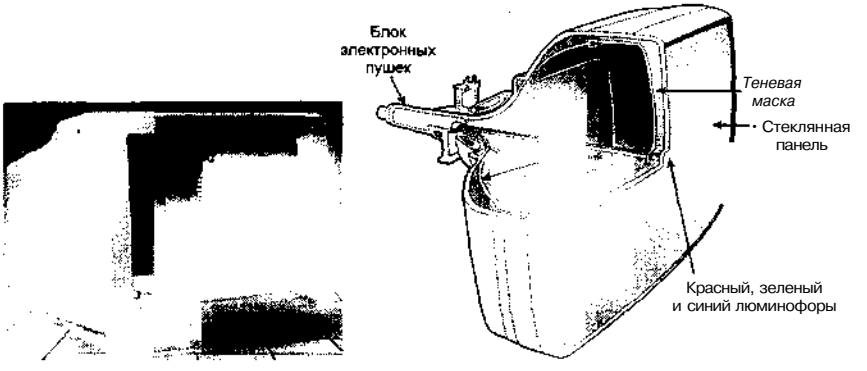
Подавляющее большинство мониторов сконструированы на базе электронно-лучевой трубки (ЭЛТ), и принцип их работы аналогичен принципу работы телевизора Мониторы бывают алфавитно-цифровые и графические, монохромные и цветного изображения Современные компьютеры комплектуются, как правило, цветными графическими мониторами

Таблица 2.11. **Некоторые примеры беспроводных клавиатур**

№	Вид клавиатуры	Описание
1		Беспроводный набор A4-Tech KBS-1527 R, PS/2, беспроводная клавиатура RSI, беспроводная механическая мышь
2		Беспроводный набор M-Tech KBS-2350 RP, USB, беспроводная клавиатура RSI + беспроводная оптическая мышь, зарядное устройство в приемнике
3		Беспроводный набор M-Tech KBS-2548 RP, USB, беспроводная клавиатура RSI + беспроводная оптическая мышь, зарядное устройство в приемнике

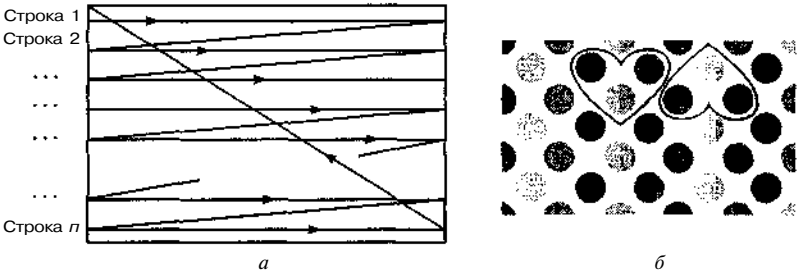
Используемая в этом типе мониторов технология была разработана много лет назад и первоначально создавалась в качестве специального инструментария для измерения переменного тока, т. е. для осциллографа.

Мониторы на основе ЭЛТ. Принцип действия таких устройств заключается в том, что испускаемый электронной пушкой пучок электронов, попадая на экран, покрытый специальным веществом, люминофором, вызывает его свечение. Конструктивно ЭЛТ-монитор представляет собой стеклянную трубку, внутри которой находится вакуум (рис. 2.48). С фронтальной стороны внутренняя часть стекла трубки покрыта люминофором. В качестве люминофоров для цветных ЭЛТ используются довольно сложные составы на основе редкоземельных металлов — иттрия, эрбия и др. Люминофор — это вещество, которое испускает свет при бомбардировке его заряженными частицами. Для создания изображения в ЭЛТ-мониторе используется электронная пушка, которая испускает поток электронов сквозь металлическую маску или решетку на внутреннюю поверхность стеклянного экрана монитора, которая покрыта разноцветными люминофорными точками. Электроны попадают на люминофорный



а б

Рис. 2.48. Общий вид (а) и схематическое устройство (б) ЭЛТ-монитора



а *б*

Рис. 2.49. Ход электронного пучка по экрану (*а*), пиксельные триады (б)

слой, после чего энергия электронов преобразуется в свет, т. е. поток электронов заставляет точки люминофора светиться. Эти светящиеся точки люминофора формируют изображение на мониторе (рис. 2.49).

Мониторы подразделяются на монохромные (Monochrome или Mono) и цветные (Colour или Color). Монохромные мониторы могут быть как черно-белыми, так и черно-зелеными или черно-желтыми. Люминофор с желтым и зеленым свечением применялся в первых мониторах, предназначенных для адаптеров MDA и HGC. Эти мониторы обеспечивали передачу лишь трех уровней градации яркости и имели довольно длительное послесвечение.

Плоскопанельные мониторы. Мониторы на основе ЭЛТ в настоящее время являются наиболее распространенными, однако они обладают рядом недостатков: значительные масса, габариты и энергопотребление; наличие тепловыделения и излучения, вредного для здо-

ровья человека. В связи с этим на смену ЭЛТ-мониторам постепенно приходят плоскопараллельные мониторы: жидкокристаллические — ЖК-мониторы, плазменные, электролюминесцентные, мониторы электростатической эмиссии, органические светодиодные мониторы.

• ЖК-мониторы (LCD — Liquid Crysta Display) составляют основную долю рынка плоскопанельных мониторов с экраном размером 13—17". Первое свое применение жидкие кристаллы нашли в дисплеях для калькуляторов и кварцевых часах, а затем их стали использовать в мониторах для портативных компьютеров. Сегодня в результате прогресса в этой области начинают получать все большее распространение LCD-мониторы для настольных компьютеров (рис. 2.50).

Рис. 2.50. Монитор с жидкокристаллической панелью

Основным элементом ЖК-монитора является ЖК-экран, состоящий из двух панелей, выполненных из стекла, между которыми размещен слой жидкокристаллического вещества, которое находится в жидком состоянии, но при этом обладает некоторыми свойствами, присущими кристаллическим телам. Фактически это жидкости, обладающие анизотропией (в частности оптической), связанной с упорядоченностью в ориентации молекул. Молекулы жидких кристаллов под воздействием электричества могут изменять свою ориентацию и вследствие этого изменять свойства светового луча, проходящего сквозь них. Следовательно, формирование изображения в ЖК-мониторах основано на взаимосвязи между изменением электрического напряжения, приложенного к жидкокристаллическому веществу, и изменением ориентации его молекул.

Экран ЖК-монитора представляет собой массив отдельных ячеек (называемых пикселями), оптические свойства которого могут меняться при отображении информации. Панели ЖК-монитора имеют несколько слоев, среди которых ключевую роль играют две панели, выполненные из свободного от натрия и очень чистого

стеклянного материала, между которыми и расположен тонкий слой жидких кристаллов. На панели нанесены параллельные бороздки, вдоль которых ориентируются кристаллы. Бороздки на подложках перпендикулярны между собой. Технология получения бороздок состоит в нанесении на стеклянную поверхность тонких пленок из прозрачного пластика. Соприкасаясь с бороздками, молекулы в жидких кристаллах ориентируются во всех ячейках одинаково.

Жидкокристаллическая панель освещается источником света (в зависимости от того, где он расположен, жидкокристаллические панели работают на отражение или на прохождение света). В качестве источников света используются специальные электролюминесцентные лампы с холодным катодом, характеризующиеся низким энергопотреблением. Молекулы одной из разновидностей жидких кристаллов в отсутствие напряжения на подложках поворачивают вектор электрической напряженности электромагнитного поля в световой волне, проходящей через ячейку, на некоторый угол в плоскости, перпендикулярной оси распространения пучка. Нанесение бороздок позволяет обеспечить одинаковые углы поворота для всех ячеек. Фактически каждая ЖК-ячейка представляет собой электронно-управляемый светофильтр, принцип действия которого основан на эффекте поляризации световой волны.

Чтобы поворот плоскости поляризации светового луча был заметен для глаза, на стеклянные панели дополнительно наносят два слоя, представляющих собой поляризационные фильтры. Эти фильтры выполняют функции поляризатора и анализатора.

Принцип действия ячейки ЖК-монитора состоит в следующем. При отсутствии напряжения между подложками ячейка ЖК-монитора прозрачна, поскольку вследствие перпендикулярного расположения бороздок на подложках и соответствующему закручиванию оптических осей жидких кристаллов вектор поляризации света поворачивается, и проходит без изменения через систему поляризатор-анализатор. Ячейки, у которых ориентирующие канавки, обеспечивающие соответствующее закручивание молекул жидкокристаллического вещества, расположены под углом 90°, называются «твистированными тематическими». При приложении между подложками напряжения 3—10 В молекулы жидкокристаллического вещества расположатся параллельно силовым линиям поля. «Твистированная структура» жидкокристаллического вещества нарушается и поворота плоскости поляризации проходящего через него света не происходит. В результате плоскость поляризации света не совпадает с плоскостью поляризации анализатора, и ЖК-ячейка оказывается непрозрачной.

• Другие типы плоскопараллельных дисплеев.

Плазменные дисплеи (Plasma Display Panel PDF) создаются путем заполнения пространства между двумя стеклянными поверхностями инертным газом, например аргоном или неоном. Затем на стеклянную поверхность наносят миниатюрные прозрачные электроды, на которые подается высокочастотное напряжение. Под действием этого напряжения в прилегающей к электроду газовой области возникает электрический разряд. Плазма газового разряда излучает свет в ультрафиолетовом диапазоне, который вызывает свечение частиц люминофора в диапазоне, видимом человеком. Фактически каждый пиксель на экране работает как обычная лампа дневного света. Высокая яркость и контрастность наряду с отсутствием дрожания являются важнейшими преимуществами таких мониторов. Кроме того, угол по отношению к нормали, под которым можно увидеть изображение на плазменных мониторах, существенно больше, чем у ЖК-мониторов.

Основными недостатками такого типа мониторов является достаточно высокая потребляемая мощность, возрастающая при увеличении диагонали монитора, и низкая разрешающая способность (не более 1024 x 768), обусловленная большим размером элемента изображения. Кроме этого, свойства люминофорных элементов со временем ухудшаются, и экран становится менее ярким, поэтому срок службы плазменных мониторов ограничен 10 000 ч, что составляет около 5 лет при интенсивном использовании. Из-за этих ограничений, такие мониторы используют пока только для конференций, презентаций, информационных щитов, т. е. там, где требуются большие размеры экранов для отображения информации. Такие крупнейшие производители, как Fujitsu, Matsushita, Mitsubishi, NEC, Pioneer и другие, начали производство плазменных мониторов с диагональю 40" и более.

Плазменные панели гораздо чаще используются как экраны для коллективного просмотра изображения с одного и того же компьютера, чем как дисплей для персональной ЭВМ.

Электролюминесцентные мониторы (ElectricLuminiescent displays ELD) по своей конструкции аналогичны ЖК-мониторам. Принцип действия электролюминесцентных мониторов основан на явлении испускании света при возникновении туннельного эффекта в полупроводниковом *p-n*-переходе. Такие мониторы имеют высокие частоты развертки и яркость свечения, кроме того, они надежны в работе. Вместе с тем они уступают ЖК-мониторам по энергопотреблению, поскольку на ячейки подается относительно высокое напряжение — около 100 В. При ярком освещении цвета электролюминесцентных мониторов тускнеют.

Мониторы электростатической эмиссии (Field Emission Displays, FED) являются сочетанием традиционной технологии, основанной на использовании ЭЛТ, и жидкокристаллической технологии. Мониторы FED основаны на процессе, который несколько похож на тот, что применяется в ЭЛТ-мониторах, так как в обоих методах применяется люминофор, светящийся под воздействием электронного луча. В качестве пикселей применяются такие же зерна люминофора, как и в ЭЛТ-мониторе, что позволяет получить чистые и сочные цвета, свойственные обычным мониторам. Однако активизация этих зерен производится не электронным лучом, а электронными ключами, подобными тем, что используются в ЖК-мониторах, построенных по TFT-технологии. Управление этими ключами осуществляется специальной схемой, принцип действия которой аналогичен принципу действия контроллера ЖК-монитора.

Органические светодиодные мониторы (Organic Light-Emitting Diode displays, OLEDs), или *LEP-мониторы* (Light Emission Plastics — светоизлучающий пластик), по своей технологии похожи на ЖК- и ELD- мониторы, но отличаются материалом, из которого изготавливается экран: в LEP-мониторах используется специальный органический полимер (пластик), обладающий свойством полупроводимости. При пропускании электрического тока такой материал начинает светиться.

Основными преимуществами LEP-технологии по сравнению с рассмотренными ранее являются:

- низкое энергопотребление (подводимое к пикселу напряжение менее 3 В);
- простота конструкции и технологии изготовления;
- тонкий (около 2 мм) экран;
- малая инерционность (менее 1 мкс).

К существенным недостаткам этой технологии следует отнести: малую яркость свечения экрана; монохромность изображения (черно-желтые экраны); малый размер экрана. LEP-мониторы используются пока только в портативных устройствах, например в сотовых телефонных трубках.

Манипуляторы — мыши, трекболы. Различают относительные и абсолютные устройства. К относительным относятся мышь (mouse), трекбол (trackball), джойстик (joystick), точпад (touchpad). К абсолютным — дигитайзер (digitizer).

В *абсолютных манипуляторах* при перемещении указателя на некоторую точку получают ее представление в виде конкретной позиции экрана или конкретного выбора меню (например, если нужно выбрать центр экрана, т. е. курсор поместить в центр экрана мони-

тора, то достаточно переместить указатель дигитайзера на центр планшета).

В *относительном манипуляторе* нельзя указывать абсолютные позиции. Здесь перемещение экранного указателя на некоторое расстояние относительно его текущей позиции возможно получить перемещением указателя манипулятора на то же относительное расстояние. Например, при использовании мыши, если нужно передвинуть курсор в центр экрана, то, видя текущую позицию курсора, необходимо передвинуть мышь из текущей позиции в таком направлении, при котором курсор будет передвигаться к центру.

Манипулятор мышь. Одним из традиционных компьютерных устройств ввода является манипулятор мышь (mouse), в ранних советских ЭВМ фигурировавшая под названием «колобок». Это устройство было изобретено достаточно давно — еще в 1970-х гг. Самые первые серийные мыши выпускала корпорация Xerox.

По типу их устройств и способу функционирования мыши разделяются на механические, оптомеханические, оптические (рис. 2.51).

Механическая мышь — движение фиксируется механически и связано с перемещением частей устройства (у оптических мышей движение определяется оптически). Внутри корпуса располагается довольно тяжелый обрезиненный металлический шарик, который при перемещении мыши по поверхности стола перекатывается внутри корпуса. Два ролика, соприкасающиеся с этим шариком, монтируются под углом 90° относительно друг друга и также вращаются вокруг своих осей.

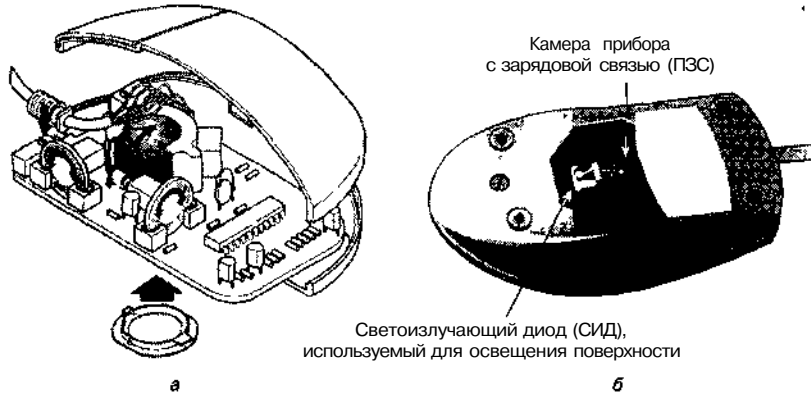


Рис. 2.51. Принцип функционирования оптомеханической *(а)* и оптической *(б)* мыши

Фактически ролики преобразуют произвольное движение шарика в движение в двух взаимно перпендикулярных направлениях (*X* и *Y*). Электронная схема, размещенная внутри корпуса, преобразует вращения роликов в электрические импульсы, передаваемые в ПК с помощью кабеля («хвоста» мыши). Кроме того, указанная электронная схема соответствующим образом реагирует на нажатие кнопок мыши. Такие мыши не очень долговечны и тяжелее перемещаются, поэтому на сегодняшний момент их выпуск прекращен.

Оптомеханическая мышь. Манипулятор в целом напоминает первый тип, но движение шарика отслеживается с помощью двух валиков с прорезями (горизонтального и вертикального) и двух оптических пар светодиод—фотодиод. В результате на оптопаре образуются импульсы, которые затем с помощью счетчиков конвертируются в числовые величины, обозначающие величину относительного перемещения мыши по горизонтальной и вертикальной осям. Эти величины вместе с состоянием кнопок мыши (нажата/отжата) передаются в ЭВМ. Для защиты обоих типов манипуляторов от проникновения пыли и грязи сквозь окошечко для шарика под мышь подкладывают специальные коврики (Mouse Pad).

Оптическая мышь. Внутри корпуса находятся две пары светодиодов и фотоэлементов (фотоэлементная пара). Один светодиод обычно излучает красный свет, а другой — инфракрасный. Фотоэлемент улавливает свет определенной частоты (один фотоэлемент мыши улавливает красный свет, а другой — инфракрасный). Светодиоды монтируются под углом к фотоэлементу. Для работы с этой мышью применяется специальный коврик. Он серебристого цвета и покрывается тонкой сеткой, состоящей из цветных горизонтальных (синего цвета) и вертикальных (серого цвета) линий.

Когда мышь устанавливается между линиями сетки, блестящая поверхность коврика отражает красные и инфракрасные лучи из светодиодов, а фотоэлементы улавливают эти лучи. При помещении мыши на синюю линию красный цвет поглощается, и чувствительный к красному свету элемент утрачивает сигнал. Аналогично при перемещении мыши на серую линию инфракрасный цвет поглощается, и сигнал на инфракрасном фотоэлементе теряется. При перемещении мыши по коврику фотоэлементы поочередно обнаруживают соответствующие им источники света. Следовательно, сигнал из чувствительного к красному цвету фотоэлемента представляет движение в направлении оси *X* (другой — оси *Y*). Эти сигналы передаются в ПК, где драйвер их использует для управления движением курсора на экране. В остальном все происходит, как в механической мыши.

«Бесхвостые» мыши (инфракрасные) для передачи сигналов используют приемник инфракрасного излучения, который кабелем соединяется с ПК и располагается или на ПК, или устанавливается где-то рядом (при этом нельзя загораживать излучатель такой мыши посторонними предметами). Преимуществом является свободное передвижение мыши.

Альтернативой является передача информации от мыши посредством радиосигнала,

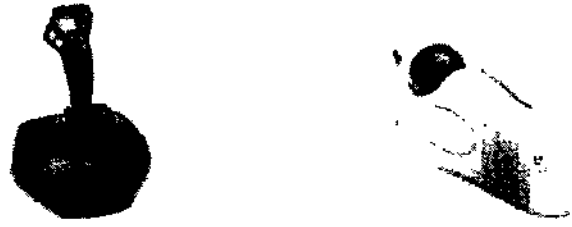
Джойстик (от англ. joy stick — веселая палочка) — обычно это стержень-ручка (рис. 2.52, *а),* применяемая для летных имитаторов или для игр, в которых оживленные объекты должны точно позиционироваться путем изменения положения ручки (влево, вправо, вверх, вниз, направо или налево вполоборота, наискосок вниз или вверх), и имеется кнопка со статусом «огонь».

В некоторых моделях в джойстик монтируется датчик давления. В этом случае, чем сильнее пользователь нажимает на ручку, тем быстрее движется курсор по экрану дисплея.

Трекбол — небольшая коробка с шариком, встроенным в верхнюю часть корпуса (рис. 2.52, *б).* Пользователь рукой вращает шарик и перемещает, соответственно, курсор. В отличие от мыши, трекбол не требует свободного пространства около компьютера, его можно встроить в корпус машины.

Известно, что подобный манипулятор использовался в советских зенитно-ракетных комплексах «Байкал», правда, диаметр шарика этого трекбола был около 8 см. На сегодняшний момент такие манипуляторы практически не используются, и даже из традиционных для них портативных компьютеров их практически вытеснили манипуляторы на основе сенсорных панелей Touch Pad (в них управление курсором идет за счет перемещения пальца по панели).

IBM производит устройство, называемое Tracpoint, которое может использоваться и как мышь (шариком вниз), и как Trackball (ша-



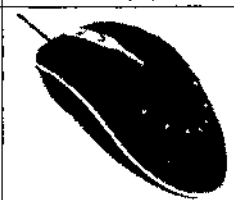
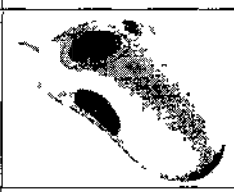
а б

Рис. 2.52. Джойстик *(а)* и трекбол *(б)*

риком вверх). В большинстве случаев в Trackball установлен шарик гораздо большего размера, чем в стандартной мыши. С точки зрения дизайна Trackball идентичен мыши по базовым функциям и электрической «начинке», но отличается ориентацией и размером шарика.

В табл. 2.12 приведены параметры некоторых современных образцов манипуляторов мышь и трекбол. Курсивом выделены особенности каждого из образцов.

Таблица 2.12. **Характеристики некоторых манипуляторов Defender и A4 Tech**

Марка	Общий вид устройства	Характеристика
DEFENDER M 2340 L Pantera		Эргономичная проводная оптическая мини-мышь со *светящимся колесом прокрутки*. Мышь имеет две кнопки и одну кнопку-колесо для быстрой прокрутки документов. Разрешение 400 dpi Разъем PS/2
MTech RP-680 UP, USB + PS/2 (new!)		Оптическая беспроводная мышь с семью кнопками и одной кнопкой-колесом. *Эргономная форма позволяетработать мышью как левой, так и правой рукой* Рабочая частота. 27 МГц Питание: 2 аккумулятора AA. Зарядное устройство встроено в приемник сигнала. Разрешение 800 dpi Радиус действия: 2 м от приемника. Разъем: PS/2, USB
A4Tech WWT-13		Трекбол с 2 колесиками для прокрутки документов и тремя кнопками, одна из которых (*кнопка для большого пальца)* может быть запрограммирована для выполнения специальных команд. Скорость прокрутки колеса может быть изменена. Разрешение. 520 dpi. Разъем: PS/2, COM, Combo и USB
M-Tech SWOP-50Z UP, USB + PS/2(new!)		Оптическая мышь Имеет четыре кнопки, *две из которых* (*Zoom-кнопки*) используются для масштабирования документов, и одну кнопку-колесо для прокрутки документов. Разрешение: 800 dpi. Разъем: PS/2 + USB

Контрольные вопросы

1. **Когда и на основании чего фон Нейман предложил новые принципы создания компьютеров?**
2. **Объясните сущность принципов, предложенных фон Нейманом. Какие функции возложены на ЦУ?**

3. Перечислите основные типы архитектуры ЭВМ.
4. Что такое процессор и АЛУ?
5. Какие типы АЛУ вам известны?
6. Что такое регистры? Назовите некоторые важные регистры и опишите их функции.
7. Назовите две основные разновидности памяти компьютера.
8. Перечислите основные компоненты внутренней памяти.
9. Что представляет собой ОЗУ? Каково его назначение?
10. В чем различие между статической и динамической памятью?
11. В чем суть принципа однородности памяти? Какие возможности он открывает?
12. В чем заключается принцип адресности?
13. Что такое команда? Что описывает команда?
14. Какого рода информацию может содержать адресная часть команды?
15. Приведите примеры команд одноадресных, двухадресных, трехадресных.
16. Каким образом процессор при выполнении программы осуществляет выбор очередной команды?
17. Опишите основной цикл процесса обработки команд.
18. Чем отличаются функции СЧАК и РК?
19. Что такое индексирование и базирование?
20. Что такое регистровая память?
21. Какие типы процессоров вам известны?
22. Что такое RISC-процессоры?
23. Что такое система команд?
24. В чем заключается технология конвейерной обработки?
25. Что такое динамическое исполнение?
26. Какие уровни памяти вам известны?
27. Что такое «прямой» и «обратный» порядок байтов?
28. Какие типы внутренних интерфейсов вам известны?
29. Какие типы внешних интерфейсов вы знаете?
30. Дайте сравнительные характеристики интерфейсов USB и IEEE 1384.
31. Опишите сходство и различие принципов записи на МЛ и МД.
32. Какие характеристики НЖМД вам известны?
33. В чем различие принципов записи-воспроизведения информации в случае CD и магнитооптики?
34. Опишите различные типы струйных принтеров.
35. Опишите различные типы лазерных принтеров.
36. Дайте классификацию сканеров.
37. Дайте классификацию плоттеров.
38. Какие интерактивные устройства вам известны?

Глава 3
ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Вычислительная система (ВС) — совокупность взаимосвязанных и взаимодействующих процессоров или ЭВМ, периферийного оборудования и программного обеспечения, предназначенная для сбора, хранения, обработки и распределения информации.

Создание ВС преследует следующие основные цели:

- повышение производительности системы за счет ускорения процессов обработки данных;
- повышение надежности и достоверности вычислений;
- предоставление пользователям дополнительных сервисных услуг и т. д.

Отличительной особенностью ВС по отношению к классическим ЭВМ является наличие в ней нескольких вычислителей, реализующих *параллельную обработку*.

Параллелизм выполнения операций существенно повышает быстродействие системы; он может также значительно повысить и надежность (при отказе одного компонента системы его функции может взять на себя другой), и достоверность функционирования системы, если операции будут дублироваться, а результаты их выполнения сравниваться.

Параллелизм в вычислениях в значительной степени усложняет управление вычислительным процессом, использование технических и программных ресурсов. Эти функции выполняет операционная система ВС.

Несмотря на то, что классическим является *многомашинный* вариант ВС, в ВС может быть только один компьютер, но агрегированный с многофункциональным периферийным оборудованием (стоимость периферийного оборудования часто во много раз превосходит стоимость центральных устройств компьютера). В компьютере может быть как несколько процессоров (тогда имеет место также классический многопроцессорный вариант ВС), так и один процессор (если не брать в расчет специализированные процессоры, входящие в состав периферийных устройств).

3.1. Основные определения. Классы архитектур вычислительных систем

Если не вдаваться в подробности, вычислительные системы (ВС) прежде всего можно различать, как:

- многомашинные;
- многопроцессорные.

Многомашинная вычислительная система. Здесь несколько процессоров, входящих в вычислительную систему, не имеют общей оперативной памяти, а имеют каждый свою (локальную). Каждый компьютер в многомашинной системе имеет классическую архитектуру, и такая система применяется достаточно широко. Однако эффект от применения такой вычислительной системы может быть получен только при решении задач, имеющих очень специальную структуру: она должна разбиваться на столько слабо связанных подзадач, сколько компьютеров в системе.

Многопроцессорная архитектура. Наличие в компьютере нескольких процессоров означает, что параллельно может быть организовано много потоков данных и много потоков команд. Таким образом, параллельно могут выполняться несколько фрагментов одной задачи. Структура такой машины, имеющей общую оперативную память и несколько процессоров, представлена на рис. 3.1. Преимущество в быстродействии многопроцессорных и многомашинных вычислительных систем перед однопроцессорными очевидно.

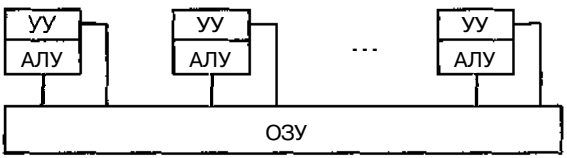


Рис. 3.1. Архитектура многопроцессорной ВС

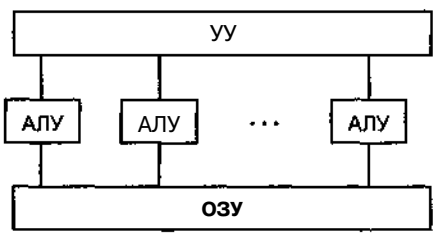


Рис. 3.2. Архитектура с параллельными процессорами

Архитектура с параллельными процессорами. Здесь несколько АЛУ работают под управлением одного УУ. Это означает, что множество данных может обрабатываться по одной программе, т. е. по одному потоку команд. Высокое быстродействие такой архитектуры можно получить только на задачах, в которых одинаковые вычислительные операции выполняются одновременно на различных однотипных наборах данных. Структура таких компьютеров представлена на рис. 3.2.

Уровни и средства комплексирования. Логические и физические уровни

В создаваемых ВС стараются обеспечить несколько путей передачи данных, что позволяет достичь необходимой надежности функционирования, гибкости и адаптируемости к конкретным условиям работы. Эффективность обмена информацией определяется скоростью передачи и возможными объемами данных, передаваемыми по каналу взаимодействия. Эти характеристики зависят от средств, обеспечивающих взаимодействие модулей, и уровня управления процессами, на котором это взаимодействие осуществляется. Сочетание различных уровней и методов обмена данными между модулями ВС наиболее полно представлено в универсальных супер-ЭВМ и больших ЭВМ, в которых сбалансированно использовались основные методы достижения высокой производительности. В этих машинах предусматривались следующие уровни комплексирования (рис. 3.3):

1) прямого управления (процессор — процессор);
2) общей оперативной памяти;
3) комплексируемых каналов ввода-вывода;
4) устройств управления внешними устройствами (УВУ);
5) общих внешних устройств.

На каждом из этих уровней используются специальные технические и программные средства, обеспечивающие обмен информацией.

Уровень прямого управления служит для передачи коротких однобайтовых приказов-сообщений. Последовательность взаимодействия процессоров сводится к следующему. Процессор-инициатор обмена по интерфейсу прямого управления передает в блок прямого управления байт-сообщение и подает команду «прямая запись». У другого процессора эта команда вызывает прерывание, относящееся к классу внешних. В ответ он вырабатывает команду «прямое чтение» и записывает передаваемый байт в свою память. Затем принятая информа-

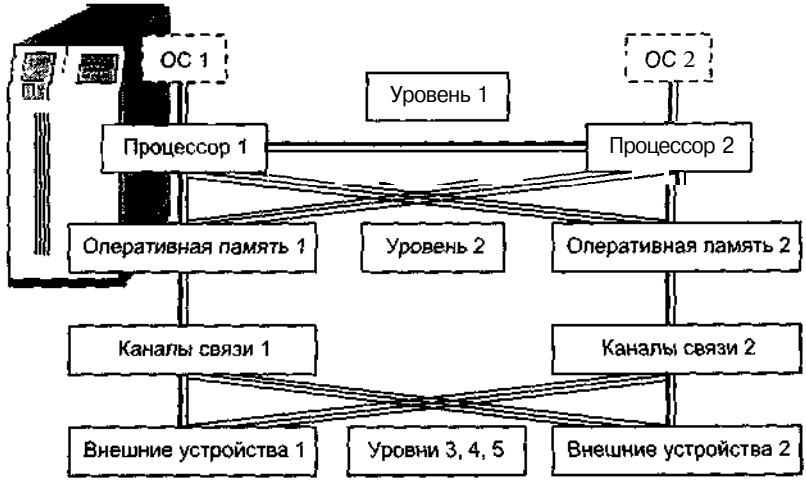


Рис. 3.3. Уровни комплексирования машин в вычислительную систему

ция расшифровывается и по ней принимается решение. После завершения передачи прерывания снимаются, и оба процессора продолжают вычисления по собственным программам. Видно, что уровень прямого управления не может использоваться для передачи больших массивов данных, однако оперативное взаимодействие отдельными сигналами широко используется в управлении вычислениями. У ПЭВМ типа IBM PC этому уровню соответствует комплексирование процессоров, подключаемых к системной шине.

Уровень общей оперативной памяти (ООП) является наиболее предпочтительным для оперативного взаимодействия процессоров. В этом случае ООП эффективно работает при небольшом числе обслуживаемых абонентов.

Уровень комплексируемых каналов ввода-вывода предназначается для передачи больших объемов информации между блоками оперативной памяти, сопрягаемых в ВС. Обмен данными между ЭВМ осуществляется с помощью адаптера «канал—канал» (АКК) и команд «чтение» и «запись». Адаптер — это устройство, согласующее скорости работы сопрягаемых каналов. Обычно сопрягаются селекторные каналы (СК) машин как наиболее быстродействующие. Скорость обмена данными определяется скоростью самого медленного канала. Скорость передачи данных по этому уровню составляет несколько мегабайт в секунду. В ПЭВМ данному уровню взаимодействия соответствует подключение периферийной аппаратуры через контроллеры и адаптеры.

Уровень устройств управления внешними устройствами (УВУ) предполагает использование встроенного в УВУ двухканального переключателя и команд «зарезервировать» и «освободить». Двухканальный переключатель позволяет подключать УВУ одной машины к селекторным каналам различных ЭВМ. По команде «зарезервировать» канал — инициатор обмена имеет доступ через УВУ к любым накопителям на дисках НМД или на магнитных лентах НМЛ. На самом деле УВУ магнитных дисков и лент — совершенно различные устройства. Обмен канала с накопителями продолжается до полного завершения работ и получения команды «освободить». Только после этого УВУ может подключиться к конкурирующему каналу. Только такая дисциплина обслуживания требований позволяет избежать конфликтных ситуаций.

На четвертом уровне с помощью аппаратуры передачи данных (АПД) (мультиплексоры, сетевые адаптеры, модемы и др.) имеется возможность сопряжения с каналами связи. Эта аппаратура позволяет создавать сети ЭВМ.

Пятый *уровень* предполагает использование *общих внешних устройств.* Для подключения отдельных устройств используется автономный двухканальный переключатель.

Пять уровней комплексирования получили название *логических* потому, что они объединяют на каждом уровне разнотипную аппаратуру, имеющую сходные методы управления. Каждое из устройств может иметь логическое имя, используемое в прикладных программах. Этим достигается независимость программ пользователей от конкретной физической конфигурации системы. Связь логической структуры программы и конкретной физической структуры ВС обеспечивается операционной системой по указаниям — директивам пользователя, при генерации ОС и по указаниям диспетчера-оператора вычислительного центра. Различные уровни комплексирования позволяют создавать самые различные структуры ВС.

Второй логический уровень позволяет создавать многопроцессорные ВС. Обычно он дополняется и первым уровнем, что позволяет повышать оперативность взаимодействия процессоров. Вычислительные системы сверхвысокой производительности должны строиться как многопроцессорные Центральным блоком такой системы является быстродействующий коммутатор, обеспечивающий необходимые подключения абонентов (процессоров и каналов) к общей оперативной памяти.

Уровни 1, 3, 4, 5 обеспечивают построение разнообразных машинных комплексов. Особенно часто используется третий в комбинации с четвертым. Целесообразно их дополнять и первым уровнем.

Пятый уровень комплексирования используется в редких специальных случаях, когда в качестве внешнего объекта используется какое-то дорогое уникальное устройство. В противном случае этот уровень малоэффективен. Любое внешнее устройство — это недостаточно надежное устройство точной механики, а значит, выгоднее использовать четвертый уровень комплексирования, когда можно сразу управлять не одним, а несколькими внешними устройствами, включая и резервные.

Чтобы дать более полное представление о многопроцессорных вычислительных системах, помимо высокой производительности необходимо назвать и другие отличительные особенности. Прежде всего это необычные архитектурные решения, направленные на повышение производительности (работа с векторными операциями, организация быстрого обмена сообщениями между процессорами или организация глобальной памяти в многопроцессорных системах и др.).

Классификация архитектуры вычислительных систем с параллельной обработкой данных (М. Флинн, 1966 г.)

Цели, которым должна служить хорошо построенная классификация архитектур:

- облегчать понимание того, что достигнуто на сегодняшний день в области архитектур вычислительных систем, и какие архитектуры имеют лучшие перспективы в будущем;
- подсказывать новые пути организации архитектур — речь идет о тех классах, которые в настоящее время по разным причинам пусты;
- показывать, за счет каких структурных особенностей достигается увеличение производительности различных вычислительных систем; с этой точки зрения классификация может служить моделью для анализа производительности.

В 1966 г. М. Флинном (Flynn) был предложен следующий подход к классификации архитектур вычислительных систем. В основу было положено понятие потока, под которым понимается последовательность элементов, команд или данных, обрабатываемая процессором. Соответствующая система классификации основана на рассмотрении числа потоков инструкций и потоков данных и описывает четыре базовых класса (табл. 3.1, рис. 3.4).

Коротко рассмотрим отличительные особенности каждой из архитектур.

Таблица 3 1 **Классификация Флинна**

Поток данных	Поток команд	
	одиночный	множественный
Одиночный	S1SD - Single Instruction stream / Single Data stream (Одиночный поток Команд и Одиночный поток Данных - ОКОД)	MISD - Multiple Instruction stream / Single Data stream (Множественный поток Команд и Одиночный поток Данных - МКОД)
Множественный	SIMD - Single Instruction stream / Multiple Data stream (Одиночный поток Команд и Множественный поток Данных - ОКМД)	MIMD - Multiple Instruction stream / Multiple Data stream (Множественный поток Команд и Множественный поток Данных - МКМД)

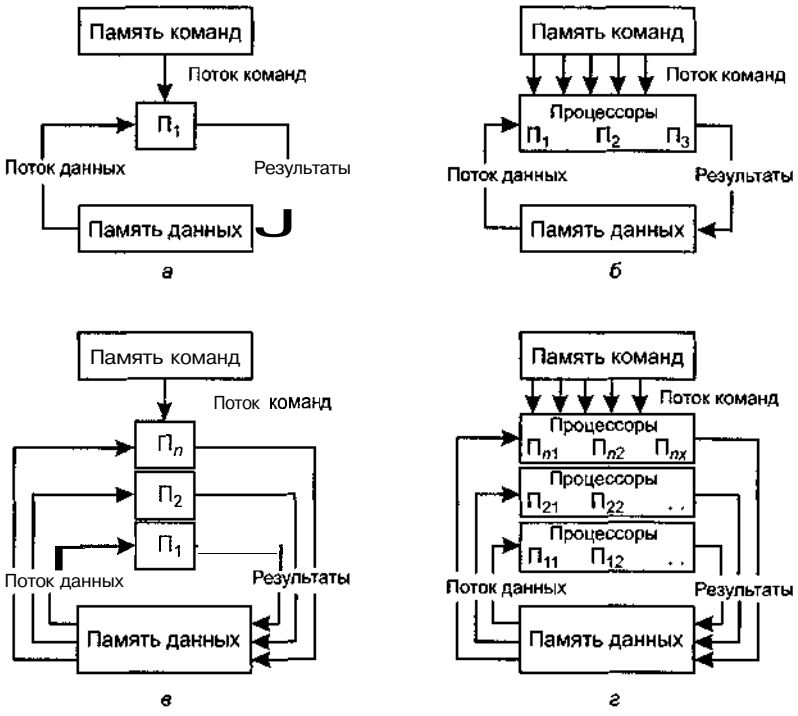


Рис. 3.4. Классификация Флинна:
а - S1SD, *б* - MISD; в - SIMD, *г* - MIMD

Архитектура ОКОД (SISD) охватывает все однопроцессорные и одномашинные варианты систем, т. е. с одним вычислителем. Все ЭВМ классической структуры попадают в этот класс. Здесь параллелизм вычислений обеспечивается путем совмещения выполнения операций отдельными блоками АЛУ, а также параллельной работой

устройств ввода-вывода информации и процессора. Закономерности организации вычислительного процесса в этих структурах достаточно хорошо изучены.

Архитектура ОКМД(SIMD) предполагает создание структур векторной или матричной обработки. Системы этого типа обычно строятся как однородные, т. е. процессорные элементы, входящие в систему, идентичны, и все они управляются одной и той же последовательностью команд. Однако каждый процессор обрабатывает свой поток данных. Под эту схему хорошо подходят задачи обработки матриц или векторов (массивов), задачи решения систем линейных и нелинейных, алгебраических и дифференциальных уравнений, задачи теории поля и др. В структурах данной архитектуры желательно обеспечивать соединения между процессорами, соответствующие реализуемым математическим зависимостям. Как правило, эти связи напоминают матрицу, в которой каждый процессорный элемент связан с соседними. По данной схеме строились системы: первая суперЭВМ — ILLIAC-IV, отечественные параллельные системы — ПС-2000, ПС-3000. Идея векторной обработки широко использовалась в таких известных суперЭВМ, как Cyber-205 и Gray-I, II, III. Узким местом подобных систем является необходимость изменения коммутации между процессорами, когда связь между ними отличается от матричной. Кроме того, класс задач, допускающих широкий матричный параллелизм, весьма узок. Структуры ВС этого типа, по существу, являются структурами специализированных суперЭВМ.

Элементы технологии SIMD реализованы в процессорах Intel начиная с Pentium MMX (1997 г.).

Третий тип архитектуры — МКОД (MISD) предполагает построение своеобразного процессорного конвейера, в котором результаты обработки передаются от одного процессора к другому по цепочке. Выгоды такого вида обработки понятны. Прототипом таких вычислений может служить схема любого производственного конвейера. В современных ЭВМ по этому принципу реализована схема совмещения операций, в которой параллельно работают различные функциональные блоки, и каждый из них делает свою часть в общем цикле обработки команды. В ВС этого типа конвейеры должны образовывать группы процессоров. Однако при переходе на системный уровень очень трудно выявить подобный регулярный характер в универсальных вычислениях. Кроме того, на практике нельзя обеспечить и «большую длину» такого конвейера, при которой достигается наивысший эффект. Вместе с тем конвейерная схема нашла применение в так называемых скалярных процессорах су-

перЭВМ, в которых они применяются как специальные процессоры для поддержки векторной обработки.

Архитектура МКМД(MIMD) предполагает, что все процессоры системы работают по своим программам с собственным потоком команд. В простейшем случае они могут быть автономны и независимы. Такая схема использования ВС часто применяется на многих крупных вычислительных центрах для увеличения пропускной способности центра. Большой интерес представляет возможность согласованной работы ЭВМ (процессоров), когда каждый элемент делает часть общей задачи. Общая теоретическая база такого вида работ практически отсутствует. Но можно привести примеры большой эффективности этой модели вычислений. Подобные системы могут быть многомашинными и многопроцессорными. Например, отечественный проект машины динамической архитектуры (МДА) — ЕС-2704, ЕС-2727 — предполагал одновременное использование сотни процессоров.

Другие подходы к классификации ВС

Наличие большого разнообразия систем, образующих класс МКМД (MIMD), делает классификацию Флинна не полностью адекватной. Действительно и 4-процессорный SX-5 компании NEC и 1000-процессорный Cray T3E попадают в этот класс, и это заставляет искать другие подходы к классификации.

Классификация Джонсона. Е. Джонсон предложил проводить классификацию MIMD-архитектур на основе структуры памяти и реализации механизма взаимодействия и синхронизации между процессорами.

По структуре оперативной памяти существующие вычислительные системы делятся на две большие группы: либо это системы с общей памятью, прямо адресуемой всеми процессорами, либо это системы с распределенной памятью, каждая часть которой доступна только одному процессору. Одновременно с этим и для межпроцессорного взаимодействия существуют две альтернативы: через разделяемые (общие) переменные или с помощью механизма передачи сообщений. Исходя из таких предположений, можно получить четыре класса MIMD-архитектур, уточняющих систематику Флинна (табл. 3.2).

Основываясь на таком делении, Джонсон вводит следующие наименования для некоторых классов:

- вычислительные системы, использующие общую разделяемую память для межпроцессорного взаимодействия и синхрониза-

Таблица 3 2 **Классификация Джонсона для систем MIMD по Флинну**

Обмен данными	Память	
	общая	распределенная
Общие данные	GMSV - General Memory-Shared variables (Общая память - разделяемые переменные) Класс 1 «Системы с разделяемой памятью»	DMSV - Distributed Memory, Shared variables (Распределенная память - разделяемые переменные) Класс 2. «Гибридная архитектура»
Передача данных	GMMP - General Memory, Message propagation (Общая память - передача сообщений)	DMMP - Distributed Memory, Message propagation) (Распределенная память - передача сообщений) Класс 3 «Архитектуры с передачей сообщений»

ции, он называет *системами с разделяемой памятью,* например CRAY Y-MP (по его классификации это класс 1);

- системы, в которых память распределена по процессорам, а для взаимодействия и синхронизации используется механизм передачи сообщений, называются *архитектурами с передачей сообщений,* например NCube (класс 3);
- системы с распределенной памятью и синхронизацией через разделяемые переменные, как в BBN Butterfly, называются *гибридными архитектурами* (класс 2).

В качестве уточнения классификации автор отмечает возможность учитывать вид связи между процессорами: общая шина, переключатели, разнообразные сети и т. п.

Классификация Базу. По мнению А. Базу (A. Basu), любую параллельную вычислительную систему можно однозначно описать последовательностью решений, принятых на этапе ее проектирования, а сам процесс проектирования представить в виде дерева. Корень дерева — это вычислительная система (рис. 3.5), и последующие ярусы дерева, фиксируя уровень параллелизма, метод реализации алгоритма, параллелизм инструкций и способ управления, последовательно дополняют друг друга, формируя описание системы.

На первом этапе определяется, какой уровень параллелизма использует вычислительная система. Одна и та же операция может одновременно выполняться над целым набором данных, определяя параллелизм на уровне данных (обозначено *D* на рис. 3.5). Способность выполнять более одной операции одновременно говорит о параллелизме на уровне команд *(О).* Если же компьютер спроектирован так, что целые последовательности команд могут быть выполнены одновременно, то будем говорить о параллелизме на уровне задач *(Т).*

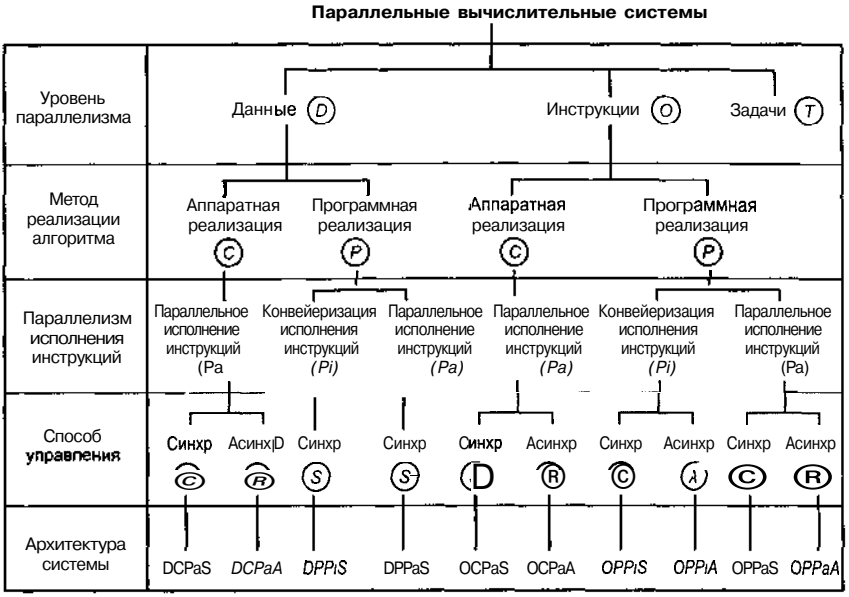


Рис. 3.5. Классификация Базу

Второй уровень в классификационном дереве фиксирует метод реализации алгоритма. С появлением сверхбольших интегральных схем (СБИС) стало возможным реализовывать аппаратно не только простые арифметические операции, но и алгоритмы целиком. Например, быстрое преобразование Фурье, перемножение матриц и другие относятся к классу тех алгоритмов, которые могут быть эффективно реализованы в СБИС. Данный уровень классификации разделяет системы с аппаратной реализацией алгоритмов (С на рис. 3.5) и системы, использующие традиционный способ программной реализации (P).

Третий уровень конкретизирует тип параллелизма, используемого для обработки инструкций машины: конвейеризация инструкций (P_i) или их независимое (параллельное) выполнение (P_a). В большей степени этот выбор относится к компьютерам с программной реализацией алгоритмов, так как аппаратная реализация всегда предполагает параллельное исполнение команд. Отметим, что в случае конвейерного исполнения имеется в виду лишь конвейеризация самих команд, разбивающая весь цикл обработки на выборку команды, дешифрацию, вычисление адресов и т. д. (возможная конвейеризация вычислений на данном уровне не принимается во внимание).

Последний уровень данной классификации определяет способ управления, принятый в вычислительной системе: синхронный (*S*) или асинхронный *(A).* Если выполнение команд происходит в строгом порядке, определяемом только сигналами таймера и счетчиком команд, то говорят о синхронном способе управления. Если же для инициации команды определяющими являются такие факторы, как, например, готовность данных, то машина попадает в класс с асинхронным управлением. Наиболее характерными представителями систем с асинхронным управлением являются компьютеры *data-driven* и *demand-driven.*

Классификация Дункана (рис. 3.6). Р. Дункан определяет тот набор требований, на который может опираться искомая классификация.

Из класса параллельных машин должны быть исключены те, в которых параллелизм заложен лишь на самом низком уровне, включая:

- конвейеризацию на этапе подготовки и выполнения команды (instruction pipelining), т. е. частичное перекрытие таких этапов, как дешифрация команды, вычисление адресов операндов, выборка операндов, выполнение команды и сохранение результата;
- наличие в архитектуре нескольких функциональных устройств, работающих независимо, в частности, возможность параллельного выполнения логических и арифметических операций;
- наличие отдельных процессоров ввода/вывода, работающих независимо и параллельно с основными процессорами.

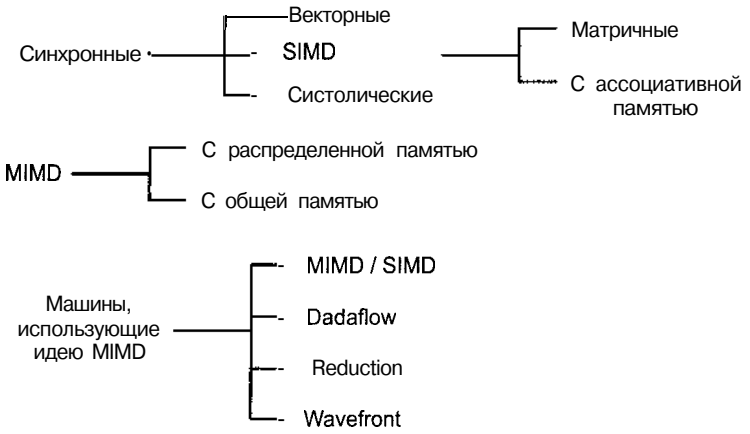


Рис. 3.6. Классификация Дункана

Причины исключения перечисленных выше особенностей автор объясняет следующим образом. Если рассматривать компьютеры, использующие только параллелизм низкого уровня, наравне со всеми остальными, то, во-первых, практически все существующие системы будут классифицированы как «параллельные» (что заведомо не будет позитивным фактором для классификации), и, во-вторых, такие машины будут плохо вписываться в любую модель или концепцию, отражающую параллелизм высокого уровня.

Классификация должна быть согласованной с классификацией Флинна, показавшей правильность выбора идеи потоков команд и данных.

Классификация должна описывать архитектуры, которые однозначно не укладываются в систематику Флинна, но тем не менее относятся к параллельным архитектурам (например, векторно-конвейерные).

Учитывая вышеизложенные требования, Дункан дает неформальное определение параллельной архитектуры, причем именно неформальность дала ему возможность включить в данный класс компьютеры, которые ранее не вписывались в систематику Флинна. Итак, *параллельная архитектура* — это такой способ организации вычислительной системы, при котором допускается, чтобы множество процессоров (простых или сложных) могло бы работать одновременно, взаимодействуя по мере надобности друг с другом. Следуя этому определению, все разнообразие параллельных архитектур Дункан систематизирует так, как это показано на рис. 3.6.

По существу систематика очень простая: процессоры системы работают либо синхронно, либо независимо друг от друга, либо в архитектуру системы заложена та или иная модификация идеи MIMD. На следующем уровне происходит детализация в рамках каждого из этих трех классов. Дадим небольшое пояснение лишь к тем из них, которые на сегодняшний день не столь широко известны.

Систолические архитектуры (их чаще называют систолическими массивами) представляют собой множество процессоров, объединенных регулярным образом (например, система WARP). Обращение к памяти может осуществляться только через определенные процессоры на границе массива. Выборка операндов из памяти и передача данных по массиву осуществляется в одном и том же темпе. Направление передачи данных между процессорами фиксированно. Каждый процессор за интервал времени выполняет небольшую инвариантную последовательность действий.

Гибридные MIMD/SIMD архитектуры, вычислительные системы *dataflow, reduction* и *wavefront*осуществляют параллельную обработку информации на основе асинхронного управления, как и MIMD-системы. Но они выделены в отдельную группу, поскольку все имеют ряд специфических особенностей, которыми не обладают системы, традиционно относящиеся к MIMD.

MIMD/SIMD– типично гибридная архитектура. Она предполагает, что в MIMD-системе можно выделить группу процессоров, представляющую собой подсистему, работающую в режиме SIMD (например, PASM, Non-Von). Такие системы отличаются относительной гибкостью, поскольку допускают реконфигурацию в соответствии с особенностями решаемой прикладной задачи.

Остальные три вида архитектур используют нетрадиционные модели вычислений. *Dataflow-машины* используют модель, в которой команда может выполняться сразу же, как только вычислены необходимые операнды. Таким образом, последовательность выполнения команд определяется зависимостью по данным, которая может быть выражена, например, в форме графа.

Модель вычислений, применяемая в *reduction*-машинах, *иная и* состоит в следующем: команда становится доступной для выполнения тогда и только тогда, когда результат ее работы требуется другой, доступной для выполнения команде в качестве операнда.

Архитектура *wavefront array* объединяет в себе идею систолической обработки данных и модель вычислений, используемой в *dataflow*-машинах. В данной архитектуре процессоры объединяются в модули и связи, по которым процессоры могут взаимодействовать друг с другом, фиксируются. Однако, в противоположность ритмичной работе систолических массивов, данная архитектура использует асинхронный механизм связи с подтверждением *(handshaking), из-за* чего «фронт волны» вычислений может менять свою форму по мере прохождения по всему множеству процессоров.

Классификация Кришнамарфи. Е. Кришнамарфи для классификации параллельных вычислительных систем предлагает использовать четыре характеристики, похожие на характеристики классификации А. Базу (рис. 3.7):

- степень гранулярности;
- способ реализации параллелизма;
- топологию и природу связи процессоров;
- способ управления процессорами.

Принцип построения классификации достаточно прост. Для каждой степени гранулярности рассматриваются все возможные способы реализации параллелизма. Для каждого полученного таким

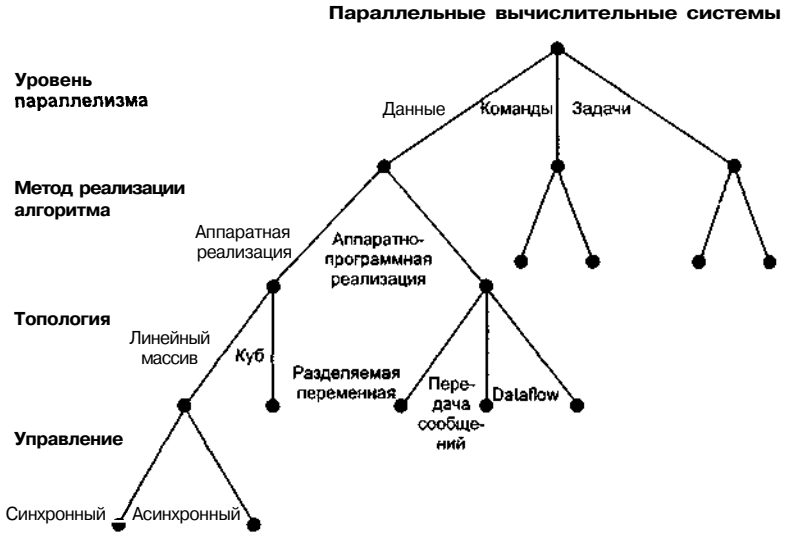


Рис. 3.7. Классификация Кришнамарфи

образом варианта устанавливаются все комбинации топологии связи и способов управления процессорами. В результате получается дерево (см. рис. 3.7), в котором каждый ярус соответствует своей характеристике, каждый лист представляет отдельную группу компьютеров в данной классификации, а путь от вершины дерева однозначно определяет значения указанных выше характеристик.

Первых два уровня практически повторяют классификацию А. Базу. Третий уровень классификации (топология и природа связи процессоров) тесно связан со вторым. Если был выбран аппаратный способ реализации параллелизма, то надо рассмотреть топологию связи процессоров (матрица, линейный массив, тор, дерево, звезда и т. п.) и степень связности процессоров между собой (сильная, слабая или средняя), которая определяется относительной долей накладных расходов при организации взаимодействия процессоров. В случае комбинированной реализации параллелизма, помимо топологии и степени связности, надо дополнительно учесть механизм взаимодействия процессоров: передача сообщений, разделяемые переменные или принцип *dataflow*(по готовности операндов).

Наконец, последний, четвертый уровень — способ управления процессорами, определяет общий принцип функционирования всей совокупности процессоров вычислительной системы: синхронный, *dataflow* или асинхронный.

На основе выделенных четырех характеристик нетрудно определить место наиболее известных классов архитектур в данной систематике.

Векторно-конвейерные компьютеры:
- гранулярность — на уровне данных;
- реализация параллелизма — аппаратная;
- связь процессоров — простая топология со средней связностью;
- способ управления — синхронный.

Классические мультипроцессоры:
- гранулярность — на уровне задач
- реализация параллелизма — комбинированная;
- связь процессоров — простая топология со слабой связностью и использованием разделяемых переменных;
- способ управления — асинхронный.

Матричные процессоры:
- гранулярность — на уровне данных;
- реализация параллелизма — аппаратная;
- связь процессоров — двухмерные массивы с сильной связностью;
- способ управления — синхронный.

Систолические массивы:
- гранулярность — на уровне данных;
- реализация параллелизма — аппаратная;
- связь процессоров — сложная топология с сильной связностью;
- способ управления — синхронный.

Архитектура типа wavefront:
- гранулярность — на уровне данных;
- реализация параллелизма — аппаратная;
- связь процессоров — двухмерная топология с сильной связностью;
- способ управления — *dataflow.*

Архитектура типа dataflow:
- гранулярность — на уровне команд;
- реализация параллелизма — комбинированная;
- связь процессоров — простая топология с сильной либо средней связностью и использованием принципа *dataflow,*
- способ управления — асинхронно-*dataflow.*

Несмотря на то, что классификация Е. Кришнамарфи построена только на четырех признаках, она позволяет выделить и описать

такие «нетрадиционные» параллельные системы, как систолические массивы, машины типа *dataflow* и *wavefron*А Однако эта же простота является и основной причиной ее недостатков: некоторые архитектуры нельзя однозначно отнести к тому или иному классу, например, компьютеры с архитектурой гиперкуба и ассоциативные процессоры. Для более точного описания таких машин потребуется ввести еще целый ряд характеристик, таких, как размещение задач по процессорам, способ маршрутизации сообщений, возможность реконфигурации, аппаратная поддержка языков программирования и другие. Вместе с тем ясно, что эти признаки формализовать гораздо труднее, поэтому есть опасность вместо ясности внести в описание лишь дополнительные трудности.

Классификация Скилликорна. Д. Скилликорн разработал подход, ориентированный как на описание свойств многопроцессорных систем, так и некоторых нетрадиционных архитектур, в частности *dataflow* и га/ис/гол-машины.

Предлагается рассматривать архитектуру любого компьютера, как абстрактную структуру, состоящую из четырех компонент:

- *процессор команд* (IP — Instruction Processor) — функциональное устройство, работающее, как интерпретатор команд; в системе, вообще говоря, может отсутствовать;
- *процессор данных* (DP — Data Processor) — функциональное устройство, работающее как преобразователь данных, в соответствии с арифметическими операциями;
- *иерархия памяти* (IM — Instruction Memory, DM — Data Memory) — запоминающее устройство, в котором хранятся данные и команды, пересылаемые между процессорами;
- *переключатель* — абстрактное устройство, обеспечивающее связь между процессорами и памятью.

Функции процессора команд во многом схожи с функциями устройств управления последовательных машин и, согласно Д. Скилликорну, сводятся к следующим:

- на основе своего состояния и полученной от DP информации IP определяет адрес команды, которая будет выполняться следующей;
- осуществляет доступ к IM для выборки команды;
- получает и декодирует выбранную команду;
- сообщает DP команду, которую надо выполнить;
- определяет адреса операндов и посылает их в DP;
- получает от DP информацию о результате выполнения команды.

Функции процессора данных делают его во многом похожим на арифметическое устройство традиционных процессоров:

- DP получает от IP команду, которую надо выполнить;
- получает от IP адреса операндов;
- выбирает операнды из DM,
- выполняет команду;
- запоминает результат в DM;
- возвращает в IP информацию о состоянии после выполнения команды.

В терминах таким образом определенных основных частей компьютера структуру традиционной фон-неймановской архитектуры можно представить в следующем виде (рис. 3.8).

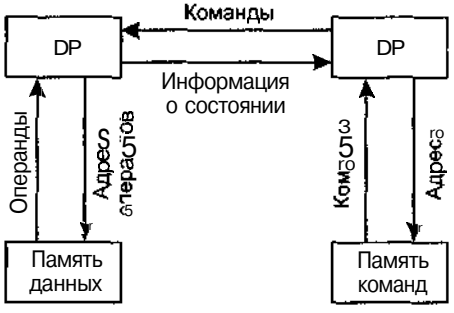


Рис. 3.8. Представление фон-неймановской архитектуры по Скилликорну

Это один из самых простых видов архитектуры, не содержащих переключателей. Для описания параллельных вычислительных систем автор зафиксировал четыре типа переключателей, без какой-либо явной связи с типом устройств, которые они соединяют:

- 1-1 — переключатель такого типа связывает пару функциональных устройств;
- *n*-*n* — переключатель связывает *i*-е устройство из одного множества устройств с *i*-м устройством из другого множества, т. е. фиксирует попарную связь;
- \-*n* — переключатель соединяет одно выделенное устройство со всеми функциональными устройствами из некоторого набора;
- *n* x *n* — каждое функциональное устройство одного множества может быть связано с любым устройством другого множества, и наоборот.

Примеров подобных переключателей можно привести очень много. Так, все матричные процессоры имеют переключатель типа

1-я для связи единственного процессора команд со всеми процессорами данных. В компьютерах семейства Connection Machine каждый процессор данных имеет свою локальную память, следовательно, связь будет описываться как *n-n*. В то же время, каждый процессор команд может связаться с любым другим процессором, поэтому данная связь будет описана как *n* x *n*.

Классификация Д. Скилликорна состоит из двух уровней. На первом уровне она проводится на основе восьми характеристик:

- количества процессоров команд (IP);
- числа запоминающих устройств (модулей памяти) команд (IM);
- типа переключателя между IP и IM;
- количества процессоров данных (DP);
- числа запоминающих устройств (модулей памяти) данных (DM);
- типа переключателя между DP и DM;
- типа переключателя между IP и DP;
- типа переключателя между DP и DP.

Используя введенные характеристики и предполагая, что рассмотрение количественных характеристик можно ограничить только тремя возможными вариантами значений: О, 1 и *n* (т. е. больше одного), можно получить 28 классов архитектур.

В классах 1—5 находятся компьютеры типа *dataflow* и *reduction,* не имеющие процессоров команд в обычном понимании этого слова. Класс 6 — это классическая фон-неймановская последовательная машина. Все разновидности матричных процессоров содержатся в классах 7—10. Классы 11 и 12 отвечают компьютерам типа MISD классификации Флинна и на настоящий момент, по мнению автора, пусты. Классы с 13-го по 28-й занимают всесозможные варианты мультипроцессоров, причем в 13—20 классах находятся машины с достаточно привычной архитектурой, в то время, как архитектура классов 21—28 пока выглядит экзотично.

На втором уровне классификации Д. Скилликорн уточняет описание, сделанное на первом уровне, добавляя возможность конвейерной обработки в процессорах команд и данных.

Классификация Хендлера. В основу классификации В Хендлер закладывает явное описание возможностей параллельной и конвейерной обработки информации вычислительной системой. При этом он намеренно не рассматривает различные способы связи между процессорами и блоками памяти и считает, что коммуникационная сеть может быть нужным образом сконфигурирована и будет способна выдержать предполагаемую нагрузку (рис. 3.9).

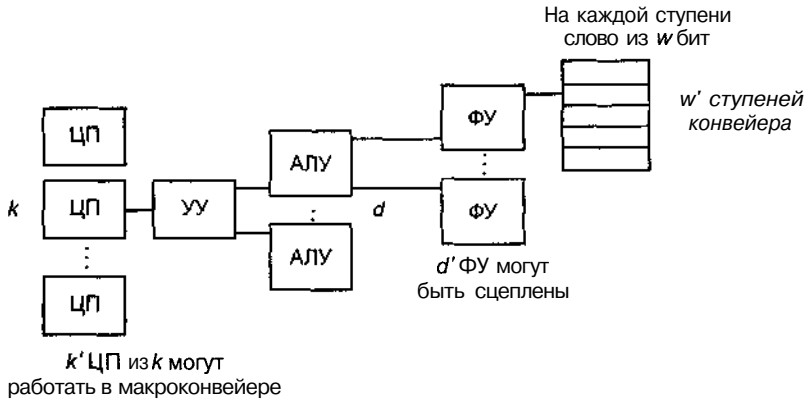


Рис. 3.9. Классификация Хендлера

Предложенная классификация базируется на различии между тремя уровнями обработки данных в процессе выполнения программ:

- уровень выполнения программы: опираясь на счетчик команд и некоторые другие регистры, устройство управления (УУ) производит выборку и дешифрацию команд программы;
- уровень выполнения команд: арифметико-логическое устройство компьютера (АЛУ) исполняет команду, выданную ему устройством управления;
- уровень битовой обработки: все элементарные логические схемы процессора (ЭЛС) разбиваются на группы, необходимые для выполнения операций над одним двоичным разрядом.

Таким образом, подобная схема выделения уровней предполагает, что вычислительная система включает какое-то число процессоров, каждый со своим устройством управления. Каждое устройство управления связано с несколькими арифметико-логическими устройствами, исполняющими одну и ту же операцию в каждый конкретный момент времени. Наконец, каждое АЛУ объединяет несколько элементарных логических схем, ассоциированных с обработкой одного двоичного разряда (число ЭЛС есть не что иное, как длина машинного слова). Если на какое-то время не рассматривать возможность конвейеризации, то число устройств управления k, число арифметико-логических устройств d в каждом устройстве управления и число элементарных логических схем w в каждом АЛУ составят тройку для описания данной вычислительной системы С:

$$t(C) = (k, d, w).$$

Классификация Хокни. Р Хокни (английский специалист в области параллельных вычислительных систем) разработал свой подход к классификации, введенной им для систематизации компьютеров, попадающих в класс MIMD.

Как отмечалось ранее (см. классификацию Флинна), класс MIMD чрезвычайно широк, причем наряду с большим числом компьютеров он объединяет и целое множество различных типов архитектур. Хокни, пытаясь систематизировать архитектуры внутри этого класса, получил иерархическую структуру, представленную на рис. 3.10.

Основная идея классификации состоит в следующем. Множественный поток команд может быть обработан двумя способами: либо одним конвейерным устройством обработки, работающем в режиме разделения времени для отдельных потоков, либо каждый поток обрабатывается своим собственным устройством. Первая возможность используется в MIMD-компьютерах, которые автор называет *конвейерными* (например, процессорные модули в Denelcor HEP). Архитектуры, использующие вторую возможность, в свою очередь опять делятся на два класса:

- MIMD-компьютеры, в которых возможна прямая связь каждой пары процессоров, которая реализуется с помощью переключателя;
- MIMD-компьютеры, в которых прямая связь каждого процессора возможна только с ближайшими соседями по сети, а

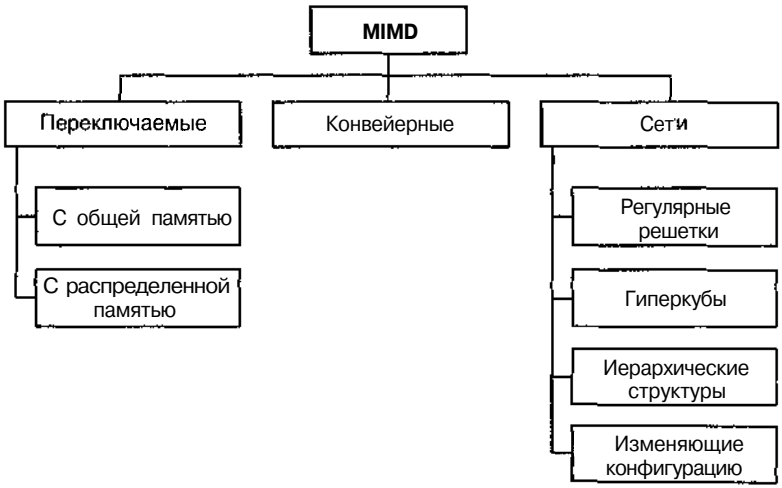


Рис. 3.10. Системы архитектуры MIMD (Флинн) в интерпретации Хокни

взаимодействие удаленных процессоров поддерживается специальной системой маршрутизации через процессоры-посредники.

Далее, среди MIMD-машин с переключателем Хокни выделяет те, в которых вся память распределена среди процессоров как их локальная память (например, PASM, PR1NGLE). В этом случае общение самих процессоров реализуется с помощью очень сложного переключателя, составляющего значительную часть компьютера. Такие машины носят название MIMD-машин *с распределенной памятью.* Если память — это разделяемый ресурс, доступный всем процессорам через переключатель, то такие MIMD-машины являются системами *с общей памятью* (CRAY X-MP, BBN Butterfly). В соответствии с типом переключателей можно проводить классификацию и далее: простой переключатель, многокаскадный переключатель, общая шина.

Многие современные вычислительные системы имеют как общую разделяемую память, так и распределенную локальную. Такие системы являются *гибридными* MIMD с переключателем.

При рассмотрении MIMD-машин с *сетевой структурой* считается, что все они имеют распределенную память, а дальнейшая классификация проводится в соответствии с топологией сети: звездообразная сеть (1CAP), регулярные решетки разной размерности (Intel Paragon, CRAY T3D), гиперкубы (NCube, Intel iPCS), сети с иерархической структурой, такой, как деревья, пирамиды, кластеры (Cm, CEDAR) и, наконец, сети, изменяющие свою конфигурацию.

Заметим, что если архитектура компьютера спроектирована с использованием нескольких сетей с различной топологией, то по аналогии с гибридными MIMD с переключателями их стоит назвать *гибридными сетевыми* M1MD, а использующие идеи разных классов — просто *гибридными* MIMD. Типичным представителем последней группы, в частности, *является* компьютер Connection Machine 2, имеющий на внешнем уровне топологию гиперкуба, каждый узел которого является кластером.

Классификация Шора. Классификация Дж. Шора, появившаяся в начале 70-х гг., интересна тем, что представляет собой попытку выделения типичных способов компоновки вычислительных систем на основе фиксированного числа базисных блоков: устройства управления, арифметико-логического устройства, памяти команд и памяти данных. Дополнительно предполагается, что выборка из памяти данных может осуществляться словами, т. е. выбираются все разряды одного слова и/или битовым слоем — по одному разряду из одной и той же позиции каждого слова (иногда этих два способа на-

зывают горизонтальной и вертикальной выборками соответственно). Конечно же, при анализе данной классификации надо делать скидку на время ее появления, так как предусмотреть невероятное разнообразие параллельных систем настоящего времени было в принципе невозможно. Итак, согласно классификации Шора все компьютеры разбиваются на шесть классов, которые так и называются: *машина типа I, II* и т. д.

Машина типа I — это вычислительная система, которая содержит устройство управления, арифметико-логическое устройство, память команд и память данных с пословной выборкой (рис. 3.11, *а).* Считывание данных осуществляется выборкой всех разрядов некоторого слова для их параллельной обработки в арифметико-логическом устройстве. Состав АЛУ специально не оговаривается, что допускает наличие нескольких функциональных устройств, может быть, конвейерного типа. По этим соображениям в данный класс попадают как классические последовательные машины (IBM 701, PDP-11, VAX 11/780), так конвейерные скалярные (CDC 7600) и векторно-конвейерные (CRAY-1).

Если в *машине типа I* осуществлять выборку не по словам, а содержимого одного разряда из всех слов, то получим *машину типа II* (рис. 3.11, *б).* Слова в памяти данных по-прежнему располагаются горизонтально, но доступ к ним осуществляется иначе. Если в *машине I* происходит последовательная обработка слов при параллельной обработке разрядов, то в *машине II* — последовательная обработка битовых слоев при параллельной обработке множества слов.

Структура *машины II* лежит в основе ассоциативных компьютеров (например, центральный процессор машины STARAN), причем фактически такие компьютеры имеют не одно арифметико-логическое устройство, а множество сравнительно простых устройств поразрядной обработки. Другим примером служит матричная система

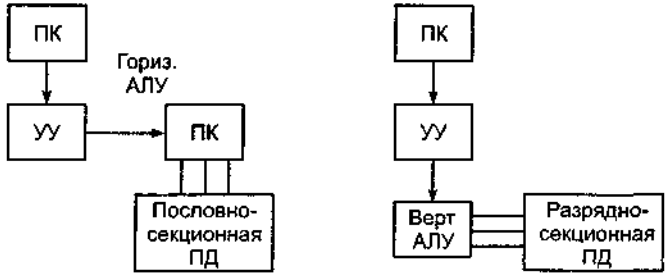


Рис. **3.11.** Машины типов I *(а)* и II *(б)* по классификации Шора

ICL DAP, которая может одновременно обрабатывать по одному разряду из 4096 слов

Если объединить принципы построения *машин I и //*, то получим *машину типа III* (рис. 3.12, *а)* Эта машина имеет два арифметико-логических устройства — горизонтальное и вертикальное, и модифицированную память данных, которая обеспечивает доступ как к словам, так и к битовым слоям. Впервые идею построения таких систем в 1960 г. выдвинул У. Шуман, называвший их *ортогональными* (если память представлять как матрицу слов, то доступ к данным осуществляется в направлении, «ортогональном» традиционному — не по словам (строкам), а по битовым слоям (столбцам)). В принципе, как машину STARAN, так и ICL DAP можно запрограммировать на выполнение функций машины III, но поскольку они не имеют отдельных АЛУ для обработки слов и битовых слоев, отнести их к данному классу нельзя. Полноправными представителями машин класса III являются вычислительные системы семейства OMEN-60 фирмы Sanders Associates, построенные в прямом соответствии с концепцией ортогональной машины.

Если в *машине I* увеличить число пар «арифметико-логическое устройство — память данных» (иногда эту пару называют *процессорным элементом),* то получим *машину типа IV* (рис. 3.12, *б).* Единственное устройство управления выдает команду за командой сразу всем процессорным элементам. С одной стороны, отсутствие соединений между процессорными элементами делает дальнейшее наращивание их числа относительно простым, но с другой — сильно ограничивает применимость машин этого класса. Такую структуру имеет вычислительная система PEPE, объединяющая 288 процессорных элементов.

Если ввести непосредственные линейные связи между соседними процессорными элементами *машины IV,* например в виде мат-

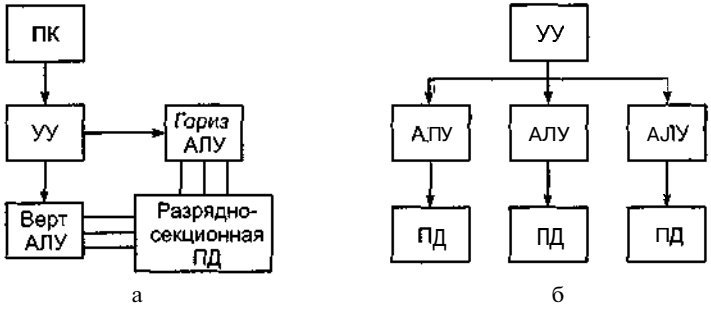


Рис. **3.12.** Машины типов III (а) и IV (б) по классификации Шора

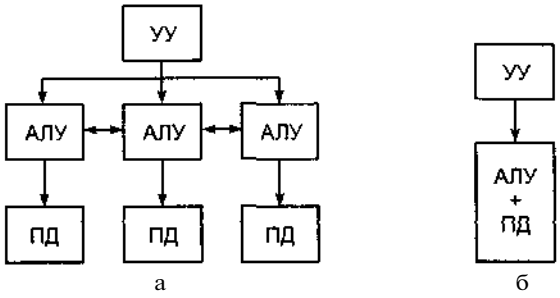


Рис. **3.13.** Машины типов V *(а)* и VI (б) по классификации Шора

ричной конфигурации, то получим схему *машины типа V* (рис. 3.13, *а)*. Любой процессорный элемент теперь может обращаться к данным как в собственной памяти, так и в памяти непосредственных соседей. Подобная структура характерна, например, для классического матричного компьютера ILLIAC IV.

Заметим, что все *машины типа I—V* придерживаются концепции разделения памяти данных и арифметико-логических устройств, предполагая наличие шины данных или какого-либо коммутирующего элемента между ними. *Машина типа VI* (рис. 3 13, *б),* названная *матрицей с функциональной памятью* (или памятью со встроенной логикой), представляет собой другой подход, предусматривающий распределение логики процессора по всему запоминающему устройству. Примерами могут служить как простые ассоциативные запоминающие устройства, так и сложные ассоциативные процессоры.

3.2. Примеры некоторых архитектур вычислительных систем

Рассмотрим далее примеры конкретных архитектур, а именно:

- асимметричную многопроцессорную (мультипроцессорную) обработку (архитектуру);
- массивно-параллельную архитектуру;
- симметричную многопроцессорную архитектуру;
- гибридную архитектуру с неоднородным доступом к памяти;
- параллельную архитектуру с векторными процессорами;
- кластерную архитектуру.

Авторы предлагают читателю самостоятельно отнести указанные типы систем к тем или иным классам из вышерассмотренных классификаций.

Асимметричная мультипроцессорная обработка — ASymmetric Multiprocessing (ASMP)

Это архитектура суперкомпьютера, в которой каждый процессор имеет свою оперативную память.

В ASMP используются три схемы (рис. 3.14) В любом случае процессоры взаимодействуют между собой, передавая друг другу сообщения, т. е. как бы образуя скоростную локальную сеть. Передача сообщений может осуществляться через общую шину (рис. 3.14, а, см. также *MPP-архитектура*) либо благодаря межпроцессорным связям. В последнем случае процессоры связаны либо непосредственно (рис. 3.14, *б),* либо через друг друга (рис. 3.14, *в).* Непосредственные связи используются при небольшом числе процессоров.

Каждый процессор имеет свою, расположенную рядом с ним оперативную память. Благодаря этому, если это необходимо, процессоры могут располагаться в различных, но рядом установленных корпусах. Совокупность устройств в одном корпусе именуется кластером. Пользователь, обращаясь к кластеру, может работать сразу с группой процессоров. Такое объединение увеличивает скорость обработки данных и расширяет используемую оперативную память. Резко возрастает также отказоустойчивость, ибо кластеры осуществляют резервное дублирование данных. Созданная таким образом система называется кластерной. В этой системе, в соответствии с ее структурой, может функционировать несколько копий операцион-

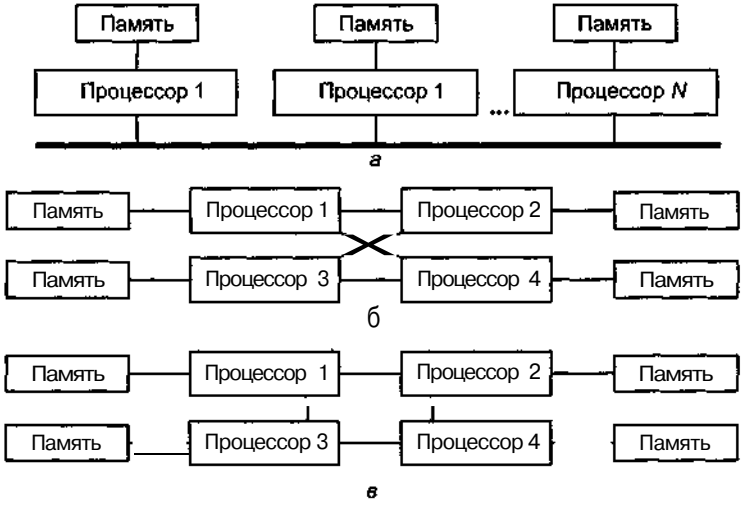


Рис. 3.14. Схемы асимметричной многопроцессорной архитектуры

ной системы и несколько копий прикладной программы, которые работают с одной и той же базой данных (БД), решая одновременно разные задачи.

MPP-архитектура (massive parallel processing) — массивно-параллельная архитектура (вариант «*а*» на рис 3 14). В этом случае система строится из отдельных модулей, каждый из которых содержит:

- процессор;
- локальный банк оперативной памяти (ОП);
- два коммуникационных процессора (рутера): один — для передачи команд, другой — для передачи данных (или сетевой адаптер);
- жесткие диски и/или другие устройства ввода/вывода.

По своей сути, такие модули представляют собой полнофункциональные компьютеры. Доступ к банку ОП из данного модуля имеют только процессоры из этого же модуля. Модули соединяются специальными коммуникационными каналами. Пользователь может определить логический номер процессора, к которому он подключен, и организовать обмен сообщениями с другими процессорами. Используются два варианта работы операционной системы на машинах MPP-архитектуры.

В первом — полноценная ОС работает только на управляющей машине (*front-end*),на каждом из остальных работает урезанный вариант ОС, обеспечивающий работу только расположенной в нем ветви параллельного приложения. Во втором варианте на каждом модуле устанавливается и работает полноценная UNIX-подобная ОС.

Главным преимуществом систем с раздельной памятью является хорошая *масштабируемость*: в отличие от SMP-систем в машинах с раздельной памятью каждый процессор имеет доступ только к своей локальной памяти, в связи с чем не возникает необходимости в потактовой синхронизации процессоров. Практически все рекорды по производительности на сегодняшний день устанавливаются на машинах именно такой архитектуры, состоящих из нескольких тысяч процессоров (ASCI Red, ASCI Blue Pacific).

Недостатки:

- отсутствие общей памяти заметно снижает скорость межпроцессорного обмена, поскольку нет общей среды для хранения данных, предназначенных для обмена между процессорами. Требуется специальная техника программирования для реализации обмена сообщениями между процессорами;
- каждый процессор может использовать только ограниченный объем локального банка памяти.

Вследствие указанных архитектурных недостатков требуются значительные усилия для того, чтобы максимально использовать системные ресурсы. Именно этим определяется высокая цена программного обеспечения для массивно-параллельных систем с раздельной памятью

Примером является сервер открытой параллельной обработки Unisys OPUS корпорации Unisys, построенный на большом числе микропроцессоров. Управление системой осуществляет модернизированная ОС Unix. Микропроцессоры образуют узлы коммуникационной сетки с пропускной способностью 175 Мбит/с. Каждый узел имеет память до 64 Мбайт. В одном корпусе могут находиться до 64 узлов.

Системы MPP выпускаются также корпорациями IBM, ICL и ATandT.

Системами с раздельной памятью являются суперкомпьютеры MBC-1000, IBM RS/6000 SP, SGI/CRAY T3E, системы ASCI, Hitachi SR8000, системы Parsytec. Машины последней серии CRAY T3E от SGI, основанные на базе процессоров Dec Alpha 21164 с пиковой производительностью 1200 Мфлопс (CRAY T3E-1200), способны масштабировать до 2048 процессоров.

Симметричная мультипроцессорная обработка Symmetric Multiprocessing (SMP)

SMP — архитектура суперкомпьютера, в которой группа процессоров работает с общей оперативной памятью (рис. 3.15).

Память является способом передачи сообщений между процессорами, при этом все вычислительные устройства при обращении к ней имеют равные права и одну и ту же адресацию для всех ячеек памяти.

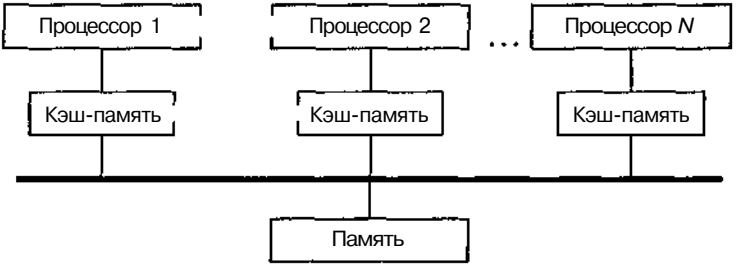


Рис. 3.15. Симметричная мультипроцессорная архитектура

Работой управляет единственная копия операционной системы. Для ускорения обработки каждый процессор может также иметь собственную кэш-память. Задания между процессами распределяются непосредственно при выполнении прикладного процесса. Нагрузка между процессорами динамически выравнивается, а обмен данными между ними происходит с большой скоростью. Достоинство этого подхода состоит в том, что каждый процессор видит всю решаемую задачу в целом. Но, поскольку для взаимодействия используется лишь одна шина, то возникают повышенные требования к ее пропускной способности. Соединение посредством шины применяется при небольшом (4—8) числе процессоров.

В подобных системах возникает проблема организации *когерентности многоуровневой иерархической памяти.*

Когерентность кэшей означает, что все процессоры получают одинаковые значения одних и тех же переменных в любой момент времени. Действительно, поскольку кэш-память принадлежит отдельному компьютеру, а не всей многопроцессорной системе в целом, данные, попадающие в кэш одного компьютера, могут быть недоступны другому. Чтобы избежать этого, следует провести синхронизацию информации, хранящейся в кэш-памяти процессоров.

Для обеспечения подобной когерентности кэшей существуют несколько возможностей:

- использовать механизм отслеживания шинных запросов (snoopy bus protocol), в котором кэши отслеживают переменные, передаваемые к любому из центральных процессоров, и при необходимости модифицируют собственные копии таких переменных;
- выделять специальную часть памяти, отвечающую за отслеживание достоверности всех используемых копий переменных.

Возможность увеличения числа процессоров в SMP ограничена из-за использования общей памяти. Более того, по той же причине все процессоры должны располагаться в одном корпусе. Между тем преимуществом SMP является то, что она может работать с прикладными программами, разработанными для однопроцессорных систем. Кроме этого, SMP использует обычные операционные системы. Например, операционную систему OS/2, сетевую операционную систему Windows NT. Корпорация Data General создала для SMP специальную версию операционной системы UNIX.

Наиболее известными SMP-системами являются SMP-серверы и рабочие станции на базе процессоров Intel (IBM, HP, Compaq, Dell, ALR, Unisys, DG, Fujitsu и др.) Обычно система работает под управлением единой ОС (как правило, UNIX-подобной, но для

Intel-платформ поддерживается Windows NT). ОС автоматически (в процессе работы) распределяет процессы по процессорам, но иногда возможна и явная привязка.

Основные преимущества SMP-систем·

- простота и универсальность для программирования. Архитектура SMP не накладывает ограничений на модель программирования, используемую при создании приложения: обычно используется модель параллельных ветвей, когда все процессоры работают абсолютно независимо друг от друга; однако можно реализовать и модели, использующие межпроцессорный обмен. Использование общей памяти увеличивает скорость такого обмена, пользователь также имеет доступ сразу ко всему объему памяти. Для SMP-систем существуют сравнительно эффективные средства автоматического распараллеливания;
- легкость в эксплуатации. Как правило, SMP-системы используют систему охлаждения, основанную на воздушном кондиционировании, что облегчает их техническое обслуживание;
- относительно невысокая цена.

Недостатки: системы с общей памятью, построенные на системной шине, плохо масштабируются.

Этот важный недостаток SMP-систем не позволяет считать их по-настоящему перспективными. Причины плохой масштабируемости состоят в том, что в каждый момент времени шина способна обрабатывать только одну транзакцию, вследствие чего возникают проблемы разрешения конфликтов при одновременном обращении нескольких процессоров к одним и тем же областям общей физической памяти. Вычислительные элементы начинают друг другу мешать. Когда произойдет такой конфликт, зависит от скорости связи и от количества вычислительных элементов. В настоящее время конфликты могут происходить при наличии 8—24 процессоров. Кроме того, системная шина имеет ограниченную (хоть и высокую) пропускную способность и ограниченное число слотов. Все это с очевидностью препятствует увеличению производительности при увеличении числа процессоров и числа подключаемых пользователей. В реальных системах можно использовать не более 32 процессоров. Для построения масштабируемых систем на базе SMP используются *кластерные или NUMA-архитектуры* При работе с SMP-системами используют так называемую парадигму программирования с общей памятью *(shared memory paradigm).*

Гибридная архитектура (NUMA). Организация когерентности многоуровневой иерархической памяти

Главная особенность гибридной архитектуры NUMA (*nonuniform memory access)* — неоднородный доступ к памяти.

Гибридная архитектура воплощает в себе удобства систем с общей памятью и относительную дешевизну систем с раздельной памятью. Суть этой архитектуры — в методе организации памяти, а именно: память является физически распределенной по различным частям системы, но логически разделяемой, так что пользователь видит единое адресное пространство. Система состоит из однородных базовых модулей (плат), состоящих из небольшого числа процессоров и блока памяти. Модули объединены с помощью высокоскоростного коммутатора. Поддерживается единое адресное пространство, аппаратно поддерживается доступ к удаленной памяти, т. е. к памяти других модулей. При этом доступ к локальной памяти осуществляется в несколько раз быстрее, чем к удаленной. По существу архитектура NUMA является MPP (массивно-параллельной) архитектурой, где в качестве отдельных вычислительных элементов берутся SMP-узлы.

Пример структурной схемы компьютера с гибридной сетью (рис. 3.16): четыре процессора связываются между собой с помощью кроссбара в рамках одного SMP-узла. Узлы связаны сетью типа «бабочка» (Butterfly).

Впервые идею гибридной архитектуры предложил и воплотил в системах серии Exemplar Стив Воллох. Вариант Воллоха — система, состоящая из восьми SMP-узлов. Фирма HP купила идею и реализовала на суперкомпьютерах серии SPP Идею подхватил Сеймур Крей (Seymour R. Cray), добавил новый элемент и воплотил в систе-

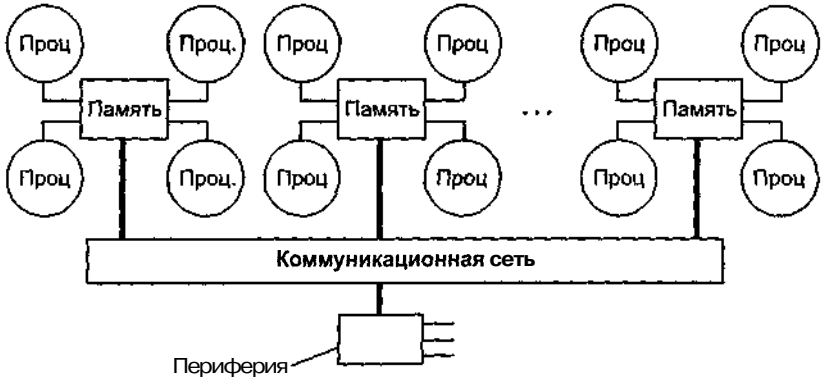


Рис. 3.16. Гибридная архитектура

мах серии Exemplar — когерентный кэш, создав так называемую архитектуру cc-NUMA (Cache Coherent Non-Uniform Memory Access), которая расшифровывается как «неоднородный доступ к памяти с обеспечением когерентности кэшей».

Известны также гибридные структуры с коммутатором (рис. 3.17). Здесь каждый процессор работает со своей памятью, но модули устройств памяти связаны друг с другом с помощью коммутатора (рис. 3.17, *а).* Коммутаторы, именуемые также узлами, могут также включаться между группами процессоров (ПР) и модулей памяти (П). Здесь сообщения между процессорами и памятью передаются через несколько узлов (рис. 3.17, *б*).

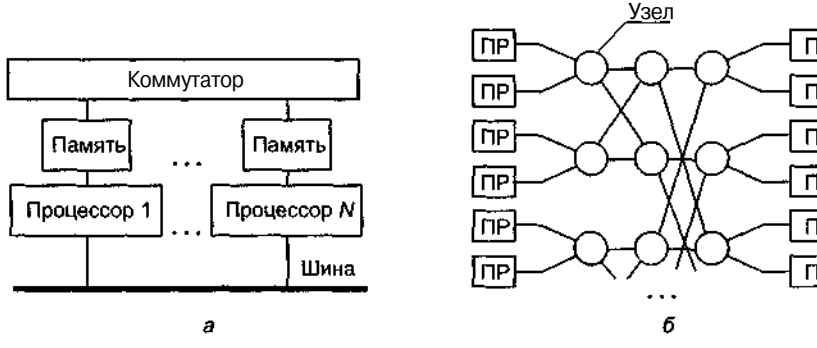


Рис. 3.17. Гибридная архитектура с коммутатором *(а);* многокаскадная коммутация *(б)*

Наиболее известными системами архитектуры cc-NUMA являются: HP 9000 V-class в SCA-конфигурациях, SGI Origin3000, Sun HPC 15000, IBM/Sequent NUMA-Q 2000. На настоящий момент максимальное число процессоров в cc-NUMA-системах может превышать 1000 (серия Origin3000). Обычно вся система работает под управлением единой ОС, как в SMP. Возможны также варианты динамического «расслоения» системы, когда отдельные «разделы» системы работают под управлением разных ОС. При работе с NUMA-системами, также как с SMP, используется парадигма программирования с общей памятью.

PVP-архитектура

PVP (Parallel Vector Process) — параллельная архитектура с векторными процессорами. Основным признаком PVP-систем является наличие *векторно-конвейерных процессоров,* в которых предусмотре-

ны команды однотипной обработки векторов независимых данных, эффективно выполняющиеся на конвейерных функциональных устройствах. Как правило, несколько таких процессоров (1—16) работают одновременно с общей памятью (аналогично SMP) в рамках многопроцессорных конфигураций. Несколько таких узлов могут быть объединены с помощью коммутатора (аналогично MPP). Поскольку передача данных в векторном формате осуществляется намного быстрее, чем в скалярном (максимальная скорость может составлять 64 Гбайт/с, что на два порядка быстрее, чем в скалярных машинах), то проблема взаимодействия между потоками данных при распараллеливании становится несущественной. И то, что плохо распараллеливается на скалярных машинах, хорошо распараллеливается на векторных. Таким образом, системы PVP-архитектуры могут являться машинами общего назначения *(general purpose systems)*. Однако, поскольку векторные процессоры весьма дороги, эти машины не являются общедоступными.

Наиболее популярны следующие машины PVP-архитектуры:

1) CRAY SV-2, SMP-архитектура. Пиковая производительность системы в стандартной конфигурации может составлять десятки Тфлопс (терафлопс — 10^{12} операций с плавающей точкой в секунду);

2) NEC SX-6, NUMA-архитектура. Пиковая производительность системы может достигать 8 Тфлопс, производительность одного процессора составляет 8 Гфлопс. Система масштабируется до 128 узлов;

3) Fujitsu-VPP5000 (vector parallel processing), MPP-архитектура. Производительность одного процессора составляет 9,6 Гфлопс, пиковая производительность системы может достигать 1249 Гфлопс, максимальная емкость памяти — 8 Тбайт. Система масштабируется до 512 узлов.

Принципы программирования на PVP-системах предусматривают векторизацию циклов (для достижения оптимальной производительности одного процессора) и их распараллеливание (для одновременной загрузки нескольких процессоров одним приложением).

За счет большой физической памяти (доли терабайта), даже плохо векторизуемые задачи на PVP-системах решаются быстрее, чем на системах со скалярными процессорами.

Кластерная архитектура

Кластер представляет собой два или более компьютеров (часто называемых узлами), которые объединяются с помощью сетевых технологий на базе шинной архитектуры или коммутатора и предо-

ставляются пользователю в качестве единого информационно-вычислительного ресурса. В качестве узлов кластера могут выступать серверы, рабочие станции или обычные персональные компьютеры. Преимущество кластеризации для повышения работоспособности становится очевидным в случае сбоя какого-либо узла: при этом другой узел кластера может взять на себя нагрузку неисправного узла, и пользователи не заметят прерывания в доступе. Возможности масштабируемости кластеров позволяют многократно увеличивать производительность приложений для большего числа пользователей технологий (Fast/Gigabit Ethernet, Myrinet) на базе шинной архитектуры или коммутатора. Такие суперкомпьютерные системы являются самыми дешевыми, поскольку собираются на базе стандартных комплектующих элементов *(«off the shelf»),* процессоров, коммутаторов, дисководов и внешних устройств.

Кластеризация может быть осуществлена на разных уровнях компьютерной системы, включая аппаратное обеспечение, операционные системы, программы-утилиты, системы управления и приложения. Чем больше уровней системы объединены кластерной технологией, тем выше надежность, масштабируемость и управляемость кластера.

Типы кластеров. Условное деление на классы предложено Я. Радаевским и Д. Эдлайном:

Класс I. Машина строится целиком из стандартных деталей, которые продают многие продавцы компьютерных компонент (низкие цены, простое обслуживание, аппаратные компоненты доступны из различных источников).

Класс II. Система включает эксклюзивные или не широко распространенные детали. Этим можно достичь очень хорошей производительности, однако при более высокой стоимости.

Как уже указывалось ранее, кластеры могут существовать в различных конфигурациях. Наиболее употребляемыми типами кластеров являются:

- системы высокой надежности;
- системы для высокопроизводительных вычислений;
- многопоточные системы.

Отметим, что границы между этими типами кластеров до некоторой степени размыты, и часто существующий кластер может иметь такие свойства или функции, которые выходят за рамки перечисленных типов. Более того, при конфигурировании большого кластера, используемого как система общего назначения, приходится выделять блоки, выполняющие все перечисленные функции.

Кластеры для *высокопроизводительных вычислений* предназначены для параллельных расчетов. Эти кластеры обычно собраны из большого числа компьютеров. Разработка таких кластеров является сложным процессом, требующим на каждом шаге аккуратных согласований таких вопросов как инсталляция, эксплуатация и одновременное управление большим количеством компьютеров, технические требования параллельного и высокопроизводительного доступа к одному и тому же системному файлу (или файлам) и межпроцессорная связь между узлами и координация работы в параллельном режиме. Эти проблемы проще всего решаются при обеспечении единого образа операционной системы для всего кластера. Однако реализовать подобную схему удается далеко не всегда, и она обычно применяется лишь для не слишком больших систем.

Многопоточные системы используются для обеспечения единого интерфейса к различным ресурсам, которые могут со временем произвольно наращиваться (или сокращаться) в размере. Наиболее общий пример этого представляет собой группа Web-серверов.

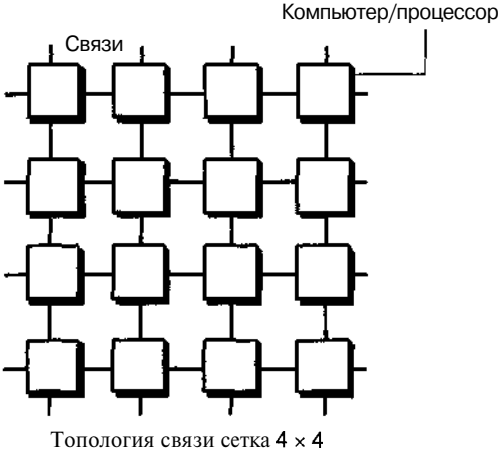
В 1994 г. Т. Стерлинг (Sterling) и Д. Беккер (Becker) создали 16-узловой кластер из процессоров Intel DX4, соединенных сетью Ethernet (10 Мбит/с) с дублированием каналов. Они назвали его «Beowulf» по названию старинной эпической поэмы. Кластер возник в центре NASA Goddard Space Flight Center для поддержки необходимыми вычислительными ресурсами проекта Earth and Space Sciences. Проектно-конструкторские работы над кластером быстро превратились в то, что известно сейчас под названием проект Beowulf.

Проект стал основой общего подхода к построению параллельных кластерных компьютеров и описывает многопроцессорную архитектуру, которая может с успехом использоваться для параллельных вычислений. Beowulf-кластер, как правило, является системой, состоящей из одного серверного узла (который обычно называется головным узлом), а также одного или нескольких подчиненных узлов (вычислительных узлов), соединенных посредством стандартной компьютерной сети. Система строится с использованием стандартных аппаратных компонент, таких, как ПК, стандартных сетевых адаптеров (например, Ethernet) и коммутаторов. Не существует особого программного пакета, называемого «Beowulf», вместо этого имеется несколько фрагментов программного обеспечения, которые многие пользователи нашли пригодными для построения кластеров Beowulf (такие программные продукты, как ОС Linux, системы передачи сообщений PVM, MPI, системы управления очередями заданий и другие стандартные продукты). Серверный узел контролирует весь кластер и обслуживает файлы, направляемые к клиентским узлам.

Связь процессоров в кластерной системе. Архитектура кластерной системы (способ соединения процессоров друг с другом) определяет ее производительность в большей степени, чем тип используемых в ней процессоров. Критическим параметром, влияющим на величину производительности такой системы, является расстояние между процессорами. Так, соединив вместе 10 персональных компьютеров, можно получить систему для проведения высокопроизводительных вычислений. Проблема, однако, будет состоять в нахождении наиболее эффективного способа соединения стандартных средств друг с другом, поскольку при увеличении производительности каждого процессора в 10 раз производительность системы в целом в 10 раз не увеличится.

Рассмотрим пример построения симметричной 16-процессорной системы, в которой все процессоры были бы равноправны. Наиболее естественным представляется соединение в виде плоской решетки, где внешние концы используются для подсоединения внешних устройств (рис. 3.18).

При таком типе соединения максимальное расстояние между процессорами окажется равным 6 (количество связей между процессорами, отделяющих самый ближний процессор от самого дальнего). Теория же показывает, что если в системе максимальное расстояние между процессорами больше 4, то такая система не может работать эффективно. Поэтому, при соединении 16 процессоров друг с другом плоская схема является неэффективной. Для получения более компактной конфигурации необходимо решить задачу о



Топология связи сетка 4 × 4

Рис. 3.18. Схема соединения процессоров в виде плоской решетки

нахождении фигуры, имеющей максимальный объем при минимальной площади поверхности.

В трехмерном пространстве таким свойством обладает шар. Но поскольку необходимо построить узловую систему, то вместо шара приходится использовать куб (если число процессоров равно 8, рис. 3.19, *а)* или гиперкуб, если число процессоров больше 8. Размерность гиперкуба будет определяться в зависимости от числа процессоров, которые необходимо соединить. Так, для соединения 16 процессоров потребуется 4-мерный гиперкуб (рис. 3.19, *б).* Для его построения следует взять обычный 3-мерный куб, сдвинуть в еще одном направлении и, соединив вершины, получить гиперкуб.

Архитектура гиперкуба является второй по эффективности, но самой наглядной. Используются и другие топологии сетей связи: трехмерный тор, «кольцо», «звезда» и другие (рис. 3.20).

Наиболее эффективной считается архитектура с топологией «толстого дерева» (fat-tree). Архитектура «fat-tree» (hypertree) предложена Лейзерсоном (Charles E. Leiserson) в 1985 г. Процессоры локализованы в листьях дерева, в то время как внутренние узлы дерева

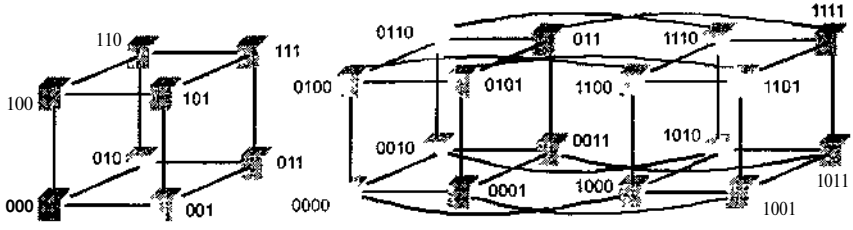


Рис. 3.19. Топологии связи:
а — трехмерный куб; *б* — четырехмерный гиперкуб

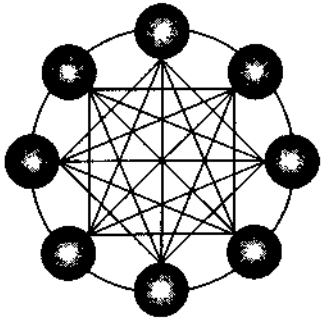
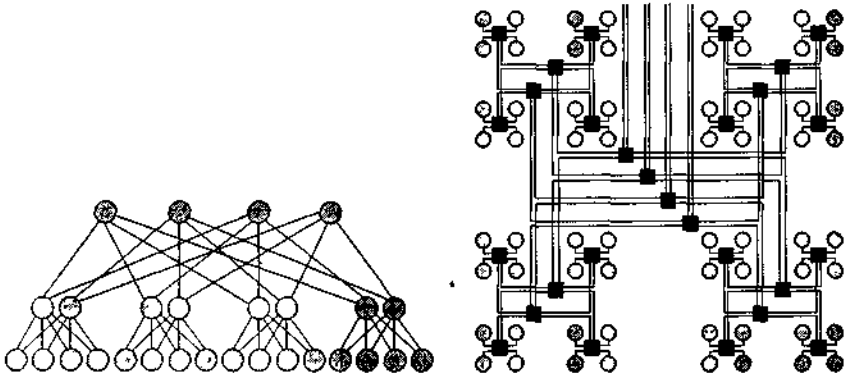


Рис. 3.20. Архитектура кольца с полной связью по хордам (Chordal Ring)



а *б*

Рис. 3.21. Кластерная архитектура «Fat-tree»
а — вид «сбоку», *б* — вид «сверху»

скомпонованы во внутреннюю сеть (рис 3.21). Поддеревья могут общаться между собой, не затрагивая более высоких уровней сети.

Поскольку способ соединения процессоров друг с другом больше влияет на производительность кластера, чем тип используемых в ней процессоров, то может оказаться более рентабельным создать систему из большего числа дешевых компьютеров, чем из меньшего числа дорогих В кластерах, как правило, используются операционные системы, стандартные для рабочих станций, чаще всего, свободно распространяемые — Linux, FreeBSD, вместе со специальными средствами поддержки параллельного программирования и балансировки нагрузки При работе с кластерами так же, как и с MPP системами, используют так называемую Massive Passing Programming Paradigm — парадигму программирования с передачей данных (чаще всего — MPI). Дешевизна подобных систем оборачивается большими накладными расходами на взаимодействие параллельных процессов между собой, что сильно сужает потенциальный класс решаемых задач.

3.3. Обобщенные представления об архитектуре вычислительных машин, систем и сетей

Рассматривая архитектуру ЭВМ, вычислительных систем, суперкомпьютеров и информационно-вычислительных сетей с общих позиций и абстрагируясь от деталей, можно воспользоваться следующей схемой (рис 3 22).

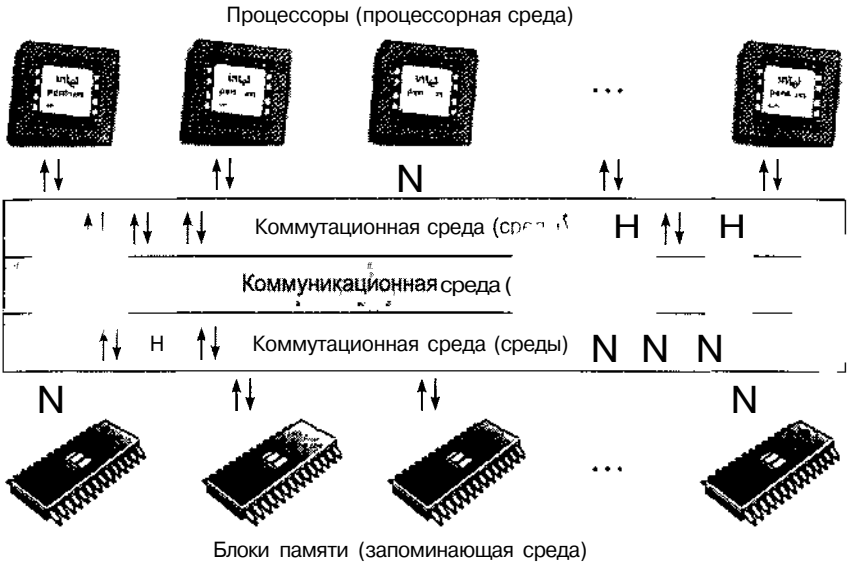


Рис. 3.22. Абстрактное представление об архитектурах ЭВМ, вычислительных систем и сетей

Здесь мы используем принципы классификации Скилликорна, построенной на следующих *элементах-объектах* (см. выше):

- *процессор команд* (IP — Instruction Processor) — функциональное устройство, работающее как интерпретатор команд; в системе, вообще говоря, может отсутствовать;
- *процессор данных* (DP — Data Processor) — функциональное устройство, работающее как преобразователь данных, в соответствии с арифметическими операциями;
- *иерархия памяти* (IM — Instruction Memory, DM — Data Memory) — запоминающее устройство, в котором хранятся данные и команды, пересылаемые между процессорами,
- *переключатель* — абстрактное устройство, обеспечивающее связь между процессорами и памятью

Таким образом, имеются:

- *процессоры и блоки памяти* — информационно-вычислительная среда,
- *средства коммутации и коммуникации* — коммуникационно-коммутационная среда.

Все эти компоненты активно присутствуют как в ЭВМ, так и в вычислительных сетях и системах (суперЭВМ).

Требования к архитектурным компонентам МВС

Эти аспекты имеют более широкий смысл, чем просто требования к техническим характеристикам компонент вычислительной системы: процессору, дисковым массивам, памяти, коммутаторам и т. п. аппаратным средствам. Гораздо более важное значение имеют требования, предъявляемые к вычислительной системе, которую собираются построить для реализации конкретных целей — решения задач определенного круга (научных, экономических, информационных систем и т. п.), модель программирования.

Разработчикам необходимо, прежде всего, проанализировать следующие связанные между собой вопросы:

- надежность и отказоустойчивость;
- масштабируемость;
- совместимость программного обеспечения;
- отношение стоимость/производительность.

Добиться дополнительного повышения производительности в МВС труднее, чем произвести масштабирование внутри узла. Основным барьером является трудность организации эффективных межузловых связей. Коммуникации, которые происходят между узлами, должны быть устойчивы к большим задержкам программно поддерживаемой когерентности. Приложения с большим количеством взаимодействующих процессов работают лучше на основе SMP-узлов, в которых коммуникационные связи более быстрые. В кластерах, как и в MPP-системах, масштабирование приложений более эффективно при уменьшении объема коммуникаций между процессами, работающими в разных узлах. Это обычно достигается путем разбиения данных.

Именно такой подход используется в наиболее известном приложении на основе кластеров OPS (Oracle Parallel Server).

Появление любого нового направления в вычислительной технике определяется требованиями компьютерного рынка. Поэтому у разработчиков компьютеров нет одной единственной цели. Большая универсальная вычислительная машина (мейнфрейм) или суперкомпьютер стоят дорого. Для достижения поставленных целей при проектировании высокопроизводительных конструкций приходится игнорировать стоимостные характеристики.

Суперкомпьютеры фирмы Cray Research и высокопроизводительные мейнфреймы компании IBM относятся именно к этой категории компьютеров. Другим крайним примером может служить конструкция, где производительность принесена в жертву для достижения низкой стоимости. К этому направлению относятся персональные компьютеры различных клонов IBM PC. Между этими

двумя крайними направлениями находятся конструкции, основанные на отношении стоимость/производительность, в которых разработчики находят баланс между стоимостными параметрами и производительностью. Типичными примерами такого рода компьютеров являются мини-компьютеры и рабочие станции

Для сравнения различных компьютеров между собой обычно используются стандартные методики измерения производительности. Эти методики позволяют разработчикам и пользователям использовать полученные в результате испытаний количественные показатели для оценки тех или иных технических решений, и, в конце концов, именно производительность и стоимость дают пользователю рациональную основу для решения вопроса, какой компьютер выбрать.

Масштабируемость. Масштабируемость представляет собой возможность наращивания числа и мощности процессоров, объемов оперативной и внешней памяти и других ресурсов вычислительной системы. Масштабируемость должна обеспечиваться архитектурой и конструкцией компьютера, а также соответствующими средствами программного обеспечения.

Так, например, возможность масштабирования кластера ограничена значением отношения скорости процессора к скорости связи, которое не должно быть слишком большим (реально это отношение для больших систем не может быть более 3—4, в противном случае не удается даже реализовать режим единого образа операционной системы). С другой стороны, последние годы развития процессоров и коммуникаторов показывают, что разрыв по скорости между ними все увеличивается.

Добавление каждого нового процессора в действительно масштабируемой системе должно давать прогнозируемое увеличение производительности и пропускной способности при приемлемых затратах. Одной из основных задач при построении масштабируемых систем является минимизация стоимости расширения компьютера и упрощение планирования. В идеале добавление процессоров к системе должно приводить к линейному росту ее производительности. Однако это не всегда так. Потери производительности могут возникать, например, при недостаточной пропускной способности шин из-за возрастания трафика между процессорами и основной памятью, а также между памятью и устройствами ввода/вывода. В действительности реальное увеличение производительности трудно оценить заранее, поскольку оно в значительной степени зависит от поведения прикладных задач.

Возможность масштабирования системы определяется не только архитектурой аппаратных средств, но и зависит от заложенных

свойств программного обеспечения. Масштабируемость программного обеспечения затрагивает все его уровни — от простых механизмов передачи сообщений до работы с такими сложными объектами, как мониторы транзакций и вся среда прикладной системы. В частности, программное обеспечение должно минимизировать трафик межпроцессорного обмена, который может препятствовать линейному росту производительности системы. Аппаратные средства (процессоры, шины и устройства ввода/вывода) являются только частью масштабируемой архитектуры, на которой программное обеспечение может обеспечить предсказуемый рост производительности. Важно понимать, что простой переход, например, на более мощный процессор может привести к перегрузке других компонентов системы. Это означает, что действительно масштабируемая система должна быть сбалансирована по всем параметрам.

Совместимость и мобильность программного обеспечения. Концепция программной совместимости впервые в широких масштабах была применена разработчиками системы IBM/360. Основная задача при проектировании всего ряда моделей этой системы заключалась в создании такой архитектуры, которая была бы одинаковой с точки зрения пользователя для всех моделей системы независимо от цены и производительности каждой из них. Огромные преимущества такого подхода, позволяющего сохранять существующий задел программного обеспечения при переходе на новые (как правило, более производительные) модели, были быстро оценены как производителями компьютеров, так и пользователями и, начиная с этого времени, практически все фирмы-поставщики компьютерного оборудования взяли на вооружение эти принципы, поставляя серии совместимых компьютеров. Следует заметить, однако, что со временем даже самая передовая архитектура неизбежно устаревает и возникает потребность внесения радикальных изменений в архитектуру и способы организации вычислительных систем.

В настоящее время одним из наиболее важных факторов, определяющих современные тенденции в развитии информационных технологий, является ориентация компаний-поставщиков компьютерного оборудования на рынок прикладных программных средств. Это объясняется, прежде всего, тем, что для конечного пользователя, в конце концов, важно программное обеспечение, позволяющее решить его задачи, а не выбор той или иной аппаратной платформы. Переход от однородных сетей программно-совместимых компьютеров к построению неоднородных сетей, включающих компьютеры разных фирм-производителей, в корне изменил и точку зрения на саму сеть: из сравнительно простого средства обмена информа-

цией она превратилась в средство интеграции отдельных ресурсов — мощную распределенную вычислительную систему, каждый элемент которой (сервер или рабочая станция) лучше всего соответствует требованиям конкретной прикладной задачи.

Этот переход выдвинул ряд новых требований. Прежде всего, такая вычислительная среда должна позволять гибко менять количество и состав аппаратных средств и программного обеспечения в соответствии с меняющимися требованиями решаемых задач. Во-вторых, она должна обеспечивать возможность запуска одних и тех же программных систем на различных аппаратных платформах, т. е. обеспечивать мобильность программного обеспечения. В-третьих, эта среда должна гарантировать возможность применения одних и тех же человеко-машинных интерфейсов на всех компьютерах, входящих в неоднородную сеть. В условиях жесткой конкуренции производителей аппаратных платформ и программного обеспечения сформировалась концепция открытых систем, представляющая собой совокупность стандартов на различные компоненты вычислительной среды, предназначенных для обеспечения мобильности программных средств в рамках неоднородной, распределенной вычислительной системы.

Одним из вариантов моделей открытой среды является модель OSE (Open System Environment), предложенная комитетом IEEE POSIX. На основе этой модели национальный институт стандартов и технологии США выпустил документ «Application Portability Profile (APP). The U.S. Government's Open System Environment Profile OSE/1 Version 2.0», который определяет рекомендуемые для федеральных учреждений США спецификации в области информационных технологий, обеспечивающие мобильность системного и прикладного программного обеспечения. Все ведущие производители компьютеров и программного обеспечения в США в настоящее время придерживаются требований этого документа.

Рассмотрим далее по порядку *процессоры, коммуникационные и коммутационные среды, блоки памяти.*

3.4. Перспективные типы процессоров ЭВМ

Ассоциативные процессоры

Существующие в настоящее время алгоритмы прикладных задач, системное программное обеспечение и аппаратные средства преимущественно ориентированы на традиционную адресную обра-

ботку данных. Данные должны быть представлены в виде ограниченного количества форматов (например, массивы, списки, записи), должна быть явно создана структура связей между элементами данных посредством указателей на адреса элементов памяти, при обработке этих данных должна быть выполнена совокупность операций, обеспечивающих доступ к данным по указателям. Такой подход обусловливает громоздкость операционных систем и систем программирования, а также служит препятствием к созданию вычислительных средств с архитектурой, ориентированной на более эффективное использование параллелизма обработки данных.

Ассоциативный способ обработки данных позволяет преодолеть многие ограничения, присущие адресному доступу к памяти, за счет задания некоторого критерия отбора и проведение требуемых преобразований только над теми данными, которые удовлетворяют этому критерию. Критерием отбора может быть совпадение с любым элементом данных, достаточным для выделения искомых данных из всех данных. Поиск данных может происходить по фрагменту, имеющему большую или меньшую корреляцию с заданным элементом данных.

Исследованы и в разной степени используются несколько подходов, различающихся полнотой реализации модели ассоциативной обработки. Если реализуется только ассоциативная выборка с последующим поочередным использованием найденных данных, то говорят об ассоциативной памяти или памяти, адресуемой по содержимому. При достаточно полной реализации всех свойств ассоциативной обработки используется термин *ассоциативный процессор* (рис. 3.23).

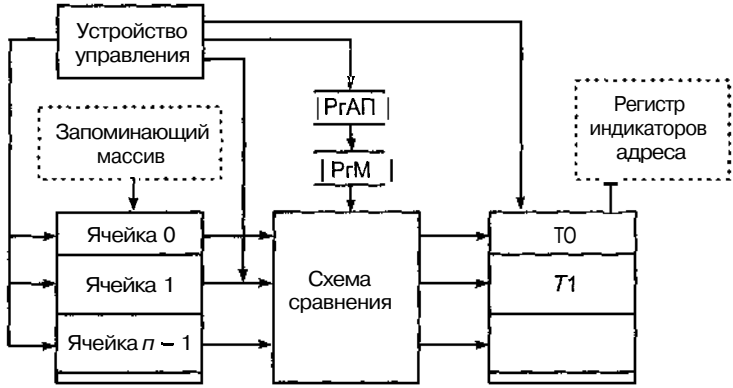


Рис. 3.23. Пример структуры ассоциативного процессора

Ассоциативные системы относятся к классу SIMD и включают некоторое множество операционных устройств, способных одновременно по командам управляющего устройства вести обработку нескольких потоков данных. В ассоциативных вычислительных системах информация на обработку поступает от ассоциативных запоминающих устройств (АЗУ), характеризующихся тем, что информация в них выбирается не по определенному адресу, а по ее содержанию.

Матричные процессоры

Наиболее распространенными из систем класса ОКМД (SIMD) являются матричные системы, которые наиболее приспособлены для решения задач, характеризующихся параллелизмом независимых объектов или данных. Организация систем подобного типа на первый взгляд достаточно проста. Они имеют общее управляющее устройство, генерирующее поток команд и большое число процессорных элементов, работающих параллельно и обрабатывающих каждая свой поток данных. Таким образом, производительность системы оказывается равной сумме производительностей всех процессорных элементов. Однако на практике, чтобы обеспечить достаточную эффективность системы при решении широкого круга задач, необходимо организовать связи между процессорными элементами, чтобы наиболее полно их загрузить. Именно характер связей между процессорными элементами и определяет различие свойств систем.

Одним из первых матричных процессоров был SOLOMON (рис. 3.24). Система SOLOMON содержит 1024 процессорных элемента, которые соединены в виде матрицы 32 х 32. Каждый элемент матрицы включает в себя процессор, обеспечивающий выполнение последовательных поразрядных арифметических и логических операций, а также оперативное ЗУ емкостью 16 Кбайт. Длина слова — переменная, от 1 до 128 разрядов, разрядность слов устанавливается программно. По каналам связи от устройства управления передаются команды и общие константы. В процессорном элементе используется так называемая *многомодальная логика,* которая позволяет каждому процессорному элементу выполнять (или нет) общую операцию в зависимости от значений обрабатываемых данных. В каждый момент все активные процессорные элементы выполняют одну и ту же операцию над данными, хранящимися в собственной памяти и имеющими один и тот же адрес.

Идея многомодальности заключается в том, что в каждом процессорном элементе имеется специальный регистр на четыре со-

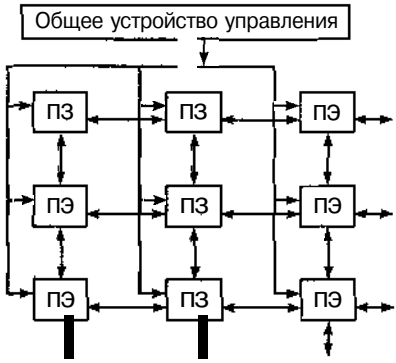


Рис. 3.24. Структура матричной вычислительной системы SOLOMON

стояния — регистр моды. Мода (модальность) заносится в этот регистр устройством управления. При выполнении последовательности команд модальность передается в коде операции и сравнивается с содержимым регистра моды. Если есть совпадения, то операция выполняется. В других случаях процессорный элемент не будет выполнять эту операцию, но может в зависимости от кода пересылать свои операнды соседнему процессорному элементу. Такой механизм позволяет выделить строку или столбец процессорных элементов, что эффективно при операциях над матрицами. Взаимодействуют процессорные элементы с периферийным оборудованием через внешний процессор.

Дальнейшим развитием матричных процессоров стала система ILLIAS-4, разработанная фирмой BARROYS. Первоначально система должна была включать в себя 256 процессорных элементов, разбитых на группы, каждый из которых должен управляться специальным процессором. Однако по различным причинам была создана система, содержащая одну группу процессорных элементов и управляющий процессор. Если в начале предполагалось достичь быстродействия в 1 млрд операций в секунду, то реальная система работала с быстродействием 200 млн операций в секунду. Эта система в течение ряда лет считалась одной из самых высокопроизводительных в мире.

Матричная система ПС-2000 (СССР)

В начале 80-х гг. в СССР была создана система ПС-2000, которая также является матричной (рис. 3.25). Основой этой системы является мультипроцессор ПС-2000, состоящий из решающего

поля и устройства управления мультипроцессором. Решающее поле строится из одного, двух, четырех или восьми устройств обработки, в каждом из которых восемь процессорных элементов. Мультипроцессор, состоящий из 64 процессорных элементов, обеспечивает быстродействие, равное 200 млн коротких операций в секунду.

Рис. 3.25. Система ПС-2000 (СССР)

В 1972—1975 гг. в Москве, в Институте проблем управления АН СССР (ИПУ РАН) была предложена структура и архитектура ПС-2000 — мультипроцессора ОКМД (рис. 3.26). Ее авторам удалось найти оригинальное структурное решение, которое соединило относительную простоту управления одним потоком команд с высокой гибкостью программирования высокопараллельной обработки информации. Найденные в ИПУ РАН структурные решения впервые сориентировали конструкторов на проектирование для таких задач недорогих высокопараллельных компьютеров с высокой производительностью в расчете на единицу стоимости. Предварительные исследования и расчеты подтвердились. Производительность серийных комплексов ПС-2000 достигла 200 млн операций в секунду.

В 1980 г. Госкомиссия приняла опытные образцы и санкционировала серийное производство ЭГВК. Сразу восемь экземпляров ЭГВК ПС-2000, демонстрировавшихся на геофизических задачах

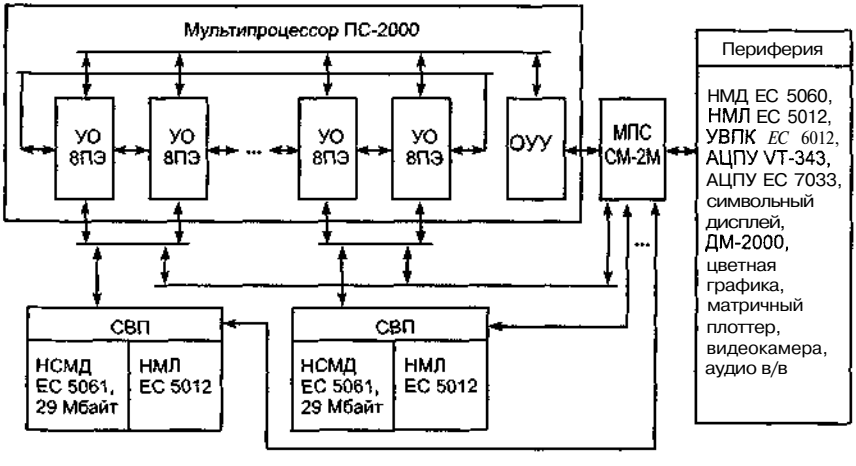


Рис. 3.26. Структура **ЭГВК** ПС-2000

(пакет программ в составе СОС-ПС (НПО «Геофизика», Москва)), давали суммарную производительность около миллиарда операций в секунду. Столь высокая производительность проблемно-ориентированных ЭГВК ПС-2000 достигалась на хорошо распараллеливаемых задачах, которые характерны для многих практических применений. При решении таких задач на ЭГВК ПС-2000 достигался рекордный «гражданский» показатель «производительность/стоимость».

С 1981 по 1988 гг. Северодонецким приборостроительным заводом Министерства приборостроения и средств автоматизации СССР было выпущено около 180 ЭГВК ПС-2000, мультипроцессоров ПС-2000 — 242 шт. Высокопроизводительные и недорогие вычислительные комплексы ПС-2000 работали в различных областях народного хозяйства во всех регионах страны и на специальных объектах.

Отечественное производство впервые в мире большим тиражом выпустило высокопроизводительную многопроцессорную вычислительную систему. В состав ЭГВК ПС-2000 входит собственно мультипроцессор, мониторная подсистема (МПС) и от одной до четырех подсистем внешней памяти (СВП). Мониторная подсистема на базе малой ЭВМ СМ-2М взяла на себя функции операционной системы, а также трансляцию, редактирование текстов, счет по вспомогательным программам, управление СВП и средствами отображения. При работе с физическими объектами в реальном времени возможно подключение потоков информации к мультипроцессору как через СВП, так и через специальные высокоскоростные каналы.

Мультипроцессор ПС-2000 ориентирован на высокопроизводительную обработку больших массивов информации по хорошо распараллеливаемым регулярным алгоритмам. Он обеспечивает однозадачный режим работы с одним потоком команд и многими потоками данных (SIMD-архитектура). Особенностью SIMD-архитектуры ПС-2000 является наличие значительных объемов регистровой памяти, в которой протекают массовые вычисления и межпроцессорные обмены, а также выполняется адресация распределенной оперативной памяти.

Мультипроцессор ПС-2000 (рис. 3.27) состоит из структурированной совокупности однотипных ПЭ и общего устройства управления. Конструктивно восемь ПЭ объединяются в УО. Каждый ПЭ содержит арифметико-логическое устройство с набором регистров общего назначения *S,* память *M,* устройство локальной адресной арифметики *L,* устройство активации ПЭ — *T,* фрагменты регулярного *B* и магистрального каналов. ОУУ содержит арифметико-логическое устройство с набором регистров общего назначения *W*, память данных *H*, адресную арифметику *HL*, память микрокоманд *G.*

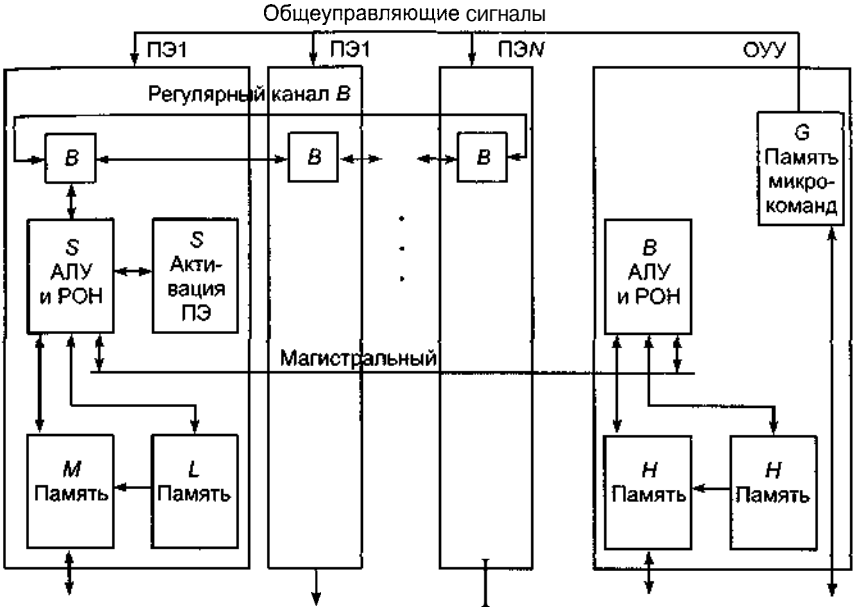


Рис. 3.27. Структура мультипроцессора ПС-2000

Устройство *S* за 0,32 мкс выполняет операции с фиксированной запятой над 24-разрядными регистровыми операндами, в нем также имеется аппаратная поддержка, обеспечивающая выполнение операций с плавающей запятой (сложение/вычитание за 0,96 мкс, умножение за 1,6 мкс). Объем памяти *M* и Я — 16 384 24-разрядных слова каждая, операции считывания или записи выполняются за 0,96 мкс. Объем памяти *G* — 16 384 64-разрядных слова. Время выдачи микрокоманды — 0,32 мкс, время выполнения команды ветвления от 1,28 до 1,92 мкс.

Все это позволяет мультипроцессору ПС-2000 с 64 ПЭ работать с эффективной производительностью 200 млн операций в секунду при выполнении расчетов с фиксированной запятой и 50 млн операций в секунду при выполнении одновременно нескольких вычислительных задач, содержащих операции с плавающей запятой. Таким образом, 64-процессорный ПС-2000, имея тактовую частоту 3 МГц, для пользователя работал с частотой 200 МГц.

Модульное конструктивное построение, неприхотливая элементная база, не требующая специальных условий охлаждения, и система программирования с гибкой системой настроек, позволяющая писать программы, не зависящие от числа ПЭ в ЭГВК

ПС-2000, обеспечили высокую живучесть и ремонтопригодность мультипроцессора ПС-2000, что позволило в условиях экспедиций обеспечить работу ЭГВК ПС-2000 более 20 ч в сутки.

Областью широкого использования ЭГВК ПС-2000 стала геофизика, которая объективно нуждалась в компьютерах такого класса. Для обработки данных сейсмической разведки месторождений нефти и газа во ВНИИ геофизики (Москва) при участии ИПУ РАН была создана система промышленной обработки геофизической информации. В отрасли успешно эксплуатировалось около 90 экспедиционных геофизических вычислительных комплексов ЭГВК ПС-2000, обеспечивающих углубленную обработку значительной части данных сейсморазведки нефти и газа.

На базе нескольких комплексов ПС-2000 были созданы высокопроизводительные (до 1 млрд операций в секунду) системы обработки гидроакустической и телеметрической информации в реальном масштабе времени.

Телеметрический вычислительный комплекс центра управления космическими полетами (ЦУП) использовал с 1986 г. вплоть до 1997 г. систему предварительной обработки телеметрической информации на базе ЭГВК ПС-2000, связанную в единый комплекс с центральной системой обработки на базе многопроцессорного вычислительного комплекса «Эльбрус-2». Высокий параллелизм обработки информации в ПС-2000 позволил реализовать новые алгоритмы обработки телеметрической информации. Первые комплексы ПС-2000 поступили в ЦУП в 1982 г., последние — в 1988 г. Всего было задействовано восемь 32-процессорных комплексов.

Клеточные и ДНК-процессоры

В настоящее время в поисках реальной альтернативы полупроводниковым технологиям создания новых вычислительных систем ученые обращают все большее внимание на биотехнологии, или *биокомпьютинг,* который представляет собой гибрид информационных и молекулярных технологий. Биокомпьютинг позволяет решать сложные вычислительные задачи, пользуясь методами, принятыми в биохимии и молекулярной биологии, организуя вычисления с помощью живых тканей, клеток, вирусов и биомолекул. Наибольшее распространение получил подход, где в качестве основного элемента (процессора) используются молекулы дезоксирибонуклеиновой кислоты. Центральное место в этом подходе занимает так называемый ДНК-процессор. Кроме ДНК в качестве биопроцессора могут быть использованы также белковые молекулы и биологические мембраны.

ДНК-процессоры. Так же, как и любой другой процессор, ДНК-процессор характеризуется структурой и набором команд. В данном случае структура процессора — это структура молекулы ДНК, а набор команд — это перечень биохимических операций над молекулами. Принцип устройства компьютерной ДНК-памяти основан на последовательном соединении четырех нуклеотидов (основных кирпичиков ДНК-цепи). Три нуклеотида, соединяясь в любой последовательности, образуют элементарную ячейку памяти — *кодон.* Кодоны затем формируют цепь ДНК. Основная трудность в разработке ДНК-компьютеров связана с проведением избирательных *однокодонных* реакций (взаимодействий) внутри цепи ДНК. Однако прогресс есть уже и в этом направлении. Уже существует экспериментальное оборудование, позволяющее работать с одним из 1020 кодонов или молекул ДНК. Другой проблемой является *самосборка* ДНК, приводящая к потере информации. Ее преодолевают введением в клетку специальных ингибиторов — веществ, предотвращающих химическую реакцию самосборки.

Теоретическое обоснование подобной возможности было сделано еще в 50-х годах прошлого века (Р. П. Фейманом). В деталях теория была проработана в 70-х годах Ч. Бенеттом и в 80-х М. Конрадом. Первый компьютер на базе ДНК был создан в 1994 г. американским ученым Л. Адлеманом. Он смешал в пробирке молекулу ДНК, в которой были закодированы исходные данные и специальным образом подобранные ферменты. В результате химической реакции структура ДНК изменилась таким образом, что в ней в закодированном виде был представлен ответ задачи. Поскольку вычисления проводились в ходе химической реакции с участием ферментов, на них было затрачено очень мало времени.

Вслед за работой Адлемана последовали другие. Л. Смит из Университета Висконсин решил с помощью ДНК задачу доставки четырех сортов пиццы по четырем адресам, которая подразумевала 16 вариантов ответа. Ученые из Принстонского университета решили комбинаторную шахматную задачу: с помощью РНК нашли правильный ход шахматного коня на доске из девяти клеток (всего их 512 вариантов).

Р Липтон из Принстона первым показал, как, используя ДНК, кодировать двоичные числа и решать проблему вычисления логического выражения. Суть ее в том, что, имея некоторое логическое выражение, включающее n логических переменных, нужно найти все комбинации значений переменных, делающих выражение истинным Задачу можно решить только перебором 2" комбинаций. Все эти комбинации легко закодировать с помощью ДНК, а дальше

действовать по методике Адлемана. Липтон предложил также способ взлома шифра DES (американский криптографический), трактуемого как своеобразное логическое выражение.

Первую модель биокомпьютера, правда, в виде механизма из пластмассы, в 1999 г. создал И. Шапиро из Вейцмановского института естественных наук. Она имитировала работу «молекулярной машины» в живой клетке, собирающей белковые молекулы по информации с ДНК, используя РНК в качестве посредника между ДНК и белком.

В 2001 г. Шапиро удалось реализовать вычислительное устройство на основе ДНК, которое может работать почти без вмешательства человека. Система имитирует машину Тьюринга — одну из фундаментальных абстракций вычислительной техники, которая теоретически может решить любую вычислительную задачу. По своей природе молекулы ДНК работают аналогичным образом, распадаясь и рекомбинируя в соответствии с информацией, закодированной в цепочках химических соединений. Разработанная установка кодирует входные данные и программы в состоящих из двух цепей молекулах ДНК и смешивает их с двумя ферментами.

Молекулы фермента выполняли роль аппаратного, а молекулы ДНК — программного обеспечения. Один фермент расщепляет молекулу ДНК с входными данными на отрезки разной длины в зависимости от содержащегося в ней кода, а другой — рекомбинирует эти отрезки в соответствии с их кодом и кодом молекулы ДНК с программой. Процесс продолжается вдоль входной цепи, и, когда доходит до конца, получается выходная молекула, соответствующая конечному состоянию системы.

Этот механизм может использоваться для решения самых разных задач. Хотя на уровне отдельных молекул обработка ДНК происходит медленно — с типичной скоростью от 500 до 1000 бит/с, что во много миллионов раз медленнее современных кремниевых процессоров, по своей природе она допускает массовый параллелизм. По оценкам Шапиро и его коллег, в одной пробирке может одновременно происходить триллион процессов, так что при потребляемой мощности в единицы нВт (10^{-9} ватт) может выполняться миллиард операций в секунду.

В конце февраля 2002 г. появилось сообщение, что фирма Olympus Optical претендует на первенство в создании коммерческой версии ДНК-компьютера, предназначенного для генетического анализа. Машина была создана в сотрудничестве с доцентом Токийского университета А. Тояма.

Компьютер, построенный Olympus Optical, имеет молекулярную и электронную составляющие. Первая из них осуществляет химические реакции между молекулами ДНК, обеспечивает поиск и выделение результата вычислений, вторая — обрабатывает информацию и анализирует полученные результаты.

Клеточные компьютеры. Клеточные компьютеры представляют собой самоорганизующиеся колонии различных «умных» микроорганизмов, в геном которых удалось включить некую логическую схему, которая могла бы активизироваться в присутствии определенного вещества. Для этой цели идеально подошли бы бактерии, стакан с которыми и представлял бы собой компьютер. Главным свойством компьютера такого рода является то, что каждая их клетка представляет собой миниатюрную химическую лабораторию. Если биоорганизм запрограммирован, то он просто производит нужные вещества. Достаточно вырастить одну клетку, обладающую заданными качествами, и можно легко и быстро вырастить тысячи клеток с такой же программой.

Основная проблема, с которой сталкиваются создатели клеточных биокомпьютеров, — организация всех клеток в единую работающую систему. На сегодняшний день практические достижения в области клеточных компьютеров напоминают достижения 20-х гг. в области ламповых и полупроводниковых компьютеров. Сейчас в Лаборатории искусственного интеллекта Массачусетского технологического университета создана клетка, способная хранить на генетическом уровне 1 бит информации. Также разрабатываются технологии, позволяющие единичной бактерии отыскивать своих соседей, образовывать с ними упорядоченную структуру и осуществлять массив параллельных операций.

В 2001 г. американские ученые создали трансгенные микроорганизмы (т. е. микроорганизмы с искусственно измененными генами), клетки которых могут выполнять логические операции И и ИЛИ.

Специалисты лаборатории Оук-Ридж, штат Теннесси, использовали способность генов синтезировать тот или иной белок под воздействием определенной группы химических раздражителей. Ученые изменили генетический код бактерий Pseudomonas putida таким образом, что их клетки обрели способность выполнять простые логические операции. Например, при выполнении операции И в клетку подаются два вещества (по сути — входные операнды), под влиянием которых ген вырабатывает определенный белок. Теперь ученые пытаются создать на базе этих клеток более сложные логические элементы, а также подумывают о возможности создания клетки, выполняющей параллельно несколько логических операций.

Потенциал биокомпьютеров очень велик. К достоинствам, выгодно отличающим их от компьютеров, основанных на кремниевых технологиях, относятся:

- более простая технология изготовления, не требующая для своей реализации столь жестких условий, как при производстве полупроводников;
- использование не бинарного, а тернарного кода (информация кодируется тройками нуклеотидов), что позволит при меньшем количестве шагов перебрать большее число вариантов при анализе сложных систем;
- потенциально исключительно высокая производительность, которая может составлять до 10^{14} операций в секунду за счет одновременного вступления в реакцию триллионов молекул ДНК;
- возможность хранить данные с плотностью, во много раз превышающей показатели оптических дисков;
- исключительно низкое энергопотребление.

Однако, наряду с очевидными достоинствами, биокомпьютеры имеют и существенные недостатки, такие, как:

- сложность со считыванием результатов — современные способы определения кодирующей последовательности не совершенны, сложны, трудоемки и дороги;
- низкая точность вычислений, связанная с возникновением мутаций, прилипанием молекул к стенкам сосудов и т. д.;
- невозможность длительного хранения результатов вычислений в связи с распадом ДНК в течение времени.

Хотя до практического использования биокомпьютеров еще очень далеко, и они вряд ли будут рассчитаны на широкие массы пользователей, предполагается, что они найдут достойное применение в медицине и фармакологии, а также с их помощью станет возможным объединение информационных и биотехнологий. Вероятно, в будущем их смогут использовать не только для вычислений, но и как своеобразные «нанофабрики» лекарств. Поместив подобное «устройство» в клетку, врачи смогут влиять на ее состояние, исцеляя человека от самых опасных недугов.

Коммуникационные процессоры

Коммуникационные процессоры — это микрочипы, являющие собой нечто среднее между жесткими специализированными интегральными микросхемами и гибкими процессорами общего назна-

чения. Коммуникационные процессоры программируются, как и привычные ПК-процессоры, но построены с учетом сетевых задач, оптимизированы для сетевой работы, и на их основе производители (как процессоров, так и другого оборудования) создают программное обеспечение для специфических приложений. Коммуникационный процессор имеет собственную память и оснащен высокоскоростными внешними каналами для соединения с другими процессорными узлами. Его присутствие позволяет в значительной мере освободить вычислительный процессор от нагрузки, связанной с передачей сообщений между процессорными узлами. Скоростной коммуникационный процессор с RISC-ядром позволяет управлять обменом данными по нескольким независимым каналам, поддерживать практически все распространенные протоколы обмена, гибко и эффективно распределять и обрабатывать последовательные потоки данных с временным разделением каналов.

Сама идея создания процессоров, предназначенных для оптимизации сетевой работы — и при этом достаточно универсальных для программной модификации, — родилась в связи с необходимостью устранить различия в подходах к созданию локальных сетей (различные подходы к архитектуре сети, классификации потоков и т. д.). Несомненно, истинной причиной бума сетевых процессоров стало ускорение темпов развития рынка. Когда рынок движется на «Internet-скорости», поставщики оборудования уже не могут тратить по два года на разработку специализированных микросхем для реализации конкретных сетевых функций. Эти два года (и вложенные деньги) будут потрачены зря, если рынок за это время уйдет в другом направлении. Выход один — разрабатывать процессоры, которые поставщики оборудования могут внедрить и выпустить в новом продукте в течение нескольких месяцев. Бум сетевых процессоров, окончательно оформившийся в середине 1999 г., не был кратким, и в последующие годы индустрия развивалась крайне бурно.

По прогнозам, рынок коммуникационных процессоров в 2004 г. составит около 2,9 млрд долл. и с увеличением объемов специальные микросхемы будут вытеснены стандартными сетевыми процессорами. Более «умеренные» аналитики считают, что у сетевых процессоров, без сомнения, есть будущее, но они смогут преобладать только на некоторых сегментах рынка, где необходимы укороченные циклы разработки, быстрота и гибкость.

Прогнозируется, что на этом рынке не будет преобладать какая-либо одна компания, как, например, Intel на рынке ПК. Однако считается, что Intel все же будет одним из ключевых игроков, разделив 2,9 млрд долл. с IBM, Motorola и дюжиной других компаний.

Серия коммуникационных процессоров INTEL IXP4xx построена на базе распределенной архитектуры XScale и включает мультимедийные возможности, а также развитые сетевые интерфейсы Ethernet. Сочетание высокой производительности и низкого энергопотребления позволяет эффективно применять коммуникационные процессоры INTEL не только в классических сетевых приложениях, но и для построения Internet-ориентированных встраиваемых систем промышленного назначения.

Эффективность работы промышленных предприятий сегодня напрямую зависит от гибкости применяемых систем автоматизированного управления. Крупные производственные установки требуют использования нескольких децентрализованных систем управления, связанных друг с другом мощной информационной сетью, способной работать в сложных промышленных условиях. Зачастую эти средства промышленной коммуникации призваны обеспечить возможность гибкого управления, программирования и контроля работы распределенных систем управления из удаленных диспетчерских пунктов. Осуществление этих целей возможно с помощью коммуникационных процессоров, предназначенных для подключения персональных компьютеров к промышленным информационным сетям.

Дополнительные возможности, обеспечиваемые коммуникационными процессорами, должны быть интересны, прежде всего, тем пользователям, которым необходимо осуществлять сложные транзакции или наладить прямую голосовую и видеопередачу в рамках сетевой инфраструктуры.

Процессоры баз данных

Процессорами (машинами) баз данных принято называть программно-аппаратные комплексы, предназначенные для выполнения всех или некоторых функций систем управления базами данных (СУБД). Если в свое время системы управления базами данных предназначались в основном для хранения текстовой и числовой информации, то теперь они рассчитаны на самые различные виды данных, в том числе графические, звуковые и видео. Процессоры баз данных выполняют функции управления и распространения, обеспечивают дистанционный доступ к информации через шлюзы, а также репликацию обновленных данных с помощью различных механизмов тиражирования.

Современные процессоры баз данных должны обеспечивать естественную связь накапливаемой в базах данных информации со средствами оперативной обработки транзакций и Internet-приложе-

ниями. Это должны быть системы, которые дают пользователям возможность в любой момент обратиться к корпоративным данным и проанализировать их вне зависимости от того, где эти данные размещаются.

Решение таких задач требует существенного увеличения производительности таких систем. Однако традиционная программная реализация многочисленных функций современных СУБД на ЭВМ общего назначения приводит к громоздким и непроизводительным системам с недостаточно высокой надежностью. Необходим поиск новых архитектурных и аппаратных решений. Интенсивные исследования, проводимые в этой области, привели к пониманию необходимости использования в качестве процессоров баз данных специализированных параллельных вычислительных систем. Создание такого рода систем связывается с реализацией параллелизма при выполнении последовательности операций и транзакций, а также конвейерной потоковой обработки данных.

Потоковые процессоры

Потоковыми называют процессоры, в основе работы которых лежит принцип обработки многих данных с помощью одной команды. Согласно классификации Флинна они принадлежат к SIMD-архитектуре. Технология SIMD позволяет выполнять одно и то же действие, например вычитание и сложение, над несколькими наборами чисел одновременно. SIMD-операции для чисел двойной точности с плавающей запятой ускоряют работу ресурсоемких приложений для создания контента, трехмерного рендеринга, финансовых расчетов и научных задач. Кроме того, усовершенствованы возможности 64-разрядной технологии MMX (целочисленных SIMD-команд); эта технология распространена на 128-разрядные числа, что позволяет ускорить обработку видео, речи, шифрование, обработку изображений и фотографий. Потоковый процессор повышает общую производительность, что особенно важно при работе с 3D-графическими объектами.

Существуют однопотоковые процессоры (Single-streaming processor — SSP) и многопотоковые процессоры (Multi-Streaming Processor — MSP).

Представителем потоковых процессоров является семейство процессоров Intel начиная с Pentium III, в основе работы которых лежит технология Streaming SIMD Extensions (SSE, потоковая обработка по принципу «одна команда — много данных»). Эта технология позволяет выполнять такие задачи, как обработка речи, коди-

рование и декодирование видео- и аудиоданных, разработка трехмерной графики и обработка изображений.

Представителями класса SIMD считаются матрицы процессоров: ILLIAC IV, ICL DAP, Goodyear Aerospace MPP, Connection Machine 1 и т. п. В таких системах единое управляющее устройство контролирует множество процессорных элементов. Процессорный элемент получает от устройства управления в каждый фиксированный момент времени команду и выполняет ее над своими локальными данными.

Нейронные процессоры

Одно из наиболее перспективных направлений разработки принципиально новых архитектур вычислительных систем тесно связано с созданием компьютеров нового поколения на основе принципов обработки информации, заложенных в искусственных нейронных сетях (НС).

Первые практические работы по искусственным нейросетям и нейрокомпьютерам начались еще в 40—50-е гг. Под *нейронной сетью* обычно понимают совокупность элементарных преобразователей информации, называемых *нейронами,* которые определенным образом соединены друг с другом каналами обмена информации *синаптическими связями.*

Одним из основных достоинств нейровычислителя является то, что его основу составляют относительно простые, чаще всего однотипные элементы, имитирующие работу нейронов мозга. Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Он обладает группой *синапсов* — однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет *аксон* — выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Общий вид нейрона приведен на рис. 3.28, *а.*

Каждый синапс характеризуется *величиной синаптической связи* или ее весом w_i, который по физическому смыслу эквивалентен электрической проводимости. Текущее состояние нейрона определяется как взвешенная сумма его входных сигналов:

$$s = \sum_{i=1}^{n} x_i w_i.$$

Выход нейрона есть функция его состояния: $y = f(s)$, которая называется *активационной.* Известны различные виды таких функций,

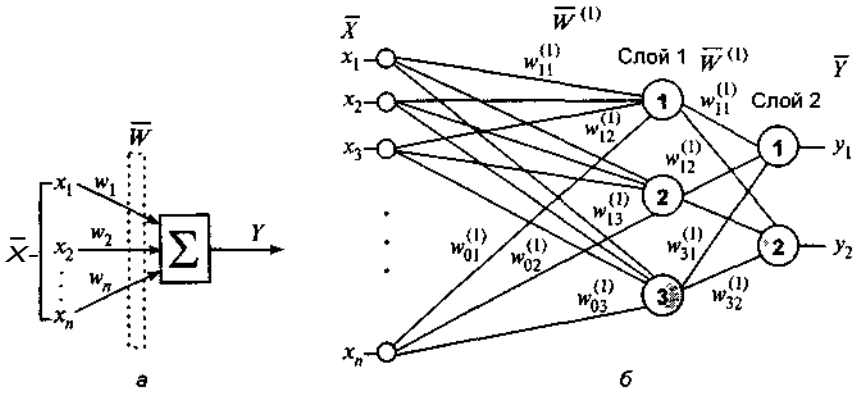


Рис. 3.28. Нейрон *(а)* и нейросеть (б)

некоторые из которых представлены на рис. 3.29. Одной из наиболее распространенных является нелинейная функция с насыщением, так называемая сигмоидальная (логистическая) функция:

$$f(x) = \frac{1}{1 + e^{-\alpha x}}.$$

Состояния нейронов изменяются в процессе функционирования и составляют кратковременную память нейросети. Каждый нейрон вычисляет взвешенную сумму пришедших к нему по синапсам сигналов и производит над ней нелинейное преобразование. При пересылке по синапсам сигналы умножаются на некоторый ве-

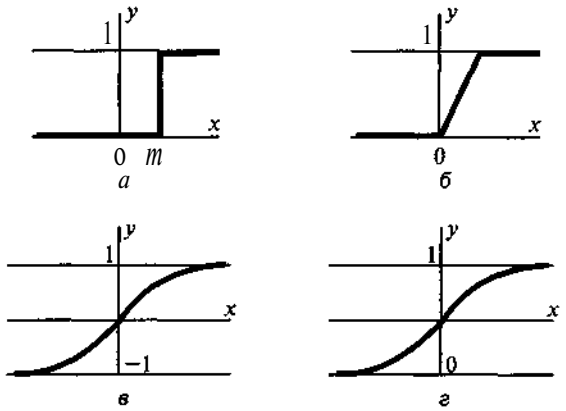


Рис. 3.29. Типовые активационные функции:
а — единичная пороговая функция; *б* — линейный порог (гистерезис); *в* — сигмоид (гиперболический тангенс); г — логистический сигмоид

совой коэффициент. В распределении весовых коэффициентов заключается информация, хранимая в ассоциативной памяти НС. При обучении и переобучении НС ее весовые коэффициенты изменяются. Однако они остаются постоянными при функционировании нейросети, формируя долговременную память.

НС может состоять из одного слоя, из двух и большего числа, однако, как правило, для решения практических задач более трех слоев в НС не требуется. Число входов НС определяет размерность гиперпространства, в котором входные сигналы могут быть представлены точками или гипероблостями из близко расположенных точек. Количество нейронов в слое сети определяет число гиперплоскостей в гиперпространстве. Вычисление взвешенных сумм и выполнение нелинейного преобразования позволяют определить, с какой стороны от той или иной гиперплоскости в гиперпространстве находится точка входного сигнала.

В нейрокомпьютерах используются принципы обработки информации, осуществляемые в реальных нейронных сетях. Это принципиально новые вычислительные средства с нетрадиционной архитектурой позволяют выполнять высокопроизводительную обработку информационных массивов большой размерности. В отличие от традиционных вычислительных систем нейросетевые вычислители, аналогично нейронным сетям, дают возможность с большей скоростью обрабатывать информационные потоки дискретных и непрерывных сигналов, содержат простые вычислительные элементы и с высокой степенью надежности позволяют решать информационные задачи обработки данных, обеспечивая при этом режим самоперестройки вычислительной среды в зависимости от полученных решений.

Вообще говоря, под термином *нейрокомпьютер* подразумевается довольно широкий класс вычислителей. Это происходит по той причине, что формально нейрокомпьютером можно считать любую аппаратную реализацию нейросетевого алгоритма от простой модели биологического нейрона до системы распознавания символов или движущихся целей. Нейрокомпьютеры не являются компьютерами в общепринятом смысле этого слова. В настоящее время технология еще не достигла того уровня развития, при котором можно было бы говорить о нейрокомпьютере общего назначения (который являлся бы одновременно искусственным интеллектом). Системы с фиксированными значениями весовых коэффициентов — вообще самые узкоспециализированные из нейросетевого семейства. Обучающиеся сети более гибки к разнообразию решаемых задач. Таким образом, построение нейрокомпьютера — это каждый раз широчай-

шее поле для исследовательской деятельности в области аппаратной реализации практически всех элементов НС.

В то же время технология интегральной электроники близка к исчерпанию своих физических возможностей. Геометрические размеры транзисторов больше нельзя физически уменьшать: при технологически достижимых размерах порядка 1 мкм и меньше проявляются физические явления, незаметные при больших размерах активных элементов — начинают сильно сказываться квантовые размерные эффекты. Транзисторы перестают работать как транзисторы.

Для аппаратной реализации НС необходим новый носитель информации. Таким новым носителем информации может быть свет, который позволит резко, на несколько порядков, повысить производительность вычислений.

Единственной технологией аппаратной реализации НС, способной в будущем прийти на смену оптике и оптоэлектронике, является нанотехнология, способная обеспечить не только физически предельно возможную степень интеграции субмолекулярных квантовых элементов с физически предельно возможным быстродействием, но и столь необходимую для аппаратной реализации НС трехмерную архитектуру.

Длительное время считалось, что нейрокомпьютеры эффективны для решения так называемых неформализуемых и плохо формализуемых задач, связанных с необходимостью включения в алгоритм решения задачи процесса обучения на реальном экспериментальном материале. В первую очередь к таким задачам относилась задача аппроксимации частного вида функций, принимающих дискретное множество значений, т. е. задача распознавания образов.

В настоящее время к этому классу задач добавляется класс задач, иногда не требующий обучения на экспериментальном материале, но хорошо представимый в нейросетевом логическом базисе. К ним относятся задачи с ярко выраженным естественным параллелизмом обработки сигналов, обработка изображений и др. Подтверждением точки зрения, что в будущем нейрокомпьютеры будут более эффективными, чем прочие архитектуры, может, в частности, служить резкое расширение в последние годы класса общематематических задач, решаемых в нейросетевом логическом базисе. К ним, кроме перечисленных выше, можно отнести задачи решения линейных и нелинейных алгебраических уравнений и неравенств большой размерности; систем нелинейных дифференциальных уравнений; уравнений в частных производных; задач оптимизации и других задач.

Процессоры с многозначной (нечеткой) логикой

Идея построения процессоров с *нечеткой логикой* (*fuzzy logic*) основывается на *нечеткой математике* Основанные на этой теории различные компьютерные системы, в свою очередь, существенно расширяют область применения нечеткой логики

Подходы нечеткой математики дают возможность оперировать входными данными, непрерывно меняющимися во времени, и значениями, которые невозможно задать однозначно, такими, например, как результаты статистических опросов В отличие от традиционной формальной логики, известной со времен Аристотеля и оперирующей точными и четкими понятиями типа «истина» и «ложь», «да» и «нет», «О» и «1», нечеткая логика имеет дело со значениями, лежащими в некотором (непрерывном или дискретном) диапазоне (рис. 3.30).

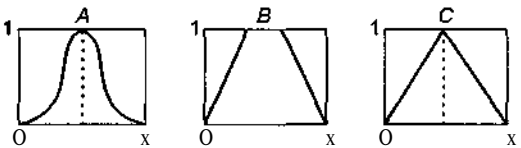


Рис. **3.30.** Различные типы функций принадлежности

Функция принадлежности элементов к заданному множеству также представляет собой не жесткий порог «принадлежит — не принадлежит», а линию, проходящую все значения от нуля до единицы. Теория нечеткой логики позволяет выполнять над такими величинами все логические операции — объединение, пересечение, отрицание и др. (рис 3.31).

Задачи с помощью нечеткой логики решаются по следующему принципу (рис. 3.32):

1) численные данные (показания измерительных приборов, результаты анкетирования) *фаззируются* (переводятся в нечеткий формат);

2) обрабатываются по определенным правилам;

3) *дефаззируются* и в виде привычной информации подаются на выход

В 1986 г. в AT&T Bell Labs создавались процессоры с «прошитой» нечеткой логикой обработки информации. В начале 90-х компания Adaptive Logic из США выпустила кристалл, сделанный по аналого-цифровой технологии. Он позволит сократить сроки конструирования многих встроенных систем управления реального вре-

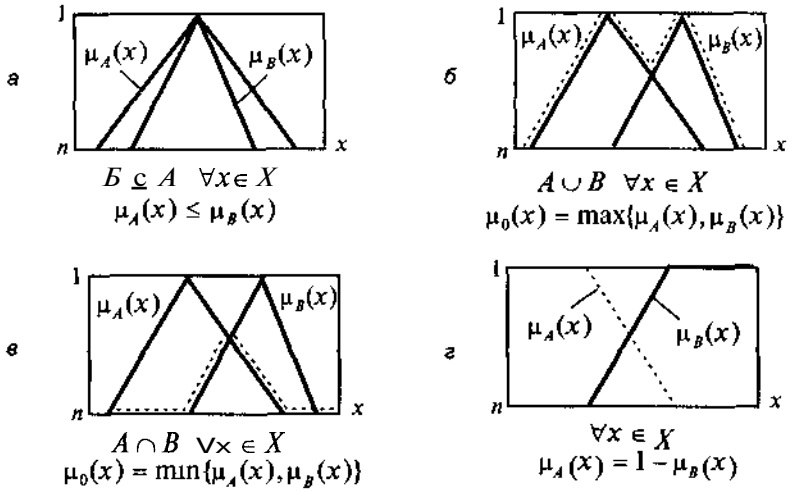


Рис. 3.31. Операции включения (*а*), объединения (б), пересечения (б) и дополнения *(г)* НМ

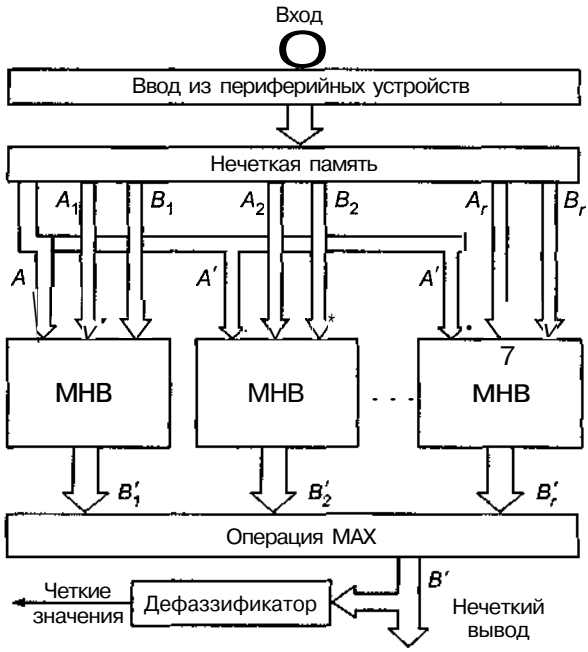


Рис. 3.32. Архитектура нечеткого компьютера
МНВ — механизм нечеткого вывода

мени, заменив собой традиционные схемы нечетких микроконтроллеров. Аппаратный процессор нечеткой логики второго поколения принимает аналоговые сигналы, переводит их в нечеткий формат, затем, применяя соответствующие правила, преобразует результаты в формат обычной логики и далее — в аналоговый сигнал. Все это осуществляется без внешних запоминающих устройств, преобразователей и какого бы ни было программного обеспечения нечеткой логики.

В Европе и США ведутся интенсивные работы по интеграции *fuzzy*-команд в ассемблеры промышленных контроллеров встроенных устройств (чипы Motorola 68HC11. 12. 21). Такие аппаратные средства позволяют в несколько раз увеличить скорость выполнения приложений и компактность кода по сравнению с реализацией на обычном ядре. Кроме того, разрабатываются различные варианты *fuzzy*-сопроцессоров, которые контактируют с центральным процессором через общую шину данных, концентрируют свои усилия на размывании/уплотнении информации и оптимизации использования правил (продукты Siemens Nixdorf).

Нечеткая логика не решит всех тех задач, которые не решаются на основе логики двоичной, но во многих случаях она удобнее, производительнее и дешевле. Разработанные на ее основе специализированные аппаратные решения (fuzzy-вычислители) позволят получить реальные преимущества в быстродействии. Если каскадировать *fuzzy*-вычислители, получается один из вариантов нейропроцессора или нейронной сети. Во многих случаях эти понятия объединяют, называя общим термином «*neuro-fuzzy logic*».

3.5. Системы памяти

Повышение производительности вычислительных систем непосредственно связано с увеличением быстродействия и емкости памяти. Емкость памяти наиболее крупных вычислительных систем возросла от 1000 байт до десятков терабайт, а время цикла уменьшилось с 20 мкс до 10 нс. Однако даже с учетом прогресса в технологии быстродействующие запоминающие устройства остаются более дорогими, чем медленные. Следовательно, с целью уменьшения стоимости ВС при той же производительности эффективнее иметь иерархию памяти с небольшим по емкости запоминающим устройством, расположенным рядом с процессором и имеющим минимальное время доступа. Такая иерархия позволяет согласовать характеристики памяти и центрального процессора.

С ростом сложности, а также размерности технических и научных задач возрастает требование к объему памяти. Рост объема памяти, в свою очередь, приводит к большим физическим размерам устройства, увеличению длины проводников и задержки распространения сигналов, т. е. уменьшается время доступа к такой памяти. Хотя быстродействие и емкость отдельных устройств памяти возросли, этого было недостаточно для того, чтобы они соответствовали характеристикам процессора. Конечно, можно реализовать память больших объемов, которая несколько быстрее имеющейся на данный момент с помощью совершеннейших технологий, однако стоимость такой памяти по отношению к производительности будет очень велика и ее использование — экономически невыгодно.

Иерархическая организация памяти

Рассмотренная ранее *двухуровневая система памяти* (регистры процессора и собственно оперативная память — ОЗУ) была характерна только на начальных этапах развития ЭВМ. В настоящее время используется *многоуровневая иерархическая модель памяти* (рис. 3.33).

Компромиссом между производительностью и объемами памяти является решение использовать иерархию запоминающих устройств, т. е. применять так называемую иерархическую модель памяти.

Иерархическая память — это система памяти, состоящая как минимум из двух запоминающих устройств, отличающихся быстродействием и емкостью. Обычно первое устройство памяти, расположенное в непосредственной близости от процессора, имеет малое

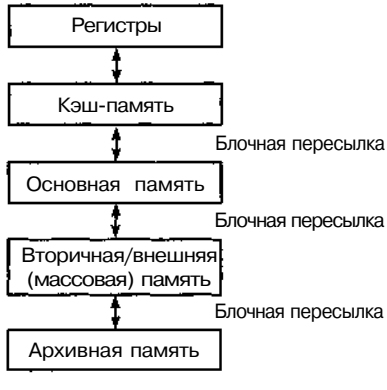


Рис. 3.33. Иерархическая модель памяти

время доступа и более высокую стоимость на бит. В связи с этим емкость более быстродействующей памяти в иерархии делается небольшой для того, чтобы оптимизировать стоимость всей системы. Технология памяти второго уровня иерархии выбирается исходя из низкой стоимости на один бит; такая память имеет большие значения времени цикла и времени доступа.

Иерархическая память управляется пользовательскими программами, аппаратным путем или системным программным обеспечением так, чтобы система памяти по времени доступа приближалась к быстродействующей памяти, а по стоимости — к медленной. Примером иерархии, управляемой системным программным обеспечением, является организация *виртуальной памяти со страничной организацией,* применяемая всеми современными ОС; недостаток емкости оперативной памяти возмещается внешней дисковой памятью.

Применение иерархических систем памяти оправдывает себя вследствие двух важных факторов — принципа локальности обращений и низкого (экономически выгодного) соотношения стоимость/производительность. Принцип *локальности обращений* состоит в том, что большинство программ обычно не выполняют обращений ко всем своим командам и данным равновероятно, а в каждый момент времени оказывают предпочтение некоторой части своего адресного пространства.

Иерархия памяти существует главным образом для того, чтобы повысить экономическую эффективность системы путем оптимального сочетания временных и стоимостных характеристик различных запоминающих устройств. Для того чтобы иерархия памяти была экономически эффективной, память должна иметь высокие эксплуатационные характеристики. Это означает, что технические параметры запоминающих устройств на разных уровнях иерархии должны существенно различаться. Невозможно достигнуть больших различий по эксплуатационным характеристикам и стоимости малой и большой памяти, если и та и другая выполнены по одной технологии и имеют близкие по значению времена доступа.

Кэш-память

Кэш-память или *cache memory* представляет собой буферное ЗУ, работающее со скоростью, обеспечивающей функционирование ЦП без режимов ожидания (рис. 3.34).

Необходимость в создании кэш-памяти возникла потому, что появились процессоры очень большого быстродействия. Между тем

Рис. 3.34. Структура кэш-памяти

для выполнения сложных прикладных процессов нужна большая память. Использование же большой сверхскоростной памяти невыгодно. Поэтому между ОЗУ и процессором стали устанавливать меньший по размерам высокоскоростной буфер, названный *кэш-памятью.* Более того, последнюю разделили на встроенную в процессор (L1) и внешнюю (L2). Встроенная кэш-память по сравнению с внешней имеет более высокое быстродействие и, естественно, стоимость. Поэтому первая меньше второй по емкости.

В кэш-память записывается часть команд и данных, содержащихся в оперативной памяти. При этом нередко используются две кэш-памяти, одна из которых хранит команды, а другая — данные. Методика выбора команд и данных, которые передаются из оперативной памяти в кэш-память, определяет скорость обработки данных, ибо может оказаться, что в некоторые моменты времени в кэш-памяти нет нужных команд или данных. В этих случаях процессор переходит в режим ожидания, а из оперативной памяти в кэш-память передается необходимая информация.

После окончания процесса обработки полученные данные должны быть записаны в память. Проще всего использовать сквозную запись для передачи этих данных в оперативную память. Но при этом оказывается, что во время этой передачи процессор находится в режиме ожидания. Чтобы избежать этого, все чаще применяют так называемую прямую обратную запись.

Применение кэширования особенно эффективно, когда доступ к данным осуществляется преимущественно в последовательном порядке. Тогда после первого запроса на чтение данных, расположенных в медленной (кэшируемой) памяти, можно заранее выполнить чтение следующих блоков данных в кэш-память для того, чтобы при следующем запросе на чтение данных почти мгновенно выдать их из кэш-памяти. Такой прием называется упреждающим чтением.

В системах с кэш-памятью быстрая и небольшая память, известная как буфер, или кэш, имеющая почти такое же быстродействие, как и регистры, расположена между процессором и основной памятью. Такой кэш согласует скорости процессора и основной памяти.

Этот буфер совершенно незаметен, недоступен пользователям и не может быть непосредственно адресован. Однако использование кэша создает иллюзию, что большая основная память работает со скоростью процессора.

Иерархия памяти обычно состоит из многих уровней, но в каждый момент времени мы имеем дело только с двумя близлежащими уровнями. Минимальная единица информации, которая может либо присутствовать, либо отсутствовать в двухуровневой иерархии, называется блоком или строкой. Размер блока обычно фиксирован, а объем памяти кратен размеру блока.

Успешное или неуспешное обращение к более высокому уровню называется соответственно *попаданием (hit) или промахом (miss).* Попадание — есть обращение к объекту в памяти, который найден на более высоком уровне, в то время как промах означает, что он не найден на этом уровне. *Доля попаданий (hit rate)* — доля обращений к данным, найденных на более высоком уровне. *Доля промахов (miss rate)* — это доля обращений к данным, которые не найдены на более высоком уровне. Частота попаданий и промахов является важной характеристикой. *Время обращения* при попадании *(hit time)* есть время обращения к более высокому уровню иерархии, которое включает в себя, в частности, и время, необходимое для определения того, является ли обращение попаданием или промахом. *Потери на промах (miss penalty)* есть время для замещения блока в более высоком уровне на блок из более низкого уровня плюс время для пересылки этого блока в требуемое устройство (обычно в процессор). Потери на промах далее включают в себя два компонента: *время доступа (access time)* — время обращения к первому слову блока при промахе и *время пересылки (transfer time)* — дополнительное время для пересылки оставшихся слов блока. Время доступа связано с задержкой памяти более низкого уровня, в то время как время пересылки связано с полосой пропускания канала между устройствами памяти двух смежных уровней.

Основные характеристики современных моделей кэш-памяти сведены в табл. 3.3.

В системе с кэш-памятью требуется идентификатор, или «признаковая память», для того, чтобы определить, какие части информации основной памяти были скопированы в кэш-память. В зависимости от организации обмена эти части могут быть словами, блоками или страницами. Специальная память, содержащая имена частей или адреса, обычно называется признаковой памятью, или справочником, и является ассоциативной.

Таблица 3.3. **Основные характеристики современных моделей кэш-памяти**

Характеристика	Типичное значение
Размер блока (строки)	4–128 байт
Время попадания (hittime)	1 -4 такта синхронизации (обычно 1 такт)
Потери при промахе (miss penalty)	8-32 такта синхронизации
Время доступа (access time)	6–10 тактов синхронизации
Время пересылки (transfer time)	2-22 такта синхронизации
Доля промахов (miss rate)	1–20 %
Размер кэш-памяти	4 Кбайт – 16 Мбайт

В буферной памяти должна быть логическая схема, определяющая порядок удаления слов или блоков из кэша при необходимости записи в него новой информации из основной памяти. Такая логическая схема называется приоритетным списком обновления данных.

Для синхронизации различных действий, например обращения в признаковую память, выборки и перемещения данных, в систему кэш-памяти вводится устройство управления.

Стратегии управления памятью

При построении систем с иерархической памятью ставится цель получения максимальной производительности подсистемы памяти при ее минимальной стоимости. Эффективность той или иной системы кэш-памяти зависит от стратегии управлений памятью. Стратегия управления памятью подразумевает ответ на такие вопросы: выбор метода отображения основной памяти в кэше; алгоритм взаимодействия между медленной основной и быстрой кэш-памятью; выбор стратегии замещения информации в кэше.

Существует три основных способа размещения блоков (строк) в кэш-памяти. По способу размещения блоков основной памяти в кэше различают кэш-память с *прямым отображением (direct-mapped cache), частично ассоциативную* кэш-память (или *множественно ассоциативную* кэш-память, *set-associative cache)* и *полностью ассоциативную* кэш-память *(fully associative cache).*

Кэш-память называется *памятью с прямым отображением,* если каждый блок основной памяти имеет только одно фиксированное место, на котором он может появиться в кэш-памяти. Это наиболее простая организация кэш-памяти. Все блоки основной памяти,

имеющие одинаковые младшие разряды в своем адресе, попадают в один блок кэш-памяти. При таком подходе справедливо соотношение:

Адрес блока кэш-памяти =
= (Адрес блока основной памяти) mod (Число блоков в кэш-памяти).

Кэш-память называется полностью ассоциативной, если некоторый блок основной памяти может располагаться на любом месте кэш-памяти. Кэш-память называется частично ассоциативной, если некоторый блок основной памяти может располагаться на ограниченном множестве мест.

В современных процессорах, как правило, используется либо кэш-память с прямым отображением, либо двух- (четырех-) канальная множественно-ассоциативная кэш-память.

Стратегии замещения информации в кэше определяет блок, подлежащий замещению при возникновении промаха. Простота при использовании кэша с прямым отображением заключается в том, что аппаратные решения здесь наиболее простые: легко реализуется сама аппаратура, легко происходит замещение данных. При замещении просто нечего выбирать — на попадание проверяется только один блок и только этот блок может быть замещен. При полностью ассоциативной или множественно-ассоциативной организации кэш-памяти имеются несколько блоков, из которых надо выбрать кандидата в случае промаха. Как правило, для замещения блоков применяются две основные стратегии: случайная и LRU-стратегия.

В первом случае, чтобы иметь равномерное распределение, блоки-кандидаты выбираются случайно. В некоторых системах, чтобы получить воспроизводимое поведение, которое особенно полезно во время отладки аппаратуры, используют псевдослучайный алгоритм замещения.

Во втором случае, чтобы уменьшить вероятность выбрасывания информации, которая скоро может потребоваться, все обращения к блокам фиксируются. Заменяется тот блок, который не использовался дольше всех (*LRU— Least-Recently Used).*

Достоинство случайного способа заключается в том, что его проще реализовать в аппаратуре. Когда количество блоков увеличивается, алгоритм LRU становится все более дорогим и часто только приближенным. В табл. 3.4 показаны различия в долях промахов при использовании алгоритма замещения LRU и случайного алгоритма.

Таблица 3.4. **Влияние стратегии замещения на долю промаха при разных размерах кэш-памяти и ассоциативности**

Ассоциативность, размер кэш-памяти, Кбайт	2-канальная стратегия		4-канальная стратегия		8-канальная стратегия	
	LRU	Random	LRU	Random	LRU	Random
16	5,18	5,69	4,67	5,29	4,39	4,96
64	1,88	2,01	1,54	1,66	1,39	1,53
256	1,15	1,17	1,13	1,13	1,12	1,12

Из таблицы видно, что при большом размере кэш-памяти стратегии замещения несущественно влияют на эффективность памяти.

Организация памяти в однопроцессорных ВС

Однопроцессорные ВС были первыми вычислительными системами, в которых была реализована иерархическая структура памяти. Как правило, однопроцессорные ВС представляют собой совокупность процессора, памяти и внешних устройств (ВНУ), соединенных с помощью общей шины (рис. 3.35).

Применение кэша в таких системах обусловлено только экономической эффективностью функционирования систем с кэш-памятью:

- алгоритм сквозной записи *(Write Through)* или сквозного накопления *(Store Through);*
- алгоритм простого свопинга *(Simple Swapping)* или обратной записи *(Write Back);*
- алгоритм свопинга с флагами *(Flag Swapping)* или обратной записи в конфликтных ситуациях с флагами *(CUX)*;
- алгоритм регистрового свопинга с флагами *(FRS).*

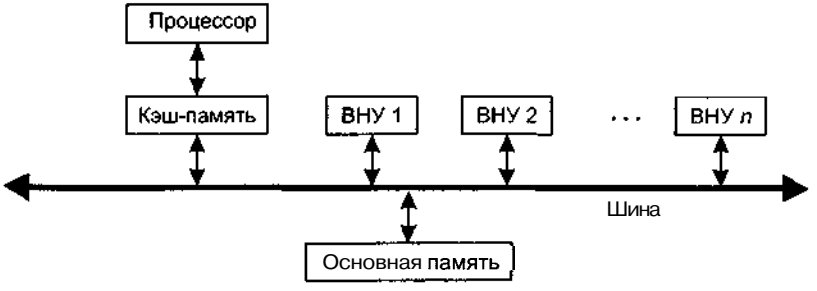


Рис. 3.35. Однопроцессорные ВС с иерархической системой памяти

Алгоритм сквозной записи. Это самый простой алгоритм свопинга. Каждый раз при появлении запроса на запись по некоторому адресу обновляется содержимое области по этому адресу как в быстрой, так и в основной памяти, даже если копия содержимого по этому адресу находится в быстром буфере. Такое постоянное обновление содержимого основной памяти, как и буфера, при каждом запросе на запись позволяет постоянно поддерживать информацию, находящуюся в основной памяти, в обновленном состоянии.

Поэтому, когда возникает запрос на запись по адресу, относящемуся к области, содержимое которой не находится в данный момент в быстром буфере, новая информация записывается просто на место блока, которое предполагается переслать в основную память (без необходимости пересылки этого слова в основную память), так как в основной памяти уже находится его достоверная копия.

Данный алгоритм несложен для реализации и понимания, поэтому он был широко распространен в системах с кэш-памятью. Данный алгоритм также просто позволяет реализовать когерентность данных для мультипроцессорных систем с раздельными кэшами и общей памятью.

Обратной стороной простоты алгоритма является его малая эффективность: в нем не заложена тенденция к минимизации доли об-

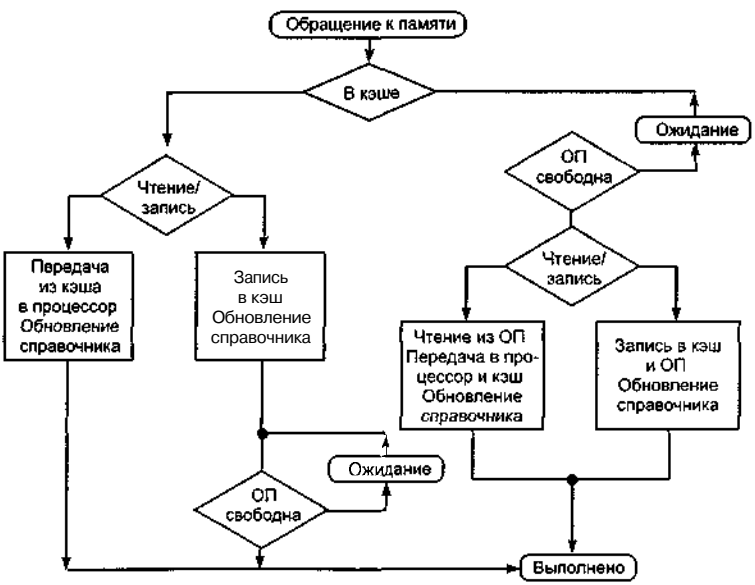


Рис. 3.36. Блок-схема алгоритма сквозной записи

ращений к основной памяти При увеличении объема кэш-памяти доля обращений к основной памяти асимптоматически приближается к доле обращений для записи в основную память (а не к нулю). На рис 3.36 приведена блок-схема алгоритма сквозной записи.

Алгоритм простого свопинга. Данный алгоритм не сложнее алгоритма сквозной записи. Обращения к основной памяти имеют место в тех случаях, когда в быстром буфере не обнаруживается нужное слово. Эта схема свопинга повышает производительность системы памяти, так как в ней обращения к основной памяти не происходят при каждом запросе на запись, что имеет место при использовании алгоритма сквозной записи. Однако в связи с тем, что содержимое основной памяти не поддерживается в постоянно обновленном состоянии, если необходимого слова в быстром буфере не обнаруживается, из буфера в основную память надо возвратить какое-либо устаревшее слово, чтобы освободить место для нового необходимого слова. Поэтому из буфера в основную память сначала пересылается какое-то слово, место которого занимает в буфере нужное слово. Таким образом, происходят две пересылки между быстрым буфером и основной памятью.

Алгоритм свопинга с флагами. Данный алгоритм является улучшением алгоритма простого свопинга. В алгоритме простого свопинга, когда в кэш-памяти не обнаруживается нужное слово, происходит два обращения к основной памяти — запись удаляемого значения из кэша и чтение нового значения в кэш. Если слово с того момента, как оно попало в буфер из основной памяти, не подвергалось изменениям, т. е. по его адресу не производилась запись (оно использовалось только для чтения), то нет необходимости пересылать его обратно в основную память, потому что в ней и так имеется достоверная его копия; это обстоятельство позволяет в ряде случаев обойтись без обращений к основной памяти. Если, однако, слово подвергалось изменениям с тех пор, когда его копия была в последний раз записана обратно в основную память, то приходится перемещать его в основную память. Отслеживать изменения слова можно, пометив слово (блок) дополнительным флаговым битом. Изменяя значение флагового бита при изменении слова, можно сформировать информацию о состоянии слова; пересылать в основную память необходимо лишь те слова, флаги которых оказываются в установленном состоянии.

Алгоритм регистрового свопинга с флагами. Повышение эффективности алгоритма свопинга с флагами возможно за счет уменьшения эффективного времени цикла, что можно получить при введении регистра (регистров) временного хранения между кэш-памятью

и основной памятью. Теперь, если данные должны быть переданы из быстрого буфера в основную память, они сначала пересылаются в регистр (регистры) временного хранения; новое слово сразу же пересылается в буфер из основной памяти, а уже потом слово, временно хранившееся в регистре, записывается в основную память. Действия в ЦП начинают опять выполняться, как только для этого возникает возможность. Алгоритм обеспечивает совмещение операций записи в основную память с обычными операциями над буфером, что обеспечивает еще большее повышение производительности.

Способы улучшения эффективности подсистемы памяти в однопроцессорных ВС. Прежде всего, эффективное быстродействие кэша должно быть примерно равно быстродействию процессора. Если полное эффективное время обращения к системе памяти с использованием буфера превышает время базового цикла процессора, то для уравнивания ее быстродействия с быстродействием процессора можно прибегнуть к конвейерной организации процесса обработки информации. Когда кэш-память работает с такой же скоростью, как процессор, никакой необходимости в конвейерной работе процессора нет. Делать быстродействие кэш-памяти более высоким, чем быстродействие процессора, не имеет смысла, так как процессор (или процессоры) требует обслуживания со стороны памяти с интервалом, не меньшим чем время его рабочего цикла.

Современные процессоры выполняют команды со скоростью от нескольких команд в такт до нескольких десятков тактов на команду. Кроме того, используется конвейерная обработка команд. Для увеличения своей производительности процессоры обычно вначале производят выборку предполагаемо необходимых данных и команд, а затем исполняют их. Исходя из таких положений, необходимо добиваться того, чтобы кэш-память действовала со скоростью тактовой частоты процессора.

На производительность памяти очень сильно влияет алгоритм свопинга. Как отмечалось выше, наиболее эффективным алгоритмом для однопроцессорных ВС является регистровый свопинг с флагами. При увеличении размера кэш-памяти стратегия замещения данных не имеет значения, поэтому можно применять наиболее простую — случайную стратегию.

Необходимо также использовать совмещение работы буферной и основной памяти.

Большой класс задач, решаемых на ЭВМ, имеет дело с большими объемами данных, хранимых в массивах. Для улучшения производительности ВС на таких задачах можно применять многомодульную систему основной памяти с горизонтальным расслоением (рис. 3.37).

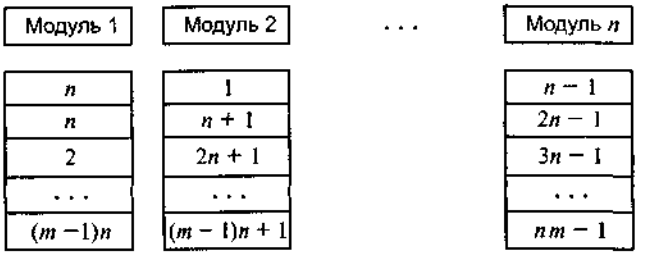


Рис. 3.37. Память с горизонтальным расслоением

При такой организации основной памяти при последовательном обращении к элементам массивов наблюдается увеличение производительности подсистемы памяти, так как кэш ждет меньшее время до освобождения памяти. Обратной стороной увеличения производительности является уменьшение надежности памяти, поскольку при выходе из строя одного модуля из адресного пространства выпадают ячейки, расположенные через некоторый шаг, что повлияет на все процессы, находящиеся в памяти. Для увеличения производительности и надежности одновременно можно применять так называемую блочно-модульную структуру памяти. Данный подход характерен для многопроцессорных ВС.

Иерархическая память многопроцессорных ВС

В структурной организации многопроцессорной системы наиболее существенен способ связи между процессорами и памятью системы.

Как известно, параллельные вычислительные системы делятся на два больших класса: SIMD и MIMD. В настоящее время осваиваются супервычисления на системах из микропроцессоров с кэш-памятью и разделяемой — логически общей и физически распределенной основной памятью.

Существующие параллельные вычислительные средства класса MIMD образуют три подкласса: симметричные мультипроцессоры (SMP), кластеры и массово-параллельные системы (MPP).

Однако степень масштабируемости SMP-систем ограничена в пределах технической реализуемости одинакового для всех процессоров доступа в память со скоростью, характерной для однопроцессорных компьютеров. На данный момент SMP-структура наиболее распространена в классе профессиональных рабочих станций на базе RISC-процессоров.

Для построения систем с большим числом процессоров применяются кластерный или MPP-подходы. Оба эти направления используют SMP как системообразующий вычислительный модуль (ВМ).

Кластерная система образуется из модулей, объединенных системой связи или разделяемыми устройствами внешней памяти, например дисковыми массивами. В настоящее время для образования кластерных систем используются либо специализированные фирменные средства (например, MEMORY CHANNEL фирмы DEC), либо универсальные локальные и глобальные сетевые технологии, такие, как Ethernet, FDDI (Fiber Distributed Data Interface), и другие сети, например, с протоколами TCP/IP (Transmission Control Protocol/Internet Protocol), либо дисковые массивы с высокоскоростными широкими двойными (Wide/Fast) и квадро PCI SCSI-контроллерами. Размер кластера варьируется от нескольких модулей до нескольких десятков модулей.

Массово-параллельные системы, в отличие от кластеров, имеют более скоростные, как правило, специализированные каналы связи между ВМ, а также широкие возможности по масштабированию. Кроме того, в MPP фиксируется некоторый достаточно высокий уровень интерфейса прикладных программ (API), поддерживаемый распределенной ОС. Однако поддержка работоспособности и оптимизация загрузки процессоров в MPP менее развита по сравнению с кластерами в силу разнообразности исполняемых программ и отсутствия функциональных связей между программами.

Начиная с 1980 г. идея SMP-архитектур, подкрепленная широким распространением микропроцессоров, стимулировала многих разработчиков на создание небольших мультипроцессоров, в которых несколько процессоров разделяют одну физическую память, соединенную с ними с помощью разделяемой шины. Из-за малого размера процессоров и заметного сокращения требуемой полосы пропускания шины, достигнутого за счет возможности реализации достаточно большой кэш-памяти, такие машины стали исключительно эффективными по стоимости. В первых разработках подобного рода машин удавалось разместить весь процессор и кэш на одной плате, которая затем вставлялась в заднюю панель, с помощью которой реализовывалась шинная архитектура. Современные конструкции позволяют разместить до четырех процессоров на одной плате, предел количества процессоров в SMP — не более 32.

Многопроцессорная ВС с общим кэшем и общей памятью. Общий вид такой многопроцессорной ВС представлен на рис. 3.38. Здесь

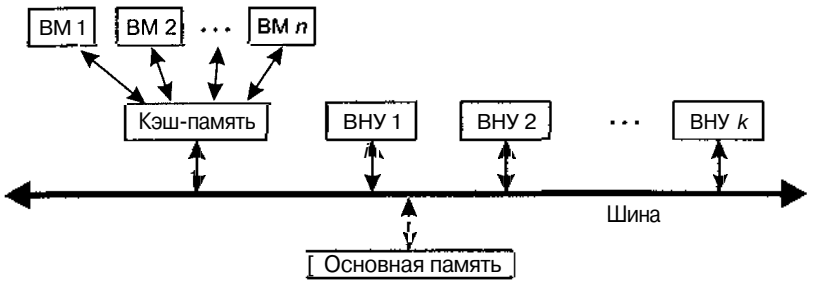


Рис. 3.38. Многопроцессорная ВС с общим кэшем и общей памятью

ВМ — вычислительный модуль (процессор), всего n штук, ВНУ — внешние устройства, всего k штук.

В качестве основной памяти может использоваться память с горизонтальным расслоением или же для повышения надежности — блочно-модульная память.

Основными проблемами в данной архитектуре являются следующие: скорость работы кэш-памяти, блокировка при совместном доступе нескольких процессоров на одновременную запись или запись/чтение данных по одному и тому же адресу. Кэш-память должна работать с тактовой частотой в n раз большей тактовой частоты процессора, так как она должна обеспечивать данными процессоры с частотой один раз в такт. В противном случае каждый процессор должен быть блокирован до момента получения (или записи) им требуемых данных, что снижает производительность многопроцессорной ВС.

Вообще такая структура сходна с однопроцессорной ВС с кэшем. Она недостаточно эффективна и, как правило, практически не применяется.

Многопроцессорная ВС с раздельными кэшами и общей памятью. Данная модель мультипроцессорной системы наиболее распространена в настоящее время. Структурная схема многопроцессорной ВС с раздельными кэшами и общей памятью представлена на рис. 3.39.

Каждый ВМ имеет собственную локальную кэш-память, имеется общая разделяемая основная память, все вычислительные модули (ВМ) подсоединены к основной памяти посредством шины. К шине подключены также внешние устройства. Все действия с использованием транзакций шины, производимые ВМ и внешними устройствами, с копиями строк, как в каждой кэш-памяти, так и в основной памяти, доступны для отслеживания всем ВМ. Это является следствием того, что в каждый момент на шине передает только один, а воспринимают все абоненты, подключенные к шине.

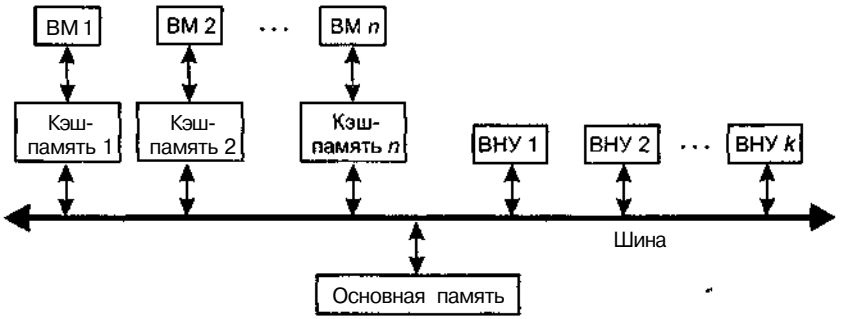


Рис. 3.39. Многопроцессорная ВС с раздельными кэшами и общей памятью

По сравнению с многопроцессорными ВС с общим кэшам и общей памятью в данной системе наряду с проблемой пропускной способности шины возникает еще и проблема когерентности кэш-памяти.

В такой вычислительной системе кэши могут содержать как разделяемые, так и частные данные. Частные данные — это данные, которые используются одним процессором, в то время как разделяемые данные используются многими процессорами, по существу обеспечивая обмен между ними Когда кэшируется элемент частных данных, их значение переносится в кэш для сокращения среднего времени доступа, а также требуемой полосы пропускания. Поскольку никакой другой процессор не использует эти данные, этот процесс идентичен процессу для однопроцессорной машины с кэш-памятью. Если кэшируются разделяемые данные, то разделяемое значение реплицируется и может содержаться в нескольких кэшах. Кроме сокращения задержки доступа и требуемой полосы пропускания такая репликация данных способствует также общему сокращению количества обменов. Таким образом, кэширование разделяемых данных вызывает проблему когерентности кэш-памяти.

3.6. Коммуникационные среды

Принципы построения коммуникационных сред

В самом общем смысле архитектуру компьютера можно определить как способ соединения процессоров между собой, с памятью и с внешними устройствами. Реализация этого соединения может идти различными путями. Конкретная реализация соединений такого рода называется коммуникационной средой компьютера.

Одна из самых простых реализаций — это использование *общей шины,* к которой подключаются как процессоры, так и память. Сама шина состоит из определенного числа линий связи, необходимых для передачи адресов, данных и управляющих сигналов между процессором и памятью. Этот способ реализован в SMP-системах. Основным недостатком таких систем, как было указано ранее, является плохая масштабируемость Увеличение, даже незначительное, числа устройств на шине вызывает заметные задержки при обмене с памятью и катастрофическое падение производительности системы в целом. Необходимы другие подходы для построения коммуникационной среды, и одним из них является разделение памяти на независимые модули и обеспечение возможности доступа разных процессоров к различным модулям одновременно посредством использования различного рода коммутаторов.

При этом возможны различные конфигурации получающихся систем связи. Так, в компьютерах семейства Cray T3D/T3E все процессоры были объединены специальными высокоскоростными каналами в трехмерный тор, в котором каждый вычислительный узел имел непосредственные связи с шестью соседями. В компьютерах IBM SP/2 взаимодействие процессоров происходит через иерархическую систему коммутаторов, также обеспечивающую возможность соединения каждого процессора с любым другим. Эти оригинальные уникальные решения значительно увеличивают цену компьютеров.

Существенно более простым и более дешевым оказалось использование системы связи на базе Ethernet, разработанной фирмой Xerox. Первоначально использовалась обычная 10-мегабитная сеть, затем стали применять Fast Ethernet, а в последнее время — Gigabit Ethernet Однако для Fast Ethernet характерна большая латентность (задержка в передаче данных), оцениваемая в 160—180 мкс, а Gigabit Ethernet отличается высокой стоимостью. Поэтому при создании многопроцессорных вычислительных систем часто предпочтение отдается технологиям SCI, Myrinet или Raceway.

Примеры построения коммуникационных сред на основе масштабируемого когерентного интерфейса SCI

SCI (Scalable Coherent Interface) принят как стандарт в 1992 г. (ANSI/IEEE Std 1596—1992). Предназначен для достижения высоких скоростей передачи с малым временем задержки, при этом обеспечивая масштабируемую архитектуру, позволяющую строить системы, состоящие из множества блоков. Представляет собой комбинацию шины и локальной сети, обеспечивает реализацию коге-

рентности кэш-памяти, размещаемой в узле SCI, посредством механизма распределенных директорий, который улучшает производительность, скрывая затраты на доступ к удаленным данным в модели с распределенной разделяемой памятью. Производительность передачи данных обычно находится в пределах от 200 до 1000 Мбайт/с на расстояниях десятков метров с использованием электрических кабелей и километров с использованием оптоволокна. SCI уменьшает время межузловых коммуникаций по сравнению с традиционными схемами передачи данных в сетях путем устранения обращений к программным уровням — операционной системе и библиотекам времени выполнения; коммуникации представляются как часть простой операции загрузки данных процессором (командами *load* или *store).*

Обычно обращение к данным, физически расположенным в памяти другого вычислительного узла и не находящимся в кэше, приводит к формированию запроса на удаленный узел для получения необходимых данных, которые в течение нескольких микросекунд доставляются в локальный кэш, и выполнение программы продолжается. Старый подход требовал формирования пакетов на программном уровне с последующей передачей их аппаратному обеспечению. Точно также происходил и прием, в результате чего задержки были в сотни раз больше, чем у SCI. Однако, для совместимости SCI имеет возможность переносить пакеты других протоколов. Другое преимущество SCI — использование простых протоколов типа RISC, которые обеспечивают большую пропускную способность (табл. 3.5). Узлы с адаптерами SCI могут использовать для соединения коммутаторы или же соединяться в кольцо. Обычно каждый узел оказывается включенным в два кольца (рис. 3.40).

Данная технология оптимизирована для работы с динамическим трафиком, однако может быть менее эффективна при работе с большими блоками данных. Протокол SCI достаточно сложен, он содержит большие возможности по управлению трафиком, но использование этих возможностей предполагает наличие развитого программного обеспечения. На коммуникационной технологии SCI

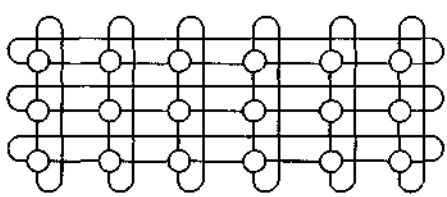


Рис. 3.40. Матрица узлов кластера на основе сети SCI

Таблица 3 5 **Основные характеристики SCI (Scalable Coherent Interface)**

Производители оборудования	Dolphin Interconnect Solutions и др
Показатели производительности	Для продуктов Dolphin пиковая пропускная способность 667 Мбайт/с Аппаратная латентность - 1 ,46 мкс
Программная поддержка	Драйверы для Linux, Windows NT, Solaris ScaMPI - реализация MPI компании Scali Computer для систем на базе SCI. SISCI API - интерфейс программирования нижнего уровня
Комментарии	SCI (ANSI/IEEE 1596–1992) - хорошо стандартизированная технология. Кроме стандартной сетевой среды, SCI поддерживает построение систем с разделяемой памятью и с когерентностью кэшей. На коммуникационной технологии SCI основаны кластерные системы компании SCALI Computer, системы семейства hpcLine компании Siemens, а также cc-NUMA-сервера Data General и Sequent Технология SCI использовалась для связи гиперузлов в системах HP/Convex Exemplar X-class

основана система связи гиперузлов CTI (Convex Torroidal Interconnect) в системах HP/Convex Exemplar X-class, кроме того, на ней построены кластерные системы SCALI Computer, системы семейства hpcLine компании Siemens, а также cc-NUMA сервера Data General и Sequent.

Традиционная область применения SCI — это коммуникационные среды многопроцессорных систем. На основе этой технологии построены, в частности, компьютеры серии hpcLine от Siemens или модульные серверы NUMA-Q от IBM, ранее известные как Sequent.

Модульные SCI-коммутаторы Dolphin позволяют потребителям строить масштабируемые, кластерные решения класса предприятия на платформах Windows NT/2000/XP, Linux, Solaris, VxWorks, LynuxWorks и NetWare с использованием стандартизованного оборудования и программного обеспечения.

Коммуникационная среда MYRINET

Сетевую технологию Myrinet представляет компания Myricom, которая впервые предложила свою коммуникационную технологию в 1994 г., а на сегодня имеет уже более 1000 инсталляций по всему миру. Технология Myrinet основана на использовании многопортовых коммутаторов при ограниченных несколькими метрами длинах связей узлов с портами коммутатора. Узлы в Myrinet соединяются друг с другом через коммутатор (до 16 портов). Максимальная длина линий связи варьируется в зависимости от конкретной реализации.

Как коммутируемая сеть, аналогичная по структуре сегментам Ethernet, соединенным с помощью коммутаторов, Myrinet может од-

новременно передавать несколько пакетов, каждый из которых идет со скоростью, близкой к 2 Гбит/с. В отличие от некоммутированных сетей Ethernet и FDDI, которые разделяют общую среду передачи, совокупная пропускная способность сети Myrinet возрастает с увеличением количества машин. На сегодняшний день Myrinet чаще всего используют как локальную сеть (LAN) сравнительно небольшого физического размера, связывая вместе компьютеры внутри комнаты или здания. Из-за своей высокой скорости, малого времени задержки, прямой коммутации и умеренной стоимости, Myrinet особенно популярен для объединения компьютеров в кластеры. Myrinet также используется как системная сеть (System Area Network, SAN), которая может объединять компьютеры в кластер внутри стойки с той же производительностью, но с более низкой стоимостью, чем Myrinet LAN. Пакеты Myrinet могут иметь любую длину. Таким образом, они могут включать в себя другие типы пакетов, включая IP-пакеты. Соединение вычислительных узлов с адаптерами Myrinet в сеть происходит с помощью коммутаторов, которые имеют сейчас 4, 8, 12 или 16 портов. В коммутаторах используется передача пакетов путем установления соединения на время передачи, для маршрутизации сообщений используется алгоритм прокладки пути (wormhole, «червоточина»). Коммутаторы, как и сетевые адаптеры, построены на специализированных микропроцессорах LANai фирмы Myricom (табл. 3.6).

На физическом уровне линки Myrinet состоят из девяти линий: 8 битов предназначены для передачи информации, интерпретируемой в зависимости от состояния девятого бита как байт данных или управляющий символ; при этом на каждом линке обеспечива-

Таблица 3.6 **Характеристики технологий Myricon**

Производители оборудования	Myricom
Показатели производительности	Пиковая пропускная способность - 2 Гбит/с, полный дуплекс. Латентность - порядка 4 мкс
Программная поддержка	Драйвера для Linux (Alpha, x86, PowerPC, UltraSPARC), Windows NT (x86), Solaris (x86, UltraSPARC) и Tru64 UNIX. GM - интерфейс программирования на нижнем урове. Пакеты HPVM (включает MPI-FM, реализацию MPI для Myrinet), BIP-MPI и др
Комментарии	Myrinet является открытым стандартом. Myricom предлагает широкий выбор сетевого оборудования по сравнительно невысоким ценам На физическом уровне поддерживаются сетевые среды SAN (System Area Network), LAN (CL-2) и оптоволокно Технология Myrinet дает высокие возможности масштабирования сети и в настоящее время очень широко используется при построении высокопроизводительных кластеров

ется управление потоком и контроль ошибок. Среда Myrinet выгодно отличается от многих других сред передачи, в частности SCI, простотой концепции и аппаратной реализацией протоколов. Она содержит ограниченный набор средств управления трафиком, использующих приливно-отливный буфер, управляющие символы и таймерные интервалы. Myrinet является открытым стандартом, компания Myricom предлагает широкий выбор сетевого оборудования по сравнительно невысоким ценам. Технология Myrinet дает большие возможности масштабирования сети и в настоящее время широко используется при построении высокопроизводительных вычислительных кластеров.

Коммуникационная среда Raceway

Коммуникационная среда Raceway обеспечивает пропускную способность на уровне 1 Гбайт/с; среда передачи создается с помощью коммутатора фирмы Cypress и соответствующих сетевых адаптеров. Коммутатор имеет шесть портов, пропускная способность каждого составляет 160 Мбайт/с. Порт состоит из 32 сигнальных линий данных и пяти управляющих линий. При начале транзакции среда Raceway предварительно устанавливает соединение, задержка в коммутаторе при установлении соединения составляет примерно 125 нс. Структуры вычислительных систем, создаваемых с помощью Raceway, аналогичны тем, которые применяются в случае использования сети Myrinet или коммутаторов и адаптеров SCI. Разница заключается в количестве портов коммутаторов, форматах передаваемых пакетов и в протоколах.

Коммуникационная среда Raceway принята в качестве стандарта (ANSI/VINA 5-1994).

Коммуникационные среды на базе транспьютероподобных процессоров

Транспьютер — это микроэлектронный прибор, объединяющий на одном кристалле микропроцессор, быструю память, интерфейс внешней памяти и каналы ввода-вывода (линки), предназначенные для подключения аналогичных приборов. Прибор спроектирован таким образом, чтобы максимально облегчить построение параллельных вычислительных систем. При соединении транспьютерных элементов между собой требуется минимальное число дополнительных интегральных схем. Связь между транспьютерами осуществля-

ется путем непосредственного соединения линка одного прибора с линком другого. Это позволяет создавать сети с различными топологиями с большим числом элементов

Транспьютер представляет собой микропроцессор, в состав которого входят:

1) ЦПУ с сокращенным набором команд (RISC);

2) 64-разрядный сопроцессор (FPU) плавающей арифметики с высокой пиковой производительностью, работающий параллельно с ЦПУ;

3) внутрикристальное ОЗУ;

4) 32-разрядная шина памяти;

5) четыре последовательные двунаправленные линии связи *(link),* обеспечивающие взаимодействие транспьютера с внешним миром, работающие параллельно;

6) таймер;

7) системные управляющие сигналы «*инициализация*», *«анализ»,* «*ошибка*», управляющие загрузкой и анализом состояния транспьютера, сигнализирующие об ошибках;

8) интерфейс внешних событий *(event),* обеспечивающий асинхронную связь внутреннего процесса и внешнего события.

Транспьютеры размещаются на транспьютерных модулях (TRAM или TPAM) — дочерних платах, содержащих транспьютер, ОЗУ, переключатели для выбора режимов, и интерфейс, включающий гнезда/штекеры питания, четыре линии связи, линии внешних событий и системных управляющих сигналов. В зависимости от состава TPAM может иметь разные физические размеры, которые стандартизованы и пронумерованы. Так, наименьший по размеру TPAM имеет номер 1, следующий — 2 и т. д.

TPAMы размешаются на объединительных платах, которые либо непосредственно включаются в некоторый компьютер, либо соединенные вместе составляют сетевой компьютер. Известно два типа объединительных плат, подключаемых к компьютеру (вычислительные транспьютерные платы):

1) загружаемые по линии связи платы общего назначения, начальная загрузка которых осуществляется программой главного компьютера по линии связи, соединяющей главный компьютер и транспьютер (корневой транспьютер), специально выделенный для взаимодействия с главным компьютером;

2) загружаемые из ПЗУ платы, предназначенные для автономных, встроенных систем.

Изначально транспьютьеры производила фирма Inmos. В настоящее время ряд зарубежных фирм пошел по пути создания

транспьютероподобных микропроцессоров, имеющих гораздо большую вычислительную мощность, чем транспьютер фирмы Inmos (например, фирма Texas Instruments выпустила сигнальный процессор TMS320C40 с производительностью 50 Мфлопс).

Возникновение высокопроизводительных параллельных вычислительных систем на базе транспьютеров и транспьютероподобных микропроцессоров в свое время потребовало создания новых эффективных операционных систем.

3.7. Коммутаторы для многопроцессорных вычислительных систем

Коммуникационные среды вычислительных систем (ВС) состоят из адаптеров вычислительных модулей (ВМ) и коммутаторов, обеспечивающих соединения между ними.

Используются как простые коммутаторы, так и составные, компонуемые из набора простых. Простые коммутаторы могут соединять лишь малое число ВМ в силу физических ограничений, однако обеспечивают при этом минимальную задержку при установлении соединения. Составные коммутаторы, обычно строящиеся из простых в виде многокаскадных схем с помощью линий «точка—точка», преодолевают ограничение на малое количество соединений, однако увеличивают и задержки.

Простые коммутаторы

Известно два основных типа простых коммутаторов:

- с временным разделением;
- с пространственным разделением.

Простые коммутаторы с временным разделением. Простые коммутаторы с временным разделением используются в системах SMP, например Power Challenge от SGI. Простые коммутаторы с временным разделением называются также шинами или шинными структурами. Все устройства подключаются к общей информационной магистрали, используемой для передачи информации между ними (рис. 3.41). Обычно шина является пассивным элементом, управление передачами осуществляется передающими и принимающими устройствами. Достоинства: простота управления и высокое быстродействие. Недостатки: малое количество входов и выходов.

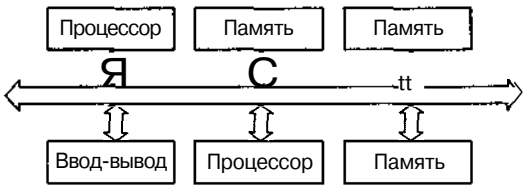


Рис. 3.41. Общая схема шинной структуры

Процесс передачи выглядит следующим образом.

Передающее устройство сначала получает доступ к шине, далее пытается установить контакт с устройством-адресатом и определить его способность к приему данных. Принимающее устройство распознает свой адрес на шине и отвечает на запрос передающего. Далее передающее устройство сообщает, какие действия должно произвести принимающее устройство в ходе взаимодействия. После этого происходит передача данных.

Так как шина является общим ресурсом, за доступ к которому соревнуются подключенные к ней устройства, то необходимы методы управления предоставлением доступа устройств к шине. Возможно использование центрального устройства для управления доступом к шине, однако это уменьшает масштабируемость и гибкость системы.

Для разрешения конфликтов, возникающих при одновременном запросе устройств на доступ к шине, используются различные приемы, в частности:

- назначение каждому устройству уникального приоритета (статического или динамического);
- использование очереди запросов FIFO;
- выделение фиксированных временных интервалов каждому устройству.

Алгоритмы арбитража.

Статические приоритеты. Каждое устройство в системе получает уникальный приоритет, при одновременном запросе нескольких устройств на передачу доступ к шине предоставляется устройству с наивысшим приоритетом. На практике часто используется соединение устройств в цепь, при котором приоритет устройства определяется местом его подключения к шине. Для контроля доступа к шине используется отдельный блок управления.

Динамические приоритеты. Так же, как и в предыдущем алгоритме, устройства получают уникальные приоритеты, однако в отличие от него эти приоритеты непостоянны во времени. Приорите-

ты динамически изменяются, предоставляя устройствам более или менее равные шансы получения доступа к шине. Наиболее часто применяются следующие способы изменения приоритетов: наивысший приоритет предоставляется устройству, наиболее долго не использовавшему шину, и циклическая смена приоритетов. Контроль доступа к шине осуществляет устройство, получившее доступ к шине в предыдущем цикле арбитража.

Фиксированные временные интервалы. Каждое устройство по порядку получает одинаковый временной интервал для осуществления передачи. Если устройство не имеет данных для передачи, то интервал тем не менее не предоставляется следующему устройству.

Очередь FIFO. Создается очередь запросов «первый пришел — первый ушел», однако сохраняется проблема арбитража между почти одновременными запросами, а также возникает необходимость поддержания очереди запросов достаточной длины. Преимуществом данного алгоритма является возможность достижения максимальной пропускной способности шины.

Особенности реализации шин. Внутри микросхем шины используются для объединения функциональных блоков микропроцессоров, микросхем памяти, микроконтроллеров. Шины используются для объединения устройств на печатных платах и печатных плат в блоках. В последнее время применяются шины следующих стандартов (см. также гл. 2):

ISA — Industry Standard Architecture;
EISA — Extended ISA;
VESA — Video Electronics Standards Association;
PCI — Peripheral Computer Interconnect;
VME — Versabus Module Europe;
I2C — Inter Integrated Circuit;
AGP — Accelerated Graphic Port.

Шины используются также в *мезонинной технологии,* где на большой плате устанавливается один или несколько шинных разъемов для установки меньших плат, так называемых *мезонинов.*

Шины, объединяющие устройства, из которых состоит вычислительная система, являются критическим ресурсом, отказ которого может привести к отказу всей системы. Шины обладают также рядом принципиальных ограничений. Возможность масштабируемости шинных структур ограничивается временем, затрачиваемым на арбитраж, и количеством устройств, подключенных к шине. При этом чем больше подключенных устройств, тем больше время, затрачиваемое на арбитраж. Время арбитража ограничивает и пропускную способность шины. Кроме того, в каждый момент времени

шина используется для передачи только одним устройством, что становится узким местом при увеличении количества подключенных устройств. Пропускная способность шины ограничивается ее шириной — количеством линий (проводников), используемых для передачи данных, — и тактовой частотой ее работы. Данные величины имеют физические ограничения.

Простые коммутаторы с пространственным разделением. Простые коммутаторы с пространственным разделением (Gigaplane) используются, например, в семействе Sun Ultra Enterprise.

Простые коммутаторы с пространственным разделением позволяют одновременно соединять любой вход с любым одним выходом (ординарные) или несколькими выходами (неординарные). Такие коммутаторы представляют собой совокупность мультиплексоров, количество которых соответствует количеству выходов коммутатора, при этом каждый вход коммутатора должен быть заведен на все мультиплексоры. Структура этих коммутаторов показана на рис. 3.42. Достоинства: возможность одновременного контакта со всеми устройствами; минимальная задержка. Недостатки: высокая сложность порядка $n \times m$, где n — количество входов, m — количество выходов; сложность обеспечения надежности.

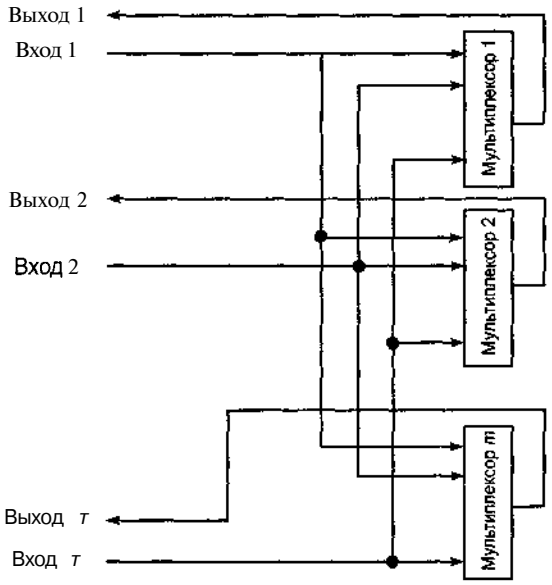


Рис. 3.42. Простой коммутатор с пространственным разделением

Составные коммутаторы

Простые коммутаторы имеют ограничения на число входов и выходов, а также могут требовать большого количества оборудования при увеличении этого числа (в случае пространственных коммутаторов). Поэтому для построения коммутаторов с большим количеством входов и выходов используют совокупность простых коммутаторов, объединенных с помощью линий «точка—точка».

Составные коммутаторы имеют задержку, пропорциональную количеству простых коммутаторов, через которые проходит сигнал от входа до выхода, т. е. числу каскадов. Однако объем оборудования составного коммутатора меньше, чем простого с тем же количеством входов и выходов.

Чаще всего составные коммутаторы строятся из прямоугольных коммутаторов 2 x 2 с двумя входами и выходами. Они имеют два состояния: прямое пропускание входов на соответствующие выходы и перекрестное пропускание. Коммутатор 2×2 состоит из собственно блока коммутации данных и блока управления. Блок управления в зависимости от поступающих на него управляющих сигналов определяет, какой тип соединения следует осуществить в блоке коммутации: прямой или перекрестный. При этом если оба входа хотят соединиться с одним выходом, то коммутатор разрешает конфликт и связывает с данным выходом только один вход, а запрос на соединение со стороны второго блокируется или отвергается.

Коммутатор Клоза (рис. 3.43). Коммутатор Клоза может быть построен в качестве альтернативы для прямоугольного коммутатора с $(m \times d)$ входами и $(m \times d)$ выходами. Он формируется из трех каскадов коммутаторов: m коммутаторов $(d \times d)$ во входном каскаде, m коммутаторов $(d \times d)$ в выходном и d промежуточных коммутаторов $(m \times m)$.

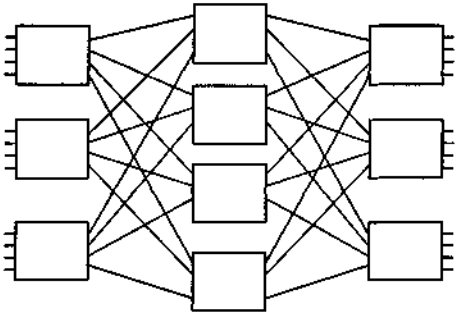


Рис. 3.43. Коммутатор Клоза 3×4

Соединения внутри коммутатора устроены следующим образом:

- *j*-й выход *i*-го коммутатора входного каскада соединен с *i*-м входом *j*-го промежуточного коммутатора;
- *j*-й вход *k*-го коммутатора выходного каскада соединен с *k*-м выходом *j*-го промежуточного коммутатора

Данный тип составных коммутаторов позволяет соединять любой вход с любым выходом, однако при установленных соединениях добавление нового соединения может потребовать разрыва и переустановления всех соединений.

Баньян-сети

Коммутаторы этого типа строятся на базе прямоугольных коммутаторов таким образом, что существует только один путь от каждого входа к каждому выходу.

Структура баньяновой сети, выполненная в виде узла на 16 входов и выходов состоит (рис. 3.44) из простых коммутирующих элементов, соединенных друг с другом. Через последовательности этих элементов передаются блоки данных. Изображенная структура имеет четыре каскада (1 — 4) коммутирующих элементов. Каждый передаваемый блок данных имеет в заголовке адрес, разрядность которого равна числу элементов баньяновой сети. Блок, поданный на

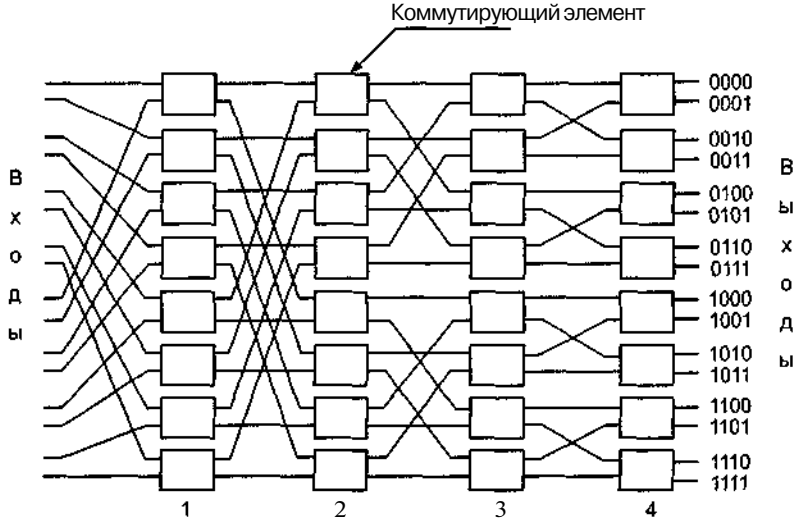


Рис. 3.44. Баньян-сеть

вход i-го каскада попадает на один из его выходов, если в i-м разряде адреса записан «О». Если в этом разряде находится «1», то блок передается на другой выход элемента. Так, по каскадам, происходит ретрансляция блоков данных, определяемая деревом выбора путей передачи.

Таким образом осуществляется самомаршрутизация блоков, определяемая их адресами. В результате баньян-сети обеспечивают большую пропускную способность, ибо блоки данных через них проходят параллельно, а функции маршрутизации выполняются аппаратно. Однако нужно иметь в виду, что в баньяновых сетях могут происходить взаимные блокировки и возникать тупиковые ситуации. Поэтому в рассматриваемых сетях должны быть приняты специальные меры, предотвращающие появление этих тупиков.

Важной разновидностью баньян-сетей является *дельта-сеть*. Она формируется из прямоугольных коммутаторов (a x b) и представляет собой n-каскадный коммутатор с n входами a и n выходами b. Составляющие коммутаторы соединены так, что для соединения любого входа и выхода образуется единственный путь одинаковой длины для всех пар входов и выходов.

Распределенные составные коммутаторы

В распределенных вычислительных системах ресурсы разделяются между задачами, каждая из которых исполняется на своем подмножестве процессоров. В связи с этим возникает понятие близости процессоров, которая является важной для активно взаимодействующих процессоров. Обычно близость процессоров выражается в различной каскадности соединений, различных расстояниях между ними.

Один из вариантов создания составных коммутаторов заключается в объединении прямоугольных коммутаторов $(v+1 \times v+1)$, $v > 1$, таким образом, что один вход и один выход каждого составляющего коммутатора служат входом и выходом составного коммутатора. К каждому внутреннему коммутатору подсоединяются процессор и память, образуя вычислительный модуль с v каналами для соединения с другими вычислительными модулями. Свободные v выходов и v входов каждого вычислительного модуля соединяются линиями «точка—точка» с входами и выходами других коммутаторов, образуя граф межмодульных связей.

Наиболее эффективным графом межмодульных связей с точки зрения организации обмена данными между вычислительными мо-

дулями является *полный граф.* В этом случае между каждой парой вычислительных модулей существует прямое соединение. При этом возможны одновременные соединения между произвольными вычислительными модулями.

Однако обычно создать полный граф межмодульных связей невозможно по различным причинам. Обмен данными приходится производить через цепочки транзитных модулей. Из-за этого увеличиваются задержки и ограничивается возможность установления одновременных соединений. Таким образом, эффективный граф межмодульных связей должен минимизировать время межмодульных обменов и максимизировать количество одновременно активизированных соединений. Кроме того, на выбор графа межмодульных связей влияет учет отказов и восстановлений вычислительных модулей и линий связи.

Матричный коммутатор состоит из множества одинаковых коммутирующих элементов (рис. 3.45). На два входа каждого из них подаются блоки данных, которые в зависимости от адресов назначения, необходимо передать на один из выходов элемента. Число коммуникационных элементов как минимум равно произведению числа входов на число выходов коммутатора. Поэтому, матричные комму-

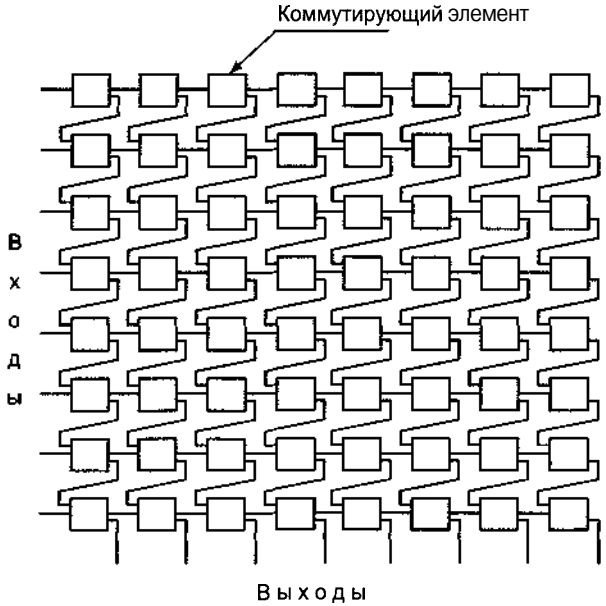


Рис. 3.45. Матричный коммутатор

таторы применяются в интегральной коммутации, число входов/выходов в которых невелико.

Граф межмодульных связей Convex Exemplar SPP1000. В качестве примера реального графа межмодульных связей рассмотрим структуру системы Convex Exemplar SPP10QO (рис. 3.46). В основе каждого составного блока системы лежит прямоугольный коммутатор (5 х 5), до 16 подобных блоков объединяются каналами «точка—точка» в кольцо (одномерный тор), состоящее из четырех независимых подканалов.

Внутри каждого блока четыре входа и выхода прямоугольного коммутатора (5 х 5) используются для взаимодействия устройств внутри блока (при этом в каждом блоке располагается по два процессора), пятые вход и выход используются для объединения блоков в кольцо. При этом каждый из четырех кольцевых каналов рассматривается как независимый ресурс, и система сохраняет работоспо-

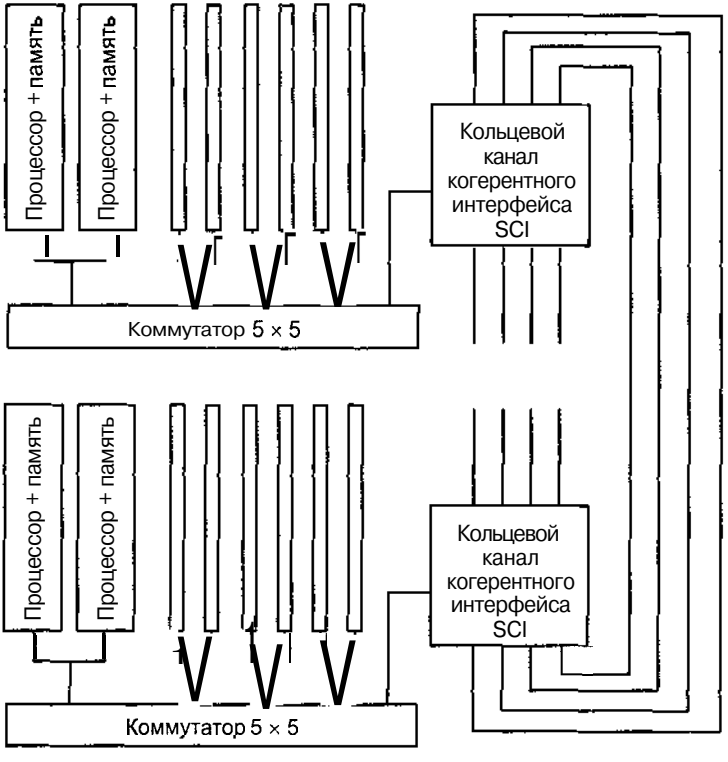


Рис. 3.46. Граф межмодульных связей Convex Exemplar SPP1000

собность до тех пор, пока существует хотя бы один функционирующий кольцевой канал.

Граф межмодульных связей МВС-100. Система МВС-100 предлагает блочный подход к построению архитектуры параллельной вычислительной системы. Структурный модуль системы состоит из 16 вычислительных узлов, образующих матрицу 4 × 4 (рис. 3.47). Угловые узлы соединяются попарно по диагонали, таким образом, максимальная длина пути между любой парой элементов равна трем. В исходной же матрице 4 × 4 эта длина равна шести. Каждый блок имеет 12 выходов, что позволяет объединять их в более сложные структуры.

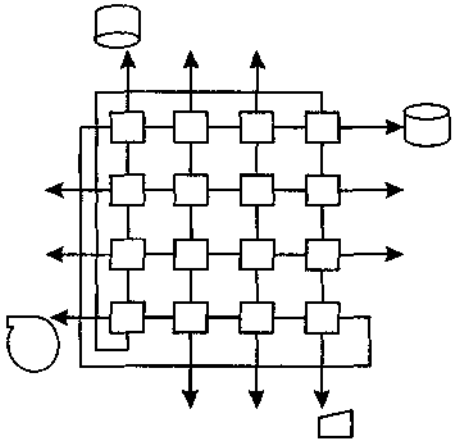


Рис. 3.47. Структурный модуль МВС-100

Для МВС-100 базовый вычислительный блок содержит 32 узла. Такой блок строится из двух структурных модулей в соответствии со схемой, приведенной на рис. 3.48, *а*. В этом случае максимальная длина пути между любой парой вычислительных узлов равна пяти. При этом остается 16 свободных связей, что позволяет продолжить объединение. При объединении двух базовых блоков по схеме, приведенной на рис. 3.48, *б* (64 вычислительных узла), максимальная длина пути составит шесть, как и в гиперкубе, количество свободных связей будет равно 16.

Граф межмодульных связей МВС-1000. Архитектура системы МВС-1000 аналогична архитектуре МВС-100. Основой системы является масштабируемый массив процессорных узлов. Каждый узел содержит вычислительный микропроцессор Alpha 21164 с произво-

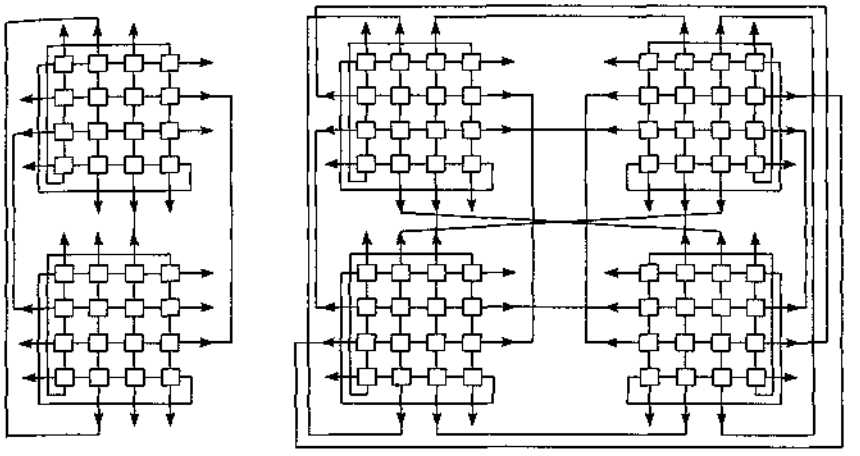


Рис. 3.48. Базовый вычислительный блок (а), объединение двух блоков *(б)* МВС-100

дительностью 2 Гфлопс при тактовой частоте 500 МГц и оперативную память объемом 128 Мбайт с возможностью расширения. Процессорные узлы взаимодействуют через коммуникационные процессоры TMS320C44 производства Texas Instruments, имеющие по четыре внешних канала (линка) с общей пропускной способностью 80 Мбайт/с (20 Мбайт/с каждый). Также разрабатывается вариант системы с использованием коммуникационных процессоров SHARC (ADSP 21060) компании Analog Devices, имеющих по шесть каналов с общей пропускной способностью до 240 Мбайт/с (40 Мбайт/с каждый).

Процессорные узлы связаны между собой по оригинальной схеме, сходной с топологией двухмерного тора (для 4-линковых узлов). Аналогично МВС-100, структурный модуль МВС-1000 состоит из 16 вычислительных модулей, образующих матрицу 4 х 4, в которой четыре угловых элемента соединяются через транспьютерные линки по диагонали попарно. Оставшиеся 12 линков предназначаются для подсоединения внешних устройств (четыре линка угловых ВМ) и соединений с подобными ВМ.

Конструктивным образованием МВС-1000 является базовый вычислительный блок, содержащий 32 вычислительных модуля. Максимальная длина пути между любыми из 32 вычислительных модулей равна пяти, при этом число свободных линков после комплектации блока составляет 16, что позволяет продолжить процеду-

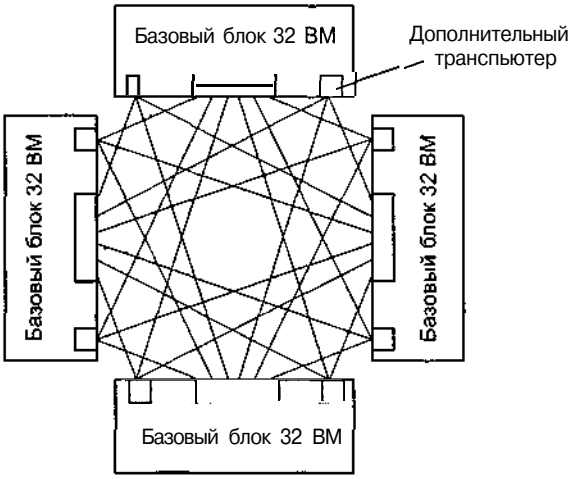


Рис. 3.49. Структура 128-процессорной системы МВС-1000, четыре базовых блока

ру объединения. Возможная схема объединения четырех базовых блоков в 128-процессорную систему приведена на рис. 3.49.

3.8. Кластерные и массивно-параллельные системы различных производителей

Развитие сетевых технологий привело к появлению недорогих, но эффективных коммуникационных решений. Это и предопределило появление кластерных вычислительных систем, фактически являющихся одним из направлений развития компьютеров с массовым параллелизмом. Классические суперкомпьютеры, использующие специализированные процессоры таких фирм, как, например, Cray, NEC (векторно-параллельные или массивно-параллельные), обычно недешевы, поэтому и стоимость подобных систем не сравнима со стоимостью систем, находящихся в массовом производстве.

Вычислительные системы, создаваемые из массово-выпускаемых компонентов, стали притягательной альтернативой традиционным суперкомпьютерным системам. При выполнении многих прикладных задач такие ВС, даже с небольшим или средним (до 128—256) числом вычислительных модулей, показывают производительность, не уступающую или даже превосходящую производитель-

ность традиционных суперкомпьютеров как с распределенной, так и с разделяемой памятью. Наряду с этим, эти ВС обладают рядом преимуществ, среди которых: более низкая стоимость, короткий цикл разработки и возможность оперативно использовать наиболее эффективные вычислительные и коммуникационные компоненты из имеющихся на рынке во время создания системы. Поэтому неудивительно, что ведущие фирмы-разработчики высокопроизводительной техники приступили к созданию кластерных систем.

Отечественные суперкомпьютеры семейства МВС

Создание и использование суперкомпьютеров в мире относится к факторам стратегического значения и входит в первую десятку приоритетов «ключевых» технологий развитых стран. В настоящее время в США эксплуатируются суперкомпьютеры с производительностью более триллиона операций в секунду, что дает возможность ускорения работ в области фундаментальных исследований, решения прикладных задач народнохозяйственного и оборонного комплекса. США, Япония и страны Западной Европы резко ограничивают возможность приобретения Россией мощных вычислительных систем. Кроме того, зарубежные вычислительные системы такого уровня непомерно дороги и их эксплуатационное освоение затруднено.

Проблема создания конкретных суперкомпьютеров требуемого уровня мощности для наиболее крупных российских вычислительных центров решается на основе сбалансированного целенаправленного сочетания закупок новейших комплектующих изделий, создания на этой основе отечественных суперкомпьютерных систем, их интеграции в информационно-вычислительные сети и необходимых усилий в области применения, т. е. в разработке прикладных программ и методов математического моделирования. Эта концепция реализована в мультипроцессорной вычислительной системе МВС-100, которая создана в кооперации научно-исследовательских институтов Российской академии наук и промышленности (головные организации НИИ «Квант» РАСУ и ИПМ РАН). Система построена на основе микропроцессоров с быстродействием порядка 100 миллионов операций в секунду, межпроцессорное взаимодействие осуществляется с помощью транспьютеров. Установки МВС-100 с суммарной производительностью более 50 миллиардов операций в секунду в течение нескольких лет успешно эксплуатируются в вычислительных центрах РАН (в Москве, Екатеринбурге, Новосибирске, Владивостоке) и в отраслевых ВЦ. С их

помощью решены сложные прикладные задачи качественно нового уровня из области аэродинамики для самолетостроения и создания реактивных двигателей, ядерной физики, управления динамическими системами, распознавания изображений при навигации движущихся объектов, сейсмогеологоразведки, нефтедобычи, метеорологии, биоинженерии и др. Показана возможность эффективного распараллеливания вычислений и обработки данных.

В сложившейся кооперации проведены работы по созданию системы нового поколения — МВС-1000 на микропроцессорах Alpha с технологическими нормами 0,35 мкм и быстродействием до 1—2 млрд оп./с. В течение 1998 г. на эксплуатируемых установках этого типа отрабатывалось программное обеспечение и решен ряд новых сложных реальных вычислительных задач. К настоящему времени введена в действие система с производительностью 200 млрд оп./с для Межведомственного суперкомпьютерного центра (Миннауки, Минобразования, РАН, РФФИ); предполагается дальнейшее наращивание мощности. Освоен режим телекоммуникационного доступа к МВС, в том числе по Internet, с обеспечением требований защиты информации. Предстоят дальнейшие разноплановые работы по техническому и математическому освоению созданных систем, развитию их программного обеспечения. Эти работы входят в соответствующие целевые научно-технические программы федерального и ведомственного уровня.

Массово-параллельные масштабируемые системы МВС предназначены для решения прикладных задач, требующих большого объема вычислений и обработки данных. Суперкомпьютерная установка системы МВС представляет собой мультипроцессорный массив, объединенный с внешней дисковой памятью и устройствами ввода-вывода информации под общим управлением персонального компьютера или рабочей станции.

МВС-1000 — система 3-го поколения, основана на использовании микропроцессоров Alpha 21164 (разработка фирмы DEC-Compaq; выпускается также заводами фирм Intel и Samsung) с производительностью до 1—2 млрд операций в секунду и присоединенной оперативной памятью объемом 0,1—2 Гбайт.

Мультипроцессорный массив системы с блоками вторичного электросилового питания и вентиляцией располагается в стойках размером 550 х 650 х 2200 мм промышленного стандарта; вес заполненной стойки — 220 кг, потребляемая мощность до 4 кВт.

В основном исполнении системы межпроцессорный обмен структурно аналогичен используемому в системе МВС-100 и реализуется в двух модификациях: на базе «транспьютероподобного»

связного микропроцессора TMS320C44 (фирма Texas Instruments), имеющего четыре канала с пропускной способностью каждого 20 Мбайт/с, либо на базе связного микропроцессора SHARC ADSP 21060 (фирма Analog Devices), имеющего шесть внешних каналов с пропускной способностью каждого — 40 Мбайт/с.

Исполнение МВС-1000К отличается использованием для межпроцессорного обмена коммутационной сети MYRINET (фирма Myricom, США) с пропускной способностью канала в дуплексном режиме 2 х 160 Мбайт/с. Кроме того, предусмотрено подключение к каждому процессору памяти на жестком диске с объемом 2—9 Гбайт.

В стандартной стойке располагается до 64 процессоров системы МВС-1000 или 24 процессора системы МВС-ЮООК. Предусмотрены средства системного объединения стоек для установок с большим числом процессоров.

В программном обеспечении МВС используются:

- языки FORTRAN и С (С++), дополнительные средства описания параллельных процессов;
- программные средства PVM и MPI (общепринятые для систем параллельной обработки);
- средства реализации многопользовательских режимов и удаленного доступа.

Примеры кластерных решений IBM

В начале 2000 г. IBM создала Linux-кластер из установленных в стойке серверов IBMxSeries, интегрировав их с соответствующими сетями, системами управления (аппаратное и программное обеспечение) и необходимыми услугами. После выпуска в 2001 г. кластера 1300 IBM представила недавно кластер 1350 на процессорах Intel Хеоп (табл. 3.7).

Таблица 3 7 **Пример конфигурации кластера 1350**

Класс	Число узлов кластера	Скорость процессора, ГГц	Память системы, Гбайт	Внутренняя память, Гбайт	Соединение кластера, Мбит/с
Начальный	8	2,0	0,512	18	10/100 Ethernet
Средний	32	2,4	1	18	10/100 Ethernet
Профессиональный	128	2,8	1	36	Gigabit Ethernet
Высокопроизводительный	64	2,8	1	36	Myrinet-2000

Стандартным вычислительным узлом для кластера 1350 является IBMxSeries 335. Это позволяет одному или двум процессорам Intel Pentium 4 (Хеоп) с быстрой динамической памятью и диском размещаться в стандартном корпусе размером «1U». Символ 1U обозначает 1,75 дюймов высоты в стандартном 19-дюймовом корпусе. X335 имеет встроенный сервисный процессор и два слота для соединения с другими компонентами системы.

Головные узлы, узлы управления и узлы запоминающих устройств обеспечивают особые функции для управления кластером (как обеспечение загрузки, управление устройствами, внешний ввод/вывод и т. д.). Сервер 2U IBM xSeries 345, основанный на процессорах Хеоп, в кластере 1350 используется как узел управления и хранения данных и может быть также использован как вычислительный узел. Коммутаторы используются для межпроцессорного соединения в параллельном программировании и для различных функций управления.

Для параллельного программирования в качестве межпроцессорного соединения обычно используется коммутатор Myrinet фирмы Myricom. Пропускная способность канала составляет приблизительно 200 Мбайт/с в каждом направлении со временем задержки 6—8 мкс.

Терминальные серверы обеспечивают удаленный доступ к консолям ОС узлов через последовательную сеть. Дополнительные функциональные возможности добавляются посредством клавиатуры, мыши, монитора.

Коммерческий программный пакет может включать в себя WebSphere, DB2, MySQL и т. д. HPC пакет может включать MPICH, PVM, Maui Scheduler, математические библиотеки, трансляторы, профилировщики и т. д.

Операционная система Linux инсталлирована на каждом узле кластера. Кластер 1350 запускается под Red Hat Linux. В дальнейшем планируется ставить ОС SuSE (4Q02).

Большинство сложившихся систем управления, называемых xCAT, были разработаны IBM для сборки кластеров на основе требований заказчика. xCAT поддерживает все требуемые функции, включая функции удаленного контроля. Отметим, что xCAT использует сервисный процессор xSeries и что xCAT не является открытым программным продуктом. Продукт поставляется свободно с кластерным пакетом IBM, включая исходные тексты.

Управление системами кластера для Linux (CSM) — это лицензионный программный продукт IBM. Он обеспечивает функции управления системами, сходными по форме с программами под-

держки параллельных систем (Parallel System Support Programs — PSSP) для AIX-систем уровня поддержки на RS/6000 SP. CSM — это стандартный программный продукт для кластера 1350.

CSM для Linux включает технологию, извлеченную из PSSP, и сейчас доступную на AIX для управления кластерами, собранными из серверов xSeries и запускаемых под Linux, серверами IBM pSeries, управляемых AIX, или комбинацией обеих операционных систем.

Другие программные продукты, как взятые из открытого доступа, так и лицензионные, могут быть выбраны и адаптированы к нуждам заказчика и инсталлированы в виде части полной системы всего кластерного решения. Образцы этого ПО включают Portable Batch Scheduler (PBS) и Maui Scheduler, взятые из открытого доступа. Другие образцы включают MPICH для параллельного программирования, математические библиотеки, инструментарий для параллельной отладки и повышения производительности и много других приложений от независимых продавцов.

Примеры кластерных решений HP

Слияние HP и Compaq обеспечило HP прочное положение лидера по продаже Linux-систем, соответствующих лучшим индустриальным стандартам на базе архитектур IA-32 и IA-64. Данная технология дополнена мощной поддержкой разработок ядра Linux на базе семейства Itanium, а также разработок с открытым кодом в целом.

Поддержка ОС Linux со стороны HP охватывает всю линейку серверов HP, основанных на архитектуре Intel (IA-32 и IA-64), включая все серверы промышленного стандарта HP ProLiant, сверхплотную блейд-архитектуру, рабочие станции HP, настольные компьютеры Evo, отдельные портативные компьютеры, серверы ProLiant для применения в качестве межсетевых экранов и даже портативные устройства iPAQ. HP также продолжает поддерживать технологию ОС Linux для архитектуры AlphaServer, разработанную компанией Compaq. Это открыло путь для современных разработок ОС Linux на базе семейства Itanium. HP поддерживает на своих серверах дистрибутивы Red Hat и SuSE, планируя осуществлять поддержку дистрибутивов операционной системы UnitedLinux после ее выпуска. HP предлагает заказчикам возможность предварительно установить любую ОС Linux на выбранные серверы ProLiant и рабочие станции Evo. Услуги по глобальному развертыванию позволяют управлять предварительной установкой операционной системы в любой точке мира.

Программное обеспечение HP может поддерживать большинство современных средств разработки и настройки производительности для кластерных решений на базе системы Linux. При выборе этих средств действуют ограничения, связанные с типами процессоров и межузловых соединений. В число программных продуктов входят:

- компилятор Intel C++ Compiler для Linux;
- компилятор Intel Fortran Compiler для Linux;
- библиотека Intel Math Kernel Library;
- Intel Vtune Performance Analyzer — средство оптимизации программного кода.

Примеры кластерных решений SGI

Компания Silicon Graphics (SGI) была создана в 1981 г. Основным направлением работы компании в течение многих лет было создание высокопроизводительных графических рабочих станций. В настоящее время ее интересы распространяются на рынок высокопроизводительных вычислений как для технических, так и для коммерческих приложений. В частности, она концентрирует свои усилия на разработке и внедрении современных технологий визуализации вычислений, трехмерной графики, обработки звука и мультимедиа.

В начале 2003 г. компания SGI представила новое семейство 64-х разрядных Linux-серверов и суперкластеров, названных SGI Altix 3000. Система SGI Altix 3000 использует процессоры Intel Itanium 2 и основана на архитектуре глобальной разделяемой памяти SGI Numaflex, которая является реализацией архитектуры неоднородного доступа к памяти (NUMA). NUMAflex появилась в 1996 г. и с тех пор использовалась в известной серии серверов и суперкомпьютеров SGI Origin, основанных на процессорах MIPS и 64-разрядной операционной системе IRIX. Дизайн NUMAflex позволяет помещать процессор, память, систему ввода/вывода, соединительные провода, графическую подсистему в модульные компоненты, иначе называемые *блоками* или *кирпичиками*. Эти кирпичики могут комбинироваться и конфигурироваться с большой гибкостью, чтобы удовлетворять потребности клиента в ресурсах и рабочей нагрузке. Используя этот дизайн третьего поколения, компания SGI смогла создать систему SGI Altix 3000 на основе традиционных блоков ввода/вывода (IX- и PX-блоки), хранения данных (D-блоки) и соединительных компонентов (маршрутизирующие блоки/R-бло-

ки). Основным отличием этой новой системы является процессорный блок (С-блок), который содержит процессоры Itanium 2.

Ключевой особенностью системы является использование каскадируемых коммутаторов в маршрутизирующих элементах. Каскадируемые коммутаторы обеспечивают системе малые времена задержки или замедление доступа к памяти, несмотря на модульную конструкцию. Это критично для машин, использующих архитектуру неоднородного доступа к памяти (NUMA). Задержки всегда были проблемой в архитектуре NUMA, так как память распределяется между узлами, а не сосредоточена в одном месте. Каскадируемые коммутаторы используют каталогизируемую схему памяти для отслеживания данных, находящихся в разных кэшах. В результате меньшие объемы данных пересылаются между частями памяти, что выливается в понижение задержек по сравнению с традиционными системами, основанными на шинах.

В недавних тестах SPECfp_rate_base2000 система SGI Altix 3000 (1 ГГц) показала мировой рекорд производительности в операциях с плавающей точкой для 64-процессорного сервера со значением 862. Наиболее близкий результат для 64-процессорных систем с единым образом операционной системы показал сервер HP Superdome (875 МГц) со значением 267 — меньше трети производительности системы SGI. По сравнению с 32-процессорными системами SGI Altix 3000 показал производительность в 1,8 раз большую, чем IBM eServer p690 (1,3 ГГц) и в 3,5 раза большую, чем HP Superdome (750 МГц). 32-процессорная система SGI получила 443 очка, IBM eServer p690 — 251, HP Superdome — 128. Результаты 32-процессорного сервера SGI Altix 3000 демонстрируют превосходство на 300 % по критерию цена/производительность по сравнению с IBM eServer p690.

Разработка процессора R10000 позволила компании перейти к объединению своих серверов Challenge (на базе процессора R4000) и PowerChallenge (на базе процессора R8000) в единую линию изделий. Благодаря повышенной производительности этого процессора на целочисленных операциях и плавающей точке, обе линии продуктов могут быть объединены без потери производительности.

Серверы Silicon Graphics работают под управлением операционной системы IRIX (ОС UNIX реального времени), построенной в соответствии с требованиями стандартов SVID (System V Interface Definition) и XPG4. Она поддерживает возможность работы нескольких машин на одном шлейфе SCSI (multi-hosted SCSI), 4-кратное зеркалирование и 128-кратное расщепление дисковых накопи-

телей. На платформе поддерживаются многие продукты компаний Oracle, Informix и Sybase.

Компьютеры Challenge DM/L/XL ориентированы в первую очередь на коммерческие применения, а Power Challenge L/XL — на работу с плавающей запятой. Системы Challenge DM относятся к среднему классу.

Power Challenge относится к классу симметричных мультипроцессорных ЭВМ (SMP-системы), базирующихся на поколении суперскалярных процессоров MIPS R8000 фирмы Silicon Graphics.

Отличительными особенностями этих систем являются:

- масштабируемость суперкомпьютинга;
- использование большой динамической памяти (адресация у POWER CHALLENGE до 16 Гбайт - в 2 раза выше, чем у Cray T90/C90/J90);
- 64-разрядная архитектура (в отличие от машин фирм IBM, HP, Sun и Thinking Machines), как у машин Cray и Convex;
- бинарная совместимость со всем семейством компьютеров SGI, включая рабочие станции Indy.

В заключение главы приведем сведения о некоторых отечественных и зарубежных суперкомпьютерах.

В табл. 3.8 приводится список, в который включены вычислительные центры (ВЦ) РФ, работающие в научно-технической области и имеющие суперкомпьютерные ресурсы.

Таблица 3.8. **Суперкомпьютерные центры России**

Компьютеры	ОП, Мбайт	$R_{реак}$, Мфлопс	Рейтинг, Мфлопс
Гидрометеоцентр РФ (Москва)			
SGI/Cray Y-MP, 8 процессоров	2048	2664	8 ×161 = 1288
Институт высокопроизводительных систем и баз данных (С.-Петербург)			
HP/Convex SPP 1000 SCPU	–	1920	8 x 65 = 520
Parsytec CC, 16 процессоров (Power PC 604/133 МГц)	512	4272	16 × 28= 448
HP/Convex C3820, 2 процессора	1024	480	2 x 44 = 88
HP/Convex C3440, 4 процессора	512	800	4 x 19 = 76
Институт физики высоких энергий (Протвино)			
DEC/Alpha Server 8200 5/300, 6 процессоров	1024	3600	6x 140 = 840
Институт органической химии РАН (Москва)			
SGI/Power Challenge L (90 МГц), 6 процессоров	512	2160	6 ×128=758

Окончание табл. 3 8

Компьютеры	ОП, Мбайт	$R_{реак}$, Мфлопс	Рейтинг, Мфлопс
ВНИИ неорганических материалов (Москва)			
SGI/Power Challenge L (90 МГц), 2 процессора	128	720	2 × 126 = 252
НИКИ энергетической техники (Москва)			
SGI/Power Challenge L (90 МГц), 2 процессора	128	720	2 × 126 = 252
Институт прикладной математики РАН (Москва)			
HP/Convex SPP 1000/CD, 4 процессора	512	800	4 × 48 = 192
Институт математического моделирования РАН (Москва)			
Parsytec CC, 12 процессоров (PPC 601/100 МГц)	96	2400	16x 12=192

Рейтинг ВЦ рассчитывается как сумма рейтингов, инсталлированных на этом ВЦ суперкомпьютерных систем. В табл. 3.9 приводятся данные по наиболее мощным суперкомпьютерам. Рейтинг каждого суперкомпьютера рассчитывается как произведение числа его процессоров на производительность процессора на тестах LinPack при N (размерности системы линейных уравнений), равном 100.

ПК, однопроцессорные рабочие станции и серверы, а также вычислительные системы, имеющие пиковую производительность не выше 200 Мфлопс или рейтинг не выше 62 Мфлопс (производительность Intel Pentium Pro с тактовой частотой 200 МГц), не учитываются.

Таблица 3 9 **Рейтинг суперкомпьютеров (по состоянию на 2004 г.)**

Рейтинг	Производитель системы/количество процессоров	R_{max} $R_{реак}$, Тфлопс	Расположение инсталляции, страна/год
1	NEC SX-8 / 4096 / ОЗУ - 64 Тбайт	65,000	NEC, Japan / 21 10.2004
2	Silicon Graphic Columbia / 20 x 512 = 10240 Intel Itanium 2*	42,7 53,0	NASA, USA / 27. 10 2004
3	IBM BLUE GENE / L / 16 000	36,010	IBM, USA / 30 09 2004
4	NEC Earth-Simulator / 5120	35,860 40,960	Earth Simulator Center, Japan / 2002
5	Hewlett-Packard ASCI Q - AlphaServer SC ES45 / 1,25 ГГц / 4096	7,727 10,240	Los Alamos National Laboratory, USA / 2002

Окончание табл 3 9

Рейтинг	Производитель системы/количество процессоров	R_{max} $R_{реак}$, Тфлопс	Расположение инсталляции, страна/год
6	IBM ASCI White, SP Power, 375 МГц /8192	7,226 12,288	Lawrence Livermore National Laboratory, USA/2000
7	Linux NetworX MCR Linux Cluster Xeon 2,4 ГГц - Quadrics / 2304	5,694 11,060	Lawrence Livermore National Laboratory, USA/2002
8	Hewlett-Packard AlphaServer SC ES45 / 1 ГГц / 3016	4,463 6,032	Pittsburgh Supercomputing Center, USA/2001
9	Hewlett-Packard AlphaServer SC ES45 / 1 ГГц/ 2560	3,980 5,120	Commissariat a l'Energie Atomique (CEA), France/2001
10	HPTi Dual Xeon 2,2 ГГц - Myrinet2000 / 1536	3,337 6,758	Forecast Systems Laboratory - NOAA, USA/2002
11	IBM pSeries 690 Turbo 1 ,3 ГГц / 1280	3,241 6,656	HPCx UK/2002
12	IBM pSenes 690 Turbo 1,3 ГГц / 1216	3,164 63,23	NCAR (National Center for Atmospheric Research) USA/2002
74	Self-made MVS1000M EV67 667 МГц / 768	0,734 1,024	Joint Supercomputer Center Russian Federation/2002

* Система Columbia представляет собой кластер 20 машин SGI Altix, содержащих по 512 процессоров Intel Itanium 2 (см табл 4 2) Каждая машина работает под управлением экземпляра ОС Linux

Контрольные вопросы

1. Охарактеризуйте одиночный поток команд — одиночный поток данных (ОКОД).
2. Охарактеризуйте одиночный поток команд — множественный поток данных (ОКМД).
3. Охарактеризуйте множественный поток команд — одиночный поток данных (МКОД).
4. Охарактеризуйте множественный поток команд — множественный поток данных (МКМД).
5. Чем многомашинные ВС отличаются от многопроцессорных?
6. Приведите характеристику каждого из четырех классов архитектуры ВС согласно классификации по режиму выполнения?

7 Какие уровни комплексирования ЭВМ вам известны?
8. Чем отличаются многомашинные ВС от многопроцессорных ВС.
9. На какие классы подразделяются многопроцессорные параллельные ВС?
10. Что такое кластеры и какими преимуществами они обладают?
11. Что такое коммутационные среды? Приведите примеры коммутаторов.
12. Охарактеризуйте стратегии управления иерархической памятью.
13. Что такое вычислительные системы и каковы их разновидности?
14. Охарактеризуйте принципы функционирования машин типа *wavefront* и *reduction*.
15. Назовите основные классы и подклассы вычислительных машин и дайте их сравнительную характеристику.
16. Дайте общую характеристику и определите область использования суперЭВМ и мэйнфреймов.

Глава 4
ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ

История персональных компьютеров (ПК) началась в 80-е гг. XX в., когда практически одновременно компании Motorola, Zilog и Intel выпустили на рынок достаточно мощные микропроцессоры M68000, Z80 и Intel 8086.

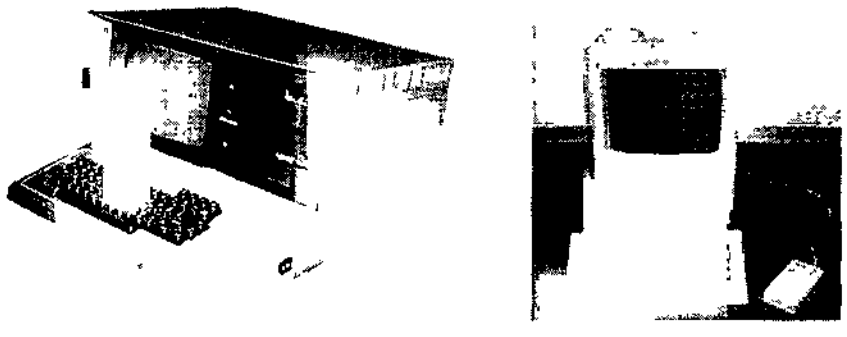
На этих микропроцессорах были построены первые микрокомпьютеры (ПК):

- Kaypro II (Zilog);
- Macintosh 128K (Motorola);
- IBM PC XT (Intel — INTegrated ELectronics).

Поскольку в дальнейшем основное внимание будет уделено IBM/PC-совместимым ПК и их «потомкам», вначале вкратце остановимся на параллельных ветвях развития ПК.

Kaypro II был представлен публике в августе 1982 г. (рис. 4.1, *а*). Несмотря на название, это была первая модель компании Э. Кея (Andrew F. Kay) Non-Linear Systems, Inc., позже переименованной в Kaypro Corporation.

При весе более десяти килограмм, Kaypro II позиционировался как переносная система. Возможность работы в полевых условиях



а *б*

Рис. 4.1. Первые ПК
а — Kaypro II, *б* — Macintosh 128K

была подтверждена во время ралли Париж-Дакар в 1984 г , на котором организаторами использовалось десять компьютеров Kaypro II.

Технические характеристики этой модели близки к системе Osborne 1, выпускаемой фирмой А Осборна, и HP-85 (Hewlett-Packard) Следует, однако, отметить, что при близости возможностей Kaypro II был почти вдвое легче Osborne 1, что для переносной системы имеет первоочередную важность, и на два года «моложе» HP-85.

Kaypro II оснащался процессором Zilog Z80 с тактовой частотой 2,5 МГц. Объем ОЗУ составлял 64 Кб, ПЗУ — 2 Кб. Встроенный дисплей размером 9" был способен работать в текстовом режиме 80 x 24, поддержка графического режима не предусматривалась Клавиатура компьютера была довольно удобной, с курсорными и цифровыми клавишами. Устройством хранения данных служили два дисковода для односторонних 5,25" дисков двойной плотности емкостью 190 Кб. Работал Kaypro II под управлением ОС CP/M (Control Program for Microprocessor) и стоил на момент выхода 1795 долл.

Модель 128K — первый продукт в семействе компьютеров Macintosh (рис. 4.1, *б)*. Корпорация Apple представила ее в январе 1984 г., развернув интенсивную маркетинговую кампанию. «Если уж компьютеры так умны, — говорилось в одной из реклам, — то не лучше ли научить их общаться с человеком, чем учить людей общаться с компьютером?»

Macintosh 128K базировался на 32-разрядном микропроцессоре Motorola 68000 с тактовой частотой 8 МГц, имел 128 Кб ОЗУ, 64 Кб ПЗУ, односторонний флоппи-дисковод (400 Кбайт; 3,5"), встроенный 9" черно-белый экран с графическим разрешением 512 x 342 точки, оснащался, помимо клавиатуры, манипулятором «мышь» и весил 20 фунтов (8 кг).

Многое в новом компьютере было заимствовано из разработок исследовательского центра PARC компании Xerox, 15 специалистов которой в начале 1980-х гг. перешли на работу в Apple. Именно благодаря достижениям PARC возникли система «окон» на экране, интерфейс, основанный на символических изображениях действий, устройство «мышь».

Первый Macintosh стоил в пять раз дороже, чем требовали спецификации проекта — 2495 долл. вместо 500 долл Но продажи шли очень успешно и за первые 9 месяцев было реализовано 275 тыс. компьютеров

«Мак», как вскоре окрестили поклонники новую машину, принес Apple грандиозный успех. Владельцы компьютеров других типов завидовали графическим средствам «Мака» и простоте обращения с

ним. Программы, позволявшие использовать «окна» и «мышь» в компьютерах других моделей, раскупались нарасхват.

Однако очень скоро пользователи Macintosh 128K столкнулись с проблемой слишком малого объема оперативной памяти, при том, что возможностей для расширения ОЗУ конструкцией модели предусмотрено не было. Посыпались многочисленные нарекания, побудившие корпорацию Apple усовершенствовать свой продукт, и уже осенью того же 1984 г. был представлен новый «Мак» — модель Macintosh 512K (модель Macintosh 128K была снята с производства в октябре 1985 г.).

4.1. Устройство ПК на процессорах Intel

Упрощенная блок-схема, отражающая основные функциональные компоненты компьютерной системы и их взаимосвязи, изображена на рис. 4.2.

Конструктивно ПК чаще всего выполнены в виде центрального (системного) блока, к которому через разъемы (стыки) подключаются внешние устройства: клавиатура, дисплей, принтер и т. д.

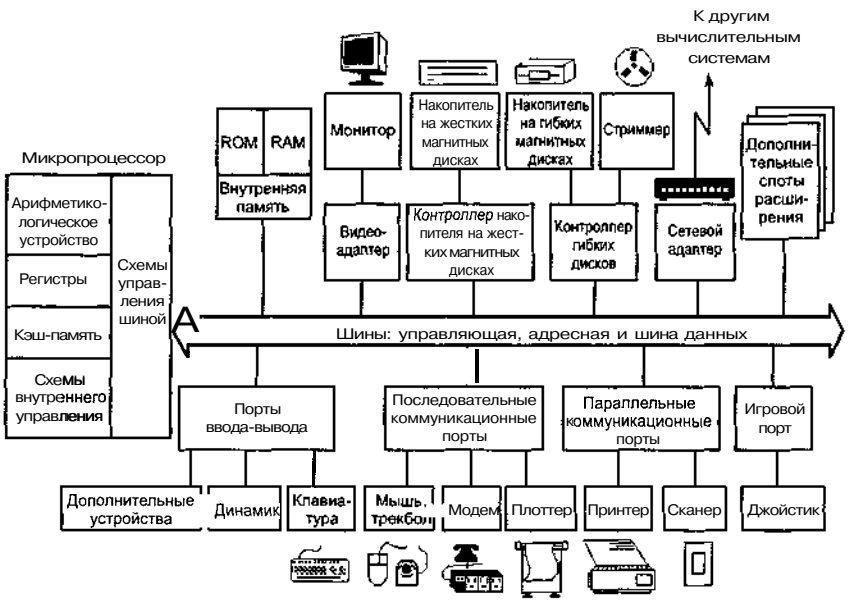


Рис. 4.2. Общая структура персонального компьютера с подсоединенными периферийными устройствами

Системный блок

Системный блок обычно включает в себя: системную плату, блок питания, накопители на дисках, разъемы для дополнительных устройств; платы расширения с контроллерами — адаптерами внешних устройств.

В зависимости от его конфигурации и размеров *корпуса* определяются такие характеристики ПК, как возможность дальнейшего расширения, транспортировка, доступ к компонентам и т. д. Типы корпусов: Slimline, Desktop, Tower (Mini-Tower, Midi-Tower, Super-Big-Tower), File Server (рис. 4.3).

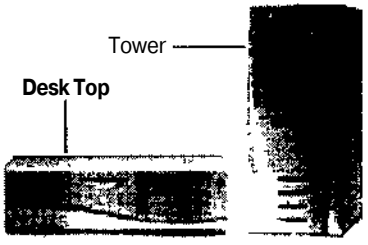


Рис. 4.3. Различные типы корпусов ПК

Системная плата (рис. 4.4) является основой системного блока, который обеспечивает внутренние связи, взаимодействует через прерывания с внешними устройствами и содержит компоненты, определяющие архитектуру ПК. Системные платы с различными микропроцессорами отличаются друг от друга по типам применяемых элементов и модулей памяти, возможностям конфигурирования и т. д. Набор микросхем на системной плате, обеспечивающий работу ЦП по обмену данными с периферийными устройствами, называют Chipset.

Чипсет

Chipset, или «PCIset» — совокупность микросхем, размещенных на системной плате, которые организуют потоки команд и данных в ПЭВМ. Сюда входят: основная память, вторичная кэш-память и устройства, связанные с шинами ISA и PCI. Кроме того, чипсет контролирует потоки данных НЖМД и других устройств, соединенных с каналом IDE. Иногда в состав чипсета включают и сам микропроцессор.

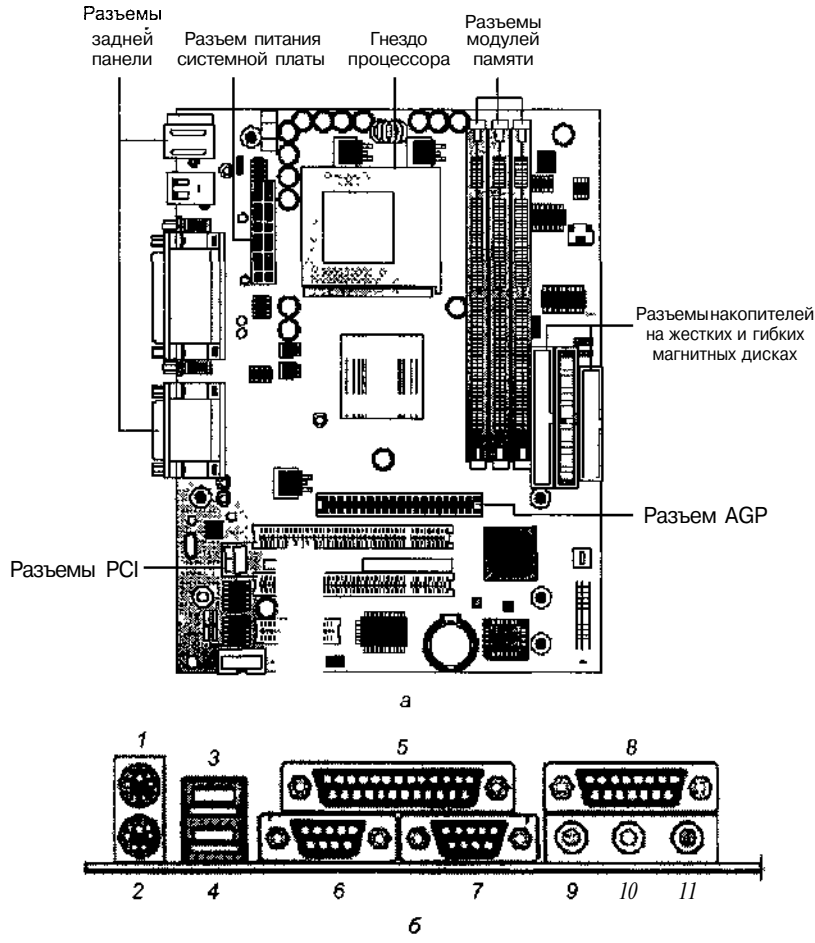


Рис. 4.4. Системная плата (Motherboard) ATX *(а)* и ее основные разъемы *(б): 1, 2* — разъемы PS/2 (мышь и клавиатура); *3, 4* — разъемы USB; 5 — принтер (LPT1); *6—9* — штырьковые (D9) разъемы последовательных портов; *8* — игровой порт (joistik); *9, 10, 11* — разъемы звуковой платы (микрофон, линейный вход, линейный выход)

Чипсеты производят различные фирмы — SIS, VIA, OPTI, однако в течение многих лет наиболее популярными остаются чипсеты Intel Triton. Чипсеты Triton позволяют соединять и эффективно использовать такие устройства, как процессор Pentium, устройства, связанные с каналами EIDE, ISA, а также поддерживает технологии памяти, такие, как EDO и SDRAM.

Рассмотрим вкратце структуру и характеристики Triton 430TX, который оптимизирован для использования с процессорами Pentium MMX (рис. 4.5).

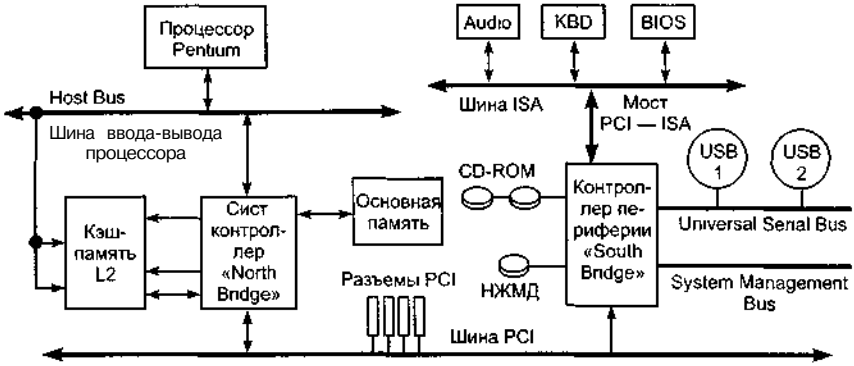


Рис. 4.5. Архитектура Triton 430 TX (Northbridge/Southbridge)

Основой Triton 430TX являются две микросхемы высокой интеграции, которые были введены еще в предыдущей версии чипсета (Triton 430HX), а именно:

- системный контроллер 82439TX (MTXC);
- контроллер периферии 82371 AB PCI ISA IDE Xcelerator (PIIX4).

Первый интегрирует функции основной и кэш-памяти второго уровня, управляет взаимодействием ЦП, кэш-памяти, основной памяти и шиной PCI. Второй представляет собой многофункциональное устройство PCI, которое реализует функции моста между PCI и ISA, PCI и IDE, а также управляет потреблением электроэнергии (функции Enhanced power management). Поскольку системный контроллер получил название «Northbridge», периферийный — «Southbridge», данная архитектура известна как Northbridge/ Southbridge.

В дальнейшем данный тип архитектуры использовался в чипсетах Intel Triton 440LX, EX, BX, ZX, GX, 450 NX, а затем был заменен (чипсеты 820, 815, 850 и др.) на архитектуру «Accelerated hub», включающую три компонента (рис. 4.6):

- контроллер памяти (Memory Controller Hub);
- контроллер ввода-вывода (I/O Controller Hub);
- постоянная память (ПЗУ — Firmware Hub).

Контроллер памяти обеспечивает высокоскоростное взаимодействие между ЦП, ОЗУ и AGP и позволяет управлять памятью разме-

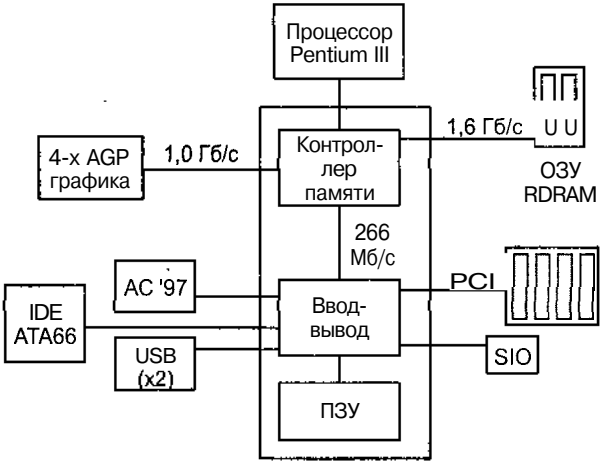


Рис. 4.6. Чипсет Intel 820 архитектуры Accelerated hub

ром до 1 Гбайт. При этом в системе могут использоваться как один, так и несколько процессоров. Контроллер ввода-вывода поддерживает прямую связь между внешними устройствами и ОЗУ; кроме того, он выполняет функции контроллера USB, IDE и AC'97. Система ПЗУ содержит схемы BIOS и видео BIOS, а также в нее вмонтирован датчик случайных чисел (Intel RNG). В отличие от программных датчиков псевдослучайных чисел (ПСЧ), Intel RNG генерирует числа, действительно случайные, для чего используется эффект теплового шума.

Интерфейсы ПК

Об интерфейсах речь шла ранее, здесь лишь отметим, что на сегодняшний день имеется несколько основных спецификаций (стандартов) компьютеров. Спецификациями де-факто считаются созданные совместно Intel и Microsoft PC98, PC99 и последняя разработка PC2001.

Данные стандарты были разработаны для обобщения архитектуры PC. Это позволяет предотвратить спад в скорости развития компьютерных технологий (все производители «железа» приостанавливают собственные разработки в High-end секторе и занимают выжидательную позицию по отношению к своим конкурентам).

Материнская плата имеет немаловажное значение при определении соответствия вашего PC одной из спецификаций. Следова-

тельно, необходимо рассмотреть требования данных спецификаций, возможности, ими предоставляемые, и перспективы развития.

Крупные производители не видят компьютера будущего без возможности объединения в глобальную информационную сеть (в первой же из приведенных спецификаций содержится обязательное требование сетевого адаптера или модема).

Одно из наиболее быстро и динамично развивающихся направлений — это обработка мультимедиа-информации. Требования к видеосистемам растут непропорционально быстро по отношению к другим ресурсам PC.

Основной тенденцией, проявляющейся в каждой последующей спецификации, является поэтапный отказ от ISA-устройств. Как в виде отдельных плат, так и от микросхем поддержки ISA-интерфейса, интегрированных на материнской плате. Первым шагом (уже совершенным) было обязательное присутствие PCI-слотов. Далее была разработана спецификация PC98 с обязательным отказом от ISA-интерфейса. Результат, к которому должны прийти PC — отсутствие ISA-слотов, PS/2-портов и накопителей 1,2 Mb, 1,44 Mb. Основным внешним интерфейсом становится Fireware (IEEE 1394). Уже нередки ситуации, когда производители периферии снимают с производства модели, не поддерживающие новые спецификации интерфейсов.

Дополнительные интегральные микросхемы

К системной шине и к МП ПК наряду с типовыми внешними устройствами могут быть подключены и некоторые дополнительные интегральные микросхемы, *расширяющие* и улучшающие функциональные возможности микропроцессора:

- математический сопроцессор;
- контроллер прямого доступа к памяти;
- сопроцессор ввода-вывода;
- контроллер прерываний и т. д.

Математический сопроцессор широко используется для ускоренного выполнения операций над двоичными числами с фиксированной и плавающей запятой, над двоично-кодированными десятичными числами, для вычисления некоторых трансцендентных, в том числе тригонометрических функций. Математический сопроцессор имеет свою систему команд и работает параллельно (совмещенно во времени) с основным МП, но под управлением последнего. Ускорение операций происходит в десятки раз. Современные

модели МП начиная с МП80486 DX включают сопроцессор в свою структуру.

Контроллер прямого доступа к памяти (DMA — Direct Memory Access) обеспечивает обмен данными между внешними устройствами и оперативной памятью без участия микропроцессора, что существенно повышает эффективное быстродействие ПК. Иными словами, режим DMA позволяет освободить процессор от рутинной пересылки данных между внешними устройствами и ОП, отдав эту работу контроллеру ОМА; процессор в это время может обрабатывать другие данные или другую задачу в многозадачной системе.

Сопроцессор ввода-вывода за счет параллельной работы с МП существенно ускоряет выполнение процедур ввода-вывода при обслуживании нескольких внешних устройств (дисплея, принтера, НЖМД, НГМД и т. д.); освобождает МП от обработки процедур ввода-вывода, в том числе реализует и режим прямого доступа к памяти.

Контроллер прерываний обслуживает процедуры прерывания. Прерывание — временная приостановка выполнения одной программы с целью оперативного выполнения другой, в данный момент более важной (приоритетной) программы. Контроллер принимает запрос на прерывание от внешних устройств, определяет уровень приоритета этого запроса и выдает сигнал прерывания в МП. Микропроцессор, получив этот сигнал, приостанавливает выполнение текущей программы и переходит к выполнению специальной программы обслуживания того прерывания, которое запросило внешнее устройство. После завершения программы обслуживания восстанавливается выполнение прерванной программы. Контроллер прерываний является программируемым. Прерывания возникают при работе компьютера постоянно, достаточно сказать, что все процедуры ввода-вывода информации выполняются по прерываниям. Например, в компьютерах IBM PC прерывания от таймера возникают и обслуживаются контроллером прерываний 18 раз в секунду (длятся эти прерывания тысячные доли секунды и поэтому пользователь их не замечает).

Основная память

Основная память (ОП) предназначена для хранения и оперативного обмена информацией с прочими блоками машины. ОП содержит два вида запоминающих устройств:

- постоянное запоминающее устройство (ПЗУ, ROM — Read Only Memory) предназначено для хранения неизменяемой (по-

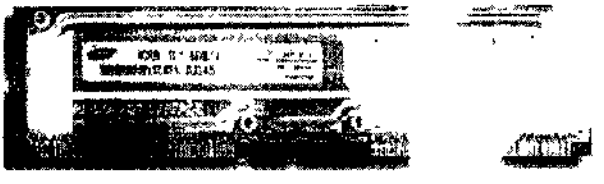
стоянной) программной и справочной информации; позволяет оперативно только считывать информацию, хранящуюся в нем (изменить информацию в ПЗУ нельзя);

- оперативное запоминающее устройство (ОЗУ, RAM — Random Access Memory) предназначено для оперативной записи, хранения и считывания информации (программ и данных), непосредственно участвующей в информационно-вычислительном процессе, выполняемом ПК в текущий период времени.

Большинство современных компьютеров комплектуются модулями типа DIMM (Dual-In-line Memory Module — модуль памяти с двухрядным расположением микросхем). В компьютерных системах на самых современных процессорах используются высокоскоростные модули Rambus DRAM (RIMM) и DDR DRAM.

Модули памяти характеризуются такими параметрами, как объем (16, 32, 64, 128, 256 или 512 Мбайт), число микросхем, паспортная частота (100 или 133 МГц), время доступа к данным (6 или 7 нс) и число контактов (72, 168 или 184). С 2001 г. производится выпуск модулей памяти на 1 Гбайт и опытных образцов модулей на 2 Гбайта.

Основная память соединяется с процессором через *адресную шину* и *шину данных*. Каждая шина состоит из множества электрических цепей или бит. Ширина (разрядность) адресной шины определяет, сколько адресов может быть в ОЗУ (адресное пространство), а шины данных — сколько данных может быть передано за 1 цикл.



а

б

Рис. 4.7. Микросхемы памяти RIMM (*а*) и DIMM *(б)*

Например, в 1985 г. процессор Intel 386 имел 32-разрядную адресную шину, что дало возможность поддерживать адресное пространство в 4 Гбайт. В процессоре Pentium (1993 г.) ширина шины данных была увеличена до 64 бит, что позволило передавать 8 байт информации одновременно.

Каждая передача данных между процессором и памятью называется *циклом шины.* Количество бит, которое процессор может передать за один цикл шины влияет на производительность компьютера и определяет, какой тип памяти требуется.

Рассмотрим пример — считывание из памяти и запись в память байта с адресом 7 (рис. 4.8).

При считывании данных из памяти некоторая внешняя (по отношению к запоминающему устройству) система (например, микропроцессор) устанавливает на шине чтения-записи сигнал 1, сообщая таким образом, что должна производиться операция считывания. Кроме того, эта внешняя система помещает на адресную шину значение 0000000000000111 в двоичной системе счисления (или, что то же самое, 7 — в десятичной системе счисления). Тем самым запоминающему устройству сообщается, что требуется считать байт информации из ячейки 7. Очевидно, что в результате операции считывания содержимое указанного байта, равное 10101010, появится на информационной шине.

При записи на шине чтения-записи устанавливается сигнал «О» и на адресную и информационную шины помещаются адреса и записываемые данные. В качестве адреса запоминающее устройство получает число 0000000000000111, или 7, а в качестве записываемых данных — число 10101010. Поскольку на шине чтения-записи установлен сигнал 0, данные с информационной шины заносятся в ячейку 7.

Кэш (англ. *cache),* или сверхоперативная память — очень быстрое ЗУ небольшого объема, которое используется при обмене данными между микропроцессором и оперативной памятью для компенсации разницы в скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Кэш-памятью управляет специальное устройство — контроллер, который, анализируя выполняемую программу, пытается предвидеть, какие данные и команды вероятнее всего понадобятся в ближайшее время процессору, и подкачивает их в кэш-память. При этом возможны как «попадания», так и «промахи». В случае попадания, то есть, если кэш содержит нужные данные, извлечение их из памяти происходит без задержки. Если же требуемая информация в кэше отсутствует, то процессор считывает ее непосредственно из

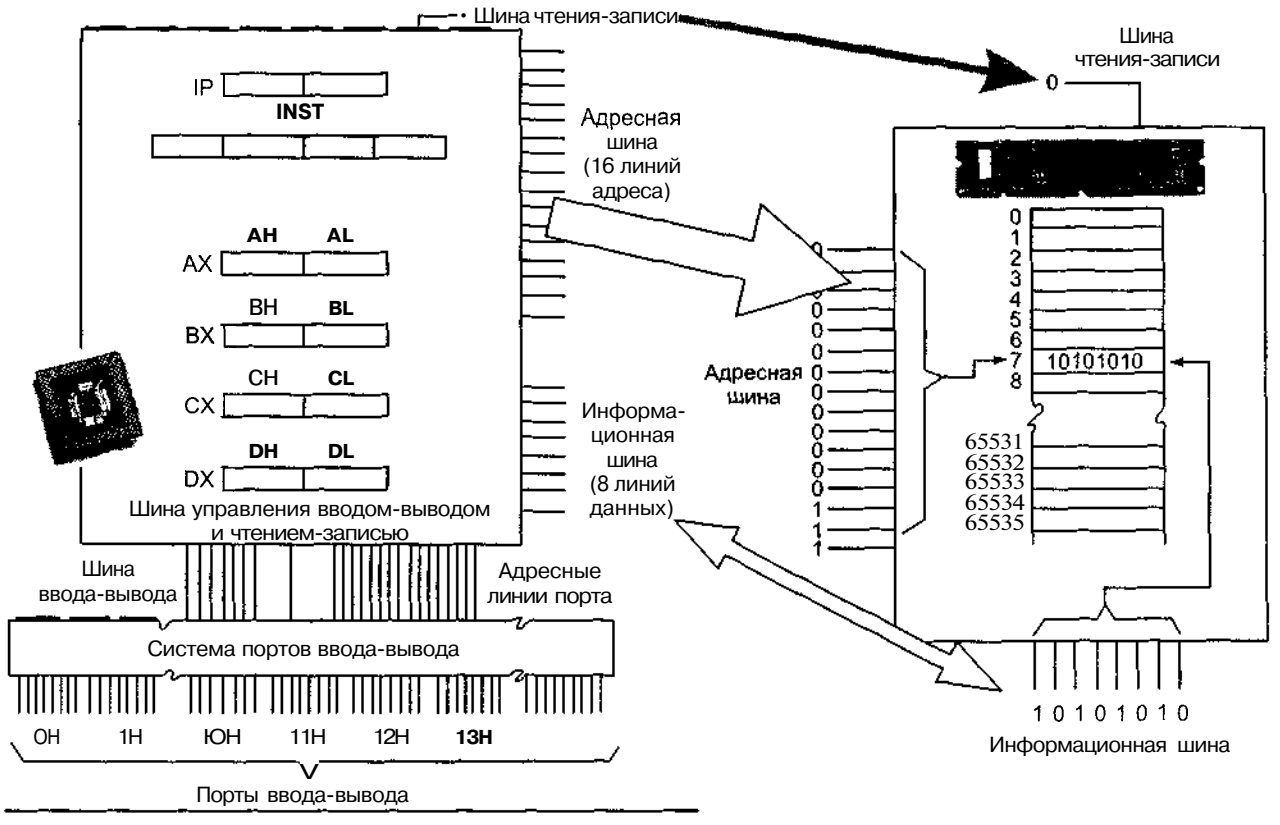


Рис. 4.8. Взаимодействие процессора и ОЗУ

оперативной памяти. Соотношение числа попаданий и промахов определяет эффективность кэширования.

Кэш-память реализуется на микросхемах статической памяти SRAM (Static RAM), более быстродействующих, дорогих и малоемких, чем DRAM (SDRAM). Современные микропроцессоры имеют встроенную кэш-память, так называемый кэш первого уровня (L1) размером 8, 16 или 32 Кбайт. Кроме того, на системной плате компьютера может быть установлен кэш второго уровня (L2) емкостью 256, 512 Кбайт и выше.

Специальная память

К устройствам специальной памяти относятся постоянная память (ROM), перепрограммируемая постоянная память (Flash Memory), память CMOS RAM, питаемая от батарейки, видеопамять и некоторые другие виды памяти.

Постоянная память (ПЗУ, англ. ROM, Read Only Memory — память только для чтения) — энергонезависимая память, используется для хранения данных, которые никогда не потребуют изменения. Содержание памяти специальным образом «зашивается» в устройстве при его изготовлении для постоянного хранения. Из ПЗУ можно только читать.

Перепрограммируемая постоянная память (Flash Memory) — энергонезависимая память, допускающая многократную перезапись своего содержимого.

Прежде всего в постоянную память записывают программу управления работой самого процессора. В ПЗУ находятся программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

Важнейшая микросхема постоянной или Flash-памяти — модуль BIOS. Роль BIOS двоякая: с одной стороны, это неотъемлемый элемент аппаратуры, а с другой стороны — важный модуль любой операционной системы.

BIOS (Basic Input/Output System — базовая система ввода-вывода) — совокупность программ, предназначенных для автоматического тестирования устройств после включения питания компьютера и загрузки операционной системы в оперативную память.

Разновидность постоянного ЗУ — CMOS RAM (рис. 4.9).

CMOS RAM — это память с невысоким быстродействием и минимальным энергопотреблением от батарейки. Используется для

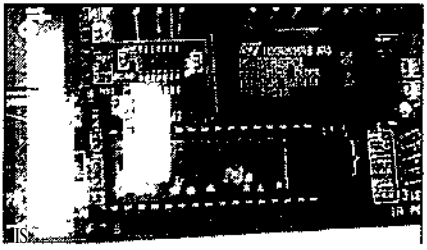


Рис. 4.9. Интегральные схемы BIOS и CMOS

хранения информации о конфигурации и составе оборудования компьютера, а также о режимах его работы.

Содержимое CMOS изменяется специальной программой Setup, находящейся в BIOS (англ. Set-up — устанавливать, читается «сетап»).

Для хранения графической информации используется видеопамять.

Видеопамять (VRAM) — разновидность оперативного ЗУ, в котором хранятся закодированные изображения. Это ЗУ организовано так, что его содержимое доступно сразу двум устройствам — процессору и дисплею. Поэтому изображение на экране меняется одновременно с обновлением видеоданных в памяти.

Система прямого доступа к памяти

Как уже отмечалось, процессор способен считывать данные из некоторого устройства и записывать их в память. Подобные действия может выполнять и система прямого доступа к памяти и таким образом освобождать процессор от этой работы. Пересылка байтов данных является простой задачей, не требующей особых ухищрений. Система прямого доступа к памяти решает ее, пока процессор продолжает выполнение.

Direct Memory Access (DMA) — метод обращения внешнего устройства к памяти компьютера без участия ЦП. Внешнее устройство, которое может с помощью DMA напрямую обратиться к памяти, обладает нужным для этого «интеллектом». Для этого устройство имеет свой процессор или способно выполнять необходимые логические операции.

На рис. 4.10 показано, каким образом процессор и система прямого доступа к памяти связаны с запоминающим устройством. Обе системы могут посылать и принимать данные из некоторого периферийного устройства. Для доступа к этому устройству процессор

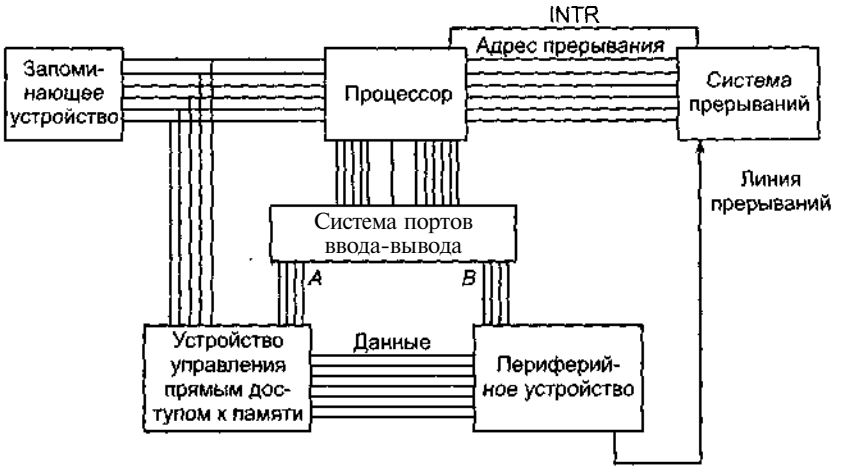


Рис. 4.10. Прямой доступ к памяти

использует порт *B*, а система прямого доступа — шину, обозначенную как «*Данные*». Наконец, процессор может вести обмен данными с системой прямого доступа к памяти через порт *A*.

Рассмотрим в качестве примера процесс считывания нескольких байтов данных из периферийного устройства и занесения их в память. Процесс начинается, когда процессор посылает одну команду в систему прямого доступа к памяти через порт *A,* а другую — в устройство через порт *B.* По команде, направленной в устройство, последнее должно переслать несколько байтов данных в систему прямого доступа к памяти. Согласно команде, посланной в систему прямого доступа, эта система должна принять байты данных из устройства и записать их в память. Во время выполнения описанных пересылок процессор может продолжать считывание и выполнение команд.

Может возникнуть вопрос, каким образом система прямого доступа и процессор могут одновременно осуществлять доступ к памяти? На самом деле такой возможности нет. Запоминающее устройство в каждый момент времени обрабатывает только один запрос. Однако система прямого доступа к памяти достаточно «интеллектуальна» и в состоянии задержать запрос процессора, пока сама реализует обращение. Таким образом, процессор и система прямого доступа к памяти поочередно работают с запоминающим устройством, причем процессор время от времени находится в состоянии ожидания, пока освободится память. С запросами периферийного

устройства на передачу данных затруднений обычно не возникает, так как они поступают существенно реже, чем запросы процессора.

Система прямого доступа лишь иногда обращается к памяти, в то время как процессор постоянно требует доступа к запоминающему устройству. Поэтому создается впечатление, что и процессор, и система прямого доступа работают с памятью одновременно.

4.2. Процессоры Intel

С развитием микроэлектронной технологии и увеличением степени интеграции элементов размещенный в одной электронной схеме (чипе) «процессор» стал называться «микропроцессором» (МП). По мере развития МП его состав, архитектура и параметры естественно менялись. Стержнем материнской (системной) платы ПК является именно МП, который и называется центральным процессорным устройством — ЦП. Во время работы он все время находится в постоянном взаимодействии с другими элементами. МП характеризуется следующими основными параметрами:

- тактовой частотой (элементы, расположенные на материнской плате, работают строго в определенные моменты времени, в соответствии с тактовой частотой, чтобы координировать друг с другом отдельные шаги работы и синхронизировать весь процесс — в общем, обработка информации происходит тем быстрее, чем выше тактовая частота) (рис. 4.11);
- степенью интеграции микросхемы (сколько транзисторов содержится в чипе) (рис. 4.12);
- внутренней разрядностью данных (количество бит, которые МП может обрабатывать одновременно);
- внешней разрядностью данных (количество одновременно передаваемых бит в процессе обмена данными ЦП с другими элементами);
- адресуемой памятью (зависит от числа адресных бит).

Этапы развития МП, соответствующие достижения, их основные архитектурные и иные характеристики естественно рассмотреть на основе МП фирмы Intel (INTegrated ELectronics), создавшей более 25 лет тому назад первый полупроводниковый прибор, который с фантастической скоростью превратил компьютер в повседневный рабочий инструмент десятков миллионов людей и оказал большое влияние практически на все сферы человеческой деятельности (табл. 4.1).

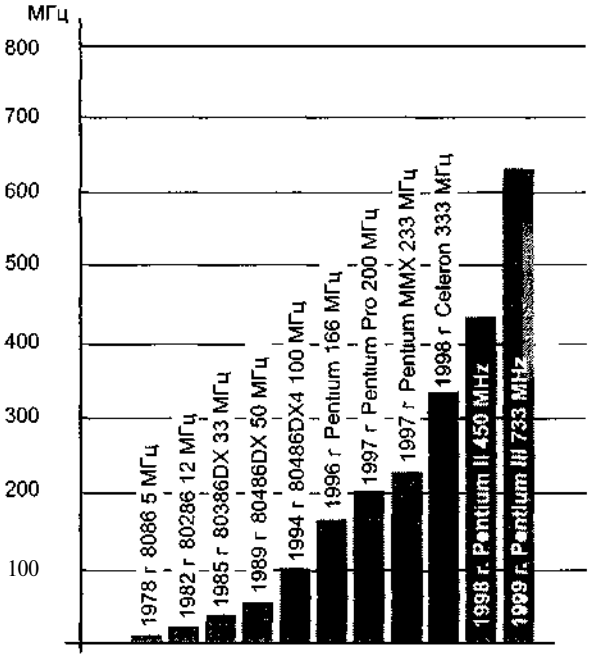


Рис. 4.11. Динамика роста тактовой частоты процессоров Intel с 1978 по 1999 г

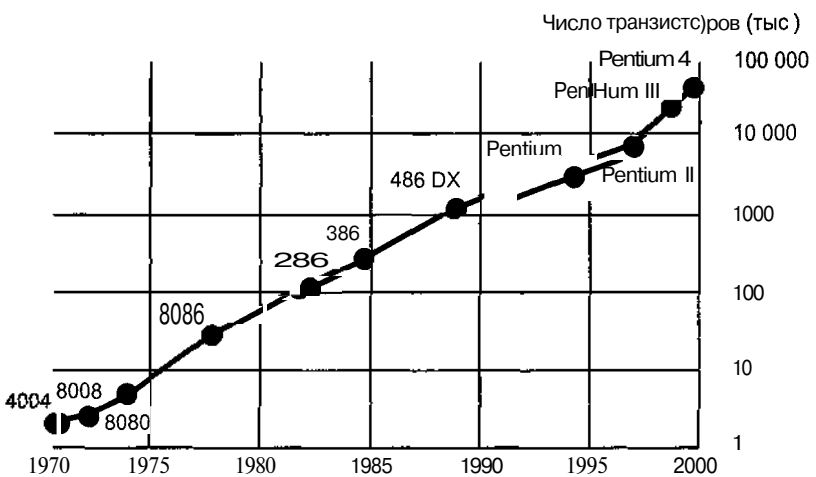


Рис. 4.12. Правило Мура (количество транзисторов в интегральной схеме удваивается каждые 18 месяцев)

Таблица 4 / Характеристики процессоров Intel и AMD

Тип процессора	Поколение	Год выпуска	Разрядность шины данных	Разрядность шины адреса	Первичная кэш-память, Кбайт		Тактовая частота шины, МГц	Тактовая частота процессора, МГц	Количество транзисторов, млн	Размер минимальной структуры, мкм
					Команды	Данные				
8088	1	1979	8	20	Нет		4,77-8	4,77-8		
8086	1	1978	16	20	Нет		4,77-8	4,77–8	0,029	3,0
80286	2	1982	16	24	Нет		6-20	6-20	0,130	1,5
80386DX	3	1985	32	32	Нет		16-33	16-33	0,27	1,0
80386SX	3	1988	16	32	8		16-33	16-33	0,27	1,0
80486DX	4	1989	32	32	8		25-50	25-50	1 2	1,0-0,8
80486SX	4	1989	32	32	8		25-50	25-50	1,1	0,8
80486DX2	4	1992	32	32	8		25-40	50-80		
80486DX4	5	1994	32	32	8	8	25-40	75-120		
Pentium	5	1993	64	32	8	8	60-66	60-200	3 1–3 3	0,8-0,35
P-MMX	5	1997	64	32	16	16	66	166-233	4,5	0,6-0,35
Pentium Pro	6	1995	64	32	8	8	66	150-200	55	0,35
Pentium II	6	1997	64	32	16	16	66	233-300	75	0,35-0,25
Pentium II Celeron	6	1998	64	32	16	16	66/100	266-533	75-19	0,25
Pentium Xeon	6-7	1998					100	400-1700		018
Pentium III	6	1999	64	32	16	16	100	450–1200	9,5-44	0,25-0,13
AMD Athlon	7	1999	64	32	64	64	266	500-2200	22	0,25
Pentium 4	7	2000	64	32	12	8	400	1,4–3,4 ГГц	42-125	0,18-0,09
AMD Athlon 64	8	2003	64	64	64	64	400	2 ГГц	54-106	0,13-0,09

Intel 4004 (1971 г.)

Этот прибор послужил отправной точкой абсолютно новому классу полупроводниковых устройств. Чип представлял 4-разрядный процессор с классической архитектурой ЭВМ гарвардского типа (название «гарвардская архитектура» связано с компьютером «Марк-I», который имел отдельную память для команд — 1950 г.), насчитывал 2300 транзисторов и работал на тактовой частоте 750 кГц (длительность цикла команды 10,8 мкс).

Чип имел адресный стек, содержащий счетчик команд и три регистра стека типа LIFO (Last In — First Out — «последним поступил — первым выводится», специальная память с ограниченной емкостью с указанной дисциплиной обслуживания); блок регистров общего назначения — РОН (регистры сверхоперативной памяти или регистровый файл); 4-разрядное параллельное АЛУ; аккумулятор; регистр команд; дешифратор команд; схему управления; схему связи с периферийными устройствами. Эти функциональные узлы объединялись 4-разрядной внутренней шиной данных. Блок РОН'ов состоял из шестнадцати 4-разрядных регистров, которые можно было использовать и как восемь 8-разрядных регистров. Такая организация РОН сохранилась и в последующих МП фирмы Intel. Система команд содержала 46 универсальных инструкций. Цикл команды МП состоял из 8 тактов задающего генератора (так как чип монтировался в корпусе с 16 выводами и имел узкий интерфейс с «внешним миром», то приходилось применять мультиплексирование шины адреса и данных, причем 12-разрядный адрес выдавался порциями по 4 разряда, а прием команды требовал еще двух тактов, на выполнение самой инструкции затрачивалось всего три такта). Адресуемая память команд достигала 4 Кбайт (для сравнения: объем памяти мини-ЭВМ в начале 70-х гг. редко превышал 16 Кбайт).

Недостатки были в скором времени устранены в МП 4040, где количество РОН было доведено до 24, причем они были разделены на две области, выбираемые с помощью специальных команд, т. е. процессор теперь мог обращаться к двум блокам памяти команд емкостью 4 Кбайт, и за каждым из них можно было закрепить свою область регистров (8 РОН были всегда доступны для использования). Можно было разрабатывать самостоятельные программные модули, способные взаимодействовать через общую часть регистрового файла. Но самое главное — это обработка одноуровневых прерываний, что превратило МП в прибор, применяемый в системах реального масштаба времени. «Остановка» создала возможность синхронизации МП с некоторыми внешними событиями. 60 инст-

рукций, 8 Кбайт памяти команд, обработка прерываний стали достоинством этого МП и вывели его на первое место на рынке МП. Тем не менее специальных команд работы со стеком пока не было.

Intel 8008 (1972 г.)

Первый 8-разрядный МП. Чип содержал уже 3500 транзисторов, работал на частоте 500 кГц при длительности машинного цикла 20 мкс (10 периодов задающего генератора) и в отличие от предшественников имел архитектуру ЭВМ принстонского типа (компьютер с единой памятью для команд и данных — архитектура Джона фон Неймана). В нем допускалось применение комбинаций постоянной и оперативной памяти. Значительные изменения (кроме увеличения разрядности) произошли и в регистровом файле. Из-за ограниченных возможностей применяемой технологии в качестве блока РОН была применена динамическая память, которая требовала производить регенерацию (были введены дополнительные аппаратные средства). МП большинство команд выполнял за 1—3 машинных цикла. Для работы с медленно действующими устройствами был введен сигнал готовности *(ready)*. Система команд содержала в общем 65 инструкций и отличалась значительным количеством команд условного перехода, логических команд и команд сдвигов. Теперь МП мог адресоваться к памяти объемом 16 Кбайт. Однако объем и организация стека остались прежними, реализация операций со стеком по-прежнему возлагалась на программиста, узкий интерфейс с «внешним миром» требовал применения около 20 схем средней интеграции для сопряжения процессора с памятью и устройствами ввода-вывода.

Intel 8080 (1974 г.)

Этот микропроцессор практически по всем параметрам разительно отличался от своих предшественников. Он работал с тактовой частотой 2 МГц, цикл команды — 2 мкс. Адресуемый объем памяти достиг 64 Кбайт, был внедрен эффективный механизм обработки прерываний, в результате использования корпуса с 40 выводами удалось разделить адресную и информационную шины процессора.

В регистровый файл МП были введены указатель стека (активно использовался при обработке прерываний) и два программно-недоступных регистра для внутренних пересылок. В систему команд были

включены инструкции, адресующие память с использованием трех пар регистров Блок РОН'ов был построен на основе статической, а не динамической памяти. Аккумулятор из регистрового файла был перенесен в состав АЛУ В результате упростилась схема управления внутренней шиной (отпала необходимость использования этой шины для передачи данных между сверхоперативной памятью и АЛУ при выполнении арифметических и логических операций)

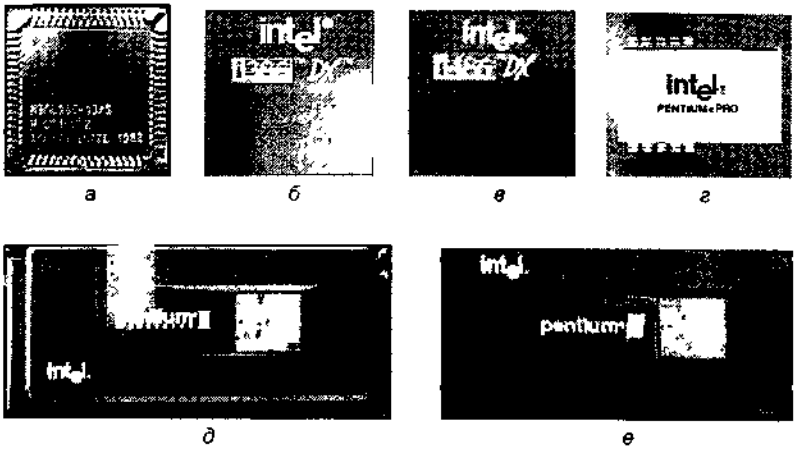
Новым в архитектуре МП стало использование многоуровневой системы прерываний по вектору (общее число источников прерываний достигло 256). Но здесь для реализации прерываний пока требовалось применить до 10 дополнительных чипов (это требование отпало после появления специализированных БИС контроллеров прерываний) В качестве стека можно было использовать отдельную память емкостью до 64 Кбайт и тем самым повысить эффективность использования оперативной памяти (экономия памяти)

Из универсальных ЭВМ в 8080 был перенесен механизм прямого доступа к памяти (ПДП-DMA), произошло освобождение ЦП от управления внешними устройствами и обменом данными между памятью и периферией (минуя ЦП). ПДП дал возможность применять в микроЭВМ накопители на магнитных дисках и лентах, дисплеи на ЭЛТ, и все это превратило микроЭВМ в полноценную вычислительную систему.

МП был окружен целым семейством новых микросхем (БИС контроллера ПДП, контроллера прерываний и др.). В результате этого проектирование микроЭВМ на базе семейства БИС значительно упростилось. МП стал стандартом де-факто, но были и недостатки (например, уже имелись МП других фирм, которые оказались более прозрачными для программистов). Далее была выпущена серия периферийных контроллеров. Но самым значительным достижением было то, что компания создала системное программное обеспечение (ПО) — однопользовательскую операционную систему (ОС) ISIS II и ОС реального времени iRMX-80 (мощнейшая в то время программная поддержка своих изделий). Фирма в 1976 г. приступила к выпуску одноплатных микроЭВМ серии iSBC на базе своих МП-комплектов.

Intel 8086 (объявлен 8 июня 1978г.)

МП содержал 29 тыс. транзисторов (рис 4.13). Высокое быстродействие элементов обеспечило тактовую частоту 5 МГц, а 16-разрядная архитектура и машинный цикл 200 нс — производительность, превышающую аналогичный параметр 8080 на порядок.



а б в г д е

Рис. 4.13. Процессоры Intel:
а — 8086, *б* — 80386, *в* — 80486, г — Pentium Pro, *д* — Pentium II, *е* — Pentium III

Именно стратегия эволюционного, а не революционного развития, выбранная фирмой Intel, была верна и давала свои плоды. Программная совместимость была исключительно важной характеристикой, которая объединяла 86-й кристалл с его предшественниками. Структура МП была полностью перестроена, он как бы был разделен на два одновременно работающих функциональных блока. Это операционный блок (EU — Execution Unit) и блок интерфейса (BIU — Bus Interface Unit). В результате исполнение одной команды совмещалось во времени с выборкой следующей команды или данных из памяти. Таким образом, из универсальных ЭВМ МП позаимствовали еще одно техническое решение — реализацию принципов параллелизма В ЦП появился небольшой буфер команд, что давало дополнительную экономию времени при обращениях к памяти

Адресация 1 Мбайт оперативной памяти (благодаря 20 адресным линиям) и ее сегментация могут быть отнесены к одним из наиболее существенных достижений. Сегментация памяти и большое число уровней прерываний были ориентированы на работу системы в многозадачном режиме. Однако механизм защиты памяти пока реализован не был, и это в ряде случаев существенно усложняло разработку программного обеспечения.

Наряду с поддержкой ввода-вывода по каналу ПДП дополнительно обеспечивалась адресация до 64 К портов программно-управляемого ввода-вывода.

МП 8086 мог работать в двух режимах: минимальном (рассчитанном на использование в небольших системах без применения

БИС контроллера шины) и максимальном (ориентированном на применение МП в сложных системах с использованием БИС контроллера шины). В систему команд входило 147 инструкций, позволяющих решать задачи управления практически любой сложности. Появились операции умножения и деления 16-разрядных чисел со знаком и без знака, команды обработки массивов данных, программно-управляемые прерывания и др., что превратило чип в универсальный прибор, который мог успешно применяться как для построения сложных контроллеров, так и в качестве ЦП ЭВМ общего назначения. Кроме того, МП вышел в мощном сопровождении средств поддержки (вспомогательных БИС, средств разработки и отладки аппаратуры и системного ПО и т. д.). Наряду с этим фирмой Intel был выпущен процессор 8088 с 8-разрядной внешней шиной данных. На основе применения экономичных 8-разрядных микросхем появился ПК с МП 8088 (фирма IBM, ПК класса XT — Extended Technology — расширенная технология, тактовая частота 4,77 МГц). На базе 8086 были выпущены младшие модели 25 и 30 семейства PS/2.

Использование чипов 8086 в IBM PC предопределило дальнейшее развитие корпорации Intel как разработчика и изготовителя универсальных процессоров общего назначения. Был изготовлен 16-разрядный арифметический сопроцессор 8087, который позволил превратить ПК еще и в достаточно мощный инструмент для решения задач вычислительного характера.

Intel 80286 (1 февраля 1982 г.)

В этом процессоре было введено большое количество новшеств. В 1984 г. фирма IBM использовала этот МП в PC AT (AT — Advanced Technology — улучшенная технология). Вот основные новшества этого чипа:

- адресное пространство составляло 16 Мбайт (вместо 1 Мбайт у предшественников), так как использовалась 24-разрядная шина адресов;
- поддержка виртуальной памяти (это позволяло использовать внешнюю память для имитации большой реальной внутренней памяти емкостью до 1 Гбайт);
- аппаратная мультизадачность (эта архитектурная особенность позволяла в ПК одновременно выполнять несколько задач с большой скоростью переключения с одной на другую);
- повышенное быстродействие (4 МГц, однако вскоре рабочая частота была повышена до 8 МГц и стала стандартной, хотя

производители клонов эту частоту довели до 10; 12,5; 16 и 20 МГц);

- встроенная система управления памятью и средства ее защиты (открывали широкие возможности использования МП в многозадачных средах);
- дополнение системы команд 16 новыми инструкциями,
- размещение в одном кристалле контроллеров прерываний и ПДП, а также таймера и системного генератора.

МП мог работать в двух режимах — *реальном* (МП действовал как 8086, что обеспечивало совместимость с DOS и существующим ПО) и *защищенном* (в этом режиме МП реализовывал режим виртуальной памяти, аппаратную мультизадачность и адресацию к большему пространству памяти);

Операционная система MS DOS может работать только в реальном режиме. Другие операционные системы, например, OS/2 (предложенная фирмой IBM — альтернатива DOS) и системы UNIX (XENIX и AIX) могут использовать защищенный режим и, следовательно, расширенные возможности 80286. Многие новшества, введенные в этот чип, впоследствии переходили от поколения к поколению МП фирмы Intel. Имелись и определенные нерешенные проблемы, связанные с многозадачностью, повышением производительности, совершенствованием тракта процессор—память и устройства управления памятью (для эффективного функционирования ПК под управлением многозадачных ОС).

Intel 80386 (17 октября 1985 г.)

Данный МП был процессором для ЭВМ общего назначения. Размещение на кристалле 275 тыс. транзисторов дало возможность полностью реализовать 32-разрядную архитектуру.

Главные особенности:

- обеспечивает 32-разрядный ввод-вывод, 32-битовую адресацию основной памяти (адресуемая реальная память — до 4 Гбайт, т. е. $2^{32} = 4\ 294\ 967\ 296$ байт) и емкость до 64 Тбайт (64×2^{40} байт) виртуальной памяти;
- рабочая тактовая частота равнялась 33 МГц.

В МП были встроены система управления памятью и защиты (регистры преобразования адреса, механизмы защиты оперативной памяти, улучшенные аппаратные средства поддержки многозадачных ОС), средства работы с виртуальной памятью со страничной организацией памяти (в устройство менеджера памяти — MMU

(Memory Management Unit) помимо блока сегментации — SU (Segmentation Unit) был включен блок управления страницами — PU (Paging Unit)), благодаря чему относительно просто реализовывались процессы свопинга (перестановка сегментов из одного места памяти в другое).

Предварительная выборка команд, буфер для команд (внутренняя кэш-память) 16 байт, конвейер команд и механизмы выполнения функций преобразования адреса значительно уменьшили среднее время выполнения команды (3—4 млн команд в секунду, что в 6—8 раз превышало аналогичный показатель 8086). Как и раньше, новый чип был совместим со своими предшественниками на уровне объектных кодов

Наиболее существенной особенностью 80386 было использование кэш-памяти, значительно повышающей производительность системы (еще один атрибут универсальных ЭВМ, который начал применяться в МП-системах). Для управления этой памятью был разработан специальный контроллер, с помощью которого формировался двухвходовый множественный ассоциативный кэш (обеспечивал буфер емкостью до 32 Кбайт и высокий коэффициент удачных обращений). Но математический сопроцессор был еще автономным на отдельном кристалле (80387).

Реализованы три режима работы 80386: реальный, защищенный и виртуальный МП 8086. Процессор 80386 как бы включает в себя три разных процессора.

В рабочем режиме — Real Mode (реальный режим — стартовый) — он ведет себя как 8086 с расширенным набором команд и имеющий доступ к первому Мбайту памяти (при этом возможности 80386 используются не полностью, но на ней могут выполняться все программы, написанные для 8086/8088, и причем значительно быстрее).

Защищенный режим (Protected Mode) 80386 соответствует аналогичному режиму 80286, имеет доступ к 16 Мбайт памяти и расширенному набору команд, а также имеет возможность использовать систему мультипрограммирования (в основном могут выполняться несколько прикладных программ, чаще работающих в среде Windows и поддерживающих защищенный режим) Если заменить ОС на OS/2 (разработка специально для защищенного режима), то появится возможность полностью использовать функциональные возможности этого режима.

Последний, третий режим 80386 — Virtual Real Mode. В этом режиме он одновременно заменяет некоторое количество параллельно работающих 8086/8088, т. е. одновременно могут быть задействова-

ны несколько программ, которые выполняются соответствующими процессорами 8086/8088. Здесь нет ограничения 1 Мбайт на память. Ядром многозадачности является основная программа, переключающая процессор в виртуальный режим и контролирующая текущие процессы выполнения различных программ (например, система Windows)

80386 внутренне одновременно оперирует 32 битами и имеет внешний 32-битовый интерфейс, но к тому времени большинство устройств и микросхем были 16-битовыми и не могли использовать эту возможность МП. Intel повторила опыт МП 8088 (8086 — 16 бит внутренние и внешние, а 8088 — 16 бит внутренние и 8 бит внешние) и создала 80386 с 16-битовым интерфейсом (он получил название 80386 SX), который оказался меньше и дешевле. Полноразрядный 80386 получил название 80386 DX.

Intel 80386 SX (февраль 1989 г.) имел 16-разрядную шину данных и 24-разрядную адресную шину (память $2^{24} = 16\,777\,200$ байт), он считал так же быстро, как и 80386 DX, однако при обращениях к имеющимся периферийным устройствам или платам расширения, а также к оперативной памяти он ограничивался 16 разрядами. Поэтому 80386 SX примерно на 10 % медленнее 80386 DX с той же тактовой частотой Истинные преимущества процессора заключались в возможностях, связанных с многозадачностью.

Intel 80386 SL (20 МГц — октябрь 1990 г) по сравнению с 80386 SX имел большую производительность за счет: кэш-контроллера на чипе; усовершенствованного диспетчера памяти MMU; увеличенной адресной шины, позволяющей осуществлять управление памятью 32 Мбайт, аппаратных средств EMS 4.0 (Expanded Memory Specification 4.0 — спецификация дополнительной памяти), позволяющей расширение основной памяти за пределы 640 Кбайт; малого расхода электроэнергии (за счет применения КМОП-структур); оптимального режима питания (которое подается только на компоненты, которые в данный момент времени задействованы в выполнении основных функций процессора, а остальные компоненты до начала своей активной работы функционируют в «спящем» режиме). Этот процессор по ряду параметров идеально подходил для ПК типа Laptop или Notebook, работающих на батареях. Он удовлетворял требованиям программы экономии энергии Агентства защиты окружающей среды (EPA — Environmental Protection Agency).

80386 SL могли работать с диском на базе флэш-памяти (так фирма Intel называет энергонезависимую память с произвольным доступом относительно медленного действия) объемом до 16 Мбайт. В 80386 SL можно было уменьшить потребление элек-

троэнергии путем снижения частоты системной синхронизации при выполнении операций, в которых высокая частота не требуется (чем выше частота синхронизации, тем больше потребляется энергии). Например, при вводе данных с клавиатуры частоту можно снизить в 2, 4 или 8 раз. При выполнении программы возобновляется обычная работа ПК.

80386 SLC представляет собой маломощный МП с внутренней кэш-памятью на 8 Кбайт, работающий на базе системы команд 80486 SX (см. далее) с оптимизацией наиболее часто встречающихся в современных сложных приложениях команд, которые выполняются за меньшее число тактов синхронизации по сравнению с обычными 80386.

Intel 80486 (10 апреля 1989 г.)

Здесь в результате повышения степени интеграции до 1,2 млн транзисторов открылась возможность реализовать на одном кристалле не только кэш-память, но и математический сопроцессор. Для кэш-памяти использовался более эффективный четырехвходовый статический буфер, который, будучи размещенным в чипе, мог работать на тактовой частоте МП (намного быстрее, чем ОП). Здесь применялся «групповой режим» — самый скоростной режим доступа к шине, обеспечивающий быстрое заполнение кэш-памяти МП. Интегрированные в чип МП 8 Кбайт кэш-памяти, управляемой через контроллер, называется внутренней (Internal Cache), т. е. Level 1 (L1). Имеется также внешняя кэш-память (External Cache), т. е. Level 2 (L2). 80 % команд могут выполняться за один такт (применяется конвейерная обработка). Этот прибор так же, как и предыдущие МП, функционировал в трех режимах и был ориентирован на многозадачные среды. Производительность в задачах вычислительного характера возросла в 3—4 раза (за счет интеграции МП и сопроцессора).

В зависимости от режима 80486 работает с различной частотой.

Intel 80486 DX, основным отличием которого является отсутствие внутреннего математического сопроцессора, рабочие частоты — 25 и 33 МГц.

Intel 80486 SL (ноябрь 1992 г.) предназначен для laptop и портативных ПК (напряжение питания 3,3 В, имеет внутреннюю схему выключения, предусмотрена схема снижения частоты синхронизации в определенных условиях и т. д.).

МП 80486 SLC2 разработан фирмой IBM (применена схема удвоения частоты — на внешнем уровне она работает с обычной так-

товой частотой, а на внутреннем уровне — с удвоенной частотой; используется напряжение питания 3,3 В, по сравнению с 80386 SLC удвоена емкость внутреннего кэша до 16 Кбайт).

Intel Rapid CAD — модифицированный процессор 80486, который с дополнительным чипом работает в любом вычислительном устройстве, рассчитанном на ЦП 80386. Этот специальный набор микросхем делает из ПК с ЦП 80386 почти полноценный ПК 486 (быстрота, как у 80486 плюс дополнительная производительность сопроцессора).

Intel 80486 DX2 (март 1992 г.) — усовершенствованный вариант 80486 DX (он на внешнем уровне представляет собой МП с тактовой частотой 25 или 33 МГц, который однако на внутреннем уровне работает с тактовой частотой 50 или соответственно 66 МГц). Тогда команды, которые не используют передачу данных на внешнюю шину, выполняются почти в 2 раза быстрее (это на практике означает повышение производительности на 50—95 %). Эти МП обеспечили новую технологию, при которой скорость работы внутренних блоков МП в 2 раза выше скорости остальной части системы (появилась возможность объединения высокой производительности МП с внутренней тактовой частотой 50 (66) МГц и эффективной по стоимости системой на 25/33 МГц).

Intel 80486 SX2. Процессор 486SX с внутренним удвоением частоты. Тактовая частота 50 МГц.

Intel 80486 DX4 (март 1994 г.). Процессор 486DX с внутренним утроением частоты совместим с 486DX, кэш объемом 16 Кб, 1,6 млн транзисторов, технология 0,6 мкм, 75 МГц, 53 млн операций в секунду; 100 МГц, 70 млн операций в секунду.

Pentium — пятое поколение МП (22 марта 1993 г.)

Отметим существенные отличительные параметры и свойства Pentium (МП пятого поколения) (табл. 4.2).

Pentium представляет собой суперскалярный (атрибут мэйнфреймов) процессор с 32-битной адресной шиной и 64-битной шиной данных, изготовленный по субмикронной технологии с комплиментарной МОП-структурой (0,8 мкм — 1/100 толщины человеческого волоса) и состоящий из 3,1 млн транзисторов (на площади в один квадратный дюйм — 625 мм^2).

Производительность МП при тактовой частоте 66 МГц составляет около 112 млн инструкций в секунду (MIPS). Пятикратное повышение (по сравнению с 80486 DX) достигалось благодаря двум

Таблица 4.2. **Процессоры Pentium**

Тип процессора	Год выпуска	Рабочее кодовое наименование	Число транзисторов, млн	Размер платы, мм^2	Кэш L2 (на плате)	Размер минимальной структуры, мкм	Тактовая частота
Pentium	1993	P5	3,1			0,80	60/66
	1994	P54	3,2			0,50	75/90/100/120
	1995	P54	3,3			0,35	120/133
	1996	P54	3,3			0,35	150/166/200
Pentium PRO	1995	P6	5,5			0,50	150
	1995	P6	5,5			0,35	160/180/200
	1997	P6	5,5			0,35	200
MMX	1997	P55	4,5			0,28	166/200/233
	1998	P55	4,5			0,25	266
Pentium II	1997	Klamath	7,5			0,28	233/266/300
	1998	Deschutes	4,5			0,25	333/350/400
Pentium II Celeron	1998	Covington	7,5			0,25	266/300
	1998	Mendocino	19,0			0,25	300/333
	1999	Mendocino	19,0			0,25	366 до 500
	2000	Mendocino	19,0			0,25	533
Pentium III	1999	Katmai	9,5		512 Кбайт	0,25	450/500/550
	1999	Coppermine	28,1		256 Кбайт	0,18	533 до 733 МГц
	1999–2000	Itanium	30,0		2 Мбайта L3	0,18	800 МГц до 1 ГГц и более
	2000	Coppermine	28,1		256 Кбайт	0,18	850 МГц до 1 ГГц
	2001	Tualatin	44,0		256 Кбайт	0,13	1,2ГГц ДО 1,4ГГц
Pentium IV	2000	Willamette	42,0	217	256 Кбайт	0,18	1,4 до 2,0 ГГц
	2002	Northwood	55,0	146	512 Кбайт	0,13	2,0 до 3,4 ГГц
	2004	Prescott	125,0	112	1 МБайт	0,09	2,8 ГГц и более

5-ступенчатым конвейерам, позволяющим выполнить одновременно несколько инструкций.

Для постоянной загрузки этих конвейеров из кэш-памяти требуется широкая полоса пропускания. Естественно, для отмеченного случая совмещенный буфер команд и данных не подходит. Pentium имеет разделенный буфер команд и данных — двухвходовые (атрибут из арсенала RISC-процессоров). Обмен данными через кэш данных выполняется независимо от процессорного ядра, а буфер инструкций связан с ним через высокоскоростную 256-разрядную внутреннюю шину. Каждый из кэшей имеет емкость 8 Кбайт, и они допускают одновременную адресацию. Поэтому программа в одном такте синхронизации может извлечь 32 байта (256 : 8 = 32) команд и произвести два обращения к данным (32 x 2 = 64). Еще одним новым средством является буфер предсказания переходов BTB (Branch Target Buffer).

Обработка графической информации, мультимедиа-приложений и интенсивное использование ПК для решения вычислительных задач требуют высокой производительности при выполнении операций с плавающей точкой. Аппаратная реализация (вместо микропрограммной) основных арифметических операций (+, ? и /) выполняется автономными высокопроизводительными блоками, и 8-ступенчатый конвейер позволяет выдавать результаты через каждый такт.

В процессор Pentium введен (как и в 80486) режим управления системой SMM (System Management Mode). Этот режим дает возможность реализовывать системные функции очень высокого уровня, включая управление питанием или защиту, прозрачные для ОС и выполняющихся приложений.

Средствами SMM управляет микропрограмма в ROM, которая встроена в МП. В режим SMM МП может переходить из любого режима и, выполнив необходимые действия, вернуться в первоначальный режим. Такая возможность достигается за счет использования специальной области памяти, называемой RAM управления системой SMRAM (System Management Random Access Memory), которая применяется для хранения самой программы SMM и сохранения состояния регистров первоначальной программы. Запрещаются все другие прерывания, что предоставляет SMM полное управление системой. Программа SMM может перевести МП или периферийное устройство в пассивное состояние после определенного времени бездействия и активизировать их при нажатии клавиши или движении мыши. Эта программа полностью управляет вводом-выводом и осуществляет адресацию всех 4 Гбайт RAM.

Естественно, переход на тактовую частоту 60 МГц и выше был значительным достижением (преодолевался магический предел 100 MIPS), и были соответственным образом решены проблемы охлаждения (поверхность процессора при этом нагревается до 85 °С). Начав в 1993 г. с отметки в 60 МГц и быстро миновав промежуточные барьеры (66, 75, 100, 133 и 166 МГц), в 1996 г. Intel довела тактовую частоту Pentium до 200 МГц и выше.

Pentium Pro (1 ноября 1995 г.)

Степень интеграции позволяла выжать из CISC-архитектуры практически все, перенять почти все технические решения, которые ранее применялись в суперЭВМ и мэйнфреймах. Pentium Pro (шестое поколение МП) стал таким микропроцессором. Новый ЦП имеет три конвейера, каждый из которых состоит из 14 ступеней. Для постоянной загрузки имеется высокоэффективный четырехвходовый кэш команд и высококачественная система предсказания ветвлений на 512 входов. Дополнительно для повышения производительности была применена буферная память (кэш) второго уровня емкостью 256 Кбайт, расположенная в отдельном чипе и смонтированная в корпусе ЦП. Этот кэш связан с ЦП собственной синхронной 64-разрядной шиной, работающей на тактовой частоте МП. В результате стала возможной эффективная разгрузка пяти исполнительных устройств: два блока целочисленной арифметики; блок загрузки; блок записи; FPU (Floating-Point Unit — устройство арифметических операций с плавающей точкой).

Архитектура Pentium Pro позволяет соединять между собой множество процессоров, создавая таким образом гибкую масштабируемость, реализовав преимущества SMP (Symmetric Multi-Processing — симметричная мультипроцессорная система) новейших ОС.

Некоторые команды предназначены для повышения быстродействия (например, команда умножения требует 20 тактов, но если нет команды, то нужна соответствующая программа, которая выполнялась бы в течение сотен тактов). Однако такие команды выполняются не очень часто (зависит от задач). Необходимо отметить, что каждая команда при своем выполнении требует исполнения определенного количества микрокоманд, и если уменьшить их количество, то можно повысить быстродействие. Но это возможно при разработке новых ЦП путем оптимизации микрокода.

Одной из причин, снижающих быстродействие, является «состояние ожидания» ЦП. Состояние ожидания происходит в тех случаях, когда цикл доступа к внешнему устройству или памяти превы-

шает стандартное значение. Время регенерации динамической памяти и обращения к ней (конфликтная ситуация) может порождать несколько состояний ожидания.

На производительность влияют также задержки, вносимые дисководами, соответствующим типом принтера и используемым видеоинтерфейсом.

Pentium P55 (Pentium MMX), 8 января 1997 г.

Pentium MMX — версия Pentium с дополнительными возможностями, технология должна была добавить/расширить мультимедиа возможности компьютеров.

Реализована методика SIMD (ОКМД), ориентированная на алгоритмы и типы данных, характерные для программного обеспечения мультимедиа. Физически данные мультимедиа упаковываются в одно 64-разрядное слово и над ним производится некое общее действие.

MMX объявлен в январе 1997 г., тактовая частота 166/200 МГц, в июне того же года появилась версия на 233 МГц. Технологический процесс 0,35 мкм (350 нм), 4,5 млн транзисторов.

Pentium II (7 мая 1997 г.)

Процессор представляет собой модификацию Pentium Pro с поддержкой возможностей MMX. Первые Pentium II объявлены как процессоры для настольных high-end компьютеров. Была изменена конструкция корпуса: пластину с контактами (разъем Socket 7) заменили на картридж (разъем Slot 1), увеличена частота шины и тактовая частота, расширены MMX инструкции. Первые модели 233/300 МГц производились по технологии 0,35 мкм, следующие — 0,25 мкм. Модели с частотой 333 МГц выпущены в январе 1998 г. и содержали 7,5 млн транзисторов. В апреле того же года появились версии 350 и 400 МГц, а августе — на 450 МГц. Все Pentium II имеют кэш второго уровня объемом 512 Кбайт. Известна также модель для ноутбуков — Pentium II PE и для рабочих станций Pentium II Хеоп (450 МГц).

Celeron (15 апреля 1998 г.)

Celeron — упрощенный вариант Pentium II для дешевых компьютеров. Основные отличия этих процессоров — в объеме кэша второго уровня и частоте шины. Первые, выпущенные в апреле и июне

1998 г., процессоры Celeron на 266 МГц и 300 МГц не имели кэша вообще при частоте шины 66 МГц и выполнены в конструктиве Slot 1. Следущие модели имеют 128 Кбайт кэша и выпускаются как для Slot 1, так и для Socket370 (PPGA), в их названии присутствует буква А (например, Celeron 333A) Тактовая частота — 266, 300, 333, 366, 400, 433, 466, 500, 533 МГц. Все эти процессоры выполнены по технологии 0,25 мкм и имеют от 7,5 до 19 млн транзисторов.

Pentium III (26 февраля 1999 г.)

РЗ — один из самых мощных и производительных процессоров Intel, но в своей конструкции он мало чем отличается от Pentium II — увеличена частота и добавлено около 70 новых инструкций. Первые модели объявлены в феврале 1999 г., тактовая частота 450, 500, 550 и 600 МГц. Частота системной шины 100 МГц, емкость кэша второго (L2) уровня — 512 Кб, технологический процесс 0,25 мкм, 9,5 млн транзисторов. В октябре 1999 г. также выпущена версия для мобильных компьютеров, выполненная по технологии 0,18 мкм с частотой 400, 450, 500 МГц, а также модели с частотой 533, 550, 600, 650, 700 и 733 МГц по технологии 0,18 мкм.

Для рабочих станций и серверов разработан Pentium III (РЗ) Хеоп, ориентированный на системную логику GX с 512 Кбайт, 1 Мбайт или 2 Мбайт кэша второго уровня (L2). Технологический процесс 0,25 мкм, системная шина работает на частоте 100 МГц, есть версия 0,18 мкм с частотой шины 133 МГц. Известны также модели на 600, 666 и 733 МГц.

Intel Pentium IV Prescott (февраль 2004 г.)

В начале февраля Intel анонсировала четыре новых процессора Pentium IV 2,8; 3,0; 3,2 и 3,4 ГГц, основанных на ядре Prescott , которое включает ряд нововведений (рис. 4.14).

Новые процессоры имеют точно такую же конструкцию, как и процессоры, основанные на ядре Northwood, поэтому для их отличия Intel ввела новый индекс в названии процессора — Е. Например, процессор Pentium IV 3,2 С основан на ядре Northwood, имеет поддержку шины 800 МГц и технологии НТ (Hyper Threading), в то время как Pentium IV 3,2 Е выполнен на ядре Prescott, и также поддерживает шину 800 МГц и технологию НТ.

Вместе с выпуском четырех новых процессоров Intel представила процессор Pentium IV 3,4 EE (Extreme Edition), основанный на

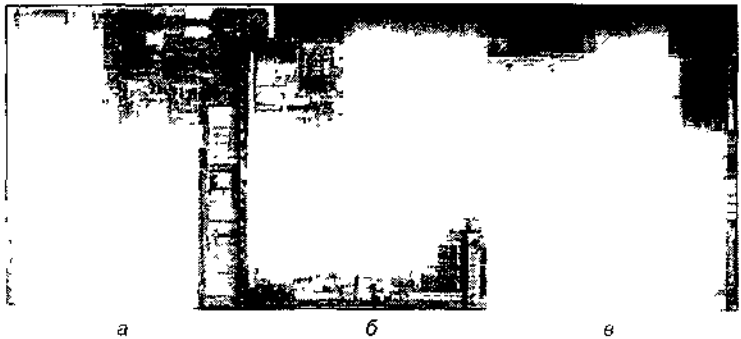


Рис. 4.14. Процессоры Pentium IV
а — Willamette (0,18 мкм), *б* — Northwood (0,13 мкм), *в* — Prescott (0,09 мкм)

ядре Northwood и имеющий кэш-память третьего уровня в 2 Мбайт, а также упрощенную версию Pentium IV 2,8A, основанную на ядре Prescott с ограниченной частотой шины (533 МГц) и отсутствием поддержки технологии HT.

Pentium IV Prescott выполнен по технологии 90 нм, что позволило уменьшить площадь кристалла, при этом общее число транзисторов было увеличено более чем в два раза. В то время как ядро Northwood имеет площадь 145 мм^2 и на нем размещено 55 млн транзисторов, ядро Prescott имеет площадь 122 мм^2, при этом на нем расположено 125 млн транзисторов.

Intel представила в новом Prescott новую технологию SSE 3, которая включает 13 новых потоковых инструкций, которые увеличат производительность некоторых операций, как только программы начнут использовать их. SSE 3 является не просто расширением SSE 2 — эта технология не только добавляет новые инструкции, она позволяет облегчить и автоматизировать процесс оптимизации готовых приложений средствами компилятора. Иначе говоря, разработчику программного обеспечения не надо будет переписывать код программы, необходимо будет только перекомпилировать ее.

Одним из важнейших, с точки зрения производительности, дополнений можно считать увеличенный до 1 Мбайт кэш второго уровня Объем кэш памяти первого уровня также был увеличен до 16 Кбайт.

Prescott имеет улучшенный механизм предвыборки данных (опережающая загрузка), и, кроме того, реализует улучшенную версию технологии HyperThreading. В новую версию включено множество особенностей, способных оптимизировать многопоточное выполнение различных операций. Единственный недостаток новой

версии заключается в необходимости перекомпиляции программного обеспечения и обновления операционной системы.

Для увеличения рабочей частоты будущих процессоров, ядро Prescott имеет увеличенную с 20 до 31 ступени длину конвейера Увеличение длины конвейера негативно сказывается на производительности в случае неправильного предсказания ветвлений. Проще говоря, если предсказание будет выполнено неправильно, то «сбрасывается» весь колвейер, и все операции будут повторены снова. Для компенсации увеличения длины конвейера была улучшена технология предсказания ветвлений.

4.3. Режимы процессора. Система команд реального режима процессоров i80x86. Интерпретация в терминах Ассемблера (MASM)

Все 32-разрядные процессоры Intel (и совместимые с ними) начиная с 80386-го могут выполнять программы в нескольких режимах. Режимы процессора предназначены для выполнения программ в различных средах; в разных режимах возможности МП неодинаковы, потому что команды выполняются по-разному.

Режимы процессора

В зависимости от режима процессора изменяется схема управления памятью системы и задачами. Процессоры могут работать в трех режимах:

- реальном;
- защищенном;
- виртуальном реальном режиме (реальном внутри защищенного).

Реальный режим. В первоначальном IBM PC использовался процессор i8086, который мог выполнять 16-разрядные команды, применяя 16-разрядные внутренние регистры, и адресовать только I Мбайт (2^{20} байт) памяти, используя 20 разрядов для адреса. Все программное обеспечение PC первоначально было предназначено для этого процессора; оно было разработано на основе 16-разрядной системы команд и модели памяти объемом 1 Мбайт. Например, DOS, все программное обеспечение DOS, Windows от 1.*x* до 3.x и все приложения для Windows от 1.*x* до 3.x написаны в расчете

на 16-разрядные команды. Эти 16-разрядные операционные системы и приложения были разработаны для выполнения на процессоре i8086.

Более поздние процессоры, например i80286, могли также выполнять те же самые 16-разрядные команды, что и первоначальный i8086, но намного быстрее. Другими словами, процессор i80286 был полностью совместим с первоначальным i8086. Шестнадцатиразрядный режим, в котором выполнялись команды процессоров i8086 и i80286, был назван *реальным режимом.* Все программы, выполняющиеся в реальном режиме, должны использовать только 16-разрядные команды, 20-разрядные адреса и поддерживаться архитектурой памяти, рассчитанной на емкость до 1 Мбайт.

Для программного обеспечения этого типа обычно используется однозадачный режим, т. е. одновременно может выполняться только одна программа. Нет никакой встроенной защиты для предотвращения перезаписи ячеек памяти одной программы или даже операционной системы другой программой; это означает, что при выполнении нескольких программ вполне могут быть испорчены данные или код одной из них, а это может привести всю систему к краху (или останову).

Защищенный режим. Несмотря на то, что процессор i80286, как и i8086, является 16-разрядным, он (в отличие от последнего) может работать в новом — защищенном — режиме и имеет аппаратурную поддержку многозадачных операционных систем, значительно ускоряющую и упрощающую процесс переключения задач. Эта поддержка активно используется всеми мультизадачными операционными системами и оболочками, разработанными для компьютера IBM PC.

Адресная шина i80286 была увеличена с 20 до 24 разрядов, что привело к расширению адресного пространства с 1 до 16 Мбайт (2^{24} байт). Новый метод адресации памяти позволил изолировать адресные пространства отдельных задач друг от друга. При этом прикладная программа, работающая в среде операционной системы, использующей защищенный режим, не может случайно или намеренно разрушить целостность самой операционной системы.

В защищенном режиме программа может записывать данные только в те области памяти, которые выделены ей операционной системой. Это повышает надежность работы мультизадачных и, в частности, мультипользовательских операционных систем. В последнем случае изолирование адресных пространств задач, принадлежащих отдельным пользователям, в хорошо спроектированной мультипользовательской операционной системе полностью исклю-

чает такую ситуацию, когда после запуска одним пользователем недостаточно отлаженной программы приходится перезапускать всю систему.

Следующие модели процессоров фирмы Intel — i80386, i80486 и i80586 (Pentium) были 32-разрядными. Помимо расширения адресного пространства до величины в 4 Гбайта (2^{32} байт) в них реализована концепция страничной виртуальной памяти, возможной только в защищенном режиме.

Механизм страничной виртуальной памяти позволяет разместить часть оперативной памяти на диске. При этом размер виртуальной памяти, предоставляемой программам, ограничивается размером свободного пространства на диске.

Перечислим кратко основные преимущества, которые получает программа, работающая в защищенном режиме процессора:

- возможность непосредственной адресации памяти за пределами первого мегабайта;
- для процессоров i80x86 реализован механизм страничной виртуальной памяти, позволяющий программам работать с памятью, размер которой может быть много больше физической оперативной памяти, установленной в компьютере;
- аппаратная поддержка мультизадачности позволяет создавать на основе процессоров, работающих в защищенном режиме, высокопроизводительные мультизадачные и мультипользовательские системы.

Виртуальный реальный режим. Помимо страничной виртуальной памяти в процессорах i80386 и более поздних реализован так называемый режим виртуального процессора i8086 или просто виртуальный режим. Этот режим реализуется в рамках защищенного режима (процессор может переключиться в виртуальный режим только из защищенного режима). В виртуальном режиме процессор способен выполнять программы, составленные для процессора i8086, находясь в защищенном режиме и используя аппаратные средства защищенного режима: мультизадачность, изолирование адресных пространств отдельных задач друг от друга, страничная виртуальная память.

Рассмотрим основные принципы функционирования процессоров i80x86 *в реальном режиме.*

Реальный режим процессоров 80x86

Это режим генерирования адресов, используемый процессором 8086. В этом режиме не может быть использована виртуальная память. Можно адресовать лишь до 1 Мбайт (2^{20} байт) оперативной

памяти, так как у процессора 8086 20-разрядная шина адреса. Так как все регистры процессора 8086 являются 16-разрядными, для представления 20-разрядного физического адреса памяти используется содержимое нескольких 16-разрядных регистров.

Оперативную память при работе в этом режиме можно разбить на логические блоки по 64 Кбайт, называемые сегментами, причем каждый сегмент может начинаться с адреса, кратного 16 байт. Таким образом, первый сегмент имеет начальный адрес 0, второй находится по адресу 16 (или 10 в шестнадцатеричной системе) и т. д. Несколько близко расположенных сегментов могут перекрываться. Это удобно при организации совместного доступа к командам и данным разными программами. Доступ к каждой ячейке в памяти происходит путем указания значения регистра сегмента (см. далее), определяющего блок размером 64 Кбайт, и положения, или смещения, этого адреса внутри этого блока.

Микропроцессор использует четыре регистра сегмента, при этом каждый регистр имеет размер, равный одному слову (16 бит):

- регистр сегмента команд cs (Code Segment), указывающий на сегмент, содержащий текущую исполняемую программу;
- регистр сегмента данных DS (Data Segment), указывающий на данные;
- регистр дополнительного сегмента ES (Extra Segment), указывающий на дополнительные данные;
- регистр сегмента стека SS (stack Segment), указывающий на стек.

У процессора 80386 и старше есть еще два сегментных регистра — FS и GS.

Содержимое каждого из этих регистров однозначно связано с местом в памяти соответствующего сегмента. Его адрес получается приписыванием справа четырех двоичных нулей к значению сегмента, что соответствует умножению на 16 (или на 10 по основанию 16). Полученное 20-битовое значение представляет собой адрес начала (или базовый адрес) сегмента в физической памяти. Для определения реального адреса команды или данных процессор добавляет затем значение смещения к базовому адресу.

Например, команда, подлежащая исполнению процессором в каждый данный момент времени, определяется из значений двух регистров: регистра cs, значение которого, будучи умножено на 16, дает адрес начала сегмента команд, и регистра указателя команд IP (instruction Pointer), указывающего положение соответствующей команды относительно начала сегмента команд.

В реальном режиме не существует никакого механизма защиты, так что любая программа может обратиться к произвольной ячейке памяти в пределах 1 Мбайт, включая область экрана или область расположения операционной системы.

Выделим два основных недостатка схемы адресации памяти реального режима:

- ограниченное адресное пространство (до 1 Мбайта и еще примерно 64 Кбайта старшей области памяти для процессоров 80286 и старше);
- свободный доступ любых программ к любым областям данных, что представляет потенциальную опасность для целостности операционной системы.

Описание форматов команд, данных, структуры памяти и процессора (см. ниже) производится с использованием ассемблерных представлений. Поэтому два слова скажем о программировании на ассемблере 8086 (MASM).

Формат команд ассемблера. Текст исходной программы состоит из операторов ассемблера, каждый из которых занимает отдельную строку этого текста. Различают два типа операторов: инструкции и директивы. Первые при трансляции преобразуются в команды процессора, которые исполняются после загрузки в память загрузочного модуля программы, имеющего расширение `.com` или `.exe`. Операторы второго типа управляют процессом ассемблирования — преобразования текста исходной программы в коды объектного модуля (расширение `.obj`). Ассемблер интерпретирует и обрабатывает операторы один за другим, генерируя последовательность из команд процессора и байтов данных.

Общий формат оператора ассемблера имеет следующий вид:

```
[Метка:]Код_операции[Операнд1[,Операнд2]][;Комментарий],
```

где элементы, указанные в квадратных скобках, могут отсутствовать.

Пробелы вводятся произвольно, но минимум один пробел должен быть после кода операции.

Метка — это идентификатор, присваиваемый первому байту того оператора, в котором она появляется.

Код_операции — это мнемоническое обозначение соответствующих команд процессора.

Операнды оператора ассемблера описываются выражениями. Выражения конструируются на основе операций над числовыми и

текстовыми константами, метками и идентификаторами переменных с использованием знаков операций и некоторых зарезервированныхслов.

Ниже приведены все определенные в ассемблере операции.

Порядок старшинства операций от высшей к низшей:

```
LENGTH, SIZE, WIDTH, MASK, (), [], ○
.PTR, OFFSET, SEG, TYPE, THIS
HIGH, LOW
+ (унарная), - (унарная)
*, /, MOD, SHL, SHR
+, -
EQ, NE, LT, LE, GT, GE
NOT
AND
OR, XOR
SHORT, .TYPE
```

Старшинство операций определяет порядок, по которому будет вычисляться выражение. Более старшие операции будут производиться раньше операций, имеющих меньшее старшинство.

Примечания

Операции, стоящие в одной строке, имеют равный приоритет. Операции равногостаршинствавычисляютсяслеванаправо.

Операции, стоящие в скобках, выполняются первыми.

Пример оператора ассемблера:

```
l0c_1: mov ax, (DAT_l+4) SHR 4.
```

Здесь использованы следующие операции ассемблера: ( ), + и SHR.

Вот примеры некоторых арифметических операторов:

«+» Сложение (бинарное) или унарный плюс

```
expressıon1 + expressıon2 (сложение)
```

или

```
+ expression (унарный плюс)
```

Бинарный «+» суммирует значения двух выражений, унарный — сохраняет знак и значение выражения.

Примечания

Оператор сложения («+») может использоваться для прибавления целого числа к операнду, перемещаемому в памяти. Операндом, перемешаемым в памяти, может быть только одно из выражений Оба выражения могут быть целыми числами

Унарная операция «+» обладает более высоким приоритетом, чем оператор сложения

«-» Вычитание (бинарное) или унарный минус

`expression1 - expression2` (вычитание)

или

`- expression` (унарный минус) .

Бинарный «–» вычитает одно выражение из другого, унарный — изменяет знак выражения.

Примечания.

Операндами оператора вычитания могут быть целые числа или операнды, перемещаемые в памяти. Если оба операнда являются адресами памяти, то они должны располагаться в одном и том же сегменте.

Унарная операция «–» обладает более высоким приоритетом, чем оператор вычитания.

«*» Умножение

`expression1 * expression2.`

Перемножает значения двух выражений.

Примечания.

Выражения должны быть целыми числами, и они не могут быть адресами, перемещаемыми в памяти.

«/» Деление

`expression1 / expression2`

Делит одно выражение на другое.

Примечания.

Выражения должны быть целыми числами, они не могут быть адресами, перемещаемыми в памяти.

«MOD» Деление по модулю

выражение1 MOD *выражение2*

Выдает остаток от деления.

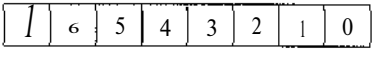
Примечания.

Оба выражения должны быть целыми числами, они не должны быть настраиваемыми адресами.

Например, `14 MOD 4` = 2, так как 14/4 дает остаток 2

Оперативная память. Объем оперативной памяти I80X86 (здесь — 8086) — 2^{20} байт (1 Мбайт). Байты нумеруются, начиная с 0, номер байта называется его *адресом.* Для ссылок на байты памяти используются 20-разрядные адреса: от `00000` до FFFFF (в шестнадцатеричной системе).

Байт содержит 8 разрядов (битов), каждый из которых может принимать значение 1 или 0. Разряды нумеруются справа налево от 0 до 7:



Байт — наименьшая адресуемая ячейка памяти. В I80X86 используются и более крупные ячейки — слова и двойные слова. *Слово* — это два соседних байта, размер слова — 16 бит (они нумеруются справа налево от 0 до 15). Адресом слова считается адрес его первого байта (с меньшим адресом); этот адрес может быть четным и нечетным. *Двойное слово* — это любые четыре соседних байта (два соседних слова), размер такой ячейки — 32 бита; адресом двойного слова считается адрес его первого байта.

Байты используются для хранения небольших целых чисел и символов, слова — для хранения целых чисел и адресов, двойные слова — для хранения «длинных» целых чисел и так называемых адресных пар `(сегмент:смещение)`.

Регистры. Помимо ячеек оперативной памяти для хранения данных (правда, кратковременного) можно использовать и регистры — ячейки, входящие в состав процессора и доступные из машинной программы. Доступ к регистрам осуществляется значительно быстрее, чем к ячейкам памяти, поэтому использование регистров заметно уменьшает время выполнения программ.

Все регистры имеют размер слова (16 битов), за каждым из них закреплено определенное имя (AX, SP и т. п.). По назначению и способу использования регистры можно разбить на следующие группы:

- регистры общего назначения (AX, BX, `CX`, DX, BP, `SI`, `DI`, SP);
- сегментные регистры (CS, DS, SS, `ES`);
- счетчик команд (IP);
- регистр флагов (Flags).

Расшифровка этих названий:

Аббревиатура	Регистр	Перевод
A	accumulator	Аккумулятор
B	base	База
C	counter	Счетчик
D	data	Данные
BP	base pointer	Указатель базы
S!	source index	Индекс источника
DI	destination index	Индекс приемника
SP	stack pointer	Указатель стека
CS	code segment	Сегмент команд
DS	data segment	Сегмент данных
SS	stack segment	Сегмент стека
ES	extra segment	Дополнительный сегмент
IP	instruction pointer	Счетчик команд

Регистры общего назначения можно использовать во всех арифметических и логических командах. В то же время каждый их них имеет определенную специализацию (некоторые команды «работают» только с определенными регистрами). Например, команды умножения и деления требуют, чтобы один из операндов находился в регистре AX или в регистрах AX и DX (в зависимости от размера операнда), а команды управления циклом используют регистр cx в качестве счетчика цикла. Регистры вх и BP очень часто используются как базовые регистры, а SI и DI — как индексные. Регистр SP обычно указывает на вершину стека, аппаратно поддерживаемого в I80X86.

Регистры AX, вх, cx и DX конструктивно устроены так, что возможен независимый доступ к их старшей и младшей половинам; можно сказать, что каждый из этих регистров состоит из двух байтовых регистров, обозначаемых AH, AL, вн и т. д. (н — high, старший; L — low, младший):

Регистр	Аббревиатура	Старший полурегистр	Младший полурегистр
Аккумулятор	AX	AH	AL
База	BX	BH	BL
Счетчик	CX	CH	CL
Данные	DX	DH	DL
Разряды	15-0	15-8	7-0

Таким образом, с каждым из этих регистров можно работать как с единым целым, а можно работать и с его «половинками». Например, можно записать слово в AX, а затем считать только часть слова из регистра AH или заменить только часть в регистре AL и т. д. Такое устройство регистров позволяет использовать их для работы как с числами, так и с символами.

Все остальные регистры не делятся на «половинки», поэтому считать или записать их содержимое (16 битов) можно только целиком.

Сегментные регистры CS, DS, SS и ES не могут быть операндами никаких команд, кроме команд пересылки и стековых команд. Эти регистры используются только для сегментирования адресов.

Счетчик команд IP всегда содержит адрес (смешение от начала программы) той команды, которая должна быть выполнена следующей (начало программы хранится в регистре CS). Содержимое регистра Ip можно изменить только командами перехода.

Флаги. Имеется особый регистр флагов. Флаг — это бит, принимающий значение «1» (флаг установлен), если выполнено некоторое условие, или «О» (флаг сброшен) в противном случае (табл. 4.3). В I80X86 используется 9 флагов, каждому из них присвоено определенное имя (ZF, CF и т. д.). Все они собраны в регистре флагов (каждый флаг — это один из разрядов регистра, часть его разрядов не используется):

Регистр флагов																
Флаги	x	x	x	x	OF	DF	IF	TF	SF	ZF	x	AF	x	PF	x	CF
Разряды	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Некоторые флаги являются флагами условий; они автоматически меняются при выполнении команд и фиксируют те или иные свойства их результата (например, равен ли он нулю).

Другие флаги называются флагами состояний; они меняются из программы и оказывают влияние на дальнейшее поведение процессора (например, блокируют прерывания).

Представление чисел. Здесь рассматривается машинное представление целых чисел, строк и адресов. Представление двоично-десятичных чисел, используемых достаточно редко, не рассматривается. Что касается вещественных чисел, то в I80X86 нет команд вещественной арифметики (операции над этими числами реализуются программным путем или выполняются сопроцессором) и потому

Таблица 4.3 **Содержание регистра флагов**

Флаги условий			
CF	carry flag	Флаг переноса	Принимает значение 1, если при сложении целых чисел появилась единица переноса, не укладывающаяся в разрядную сетку, или если при вычитании чисел без знака первое из них было меньше второго В командах сдвига в CF заносится бит, вышедший за разрядную сетку CF фиксирует также особенности команды умножения
OF	overflow flag	Флаг переполнения	Устанавливается в 1, если при сложении или вычитании целых чисел со знаком получился результат, по модулю превосходящий допустимую величину (произошло переполнение мантиссы и она «залезла» в знаковый разряд)
ZF	zero flag	Флаг нуля	Устанавливается в 1, если результат команды оказался равным 0
SF	sign flag	Флаг знака	Устанавливается в 1, если в операции над числами со знаками получился отрицательный результат
PF	parity flag	Флаг четности	Равен 1, если результат очередной команды содержит четное количество двоичных единиц Учитывается обычно только при операциях ввода-вывода
AF	auxiliary carry flag	Флаг дополнительного переноса	Фиксирует особенности выполнения операций над двоично десятичными числами
Флаги состояний			
DF	direction flag	Флаг направления	Устанавливает направление просмотра строк в строковых командах при DF = 0 строки просматриваются «вперед» (от начала к концу), при DF = 1 - в обратном направлении
IF	interrupt (lag	Флаг прерываний	При IF = 0 процессор перестает реагировать на поступающие к нему прерывания, при IF = 1 блокировка прерываний снимается
TF	trap flag	Флаг трассировки	При TF = 1 после выполнения каждой команды процессор делает прерывание (с номером 1), чем можно воспользоваться при отладке программы для ее трассировки

нет стандартного представления вещественных чисел. Кроме того, рассматриваются некоторые особенности выполнения арифметических операций.

Шестнадцатеричные числа записываются с буквой h на конце, двоичные числа — с буквой b (так принято в MASM).

Представление целых чисел. В общем случае под целое число можно отвести любое число байтов, однако система команд I80X86 поддерживает только числа размером в байт и слово и частично поддерживает числа размером в двойное слово. Именно эти форматы и будут рассмотрены.

В I80X86 делается различие между целыми числами без знака (неотрицательными) и со знаком. Это объясняется тем, что в ячейках одного и того же размера можно представить больший диапазон чисел без знаков, чем чисел со знаком, и если известно заранее, что некоторая числовая величина является неотрицательной, то выгоднее рассматривать ее как величину без знака, чем как величину со знаком

Целые числа без знака Эти числа могут быть представлены в виде байта, слова или двойного слова — в зависимости от их размера. В виде байта представляются целые от 0 до 255 (2^8 - 1), в виде слова — целые от 0 до 65 535 (2^{16} - 1), в виде двойного слова — целые от 0 до 4 294 967 295 (2^{32} - 1) Числа записываются в двоичной системе счисления, занимая все разряды ячейки.

Например, число 130 записывается в виде байта `10000010b` (82h).

Числа размером в слово хранятся в памяти в «перевернутом» виде: младшие (правые) 8 битов числа размещаются в первом байте слова, а старшие 8 битов — во втором байте (в шестнадцатеричной системе: две правые цифры — в первом байте, две левые цифры — во втором байте). Например, число 130 (0082h) в виде слова хранится в памяти так:

82	00

Однако в регистрах числа хранятся в нормальном виде:

AX	00	82
	AH	AL

Перевернутое представление используется и при хранении в памяти целых чисел размером в двойное слово: в первом его байте размещаются младшие 8 битов числа, во втором байте — предыдущие 8 битов и т. д. Например, число `12345678h` хранится в памяти так:

78	56	34	12

Другими словами, в первом слове двойного слова размещаются младшие (правые) 16 битов числа, а во втором слове — старшие 16 битов, причем в каждом из этих двух слов в свою очередь используется «перевернутое» представление.

Такое необычное представление чисел объясняется тем, что в первых моделях I80X86 за один такт можно было считать из памяти только один байт и что все арифметические операции над многозначными числами начинаются с действий над младшими цифрами, поэтому из памяти в первую очередь надо считывать младшие цифры, если сразу нельзя считать все цифры. Учитывая это, в первых I80X86 стали размешать младшие цифры числа перед старшими цифрами, а ради преемственности такое представление чисел сохранили в последующих моделях 180X86.

Конечно, «перевернутое» представление неудобно для людей, однако при использовании языка ассемблера это неудобство не чувствуется: в MASM все числа записываются в нормальном, неперевернутом виде.

Целые числа со знаком. Эти числа также представляются в виде байта, слова и двойного слова. В виде байта записываются числа от -128 до 127, в виде слова — числа от -32 768 до 32 767, а в виде двойного слова — числа от -2 147 483 648 до 2 147 483 647. При этом числа записываются в дополнительном коде: неотрицательное число записывается так же, как и число без знака (т. е. в прямом коде), а отрицательное число $-x$ $(x > 0)$ представляется как число без знака $2^8 - x$ (для байтов), $2^{16} - x$ (для слов) или $2^{32} - x$ (для двойных слов).

Например, дополнительным кодом числа -6 является байт FAh (256 - 6), слово FFFAh или двойное слово FFFFFFFAh. При этом байт `10000000b` (80h) трактуется как -128, а не как +128 (слово 8000h понимается как -32 678), поэтому левый бит дополнительного кода всегда играет роль знакового: для неотрицательных чисел он равен «О», для отрицательных — «1».

Числа со знаком размером в слово и двойное слово записываются в памяти в «перевернутом» виде (при этом знаковый бит оказывается в последнем байте ячейки). Но в MASM эти числа, как и беззнаковые, записываются в нормальной форме.

Иногда число-байт необходимо расширить до слова, т. е. получить такое же по величине число, но размером в слово. Существует два способа такого расширения — без знака и со знаком. В любом случае исходное число-байт попадает во второй (до «переворачивания») байт слова, а вот первый байт заполняется по-разному: при расширении без знака в него записываются нулевые биты (`12h` превращается в `0012h`), а при расширении со знаком в первый байт записываются нули, если число-байт было неотрицательным, и записывается восемь двоичных единиц в противном случае (`81h` перехо-

дит в FF81h). Другими словами, при расширении со знаком в первом байте слова копируется знаковый разряд числа-байта.

Аналогично происходит расширение числа-слова до двойного слова.

Арифметические операции. В I80X86 имеются команды сложения и вычитания целых чисел размером в слово и байт. Специальных команд для сложения и вычитания двойных слов нет, эти операции реализуются через команды сложения и вычитания слов. Сложение и вычитание беззнаковых чисел производится по модулю 2^8 для байтов и 2^{16} для слов. Это означает, что если в результате сложения появилась единица переноса, не вмещающаяся в разрядную сетку, то она отбрасывается. Например, при сложении байтов 128 и 130 получается число 258 (100000010b), поэтому левая двоичная единица отбрасывается и остается число 2 (10Ь), которое является результатом сложения.

Ошибка здесь не фиксируется, но в флаг переноса CF заносится «1» (если переноса не было, в CF заносится «0»). Установить такое искажение суммы можно только последующим анализом флага CF.

Искажение результата происходит и при вычитании из меньшего числа большего. И здесь не фиксируется ошибка, однако первому числу дается «заем единицы» (в случае байтов это число увеличивается на 256, для слов — на 2^{16}), после чего и производится вычитание. Например, вычитание байтов 2 и 3 сводится к вычитанию чисел 256 + 2 = 258 и 3, в результате чего получается неправильная разность — 255 (а не -1). Для того чтобы можно было обнаружить такую ситуацию, флаг переноса CF переключается на «1» (если «заема» не было, в CF записывается «0»).

Сложение и вычитание целых чисел со знаком производится по тем же алгоритмам, что и для чисел без знака (в этом одно из достоинств дополнительного кода): числа со знаком рассматриваются как соответствующие числа без знака, производится операция над этими беззнаковыми числами и полученный результат интерпретируется как число со знаком. Например, сложение байтовых чисел 1 и -2 происходит так: берутся их дополнительные коды 1 и (256 - 2) = 254, вычисляется сумма этих величин 1 + 254 = 255 и она трактуется как число со знаком -1 (255 = 256 - 1).

Если при таком сложении возникла единица переноса, то она, как обычно, отбрасывается, а флаг CF получает значение «1». Однако в данном случае это отсечение не представляет интереса — результат операции будет правильным, например: 3 + (-2) = = 3 + 254(mod 256) = 257(mod 256) = 1. Зато здесь возможна иная

неприятность: модуль суммы (ее мантисса) может превзойти допустимую границу и «залезть» в знаковый разряд, испортив его. Например, при сложении байтовых чисел 127 и 2 получается величина 129 (10000100b), представляющая дополнительный код числа -127 (256 – 129).

Хотя результат здесь получился и неправильным, процессор не фиксирует ошибку, но зато заносит «1» в флаг переполнения OF (если «переполнения мантиссы» не было, в OF записывается «0»). Анализируя затем этот флаг, можно «поймать» такую ошибку.

Таким образом, сложение (вычитание) знаковых и чисел без знака производится по одному и тому же алгоритму. При этом I80X86 «не знает», какие числа (со знаком или без) он складывает; в любом случае он складывает их как числа без знака и в любом случае формирует флаги CF и OF. Как интерпретировать слагаемые и сумму, на какой из этих флагов обращать внимание — личное дело автора программы.

Что касается умножения и деления знаковых и чисел без знака, то они выполняются по разным алгоритмам, разными машинными командами. Однако и у этих операций есть ряд особенностей. При умножении байтов (слов) первый сомножитель обязан находиться в регистре AL (AX), результатом же умножения является слово (двойное слово), которое заносится в регистр AX (регистры DX и AX). Тем самым при умножении сохраняются все цифры произведения. При делении байтов (слов) первый операнд (делимое) должен быть словом (двойным словом) и обязан находиться в регистре AX (регистрах DX и AX). Результатом деления являются две величины размером в байт (слово) — неполное частное (`div`) и остаток от деления (mod); неполное частное записывается в регистр `AL` (AX), а остаток — в регистр AH (DX).

Представление символов и строк. На символ отводится один байт памяти, в который записывается код символа — целое от 0 до 255. В I80X86 используется система кодировки ASCII (American Standard Code for Information Interchange). Она не предусматривает кодов русских букв (кириллицы), поэтому в РФ применяется некоторый вариант этой системы с кириллицей (обычно это альтернативная кодировка ГОСТа).

Некоторые особенности этих систем кодировки:

- код пробела меньше кода любой буквы, цифры и вообще любого графически представимого символа;
- коды цифр упорядочены по величине цифр и не содержат пропусков, т. е. из неравенства код(«0») < код(*C*) < код(«9») следует, что *C* — цифра;

- коды прописных латинских букв упорядочены согласно алфавиту и не содержат пропусков; аналогично со строчными латинскими буквами;
- (в альтернативной кодировке ГОСТа) коды кириллицы (как прописных, так и строчных) упорядочены согласно алфавиту, но между ними есть коды других символов

Строка (последовательность символов) размещается в соседних байтах памяти (в неперевернутом виде): код первого символа строки записывается в первом байте, код второго символа — во втором байте и т. п. Адресом строки считается адрес ее первого байта.

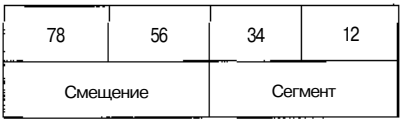
В I80X86 строкой считается также и последовательность слов (обычно это последовательность целых чисел). Элементы таких строк располагаются в последовательных ячейках памяти, но каждый элемент представлен в «перевернутом» виде.

Представление адресов. Адрес — это порядковый номер ячейки памяти, т. е. неотрицательное целое число, поэтому в общем случае адреса представляются так же, как и числа без знака. Однако в I80X86 есть ряд особенностей в представлении адресов.

Дело в том, что в I80X86 термином «адрес» обозначают разные понятия. Часто под адресом понимается 16-битовое смещение *(offset)* — адрес ячейки, отсчитанный от начала сегмента (области) памяти, которому принадлежит эта ячейка. В этом случае под адрес отводится слово памяти, причем адрес записывается в «перевернутом» виде (как и числа-слова вообще).

В другом случае под «адресом» понимается 20-битовый абсолютный адрес некоторой ячейки памяти. В силу ряда причин в I80X86 такой адрес задается не как 20-битовое число, а как пара «сегмент:смещение», где «сегмент» (*segment*) — это первые 16 битов начального адреса сегмента памяти, которому принадлежит ячейка, а «смещение» *(offset)* — 16-битовый адрес этой ячейки, отсчитанный от начала данного сегмента памяти (величина 16 х сегмент + смещение дает абсолютный адрес ячейки).

Такая пара записывается в виде двойного слова, причем (как и для чисел) в «перевернутом» виде: в первом слове размещается смещение, а во втором — сегмент, причем каждое из этих слов в свою очередь представлено в «перевернутом» виде. Например, пара 1234h: 5678h будет записана так:



Директивы определения данных. Для того чтобы в программе на MASM зарезервировать ячейки памяти под константы и переменные, необходимо воспользоваться директивами определения данных — с названиями DE (описывает данные размером в байт), DW (размером в слово) и DD (размером в двойное слово). (Директивы, или команды ассемблеру, — это предложения программы, которыми ее автор сообщает какую-то информацию ассемблеру или просит что-то сделать дополнительно, помимо перевода символьных команд на машинный язык.)

В простейшем случае в директиве DB, DW или DD описывается одна константа, которой дается имя для последующих ссылок на нее. По этой директиве ассемблер формирует машинное представление константы (в частности, если надо, «переворачивает» ее) и записывает в очередную ячейку памяти. Адрес этой ячейки становится значением имени: все вхождения имени в программу ассемблер будет заменять на этот адрес.

Имена, указанные в директивах DB, DW и DD, называются именами переменных (в отличие от меток — имен команд).

В MASM числа записываются в нормальном (неперевернутом) виде в системах счисления с основанием 10, 16, 8 или 2. Десятичные числа записываются как обычно, за шестнадцатеричным числом ставится буква h (если число начинается с «цифры» A, в, ., F, то вначале обязателен 0), за восьмеричным числом — буква q или o, за двоичным числом — буква b.

Примеры:

```
A DB 162     ;описать константу-байт 162 и дать ей имя A
B DB OA2h    ;такая же константа, но с именем B
C DW -1      ;константа-слово -1 с именем C
D DW OFFFFh ;такая же константа-слово, но с именем D
E DD -1      ;-1 как двойное слово
```

Константы-символы описываются в директиве DB двояко: указывается либо код символа (целое от 0 до 255), либо сам символ в кавычках (одинарных или двойных); в последнем случае ассемблер сам заменит символ на его код. Например, следующие директивы эквивалентны (2A — код звездочки в ASCII):

```
CH DB 02Ah
CH DB '*'
CH DB «*»
```

Константы-адреса, как правило, задаются именами. Так, по директиве

```
ADR DW CH
```

будет отведено слово памяти, которому дается имя ADR и в которое запишется адрес (смещение), соответствующий имени сн. Если такое же имя описать в директиве DD, то ассемблер автоматически добавит к смещению имени его сегмент и запишет смещение в первую половину двойного слова, а сегмент — во вторую половину.

По любой из директив DB, DW и DD можно описать переменную, т. е. отвести ячейку, не дав ей начального значения. В этом случае в правой части директивы указывается вопросительный знак:

```
F DW ? ; отвести слово и дать ему имя F, ничего в это
слово не записывать.
```

В одной директиве можно описать сразу несколько констант и/или переменных одного и того же размера, для чего их надо перечислить через запятую. Они размещаются в соседних ячейках памяти. Пример:

```
G DB 200, -5, 10h, ?, 'F'
```

Имя, указанное в директиве, считается именующим первую из констант.

Для ссылок на остальные в MASM используются выражения вида <имя> + <целое>; например, для доступа к байту с числом -5 надо указать выражение G + 1, для доступа к байту с 10h — выражение G + 2 и т. д.

Если в директиве DB перечислены только символы, например:

```
S DB 'a','+','b'
```

тогда эту директиву можно записать короче, заключив все эти символы в одни кавычки:

```
S DB 'a+b'
```

И наконец, если в директиве описывается несколько одинаковых констант (переменных), то можно воспользоваться конструкцией повторения

```
k DUP(a,b,...,c),
```

которая эквивалентна повторенной *k* раз последовательности a, *b, ..., c.*

Например, директивы

```
V1 DB 0,0,0,0,0
V2 DW ?,?,?,?,?,?,?,?,?,'a',1,2,1,2,1,2,1,2
```

можно записать более коротко таким образом:

```
V1 DB 5 DUP(0)
V2 DW 9 DUP(?), 'a', 4 DUP(1,2)
```

Представление команд. Модификация адресов

Структура команд. Исполнительные адреса. Машинные команды I80X86 занимают от 1 до 6 байтов. Код операции (КОП) занимает один или два первых байта команды. В 180X86 много различных операций, так что для них не хватает 256 различных КОПов, которые можно представить в одном байте. Поэтому некоторые операции объединяются в группу и им дается один и тот же КОП, во втором же байте этот КОП уточняется. Кроме того, во втором байте указываются типы и способ адресации операндов. Остальные байты команды указывают на операнды.

Команды могут иметь от 0 до трех операндов, у большинства команд — один или два операнда. Размер операндов — байт или слово (редко — двойное слово). Операнд может быть указан в самой команде (это так называемый непосредственный операнд), либо может находиться в одном из регистров I80X86 и тогда в команде указывается этот регистр, либо может находиться в ячейке памяти и тогда в команде тем или иным способом указывается адрес этой ячейки. Некоторые команды требуют, чтобы операнд находился на фиксированном месте (например, в регистре AX), тогда этот операнд явно не указывается в команде. Результат выполнения команды помещается в регистр или ячейку памяти, из которого (которой), как правило, берется первый операнд. Например, большинство команд с двумя операндами реализуют действие

```
op1 := op1 - op2
```

где op1 — регистр или ячейка; а op2 — непосредственный операнд, регистр или ячейка.

Адрес операнда разрешено модифицировать по одному или двум регистрам. В первом случае в качестве регистра-модификатора разрешено использовать регистр BX, BP, SI или DI (и никакой иной). Во втором случае один из модификаторов обязан быть регистром BX или BP, а другой — регистром SI или DI; одновременная модификация по BX и BP или SI и DI недопустима.

Регистры вх и ВР обычно используются для хранения базы (начального адреса) некоторого участка памяти (скажем, массива) и потому называются базовыми регистрами, а регистры SI и DI часто содержат индексы элементов массива и потому называются индексными регистрами.

Однако такое распределение ролей необязательно, и, например, в SI может находиться база массива, а в вх — индекс элемента массива.

В MASM адреса в командах записываются в виде одной из следующих конструкции:

```
A, A[M] или A[M1][M2],
```

где А — адрес; м — регистр вх, ВР, SI или DI, Ml — регистр вх или ВР, а M2 — регистр SI или DI. Во втором и третьем варианте А может отсутствовать, в этом случае считается, что А = 0.

При выполнении команды процессор прежде всего вычисляет так называемый *исполнительный* (эффективный) адрес — как сумму адреса, заданного в команде, и текущих значений указанных регистров-модификаторов, причем все эти величины рассматриваются как неотрицательные и суммирование ведется по модулю 2^{16} ([r] означает содержимое регистра r):

```
A : Аисп = A
A[M] : Аисп = A+[M] (mod 2^16)
A[M1][M2]: Аисп = A+[M1]+[M2] (mod 2^16)
```

Полученный таким образом 16-разрядный адрес определяет так называемое *смещение* — адрес, отсчитанный от начала некоторого сегмента памяти. Перед обращением к памяти процессор еще добавляет к смещению начальный адрес этого сегмента (он хранится в некотором сегментном регистре), в результате чего получается окончательный 20-разрядный адрес, по которому и происходит фактическое обращение к памяти.

Форматы команд. В I80X86 форматы машинных команд достаточно разнообразны. Для примера приведем лишь основные форматы команд с двумя операндами.

1. Формат «регистр—регистр» (2 байта):

КОП	d	w	11	reg1	reg2
7 2	1	0	76	5 3	2 0

Команды этого формата описывают обычно действие reg1 := reg1 - reg2 или reg2 := reg2 - reg1. Поле КОП первого байта указывает на операцию (-), которую надо выполнить.

Бит w определяет размер операндов, а бит d указывает, в какой из регистров записывается результат:

w = 1	— слова	d = 1	reg1 := reg1 - reg2
= 0	— байты	= 0	reg2 := reg2 - reg1

Во втором байте два левых бита фиксированы (для данного формата), а трехбитовые поля reg1 и reg2 указывают на регистры, участвующие в операции, согласно следующим правилам:

reg	w=1	w=0
000	AX	AL
001	CX	CL
010	DX	DL
011	BX	BL
100	SP	AH
101	BP	CH
110	SI	DH
111	DI	BH

2. Формат «регистр—память» (2—4 байта):

КОП		w	mod	reg	mem	адрес (0 - 2 байта)

Эти команды описывают операции reg := reg - mem или mem := mem - reg. Бит w первого байта определяет размер операндов (см. ранее), а бит d указывает, куда записывается результат: в регистр (d = 1) или в ячейку памяти (d = 0). Трехбитовое поле reg второго байта указывает операнд-регистр (см. выше), двухбитовое поле mod определяет, сколько байт в команде занимает операнд-адрес (00 — 0 байтов, 01 — 1 байт, 10 — 2 байта), а трехбитовое поле mem указывает способ модификации этого адреса. В следующей таб-

лице указаны правила вычисления исполнительного адреса в зависимости от значений полей mod и mem (a8 — адрес размером в байт, a16 — адрес размером в слово):

mem\mod	00	01	10
000	[BX]+[SI]	[BX]+[SI]+a8	[BX]+[SI]+a16
001	[BX]+[DI]	[BX]+[DI]+a8	[BX]+[DI]+a16
010	[BP]+[SI]	[BP]+[SI]+a8	[BP]+[SI]+a16
011	[BP]+[DI]	[BP]+[DI]+a8	[BP]+[DI]+a16
100	[SI]	[SI]+a8	[SI]+a16
101	[DI]	[DI]+a8	[DI]+a16
110	a16	[BP]+a8	[BP]+a16
111	[BX]	[BX]+a8	[BX]+a16

Замечания. Если в команде не задан адрес, то он считается нулевым.

Если адрес задан в виде байта (a8), то он автоматически расширяется со знаком до слова (a*16).* Случай mod = 00 и mem = 110 указывает на отсутствие регистров-модификаторов, при этом адрес должет иметь размер слова (адресное выражение [BP] ассемблер транслирует в mod = 01 и mem = 110 при a8 = 0). Случай mod = 11 соответствует формату «регистр—регистр».

3. Формат «регистр—непосредственный операнд» (3—4 байта):

КОП	s	w	11	КОП'	reg	непосред. операнд (1 - 2 б)

Команды этого формата описывают операции reg := reg - immed (immed — непосредственный операнд). Бит w указывает на размер операндов, а поле reg — на регистр-операнд (см. ранее). Поле КОП в первом байте определяет лишь класс операции (например, класс сложения), уточняет же операцию поле КОП из второго байта. Непосредственный операнд может занимать 1 или 2 байта в зависимости от значения бита w, при этом операнд-слово записывается в команде в «перевернутом» виде. Ради экономии памяти в I80X86 предусмотрен случай, когда в операции над словами непосредственный операнд может быть задан байтом (на этот случай указывает «1» в бите s при w = 1), и тогда перед выполнением операции байт автоматически расширяется (со знаком) до слова.

4. Формат «память—непосредственный операнд» (3—6 байтов):

КОП	s	w	mod	КОП"	mem	адрес (0 2б)	непоср. оп (1 2б)

Команды этого формата описывают операции типа mem := mem - immed. Смысл всех полей — тот же, что и в предыдущих форматах.

Помимо рассмотренных в I80X86 используются и другие форматы команды с двумя операндами; так, предусмотрен специальный формат для команд, один из операндов которых фиксирован (обычно это регистр AX). Имеют свои форматы и команды с другим числом операндов.

Запись команд в MASM. Ранее уже было дано краткое описание форматов ассемблерных команд, здесь же следует его конкретизировать, с учетом дополнительных сведений. Из сказанного ясно, что одна и та же операция в зависимости от типов операндов записывается в виде различных машинных команд: например, в I80X86 имеется 28 команд пересылки байтов и слов. В то же время в MASM все эти «родственные» команды записываются единообразно: например, все команды пересылки имеют одну и ту же символьную форму записи:

```
MOV op1,op2 (op1:=op2)
```

Анализируя типы операндов, ассемблер сам выбирает подходящую машинную команду.

В общем случае команды записываются в MASM следующим образом:

```
МЕТКА: МНЕМОКОД ОПЕРАНДЫ ;КОММЕНТАРИЙ
```

Метка с двоеточием, а также точка с запятой и комментарий могут отсутствовать. Метка играет роль имени команды, ее можно использовать в командах перехода на данную команду. Комментарий не влияет на смысл команды, а лишь поясняет ее. Операнды, если есть, перечисляются через запятую. Основные правила записи операндов следующие.

Регистры указываются своими именами, например:

```
MOV AX,SI  ;оба операнда – регистры
```

Непосредственные операнды задаются константными выражениями (их значениями являются константы-числа), например:

```
MOV BH,5          5 - непосредственный операнд
MOV DI,SIZE X  SIZE X (число байтов, занимаемых
                переменной X) - непосредственный операнд
```

Адреса описываются адресными выражениями (например, именами переменных), которые могут быть модифицированы по одному или двум регистрам; например, в следующих командах первые операнды задают адреса:

```
MOV X,AH
MOV X[BX][DI],5
MOV [BX],CL
```

При записи команд в символьной форме необходимо внимательно следить за правильным указанием типа (размера) операндов, чтобы не было ошибок. Тип обычно определяется по внешнему виду одного из них, например:

`MOV AH,5` Пересылка байта, так как AH - байтовый регистр

`MOV AX,5` Пересылка слова, так как AX - 16-битовый регистр-операнд (5 может быть байтом и словом, по нему нельзя определить размер пересылаемой величины)

`MOV [BX],300` Пересылка слова, так как число 300 не может быть байтом

Если по внешнему виду можно однозначно определить тип обоих операндов, тогда эти типы должны совпадать, иначе ассемблер зафиксирует ошибку. Примеры:

`MOV DS,AX` Оба операнда имеют размер слова

`MOV CX,BH` Ошибка: регистры CX и BH имеют разные размеры

`MOV DL,300` Ошибка: DL — байтовый регистр, а число 300 не может быть байтом

Возможны ситуации, когда по внешнему виду операндов нельзя определить тип ни одного из них, как, например, в команде

```
MOV [BX],5
```

Здесь число 5 может быть и байтом, и словом, а адрес из регистра вх может указывать и на байт памяти, и на слово. В подобных ситуациях ассемблер фиксирует ошибку. Чтобы избежать ее, надо уточнить тип одного из операндов с помощью оператора с названием PTR:

```
MOV BYTE PTR [BX],5   Пересылка байта
MOV WORD PTR [BX],5   Пересылка слова
```

(Операторы — это разновидность выражений языка MASM, аналогичные функциям.)

Оператор PTR необходим и в том случае, когда надо изменить тип, предписанный имени при его описании. Если, например, x описано как имя переменной размером в слово:

```
X DW 999
```

и если надо записать в байтовый регистр AH значение только первого байта этого слова, тогда воспользоваться командой вида

```
MOV AH,X
```

нельзя, так как ее операнды имеют разный размер. Эту команду следует записать несколько иначе:

```
MOV AH,BYTE PTR X
```

Здесь конструкция BYTE PTR x означает адрес x, однако рассматриваемый как адрес байта, а не слова. (Напомним, что с одного и того же адреса может начинаться байт, слово и двойное слово; оператор PTR уточняет, какого размера ячейка имеется.)

И еще одно замечание. Если в символьной команде, оперирующей со словами, указан непосредственный операнд размером в байт, как, например, в команде

```
MOV AX,80h
```

то возникает некоторая неоднозначность: что будет записано в регистр AX — число 0080h (+128) или 0FF80h (-128)? В подобных ситуациях ассемблер формирует машинную команду, где операнд-байт расширен до слова, причем расширение происходит со знаком, если операнд был записан как отрицательное число, и без знака — в остальных случаях. Например:

```
MOV AX,-128 ; => MOV AX,0FF80h (A:=-128)
MOV AX,128  ; => MOV AX,0080h (A:=+128)
MOV AX,80h  ; => MOV AX,0080h (A:=+128)
```

Сегментирование

Сегменты памяти. Сегментные регистры. Первые модели I80X86 имели оперативную память объемом 2^{16} байтов (64 Кб) и потому использовали 16-битовые адреса. В последующих моделях память была увеличена до 2^{20} байтов (1 Мбайт = 1000 Кб), для чего уже необходимы 20-битовые адреса. Однако в этих I80X86 для преемственности были сохранены 16-битовые адреса: именно такие адреса хранятся в регистрах и указываются в командах, а также получаются в результате модификации по базовым и индексным регистрам. Как же удается 16-битовыми адресами ссылаться на 1 Мбайт памяти и большие объемы?

Эта проблема решается с помощью сегментирования адресов (неявного базирования адресов). В I80X86 вводится понятие «сегмент памяти». Так называется любой участок памяти размером до 64 Кб и с начальным адресом, кратным 16. Абсолютный (20-битовый) адрес A любой ячейки памяти можно представить как сумму 20-битового начального адреса (базы) B сегмента, которому принадлежит ячейка, и 16-битового смещения D — адреса этой ячейки, отсчитанного от начала сегмента: A = в + D. (Неоднозначность выбора сегмента не играет существенной роли, главное — чтобы сумма в и D давала нужный адрес.) Адрес в заносится в некоторый регистр S, а в команде, где должен быть указан адрес A, вместо него записывается пара из регистра s и смещения D (в MASM такая пара, называемая адресной парой или указателем, записывается как `S:D`). Процессор же устроен так, что при выполнении команды он прежде всего по паре `S:D` вычисляет абсолютный адрес A как сумму содержимого регистра s и смещения `D` и только затем обращается к памяти по этому адресу A. Таким образом, заменяя в командах абсолютные адреса на адресные пары, удается адресовать всю память 16-битовыми адресами (смещениями).

В качестве регистра s разрешается использовать не любой регистр, а только один из четырех регистров, называемых сегментными: CS, DS, SS и ES. В связи с этим одновременно можно работать с четырьмя сегментами памяти: начало одного из них загружается в регистр CS и все ссылки на ячейки этого сегмента указываются в виде пар CS : D, начало другого заносится в DS и все ссылки на его ячейки задаются в виде пар DS : D и т. д. Если одновременно надо работать с большим числом сегментов, то следует своевременно сохранять содержимое сегментных регистров и записывать в них начальные адреса пятого, шестого и т. д. сегментов.

Отметим, что используемые сегменты могут располагаться в памяти произвольным образом: они могут не пересекаться или Пересе-

каться и даже совпадать. Какие сегменты памяти использовать, в каких сегментных регистрах хранить их начальные адреса — все это дело автора программы.

Как и все регистры I80X86, сегментные регистры имеют размер слова. Поэтому возникает вопрос: как удается разместить в них 20-битовые начальные адреса сегментов памяти? Ответ следующий. Поскольку все эти адреса кратны 16 (см. ранее), то в них младшие 4 бита (последняя шестнадцатеричная цифра) всегда нулевые, а потому эти биты можно не хранить явно, а лишь подразумевать. Именно так и делается: в сегментном регистре всегда хранятся только первые 16 битов (первые четыре шестнадцатеричные цифры) начального адреса сегмента (эта величина называется номером сегмента или просто сегментом). При вычислении же абсолютного адреса ($A_{абс}$) по паре S:D процессор сначала приписывает справа к содержимому регистра s четыре нулевых бита (другими словами, умножает на 16) и лишь затем прибавляет смешение D, причем суммирование ведется по модулю 2^{20}:

$$A_{абс} = 16 \times [s] + D \pmod{2^{20}}.$$

Если, например, в регистре CS хранится величина 1234h, тогда адресная пара 1234h:507h определяет абсолютный адрес, равный 16 x 1234h + 507h = 12340h + 507h = 12847h.

Сегментные регистры по умолчанию. Согласно описанной схеме сегментирования адресов, замену абсолютных адресов на адресные пары надо производить во всех командах, имеющих операнд-адрес. Однако был разработан способ, позволяющий избежать выписывания таких пар в большинстве команд. Суть его в том, что устанавливается по умолчанию, какой именно сегментный регистр на какой сегмент памяти будет указывать, и что в командах задается только смещение: не указанный явно сегментный регистр автоматически восстанавливается согласно этой договоренности. И только при необходимости нарушить эту договоренность надо полностью указывать адресную пару. Список умолчаний приводится в табл. 4.4.

С учетом такого распределения ролей сегментных регистров машинные программы обычно строятся так: все команды программы размещаются в одном сегменте памяти, начало которого заносится в регистр cs, а все данные размещаются в другом сегменте, начало которого заносится в регистр DS; если нужен стек, то под него отводится третий сегмент памяти, начало которого записывается в регистр ss. После этого практически во всех командах можно указывать не полные адресные пары, а лишь смещения, так как сегментные регистры в этих парах будут восстанавливаться автоматически.

Таблица 4 4 **Сегментные регистры по умолчанию**

Регистр	Умолчания	Комментарий
CS	Указывает на начало области памяти, в которой размещены команды программы (эта область называется сегментом команд или сегментом кодов), и потому при ссылках на ячейки этой области регистр CS можно не указывать явно, он подразумевается по умолчанию	Абсолютный адрес очередной команды, подлежащей выполнению, всегда задается парой CS IP в счетчике команд IP всегда находится смещение этой команды относительно адреса из регистра CS
DS	Указывает на сегмент данных (область памяти с константами, переменными и другими величинами программы)	Во всех ссылках на этот сегмент регистр DS можно явно не указывать, так как он подразумевается по умолчанию
SS	Указывает на стек - область памяти, доступ к которой осуществляется по принципу «последним записан - первым считан»	Все ссылки на стек, в которых явно не указан сегментный регистр, по умолчанию сегментируются по регистру SS
ES	Считается свободным, он не привязан ни к какому сегменту памяти и его можно использовать по своему усмотрению	Чаще всего он применяется для доступа к данным, которые не поместились или сознательно не были размещены в сегменте данных

Здесь, правда, возникает такой вопрос: как по смещению определить, на какой сегмент памяти оно указывает? Этот вопрос рассматривается далее, а в общих чертах ответ такой: ссылки на сегмент команд могут содержаться только в командах перехода, а ссылки практически во всех других командах (кроме строковых и стековых) — это ссылки на сегмент данных. Например, в команде пересылки

```
MOV AX,X
```

имя X воспринимается как ссылка на данное, а потому автоматически преобразуется в адресную пару `DS:X`. В команде же безусловного перехода по адресу, находящемуся в регистре BX,

```
JMP BX
```

абсолютный адрес перехода определяется парой `cs : [BX]`.

Итак, если в ссылке на какую-то ячейку памяти не указан явно сегментный регистр, то этот регистр берется по умолчанию Явно же сегментные регистры надо указывать, только если по каким-то причинам регистр по умолчанию не подходит. Если, например, в команде пересылки следует сослаться на стек (скажем, надо записать в регистр AH байт стека, помеченный именем x), тогда нас уже не будет устраивать договоренность о том, что по умолчанию операнд команды MOV сегментируется по регистру DS, и потому мы обя-

заны явно указать иной регистр — в нашем случае регистр SS, так как именно он указывает на стек:

```
MOV AH,SS:X
```

Однако такие случаи встречаются редко и поэтому в командах, как правило, указываются только смещения.

Отметим, что в MASM сегментный регистр записывается в самой команде непосредственно перед смещением (именем переменной, меткой и т. п.), однако на уровне машинного языка ситуация несколько иная. Имеются четыре специальные однобайтовые команды, называемые префиксами замены сегмента (обозначаемые как CS:, DS:, SS: и ES:). Они ставятся перед командой, операнд-адрес которой должен быть просегментирован по регистру, отличному от регистра, подразумеваемому по умолчанию. Например, приведенная ранее символическая команда пересылки — это на самом деле две машинные команды:

```
SS:
MOV AH,X
```

Сегментирование, базирование и индексирование адресов. Поскольку сегментирование адресов — это разновидность модификации адресов, то в I80X86 адрес, указываемый в команде, в общем случае модифицируется по трем регистрам: сегментному, базовому и индексному. В целом, модификация адреса производится в два этапа. Сначала учитываются только базовый и индексный регистры (если они, конечно, указаны в команде), причем вычисление здесь происходит в области 16-битовых адресов; полученный в результате 16-битовый адрес называется *исполнительным (эффективным) адресом.* Если в команде не предусмотрено обращение к памяти (например, она загружает адрес в регистр), то на этом модификация адреса заканчивается и используется именно исполнительный адрес (он загружается в регистр). Если же нужен доступ к памяти, тогда на втором этапе исполнительный адрес рассматривается как смещение и к нему прибавляется (умноженное на 16) содержимое сегментного регистра, указанного явно или взятого по умолчанию, в результате чего получается *абсолютный (физический)* 20-битовый адрес, по которому реально и происходит обращение к памяти (рис. 4.15).

Отметим, что сегментный регистр учитывается только в «последний» момент, непосредственно перед обращением к памяти, а до этого работа ведется только с 16-битовыми адресами. Если учесть к тому же, что сегментные регистры, как правило, не указываются в

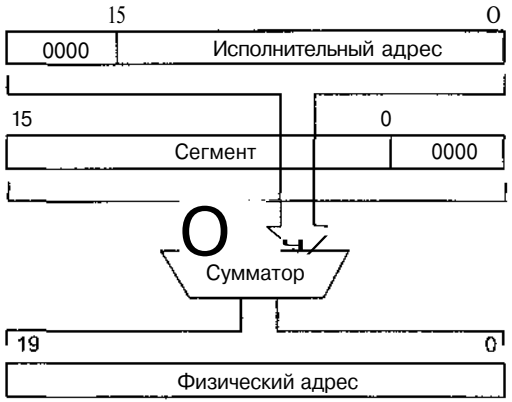


Рис. 4.15. Модификация адресов

командах, то можно в общем-то считать, что I80X86 работает с 16-битовыми адресами.

Как уже сказано, если в ссылке на ячейку памяти не указан сегментный регистр, то он определяется по умолчанию. Это делается по следующим правилам.

1. В командах перехода адрес перехода сегментируется по регистру CS и только по нему, так как абсолютный адрес команды, которая должна быть выполнена следующей, всегда определяется парой `CS:IP` (попытка изменить в таких командах сегментный регистр будет безуспешной).

Отметим, что сегментирование по регистру `CS` касается именно адреса перехода, а не адреса той ячейки, где он может находиться. Например, в команде безусловного перехода по адресу, находящемуся в ячейке x:

```
JMP X
```

имя x сегментируется по регистру `DS`, а вот адрес перехода, взятый из ячейки X, уже сегментируется по регистру CS.

2. Адреса во всех других командах, кроме строковых (STOS, `MOVS`, SCAS и CMPS), по умолчанию сегментируются.

- по регистру DS, если среди указанных регистров-модификаторов нет регистра BP,
- по регистру ss, если один из модификаторов — регистр BP.

Таким образом, адреса вида A, A[вх], `A[SI]`, A[DI], A[вх] `[SI]` и A[вх] [DI] сегментируются по регистру DS, а адреса A[BP], `A[BP][SI]` и `A[BP][DI]` — по регистру ss, т. е. адреса трех последних видов используются для доступа к ячейкам стека.

3. В строковых командах STOS, MOVS, SCAS и CMPS, имеющих два операнда-адреса, на которые указывают индексные регистры SI и DI, один из операндов (на который указывает SI) сегментируется по регистру DS, а другой (на него указывает DI) — по регистру ES.

Программные сегменты. Директива ASSUME. Рассмотрим, как осуществляется сегментирование в программах на MASM. Для того чтобы указать, что некоторая группа операторов программы образует единый сегмент памяти, они оформляются как программный сегмент: перед ними ставится директива SEGMENT, после них — директива ENDS, причем в начале обеих этих директив должно быть указано одно и то же имя, играющее роль имени сегмента. Программа в целом представляет собой последовательность таких программных сегментов, в конце которой указывается директива конца программы END, например:

```
DT1 SEGMENT ;программный сегмент с именем DT1
A DB 0
B DW ?
DT1 ENDS
;
DT2 SEGMENT ;программный сегмент DT2
C DB 'hello'
DT2 ENDS
;
CODE SEGMENT ;программный сегмент CODE
ASSUME CS:CODE, DS:DT1, ES:DT2
BEG: MOV AX,DT2
MOV DS,AX
MOV BH,C
...
CODE ENDS
END BEG ;конец текста программы
```

Предложения программного сегмента ассемблер размещает в одном сегменте памяти (в совокупности они не должны занимать более 64 Кб) начиная с ближайшего свободного адреса, кратного 16. Номер (первые 16 битов начального адреса) этого сегмента становится значением имени сегмента. В MASM это имя относится к константным, а не к адресным выражениям, в связи с чем в команде

```
MOV AX,DT2
```

второй операнд является непосредственным, поэтому в регистр AX будет записано начало (номер) сегмента DT2, а не содержимое начальной ячейки этого сегмента.

Имена же переменных (A, в, с) и метки (BEG) относятся к адресным выражениям, и им ставится в соответствие адрес их ячейки относительно «своего» сегмента:

имени A соответствует адрес 0;

имени в — адрес 1;

имени с — адрес 0, а метке BEG — адрес 0.

Все ссылки на предложения одного программного сегмента ассемблер сегментирует (по умолчанию) по одному и тому же сегментному регистру. По какому именно — устанавливается специальной директивой ASSUME. В нашем примере эта директива определяет, что все ссылки на сегмент CODE должны (если явно не указан сегментный регистр) сегментироваться по регистру CS, все ссылки на DT1 — по регистру DS, а все ссылки на DT2 — по регистру ES.

Встретив в тексте программы ссылку на какое-либо имя (например, на имя с в команде MOV `AX,с`), ассемблер определяет, в каком программном сегменте оно описано (у нас — в DT2), затем по информации из директивы ASSUME узнает, какой сегментный регистр поставлен в соответствие этому сегменту (у нас — это ES), и далее образует адресную пару из данного регистра и смещения имени (у нас — `ES:0`), которую и записывает в формируемую машинную команду. При этом ассемблер учитывает используемое в I80X86 соглашение о сегментных регистрах по умолчанию: если в адресной паре, построенной им самим или явно заданной в программе, сегментный регистр совпадает с регистром по умолчанию, то в машинную команду заносится лишь смещение. Если, скажем, в нашем примере встретится команда

```
MOV CX,B,
```

тогда по имени B ассемблер построит пару DS : 1, но поскольку операнд-адрес команды MOV по умолчанию сегментируется по регистру DS, то записывать этот регистр в машинную команду излишне и ассемблер записывает в нее только смещение 1.

Таким образом, директива ASSUME избавляет программистов от необходимости выписывать полные адресные пары не только тогда, когда используются сегментные регистры по умолчанию (как в случае с именем в), но тогда, когда в машинной команде нужно было бы явно указать сегментный регистр (как в случае с именем с). В MASM сегментный регистр в ссылке на имя требуется указывать лишь тогда, когда имя должно по каким-либо причинам сегментироваться по регистру, отличному от того, что поставлен в соответствие всему сегменту, в котором это имя описано.

Однако все это справедливо только при соблюдении следующих условий.

Во-первых, директива ASSUME должна быть указана перед первой командой программы В противном случае ассемблер, просматривающий текст программы сверху вниз, не будет знать, как сегментировать имена из команд, расположенных до этой директивы, и потому зафиксирует ошибку.

Во-вторых, в директиве ASSUME каждому сегменту следует ставить в соответствие сегментный регистр: если ассемблер встретит ссылку на имя из сегмента, которому не соответствует никакой сегментный регистр, то он зафиксирует ошибку. Правда, в обоих случаях можно избежать ошибки, но для этого в ссылках необходимо явно указывать сегментный регистр.

Начальная загрузка сегментных регистров. Директива ASSUME сообщает ассемблеру о том, по каким регистрам он должен сегментировать имена, из каких сегментов, и гарантирует, что в этих регистрах будут находиться начальные адреса этих сегментов. Однако загрузку этих адресов в регистры сама директива не осуществляет.

Сделать такую загрузку — обязанность самой программы, с загрузки сегментных регистров должно начинаться выполнение программы. Делается это так.

Поскольку в I80X86 нет команды пересылки непосредственного операнда в сегментный регистр (а имя, т. е. начало, сегмента — это непосредственный операнд), то такую загрузку приходится делать через какой-то другой, несегментный, регистр (например, AX):

```
MOV AX,DT1 ;AX := начало сегмента DT1
MOV DS,AX ;DS := AX
```

Аналогично загружается и регистр ES.

Загружать регистр cs в начале программы не надо: он, как и счетчик команд IP, загружается операционной системой перед тем, как начинается выполнение программы (иначе нельзя было бы начать ее выполнение). Что же касается регистра SS, используемого для работы со стеком, то он может быть загружен так же, как и регистры DS и ES, однако в MASM предусмотрена возможность загрузки этого регистра еще до выполнения программы.

Ссылки вперед. Встречая в символьной команде ссылку назад — имя, которое описано в тексте программы до этой команды, ассемблер уже имеет необходимую информацию об имени и потому может правильно оттранслировать эту команду. Но если в команде встретится ссылка вперед, т. е. имя, которое не было описано до

этой команды и которое, наверное, будет описано позже, то ассемблер в большинстве случаев не сможет правильно оттранслировать эту команду. Например, не зная, в каком программном сегменте будет описано это имя, ассемблер не может определить, по какому сегментному регистру надо сегментировать имя, и потому не может определить, надо или нет размещать перед соответствующей машинной командой префикс замены сегмента и, если надо, то какой именно.

В подобной ситуации ассемблер действует следующим образом:

- если в команде встретилась ссылка вперед, то он делает некоторое предположение относительно этого имени и уже на основе этого предположения формирует машинную команду;
- если затем (когда встретится описание имени) окажется, что данное предположение было неверным, тогда ассемблер пытается исправить сформированную ранее машинную команду. Однако это не всегда удается: если правильная машинная команда должна занимать больше места, чем машинная команда, построенная на основе предположения (например, перед командой надо на самом деле вставить префикс замены сегмента), тогда ассемблер фиксирует ошибку (как правило, это ошибка номер 6: Phase error between passes).

Какие же предположения делает ассемблер, встречая ссылку вперед?

Во всех командах, кроме команд перехода, ассемблер предполагает, что имя будет описано в сегменте данных и потому сегментируется по регистру DS. Это следует учитывать при составлении программы: если в команде встречается ссылка вперед на имя, которое описано в сегменте, на начало которого указывает сегментный регистр, отличный от DS, то перед таким именем автор программы должен написать соответствующий префикс. Пример:

```
CODE SEGMENT
ASSUME CS:CODE
X DW ?
BEG: MOV AX,X ;здесь вместо CS:X можно записать просто
X
MOV CS:Y,AX ;здесь обязательно надо записать CS:Y
...
Y DW ?
CODE ENDS
```

Переходы. В систему команд I80X86 входит обычный для ЭВМ набор команд перехода: безусловные и условные переходы, перехо-

ды с возвратами и др. Однако в I80X86 эти команды имеют некоторые особенности, которые здесь и рассматриваются.

Абсолютный адрес команды, которая должна быть выполнена следующей, определяется парой CS:IP, поэтому выполнение перехода означает изменение этих регистров, обоих или только одного (IP):

- если изменяется только счетчик команд IP, то такой переход называется *внутрисегментным или близким* (управление остается в том же сегменте команд);
- если меняются оба регистра CS и IP, то это *межсегментный,* или *дальний,* переход (начинают выполняться команды из другого сегмента команд).

По способу изменения счетчика команд переходы делятся на *абсолютные* и *относительные:*

- если в команде перехода указан адрес (смещение) той команды, которой надо передать управление, то это *абсолютный переход;*
- если в команде указана величина (сдвиг), которую надо добавить к текущему значению регистра IP, чтобы получился адрес перехода, и тогда это будет относительный переход, при этом сдвиг может быть *положительным* и *отрицательным,* так что возможен переход вперед и назад.

По величине сдвига относительные переходы делятся на короткие (сдвиг задается байтом) и длинные (сдвиг-слово).

Абсолютные переходы делятся на прямые и косвенные:

- при *прямом* переходе адрес перехода задается в самой команде;
- при *косвенном* — в команде указывается регистр или ячейка памяти, в котором (которой) находится адрес перехода.

Безусловные переходы. В MASM все команды безусловного перехода обозначаются одинаково:

```
JMP op
```

но в зависимости от типа операнда ассемблер формирует разные машинные команды.

1. Внутрисегментный относительный короткий переход.

```
JMP i8 (IP:=IP+i8)
```

Здесь i8 обозначает непосредственный операнд размером в 1 байт, который интерпретируется как знаковое целое от -128 до 127. Команда прибавляет это число к текущему значению регистра

IP, получая в нем адрес (смещение) той команды, которая должна быть выполнена следующей. Регистр CS при этом не меняется.

Необходимо учитывать следующую особенность регистра IP. Выполнение любой команды начинается с того, что в IP заносится адрес следующей за ней команды, и только затем выполняется собственно команда. Для команды относительного перехода это означает, что операнд `i8` прибавляется не к адресу этой команды, а к адресу команды, следующей за ней, поэтому, к примеру, команда `JMP 0` — это переход на следующую команду программы.

При написании машинной программы сдвиги для относительных переходов приходится вычислять вручную, однако MASM избавляет от этого неприятного занятия: в MASM в командах относительного перехода всегда указывается метка той команды, на которую надо передать управление, и ассемблер сам вычисляет сдвиг, который он и записывает в машинную команду. Отсюда следует, что в MASM команда перехода по метке воспринимается не как абсолютный переход, а как относительный.

По короткому переходу можно передать управление только на ближайшие команды программы, отстоящие от команды, следующей за командой перехода, до 128 байтов назад или до 127 байтов вперед.

2. Внутрисегментный относительный длинный переход используется для перехода на более дальние команды.

```
JMP i16 (IP:=IP+i16)
```

Здесь *i16* обозначает непосредственный операнд размером в слово, который рассматривается как знаковое целое от -32 768 до 32 767. Этот переход аналогичен короткому переходу.

Отметим, что, встретив команду перехода с меткой, которой была помечена одна из предыдущих (по тексту) команд программы, ассемблер вычисляет разность между адресом этой метки и адресом команды перехода и по этому сдвигу определяет, какую машинную команду относительного перехода (короткую или длинную) надо сформировать. Но если метка еще не встречалась в тексте программы, т. е. происходит переход вперед, тогда ассемблер, не зная еще адреса метки, не может определить, какую именно машинную команду относительного перехода формировать, поэтому он на всякий случай выбирает команду длинного перехода. Однако эта машинная команда занимает 3 байта, тогда как команда короткого перехода — 2 байта, и если автор программы на MASM стремится к экономии памяти и знает заранее, что переход вперед будет на близкую метку,

то он должен сообщить об этом ассемблеру, чтобы тот сформировал команду короткого перехода. Такое указание делается с помощью оператора SHORT:

```
JMP SHORT L
```

Для переходов назад оператор SHORT не нужен: уже зная адрес метки, ассемблер сам определит вид команды относительного перехода.

3. Внутрисегментный абсолютный косвенный переход.

```
JMP r16 (IP:=[r]) или
JMP m16 (IP:=[m16])
```

Здесь r16 обозначает любой 16-битовый регистр общего назначения, а m16 — адрес слова памяти. В этом регистре (слове памяти) должен находиться адрес, по которому и будет произведен переход. Например, по команде JMP BX осуществляется переход по адресу, находящемуся в регистре вх.

4. Межсегментный абсолютный прямой переход.

```
JMP seg:ofs (CS := seg, IP := ofs)
```

Здесь seg — начало (первые 16 битов начального адреса) некоторого сегмента памяти, а ofs — смещение в этом сегменте. Пара seg:ofs определяет абсолютный адрес, по которому делается переход. В MASM эта пара всегда задается конструкцией FAR PTR <метка>, которая указывает, что надо сделать переход по указанной метке, причем эта метка — «дальняя», из другого сегмента. Отметим, что ассемблер сам определяет, какой это сегмент, и сам подставляет в машинную команду его начало, т. е. seg.

5. Межсегментный абсолютный косвенный переход.

```
JMP m32 (CS := [m32 + 2], IP := [m32])
```

Здесь под т32 понимается адрес двойного слова памяти, в котором находится пара seg:ofs, задающая абсолютный адрес, по которому данная команда должна выполнить переход. Напомним, что в I80X86 величины размером в двойное слово хранятся в «перевернутом» виде, поэтому смещение ofs находится в первом слове двойного слова m32, а смещение seg — во втором слове (по адресу m32 + 2).

Команды межсегментного перехода используются тогда, когда команды программы размещены не в одном сегменте памяти, а в нескольких (например, если команд так много, что в совокупности

они занимают более 64 Кб, т. е. больше максимального размера сегмента памяти). При переходе из одного такого сегмента в другой необходимо менять не только счетчик команд IP, но и содержимое регистра CS, загружая в последний начальный адрес второго сегмента. Такое одновременное изменение обоих этих регистров и делают команды межсегментного перехода.

При записи в MASM команд перехода следует учитывать, что они могут восприниматься неоднозначно. Скажем, как воспринимать команду

```
JMP A
```

как переход по метке A или как переход по адресу, хранящемуся в ячейке с именем A?

Кроме того, какой это переход — внутрисегментный или межсегментный? Ответ зависит от того, как описано имя A, и от того, когда описано имя A — до или после команды перехода.

Пусть A описано до команды перехода *(ссылка назад).* Если именем A помечена некоторая команда текущего сегмента команд (т. е. A — метка), тогда ассемблер формирует машинную команду внутрисегментного относительного перехода. Если же A — имя переменной, тогда ассемблер формирует машинную команду косвенного перехода — внутрисегментного, если A описано в директиве DW, или межсегментного, если A описано в директиве DD.

В случае же, если имя A описано после команды перехода *(ссылка вперед),* ассемблер всегда формирует машинную команду внутрисегментного относительного длинного перехода. С учетом этого имя A обязательно должно быть меткой команды из текущего сегмента команд, иначе будет зафиксирована ошибка. Если такая трактовка ссылки вперед не удовлетворяет автора программы, то он обязан с помощью оператора SHORT или PTR уточнить тип имени A:

```
JMP SHORT A ;внутрисегментный короткий переход по метке
JMP WORD PTR A ;внутрисегментный косвенный переход
JMP DWORD PTE A ;межсегментный косвенный переход
```

Отметим, что переход по метке A из другого сегмента команд всегда должен указываться с помощью FAR PTR (независимо от того, описана метка A до или после команды перехода):

```
JMP FAR PTR A ;межсегментный переход по метке
```

Условные переходы. Практически во всех командах условного перехода проверяется значение того или иного флага (например, флага нуля ZF) и, если он имеет определенное значение,

выполняется переход по адресу, указанному в команде. Значение флага должно быть установлено предыдущей командой, например командой сравнения

```
CMP op1,op2
```

которая вычисляет разность op1 - op2, однако результат никуда не записывает, а только меняет флаги, на которые будет реагировать команда условного перехода.

В MASM команды условного перехода имеют следующую форму:

```
Jxx op
```

где xx — одна или несколько букв, в сокращенном виде отражающих проверяемое условие (обычно в предположении, что перед этой командой находится команда сравнения).

Примеры некоторых мнемонических записей:

```
JE - переход «по равно» (jump if equal)
JL - переход «по меньше» (jump if less)
JNL - переход «по не меньше» (jump if not less)
```

Особенностью всех машинных команд условного перехода является то, что они реализуют внутрисегментный относительный короткий переход, т. е. добавляют к счетчику команд IP свой операнд, рассматриваемый как число со знаком от -128 до 127. В MASM этот операнд всегда должен записываться как метка, которую ассемблер заменит на соответствующий сдвиг.

Такая особенность команд условного перехода вызывает неудобство при переходах на «дальние» команды. Например, если надо сделать переход при A < в на команду, помеченную меткой L и расположенную далеко от команды перехода, то приходится использовать команду длинного безусловного перехода:

```
MOV AX,A
CMP AX,B ;сравнение A и B
JNL M ;не меньше -> M (обход команды JMP)
JMP L ;меньше -> L (длинный переход)
M: ...
```

Команды управления циклом. В I80X86 есть несколько команд, упрощающих программирование циклов с заранее известным числом повторений. Применение этих команд требует, чтобы к началу цикла в регистр cx было занесено число шагов цикла. Сами команды размещаются в конце цикла, они уменьшают значение cx

на 1 и, если cx еще не равно 0, передают управление на начало цикла. Например, найти s — сумму элементов массива x из 10 чисел-слов можно так:

```
MOV AX,0 ;начальное значение суммы (накапливается в AX)
MOV SI,0 ;начальное значение индексного регистра
MOV CX,10 ;число повторений цикла
L: ADD AX,X[SI] ;AX:=AX+X[i]
ADD SI,2 ;SI:=SI+2
LOOP L ;CX:=CX-1; if CX<>0 then goto L
MOV S,AX ;S:=AX
```

Помимо команды LOOP есть еще две «циклические» команды — LOOPZ и LOOPNZ (они имеют синонимичные названия LOOPE и LOOPNE), которые, кроме регистра cx, проверяют еще и флаг нуля ZF; например, команда LOOPZ «выходит» из цикла, если cx = 0 или ZF = 1. Эту команду можно, например, использовать при поиске в массиве первого нулевого элемента, где должно быть предусмотрено два условия выхода из цикла: либо будет найден нулевой элемент (ZF = 1, если перед LOOPZ поставить команду сравнения очередного элемента с 0), либо будет исчерпан весь массив (cx = 0).

Отметим, что все эти «циклические» команды реализуют короткий относительный переход, как и команды условного перехода, поэтому их можно использовать только для циклов с небольшим числом команд.

В MASM есть еще две команды перехода — CALL (переход с возвратом) и RET (возврат из подпрограммы).

Строковые операции. В I80X86 под строкой понимается последовательность соседних байтов или слов. В связи с этим все строковые команды имеют две разновидности:

- для работы со строками из байтов (в мнемонику операций входит буква b);
- для работы со строками из слов (в мнемонику входит W).

Имеются следующие операции над строками:

- пересылка элементов строк (в память, из памяти, память—память);
- сравнение двух строк;
- просмотр строки с целью поиска элемента, равного заданному.

Каждая из этих операций выполняется только над одним элементом строки, однако одновременно происходит автоматическая настройка на следующий или предыдущий элемент строки. Имеются специальные команды повторения (REP и др.), которые заставляют следующую за ними строковую команду многократно по-

вторяться (до 2^{16} раз), в связи с чем такая пара команд позволяет обработать всю строку, причем намного быстрее, чем запрограммированный цикл.

Кроме того, строки можно просматривать вперед (от их начала к концу) и назад. Направление просмотра зависит от флага направления DF, значение которого можно менять с помощью команд STD (DF := 1) и CLD (DF := 0). При DF = 0 все последующие строковые команды программы просматривают строки вперед, а при DF = 1 — назад..

В строковых командах операнды явно не указываются, а подразумеваются. Если команда работает с одной строкой, то адрес очередного (обрабатываемого сейчас) элемента строки задается парой регистров DS и SI или парой ES и DI, а если команда работает с двумя строками, то адрес элемента одной из них определяется парой DS:SI, а адрес элемента другой — парой ES:DI. После выполнения операции значение регистра SI и/или DI увеличивается (при DF = 0) или уменьшается (при DF = 1) на 1 (для байтовых строк) или на 2 (для строк из слов).

Начальная установка всех этих регистров, а также флага DF должна быть выполнена до начала операции над строкой. Если сегментный регистр DS уже имеет нужное значение, тогда загрузить регистр SI можно с помощью команды

```
LEA SI,<начальный/конечный адрес строки>
```

Если же надо загрузить сразу оба регистра DS и SI, тогда можно воспользоваться командой

```
LDS SI,m32,
```

которая в регистр SI заносит первое слово, а в регистр DS — второе слово из двойного слова, имеющего адрес m32 (таким образом, по адресу т32 + 2 должен храниться сегмент, а по адресу т32 — смещение начального или конечного элемента строки). Начальную загрузку регистров ES и DI обычно осуществляют одной командой

```
LES DI,m32,
```

которая действует аналогично команде LDS.

Перечислим вкратце строковые команды 180X86:

- команда загрузки элемента строки в аккумулятор (LODSB или LODSW) пересылает в регистр AL или AX очередной элемент строки, на который указывает пара DS:SI, после чего увеличивает (при DF = 0) или уменьшает (при DF = 1) регистр SI на 1 или 2;

- запись аккумулятора в строку (STOSB или STOSW); содержимое регистра AL или AX заносится в тот элемент строки, на который указывает пара ES:DI, после чего изменяет регистр DI на 1 или 2;
- пересылка строк (MOVSB или MOVSW); элемент первой строки, определяемый парой DS:SI, заносится в элемент второй строки, определяемый парой ES:DI, после чего одновременно меняет регистры SI и DI;
- сравнение строк (CMPSB или CMPSW); сравниваются очередные элементы строк, указываемые парами DS:SI и ES:DI, и результат сравнения (равно, меньше и т. п.) фиксирует в флагах, после чего меняется содержание регистров SI и DI;
- сканирование строки (SCASB или SCASW) — сравнивается элемент строки, адрес которого задается парой ES:DI, со значением регистра AL или AX и результат сравнения фиксирует в флагах, после чего меняется содержимое регистра DI.

Перед любой строковой командой можно поставить одну из двух команд, называемых «префиксами повторения», которая заставит многократно повториться эту строковую команду. Число повторений (обычно это длина строки) должно быть указано в регистре cx.

Префикс повторения REPZ (синонимы — REPE, REP) сначала заносит 1 в флаг нуля ZF, после чего, постоянно уменьшая cx на 1, заставляет повторяться следующую за ним строковую команду до тех пор, пока в cx не окажется «О» или пока флаг ZF не изменит свое значение на 0.

Другой префикс повторения REPNZ (синоним — REPNE) действует аналогично, но только вначале устанавливает флаг ZF в 0, а при изменении его на 1 прекращает повторение строковой команды.

Пример. Пусть надо переписать 10 000 байтов начиная с адреса A в другое место памяти начиная с адреса в. Если оба этих имени относятся к сегменту данных, на начало которого указывает регистр DS, тогда эту пересылку можно сделать так:

```
CLD ;DF:=0 (просмотр строки вперед)
MOV CX,1000 ;CX - число повторений
MOV AX,DS
MOV ES,AX ;ES:=DS
LEA SI,A ;ES:SI - «откуда»
LEA DI,B ;DS:DI - «куда»
REP MOVSB ;пересылка CX байтов
```

Стек. Подпрограммы

Стек. В I80X86 имеются специальные команды работы со *стеком, т. е. областью памяти, доступ к элементам которой осуществляется по принципу «последним записан — первым считан»* (рис. 4.16). Но для того чтобы можно было воспользоваться этими командами, необходимо соблюдение ряда условий.

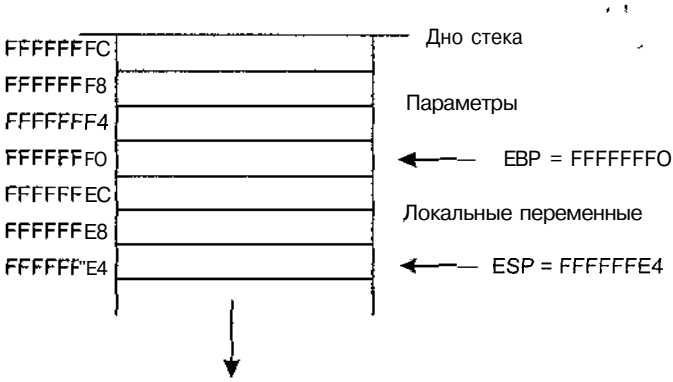


Рис. 4.16. Структура стековой памяти

Во многих случаях программе требуется временно запомнить информацию, а затем считывать ее в обратном порядке. Эта проблема в ПК решена посредством реализации стека LIFO («последним пришел — первым ушел»), называемого также стеком включения/извлечения *(stack* — кипа, например бумаг). Наиболее важное использование стека связано с процедурами. Стек обычно рассчитан на косвенную адресацию через регистр SP — указатель стека. При включении элементов в стек производится автоматическое уменьшение указателя стека, а при извлечении — увеличение, т. е. стек всегда «растет» в сторону меньших адресов памяти. Адрес последнего включенного в стек элемента называется вершиной стека (TOS).

Под стек можно отвести область в любом месте памяти. Размер ее может быть любым, но не должен превосходить 64 Кб, а начальный адрес должен быть кратным 16. Другими словами, эта область должна быть сегментом памяти (сегмент стека). Начало этого сегмента (первые 16 битов начального адреса) должно обязательно храниться в сегментном регистре SS.

Хранимые в стеке элементы могут иметь любой размер, однако следует учитывать, что в 180X86 имеются только команды записи в

стек и чтения из него слов. Поэтому для записи байта в стек его надо предварительно расширить до слова, а запись или чтение двойных слов осуществляются парой команд.

В I80X86 принято заполнять стек снизу вверх, от больших адресов к меньшим: первый элемент записывается в конец области, отведенной под стек, второй элемент — в предыдущую ячейку области и т. д. Считывается всегда элемент, записанный в стек последним. В связи с этим нижняя граница стека всегда фиксирована, а верхняя — меняется. Слово памяти, в котором находится элемент стека, записанный последним, является вершиной стека. Адрес вершины, отсчитанный от начала сегмента стека, должен находиться в указателе стека — регистре SP. Таким образом, абсолютный адрес вершины стека определяется парой `SS:SP`.

Значение «О» в регистре SP свидетельствует о том, что стек полностью заполнен (его вершина «дошла» до начала области стека). Поэтому для контроля за переполнением стека надо перед новой записью в стек проверять условие SP = 0 (сам I80X86 этого не делает). Для пустого стека значение SP должно равняться размеру стека, т. е. пара SS:SP должна указывать на байт, следующий за последним байтом области стека. Контроль за чтением из пустого стека, если это необходимо, должна делать сама программа.

Начальная установка регистров SS и SP может быть произведена программой, однако в MASM предусмотрена возможность автоматической загрузки этих регистров. Если в директиве SEGMENT, начинающей описание сегмента стека, указать параметр STACK, тогда ассемблер (точнее, загрузчик) перед тем, как передать управление на первую команду машинной программы, загрузит в регистры SS и SP нужные значения. Например, если в программе сегмент стека описан следующим образом:

```
ST SEGMENT STACK
DB 256 DUP(?)  ;размер стека - 256 байтов
ST ENDS
```

и если под этот сегмент была выделена область памяти начиная с абсолютного адреса `12340h`, тогда к началу выполнения программы в регистре ss окажется величина `1234h`, а в регистре SP — величина `100h` (=256).

Отметим, что эти значения соответствуют пустому стеку.

Основные стековые команды. При соблюдении указанных требований в программе можно использовать команды,

предназначенные для работы со стеком. Основными из них являются следующие.

1. Запись слова в стек:

```
PUSH  op
```

Здесь op обозначает любой 16-битовый регистр (в том числе сегментный) или адрес слова памяти. По этой команде значение регистра SP уменьшается на 2 (вычитание происходит по модулю 2^{16}), после чего указанное операндом слово записывается в стек по адресу `SS:SP`.

2. Чтение слова из стека:

```
POP  op
```

Слово, считанное из вершины стека, присваивается операнду op (регистру, в том числе сегментному, но не CS, или слову памяти), после чего значение SP увеличивается на 2.

3. Переход с возвратом:

```
CALL  op
```

Эта команда записывает адрес следующей за ней команды в стек и затем делает переход по адресу, определяемому операндом op. Она используется для переходов на подпрограммы с запоминанием в стеке адреса возврата. Имеются следующие разновидности этой команды (они аналогичны вариантам команды безусловного перехода `JMP`):

- *внутрисегментный относительный длинный* переход (op — непосредственный операнд размером в слово, а в MASM — это метка из текущего сегмента команд или имя близкой процедуры (см. ниже)); в этом случае в стек заносится только текущее значение счетчика команд IP, т. е. смещение следующей команды;
- *внутрисегментный абсолютный косвенный* переход (op — адрес слова памяти, в которой находится адрес (смещение) той команды, на которую и будет сделан переход); и здесь в стек записывается только смещение адреса возврата;
- *межсегментный абсолютный прямой* переход (op — непосредственный операнд вида `seg:ofs`, а в MASM — это FAR PTR `<метка>` или имя дальней процедуры); здесь в стек заносятся текущие значения регистров `CS` и `IP` (первым в стек записывается содержимое CS), т. е. абсолютный адрес возврата, после чего меняются регистры CS и IP;

- *межсегментный абсолютный косвенный* переход (op — адрес двойного слова, в котором находится пара `seg:ofs`, задающая абсолютный адрес перехода), и здесь в стеке спасается содержимое регистров CS и IP.

4. Переход (возврат) по адресу из стека:

RET op

Из стека считывается адрес и по нему производится переход. Если указан операнд (а это должно быть неотрицательное число), то после чтения адреса стек еще очищается на это число байтов (к SP добавляется это число). Команда используется для возврата из подпрограммы по адресу, записанному в стек по команде CALL при вызове подпрограммы, и одновременной очистки стека от параметров, которые основная программа занесла в стек перед обращением к подпрограмме.

Команда RET имеет две разновидности (хотя в MASM они одинаково записываются):

- в первом случае из стека считывается только одно слово — смещение адреса возврата;
- во втором — из стека считывается пара `seg:ofs`, указывающая абсолютный адрес возврата. Каким образом ассемблер определяет, какой случай имеет место, объяснено ниже.

В I80X86 стек в основном используется для организации подпрограмм и прерываний. Однако даже если программе не нужен стек, она все равно должна отвести под него место. Дело в том, что стеком будет неявно пользоваться операционная система при обработке прерываний, которые возникают (например, при нажатии клавиш на клавиатуре) в то время, когда выполняется программа. Для нужд ОС рекомендуется выделять в стеке 64 байта.

Прерывания. Иногда необходимо выполнить одну из набора специальных процедур, если в системе или в программе возникают определенные условия, например, нажата клавиша на клавиатуре. Действие, стимулирующее выполнение одной из таких процедур, называется *прерыванием*, поскольку основной процесс при этом приостанавливается на время выполнения этой процедуры

Существует два общих класса прерываний: внутренние и внешние. Первые инициируются состоянием ЦП или командой, а вторые — сигналом, подаваемым от других компонентов системы. Типичные внутренние прерывания: деление на нуль, переполнение и т. п., а типичные внешние — запрос на обслуживание со стороны какого-либо устройства ввода/вывода.

Переход к процедуре прерывания осуществляется из любой программы, а после выполнения процедуры прерывания обязательно происходит возврат в прерванную программу. Перед обращением к процедуре прерывания должно быть сохранено состояние всех регистров и флагов, используемых процедурой прерывания, а после окончания прерывания эти регистры должны быть восстановлены.

Прерывание вынуждает процессор прекратить выполнение одной последовательности команд и начать выполнение другой, при этом адрес очередной команды, которая должна была бы выполняться (содержимое регистра IP), если бы не было прерывания, запоминается. Адрес команды, которая должна выполняться после возникновения прерывания, выбирается из таблицы, хранящейся в начальной области памяти. Эта таблица называется *таблицей векторов прерываний.* В таблице записано 256 адресов. Когда устройство вызывает прерывание процессора, оно сообщает ему, какой адрес из таблицы следует использовать для перехода к новой последовательности команд.

Аппаратное прерывание. На рис. 4.17 изображены процессор, запоминающее устройство с таблицей адресов, система прерываний, внешнее устройство, требующее прерывания, и 8-разряд-

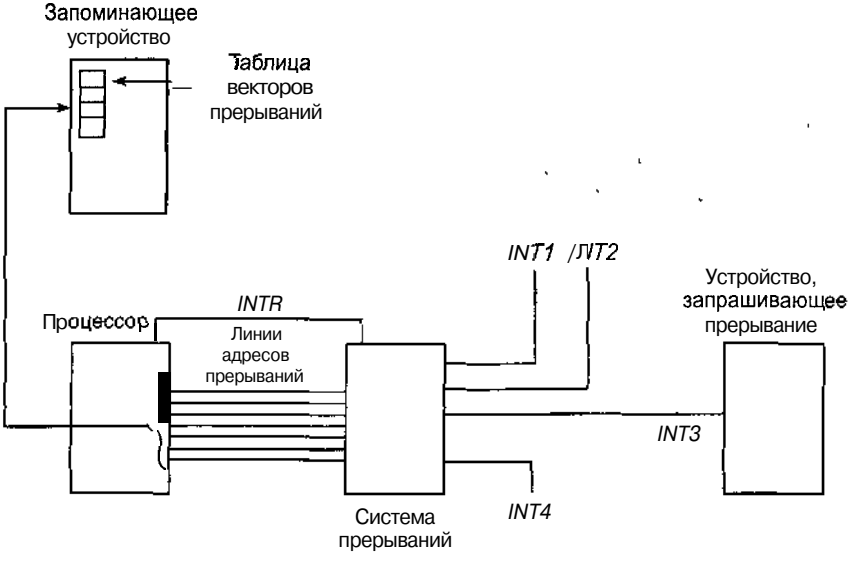


Рис. 4.17. Аппаратные прерывания

ная шина адреса прерывания, по которой из системы прерываний процессору передается указание о том, какой адрес из таблицы векторов прерываний должен быть использован при получении процессором сигнала аппаратного прерывания. Связь системы прерываний с процессором осуществляется с помощью шины прерываний *INTR,* появление на этой шине, обычно находящееся в состоянии «1», сигнала «О» приводит к прерыванию работы процессора. Справа от системы прерываний изображены четыре шины, соединяющие ее с устройствами, которые могут потребовать прерывания работы процессора, и, в частности, устройство, требующее внимания со стороны процессора и использующее для прерывания шину *INT3.*

Прерывание возникает тогда, когда устройство, требующее прерывания (например, печатающее устройство), посылает сигнал на шину *INT3.* Система прерываний выявляет наличие сигнала на шине и реагирует на него; т. е. помещает восьмибитовое число на шину адреса прерывания и устанавливает «О» на шине прерываний *INTR.* Процессор, определив, что на шине *INTR* появился сигнал «О», запоминает адрес команды, которая должна была бы выполняться следующей. Затем с адресной шины прерываний считывается число, указывающее, какой адрес необходимо извлечь из таблицы векторов прерываний, и процессор переходит к выполнению команды, начинающейся с этого адреса.

Типичным примером устройства, требующего прерывания, является клавиатура. После нажатия клавиши на клавиатуре в систему прерываний передается сигнал, приводящий к прерыванию работы процессора. Очевидно, что процессор прекращает текущую работу и, используя адрес, переданный из системы прерываний, начинает выполнение специальной программы взаимодействия с клавиатурой. Программа вводит с клавиатуры код, соответствующий нажатой клавише, и заносит его в область сохранения, расположенную в памяти. Затем программа взаимодействия с клавиатурой возвращает управление программе, которая выполнялась до прерывания.

Программные прерывания. Прерывания могут также иметь место при выполнении специальной команды: *прервать, используя адрес* X. (Число x указывает один из адресов в таблице векторов прерываний.) При выполнении команды «прервать» процессор определяет и запоминает адрес команды, которая должна была бы выполняться следующей, а затем переходит к выполнению программы, начинающейся с адреса X, извлеченного из таблицы

векторов прерываний. Такая последовательность действий называется *программным прерыванием.*

Подпрограммы. Типичная схема организации подпрограмм, обычно используемая трансляторами с языков высокого уровня для реализации процедур и функций (в частности, рекурсивных), следующая.

При обращении к подпрограмме в стек заносятся параметры для нее и адрес возврата, после чего делается переход на ее начало.

```
PUSH param1 ;запись 1-го параметра в стек
...
PUSH paramk ;запись последнего (k-го) параметра в стек
CALL subr   ;переход с возратом на подпрограмму
```

(Если необходимо вычислить параметр или если его размер отличен от слова, тогда для записи параметра в стек нужны, конечно, несколько команд, а не одна.)

Первыми командами подпрограммы обычно являются следующие:

```
PUSH BP      ;сохранить в стеке старое значение BP
MOV SP,BP    /установить BP на вершину стека
SUB SP,m     /отвести в стеке место (т байтов) под
             ; локальные величины подпрограммы.
```

Поясним эти «входные» команды. В подпрограмме для обращения к ячейкам стека, занятым параметрами, используется (как базовый) регистр BP: если в BP занести адрес вершины стека, то для доступа к этим ячейкам следует использовать адресные выражения вида i[BP] или, что то же самое, [BP + i]. (Отметим, что применять здесь регистры-модификаторы вх, SI и DI нельзя, так как формируемые по ним исполнительные адреса будут сегментироваться по умолчанию по регистру DS, а в данном случае необходимо сегментирование по ss.) Однако данная подпрограмма может быть вызвана из другой, также использующей регистр BP, поэтому прежде, чем установить BP на вершину стека, надо спасти в стеке старое значение этого регистра, что и делает первая из «входных» команд.

Вторая же команда устанавливает BP на вершину стека. Если предположить, что каждый параметр и адрес возврата занимают по слову памяти, тогда доступ к первому параметру обеспечивается адресным выражением [BP + 4], ко второму — выражением [BP + 6] и т. д.

Подпрограмме может потребоваться место для ее локальных величин. Такое место обычно отводится в стеке (а для рекурсивных подпрограмм только в стеке) «над» ячейкой, занимаемой старым значением BP. Если под эти величины нужно *m* байтов, то такой «захват» места можно реализовать простым уменьшением значения регистра SP на *m,* что и делает 3-я «входная» команда. Доступ к локальным величинам обеспечивается адресными выражениями вида `[BP-1]`. Если подпрограмме не нужно место под локальные величины, тогда третью из «входных» команд следует опустить.

Выход из подпрограммы реализуется следующими командами:

```
MOV SP,BP ;очистить стек от локальных величин
POP BP    ;восстановить старое значение BP
RET 2xk   ;возврат из подпрограммы и очистка стека от
          ;параметров (считаем, что они занимают
          ; 2xk байтов).
```

Первая из этих «выходных» команд заносит в регистр SP адрес той ячейки стека, где хранится старое значение регистра BP, т. е. происходит очистка стека от локальных величин (если их не было, то данную команду надо опустить). Вторая команда восстанавливает в BP это старое значение, одновременно удаляя его из стека. В этот момент состояние стека будет таким же, как и перед входом в подпрограмму. Третья команда считывает из стека адрес возврата (в результате чего SP «опускается» на 2 байта), затем добавляет к SP число, которое должно равняться числу байтов, занимаемых всеми параметрами подпрограммы, и затем осуществляет переход по адресу возврата. В этот момент состояние стека будет таким же, каким оно было перед обращением к подпрограмме.

Здесь описана универсальная схема организации работы подпрограмм. В конкретных случаях могут использоваться более простые схемы. Например, параметры можно передавать не через стек, а через регистры, место под локальные величины можно отводить не в стеке, а в сегменте данных и т. п.

Процедуры в ассемблере. При составлении и вызове подпрограмм необходимо следить за тем, чтобы команды CALL и RET действовали согласованно — были одновременно близкими или дальними. В MASM эта проблема снимается, если подпрограмму описать как процедуру. Процедуры имеют следующий вид:

```
имя_процедуры PROC [NEAR или FAR]
...
имя_процедуры ENDP
```

Хотя в директиве PROC после имени процедуры не ставится двоеточие, это имя относится к меткам и его можно указывать в командах перехода, в частности в команде CALL, когда надо вызвать процедуру Это же имя должно быть повторено в директиве ENDP, заканчивающей описание процедуры Предложения между этими двумя директивами образуют тело процедуры (подпрограмму) Имя процедуры является фактически меткой первой из команд тела, поэтому данную команду не надо специально метить

Если в директиве PROC указан параметр NEAR или он вообще не указан, то такая процедура считается «близкой» и обращаться к ней можно только из того сегмента команд, где она описана. Дело в том, что ассемблер будет заменять все команды CALL, где указано имя данной процедуры, на машинные команды близкого перехода с возвратом, а все команды RET внутри процедуры — на близкие возвраты.

Если же в директиве PROC указан параметр FAR, то это «дальняя» процедура: все обращения к ней и все команды RET внутри нее рассматриваются ассемблером как дальние переходы. Обращаться к этой процедуре можно из любых сегментов команд.

Таким образом, достаточно лишь указать тип процедуры (близкая она или дальняя), и всю остальную работу возьмет на себя ассемблер: переходы на нее и возвраты из нее будут автоматически согласованы с этим типом. В этом главное (и единственное) достоинство описания подпрограмм в виде процедур. (Отметим, что метки и имена, описанные в процедуре, не локализуются в ней.)

Например, вычисление ax := sign(ax) можно описать в виде процедуры следующим образом:

```
SING PROC FAR ; дальняя процедура
CMP AX,0
JE SGN1 ;AX=0 - перейти к SGN1
MOV AX,1 ;AX:=1 (флаги не изменились!)
JG SGN1 ;AX>0 - перейти к sgn1
MOV AX,-1 ;AX:=-1
SGN1: RET ;дальний возврат
SIGN ENDP
...
```

Возможный пример обращения к этой процедуре:

```
;CX:=SIGN(VAR)
MOV AX,VAR
CALL SIGN ;дальний вызов
MOV CX,AX
```

4.4. Защищенный режим

Рассмотрим работу в *защищенном режиме* процессоров 80286, 80386 и более старших моделей

Адресация в защищенном режиме 16-разрядного МП 80286

В этом режиме механизм адресации отличается от механизма адресации в реальном режиме (рис. 4.18) За счет использования 24-битной адресной шины процессора 80286 физическое адресное пространство достигает 16 Мбайт (2^M байт).

В данном режиме для определения адресов также необходимы два регистра, однако содержимое каждого регистра сегмента не соответствует непосредственно какому-либо участку оперативной памяти. Регистр сегмента, который в защищенном режиме называют селектором, в действительности становится индексом, указывающим на элемент таблицы, называемой таблицей дескрипторов (Descriptor Table). Каждый элемент этой таблицы (называемый дескриптором) характеризует один сегмент команд или один сегмент данных. Сегмент стека относят при этом к сегментам данных, поскольку он не содержит исполняемых команд (рис 4.18).

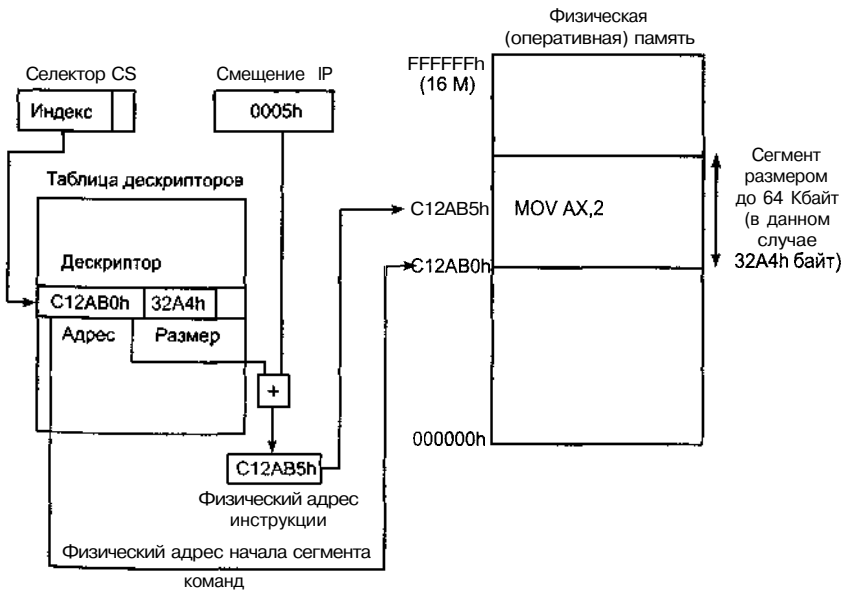


Рис. 4.18. Адресация в защищенном режиме

Дескриптор содержит физический адрес начала сегмента в памяти (базовый адрес), его размер, а также другую информацию. Базовый адрес размещается в 24 битах (а не в 20, как для реального режима), поэтому сегменты не обязательно теперь должны быть выровнены на адрес, кратный 16 Размер сегмента указывается в 16 битах, и поэтому может принимать любые значения, меньшие или равные 64 Кбайт. В этом состоит фундаментальное отличие от реального режима, где каждый сегмент по умолчанию имеет размер 64 Кбайт. После определения базового адреса сегмента к нему прибавляется значение смещения (размещаемого в 16 битах), и процессор может обратиться к соответствующей ячейке памяти. Очевидно, что метод вычисления физических адресов в защищенном режиме значительно сложнее, чем в реальном режиме. Однако эти вычисления выполняются процессором и «прозрачны» для программиста, для которого механизм адресации выглядит неизменным. Действительно, как и в реальном режиме, команды, стек и данные адресуются с помощью соответствующего регистра сегмента (называемого селектором) и смещения.

Эти данные позволяют теперь определить максимальный размер адресного пространства, доступного для каждой программы. В самом деле, мы видели, что поле индекса какого-либо регистра селектора имеет размер 13 бит, что позволяет обратиться к 8192 ($2^{13} = 8192$) дескрипторам в каждой таблице. Зная значение бита индикатора таблицы, можно обратиться к двум таблицам, которые содержат по 8192 дескриптора. Каждый из этих дескрипторов соответствует сегменту с максимальным размером 64 Кбайт ($2^{16} = 64$ Кбайт). Таким образом, полная виртуальная память, доступная для каждой задачи составляет 8192 x 2 x 64 Кбайт, что равно 1 Гбайт (2^{13} x 2 x $2^{16} = 2^{30} = 1$ Гбайт). Пересчет адреса в виртуальной памяти в реальную оперативную память производится процессором автоматически. Это соответствует тому, что физические адреса не наблюдаемы или прозрачны для программы.

Адресация в защищенном режиме МП 80386 и старше

Защищенный режим, появившийся в МП 80286, имел не все возможности, доступные в 32-разрядных процессорах, какими являются 80386, 80486 и Pentium. На рис. 4.19 показано, как в этих МП (при работе в защищенном режиме) генерируется физический адрес памяти на основе содержимого базового и индексного регистров, а также находящегося в команде значения смещения. Для получения

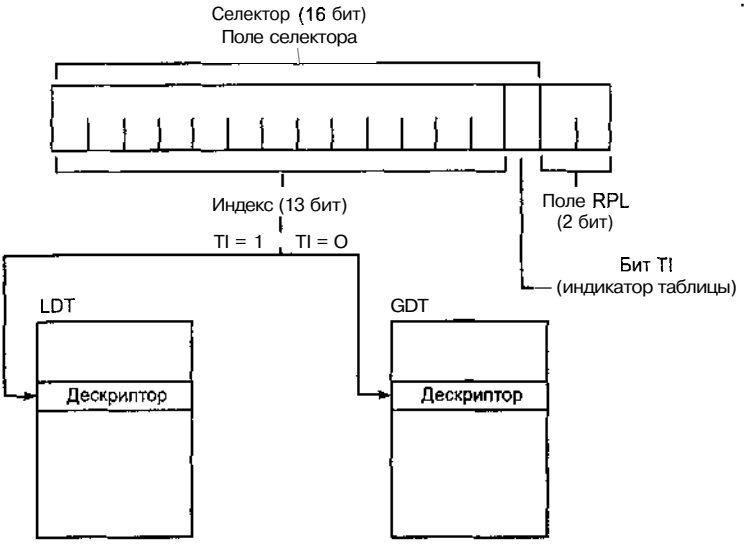


Рис. 4.19. Формирование адреса в защищенном режиме

32-разрядного исполнительного адреса значение индексного регистра умножается на коэффициент масштабирования, равный 1, 2, 4 или 8, затем результат прибавляется к содержимому базового регистра с учетом заданного в команде смещения.

Четырнадцать старших битов одного из шести сегментных регистров определяют дескриптор, используемый в качестве индекса в таблице дескрипторов, из которой извлекается 32-разрядный базовый адрес. Этот адрес прибавляется к исполнительному адресу внутри сегмента, вследствие чего получается 32-разрядный линейный адрес. Страничный блок, используя таблицу страниц, транслирует линейный адрес в 32-разрядный физический адрес.

Таблицы страниц и дескрипторов сегментов довольно велики, поэтому хранятся в основной памяти. Для обеспечения быстрой трансляции может использоваться буфер быстрого преобразования адресов TLB. В таблицах дескрипторов сегментов содержатся поля прав доступа, а также поля границ сегментов, определяющие их максимальный размер. Этими параметрами управляет операционная система. Они нужны для защиты как операционной системы, так и прикладных программ, находящихся в основной памяти. Отсюда и название данного режима работы процессора — «защищенный».

Защищенный режим предназначен для обеспечения независимости выполнения нескольких задач, что подразумевает защиту ре-

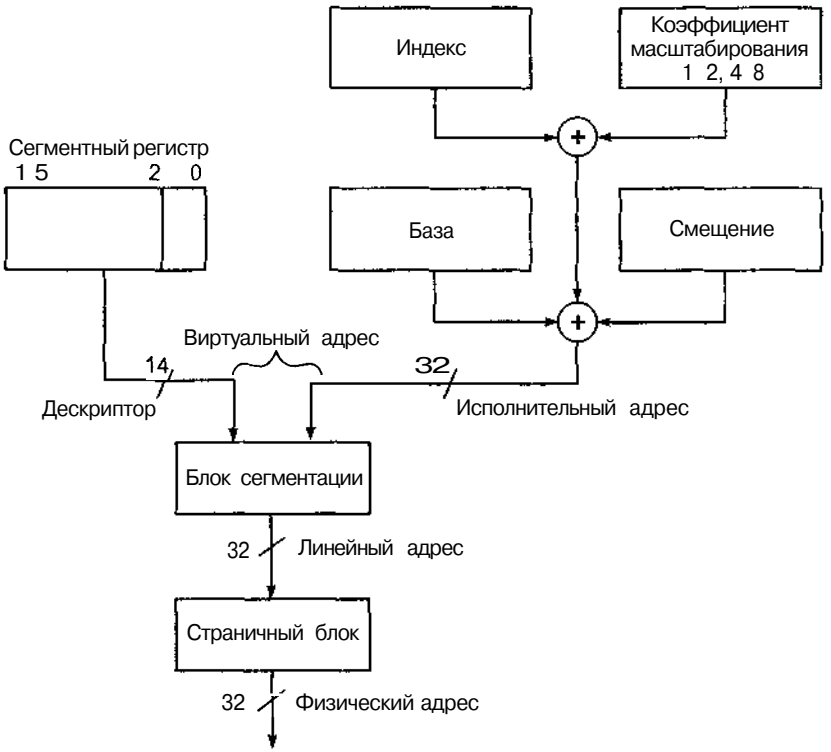


Рис. 4.20. Адресация в защищенном режиме МП 80386 и старше

сурсов одной задачи от возможного воздействия другой (под задачами подразумеваются как приложения, так и задачи операционной системы)

Основным защищаемым ресурсом является память, в которой хранятся коды, данные и различные системные таблицы (например, таблица прерываний) Защищать требуется и совместно используемую аппаратуру, обращение к которой обычно происходит через операции ввода/вывода и прерывания В защищенном режиме процессор аппаратно реализует многие функции защиты, необходимые для построения супервизора многозадачной ОС, в том числе механизм виртуальной памяти

Защита памяти основана на сегментации *Сегмент* — блок пространства памяти определенного назначения К элементам сегмента возможно обращение с помощью различных инструкций процессора, использующих разные режимы адресации для формирования ад-

реса в пределах сегмента Максимальный размер сегмента — 4 Гбайт (для процессоров 8086 и 80286 предел был 64 Кбайт) Сегменты памяти выделяет операционная система, но в реальном режиме любая задача может переопределить значение сегментных регистров, задающих положение сегмента в пространстве памяти, и «залезть» в чужую область данных или кода В защищенном режиме сегменты тоже распределяются операционной системой, но прикладная программа сможет использовать только разрешенные для нее сегменты памяти, выбирая их с помощью селекторов из предварительно сформированных таблиц *дескрипторов сегментов*

Процессор может обращаться только к тем сегментам памяти, для которых имеются дескрипторы в таблицах Механизм сегментации формирует линейный адрес по схеме, приведенной на рис 4 21

Дескрипторы выбираются с помощью 16-битных селекторов, программно загружаемых в сегментные регистры, формат селекторов приведен на рис 4 22

Индекс совместно с индикатором таблицы `TI` позволяет выбрать дескриптор из локальной (TI = 1) или глобальной (`TI` = 0)

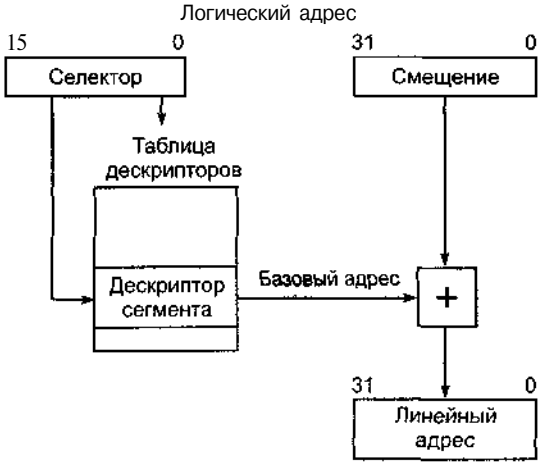


Рис. 4.21. Формирование линейного адреса

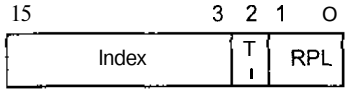


Рис. 4.22. Формат селектора адреса

таблицы дескрипторов Для неиспользуемых сегментных регистров предназначен нулевой селектор сегмента, формально адресующийся к самому первому элементу глобальной таблицы Попытка обращения к памяти по такому сегментному регистру вызовет прерывание Прерывание возникнет и при попытке загрузки нулевого селектора в регистр CS или SS, Поле RPL указывает требуемый уровеньпривилегий

Дескрипторы представляют собой 8-байтные структуры данных, используемые для определения свойств программных элементов (сегментов, вентилей и таблиц). Дескриптор определяет положение элемента в памяти, размер занимаемой им области (лимит), его назначение и характеристики защиты. Все дескрипторы хранятся в таблицах, обращение к которым поддерживается процессором аппаратно.

Защита памяти с помощью сегментации не позволяет

- использовать сегменты не по назначению (например, пытаться трактовать область данных как коды инструкций);
- нарушать права доступа (пытаться модифицировать сегмент, предназначенный только для чтения, обращаться к сегменту, не имея достаточных привилегий, и т п.);
- адресоваться к элементам, выходящим за лимит сегмента,
- изменять содержимое таблиц дескрипторов (то есть параметров сегментов), не имея достаточных привилегий.

Защищенный режим предоставляет средства переключения задач. Состояние каждой задачи (значение всех связанных с ней регистров процессора) может быть сохранено в специальном сегменте состояния задачи TSS, на который указывает селектор в регистре задачи TR. При переключении задач достаточно загрузить новый селектор в регистр задачи, и состояние текущей задачи автоматически сохранится в ее TSS, а в процессор загрузится состояние новой (возможно, ранее прерванной) задачи, и начнется (продолжится) ее выполнение.

Четырехуровневая иерархическая система привилегий (рис. 4.23) предназначена для управления использованием привилегированных инструкций и доступом к дескрипторам. Уровни привилегий нумеруются от 0 до 3, нулевой уровень соответствует максимальным (неограниченным) возможностям доступа и отводится для ядра операционной системы. Уровень 3 имеет самые ограниченные права и обычно предоставляется прикладным задачам. Систему защиты обычно изображают в виде концентрических колец, соответствующих уровням привилегий, а сами уровни привилегий иногда называют кольцами защиты. Сервисы, предоставляемые задачам, могут на-

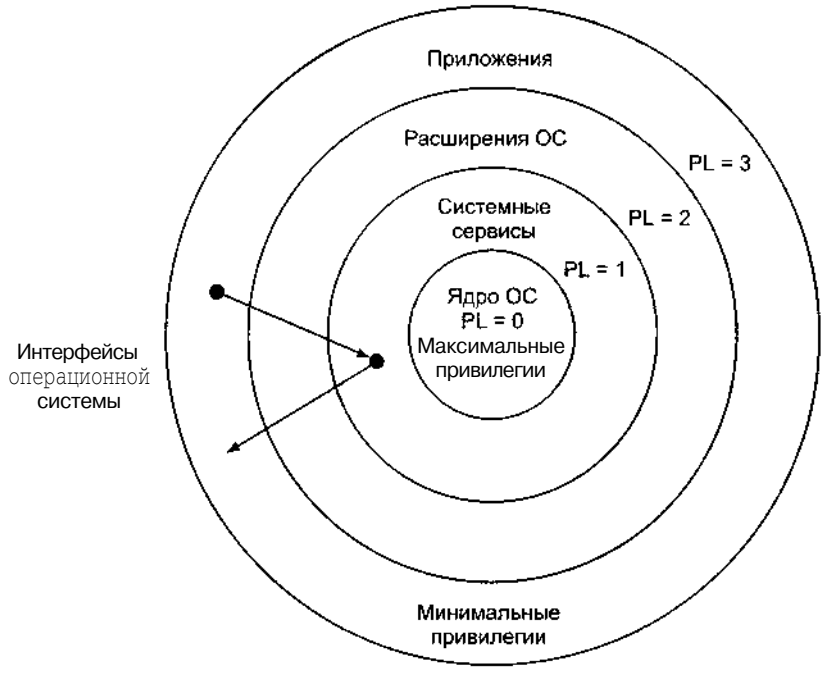


Рис. 4.23. Четырехуровневая система привилегий

ходиться в разных кольцах защиты. Передача управления между задачами контролируется *вентилями (gate),* называемыми также *шлюзами,* проверяющими правила использования уровней привилегий. Через вентили задачи могут получить доступ только к разрешенным им сервисам других сегментов.

Уровни привилегий относятся к дескрипторам, селекторам и задачам. Кроме того, в регистре флагов имеется поле привилегий ввода/вывода, с помощью которого обеспечивается управление доступом к инструкциям ввода/вывода и управлению флагом прерываний.

Дескрипторы и привилегии являются основой системы защиты: дескрипторы определяют структуры программных элементов (без которых эти элементы невозможно использовать), а привилегии определяют возможность доступа к дескрипторам и выполнения привилегированных инструкций Любое нарушение защиты приводит к возникновению специальных исключений, обрабатываемых ядром операционнойсистемы.

Механизм виртуальной памяти позволяет любой задаче использовать логическое адресное пространство размером до 64 Тбайт

(16 К сегментов по 4 Гбайт). Для этого каждый сегмент в своем дескрипторе имеет специальный бит, который указывает на присутствие данного сегмента в оперативной памяти в текущий момент времени. Неиспользуемый сегмент может быть выгружен из оперативной памяти во внешнюю (например, дисковую), о чем делается пометка в его дескрипторе. На освободившееся место из внешней памяти может восстанавливаться содержимое другого сегмента (этот процесс называется свопингом, или подкачкой), и в его дескрипторе делается пометка о присутствии в памяти. При обращении задачи к отсутствующему сегменту процессор вырабатывает соответствующее прерывание, обработчик которого заведует виртуальной памятью в операционной системе. Механизм страничной переадресации обеспечивает виртуализацию памяти, адресуемой логическим адресом, на уровне страниц фиксированного размера. После подкачки сегмента (страницы) выполнение задачи продолжается, так что виртуализация памяти для прикладных задач прозрачна (если не принимать во внимание задержку, вызванную подкачкой).

Процессор предоставляет только необходимые аппаратные средства поддержки защиты и виртуальной памяти, а их реальное использование и устойчивость работы программ и самой операционной системы защищенного режима зависят от корректности построения ОС и предусмотрительности ее разработчиков. Хорошо спроектированная операционная система защищенного режима может обеспечить устойчивость ОС даже при некорректном поведении прикладных задач.

4.5. BIOS и ее настройка

Наиболее известными фирмами-изготовителями BIOS являются AMI (American Megatrends Inc.), Award и Phoenix. Хотя функции BIOS одинаковы, но возможны отличия способов настройки, а также меню BIOS. В большинстве случаев для хранения программ BIOS используются ППЗУ. В современных системах требуются более объемные ППЗУ. Для рассмотрения вопросов, связанных с настройкой ROM BIOS, возьмем за основу программу настройки BIOS Setup фирмы AMI. Эта версия BIOS называется AMI BIOS. Настройка (конфигурирование) системной платы состоит из подстройки ПК под используемый графический режим, установки рабочей тактовой частоты, указания объема имеющейся в наличии кэш-памяти, типа встроенного сопроцессора и т. д. Существует

семь возможных вариантов настройки, из которых мы укажем только следующие

- standard-CMOS-Setup (основные установки CMOS),
- Advanced-CMOS-Setup (дополнительные установки CMOS),
- Advanced-Chipset-Setup (дополнительные установки системной платы)

Основные установки CMOS — Standard-CMOS-Setup

В процедуре Standard-CMOS-Setup в системе AMI BIOS устанавливаются: дата и текущее время, параметры жесткого диска или дисковода, изменения используемой видеокарты и характеристики клавиатуры. Данные, относящиеся к используемой оперативной памяти ПК, рассчитываются автоматически, вводятся в действие системой BIOS и не могут быть изменены (они размещаются в правом верхнем углу экрана-меню). Жесткие диски могут выбираться из имеющихся 46 типов, представленных в виде таблицы, или же браться как тип 47 (не имеющий заданных значений), данные для которого вводятся вручную.

Для начала настройки необходимо после включения ПК внимательно следить за экраном и, как появится предложение о переходе к BIOS, нажать клавишу <Del>. Вызывается программа настройки BIOS и появляется основное меню. Далее выбирается пункт Standard-CMOS-Setup, нажимается <Enter>, в результате появляется диалоговое окно программы Standard-CMOS-Setup. Если нажать <Esc>, то можно вернуться в главное меню BIOS. Если же нужно перейти к настройке, то достаточно нажать на любую клавишу (кроме <Esc>) и перейти в диалоговое окно программы настройки. Далее с использованием клавишей перемещения курсора выбирается поле ввода, в котором нужно произвести изменения. После завершения работы нажатием клавиши <Esc> происходит выход в главное меню AMI BIOS. Для заполнений новых значений необходимо выбрать пункт меню WRITE то CMOS AND EXIT. После нажатия клавиши <Y> данные запомнятся в CMOS-RAM. Далее система автоматически перезагружается с измененными или дополненными данными.

В случае неправильной конфигурации (еще раз проверяется конфигурация системы) с помощью звуковых сигналов дается оповещение об ошибках. Система выдает соответствующие сведения об ошибке, и при нажатии клавиши <F1> можно запустить программу установки Setup.

Для того чтобы повторно оказаться в диалоговом окне программы настройки Standard-CMOS, нужно выполнить следующие действия:

<Ctrl> + <Alt> + <Del>;

<Del>, вход в BIOS-Setup;

выбор Standard-CMOS-Setup, <Enter>;

<Esc> — переход в диалоговое окно.

Находимся в программе Standard-CMOS-Setup.

Хотя BIOS и обеспечивает функционирование ПК, последний может работать не на полную мощность. Поэтому она имеет возможность восстанавливать основную начальную конфигурацию (по умолчанию). Это производится в AMI BIOS запуском ПК и удержанием нажатой клавишей <Insert>. Немного погодя, ПК сообщит, что была нажата клавиша экстренной остановки, и все изменения программы Advanced-Chipset-Setup отменены. Как видите, это относится только к Advanced-Chipset-Setup.

Мы здесь не рассматриваем тонкости конфигурирования системы и не затрагиваем вопросы, связанные с конфигурированием программных средств (оптимальное согласование ОС с техническими средствами, правильное использование различных сервисных программ). Цель раздела состоит в том, чтобы показать, какую роль в этом играют средства BIOS, как запоминается информация в CMOS, как она используется при запуске системы.

Дополнительные установки CMOS — Advanced-CMOS-Setup

Программа Advanced-CMOS-Setup позволяет более тонко оптимизировать компьютер для возможного увеличения его быстродействия.

Таким образом, настройка аппаратной части приводит к созданию нужной конфигурации ПК, а настройка программной части связана с оптимальным согласованием операционной системы с техническими возможностями ПК, а также с правильным использованием различных сервисных программ. Целесообразно под рукой иметь системную (загрузочную) дискету, содержащую необходимую для запуска ПК информацию. Ею можно воспользоваться тогда, когда по каким-либо причинам ПК не загружается с винчестера.

Аппаратное конфигурирование ПК включает в себя настройку BIOS и освобождает пользователя от дальнейших забот по конфигурированию системы. При холодном старте ПК данные считываются из CMOS RAM, и, если не обнаруживаются ошибки, то работа продолжается в соответствии с настройкой BIOS Setup. Поэтому при

внесении изменений в конфигурацию ПК необходимо вводить соответствующие изменения и в настройку системы BIOS.

Диапазон памяти для ROM BIOS находится между адресами FE000H и FFFFFH и делится на соответствующие поддиапазоны (табл. 4.5).

Таблица 4.5. **Диапазоны памяти ROM BIOS**

№	Адресная область	Функция
1	FE000–FFFD9	Подпрограммы BIOS
2	FFFFO-FFFF4	Начальный адрес
3	FFFF5-FFFFC	Начальные данные BIOS
4	FFFFE-FFFFF	Идентификатор функций и идентификация фирмы-изготовителя

В области FE000—FFFD9 находятся определенные подпрограммы, которые специально подобраны для используемой системной платы. Для системы BIOS используется или 8-Кбайтовая ППЗУ типа 2764 или же 64-Кбайтовая ППЗУ типа 27512 (для AT- или EISA-систем). Обработка подпрограмм BIOS начинается с перехода на адрес FE000H. МП имеет доступ к этим программам благодаря программным прерываниям (10H ÷ 1AH). Каждое прерывание имеет доступ к определенной подпрограмме BIOS.

Область памяти объемом 384 Кбайт (между 640 Кбайт и 1 Мбайт) называется *сегментом внешних устройств* (A0000—EFFFF). В табл. 4.6 дано распределение адресов BIOS в указанном сегменте.

Таблица 4.6. **Распределение адресов BIOS в сегменте внешних устройств**

№	Шестнадцатеричный адрес	Функция
1	A0000–AFFFF	RAM графического адаптера EGA
2	BOOOO-BOFFF	RAM монохромного адаптера MGA
3	B1000-B7FFF	Зарезервировано для видеопамяти
4	B8000-BBFFF	RAM адаптера GGA
5	BC000–BFFFF	RAM адаптеров CGA или EGA
6	C0000–C3FFF	ROM адаптеров
7	C4000–C7FFF	Область для дополнительного ПЗУ
8	C8000-CCFFF	HDD и FDD
9	CD000–CFFFF	ПЗУ ввода-вывода
10	DOOOO-EFFFF	Область для дополнительного ПЗУ

AMI BIOS (другое название — Hi-Flex BIOS) в своих новых версиях имеет семь вариантов настройки:

- Standard CMOS Setup отвечает за установку стандартных встроенных аппаратных компонентов, определение оперативной памяти, а также за установку времени и даты;
- Advanced CMOS Setup обеспечивает конфигурацию различных установок при старте ПК и, кроме того, позволяет расположить в верхней части стандартной памяти системную область ROM BIOS;
- Advanced Chipset Setup служит для установки опций Chipset, что может ускорить или замедлить работу ПК;
- Autoconfiguration with BIOS default возвращает Setup стандартные значения, которые жестко «прошиты» в ROM BIOS (обычно опции выставляются так, чтобы начальным устройством загрузки ПК являлся дисковод A:);
- Autoconfiguration with power-on defaults восстанавливаются установки, которые имели место при последнем включении ПК;
- Change password — замена пароля (с помощью пароля можно защитить CMOS-Setup от нежелательного доступа);
- Autodetect hard disk служит для автоматического опознавания BIOS винчестера и установки его параметров (в старых версиях была опция для форматирования жесткого диска);
- Write to CMOS and Exit подтверждает установленные (а также измененные) значения параметров, производит выход из Setup и перезагружает ПК (с новыми значениями);
- Do not write to CMOS and Exit - противоположный последнему вариант — новые установки игнорируются, и ПК стартует со старыми установками (тот же эффект можно получить при нажатии клавиши <Esc>).

Имеется опция Power management Setup, которая дает возможность с помощью содержащихся в ней установок заставить ПК через определенный промежуток времени перейти в режим stand by Mode (ослабленный режим: гасится экран монитора, деактивизируется винчестер и осуществляется переход к более низкой тактовой частоте).

Текущее значение соответствующего параметра изменяется клавишами <Page Up> и <Page Down>.

Пункт меню: Typematic Rate Programming (программирование параметров автоповтора). Назначение: если вы включаете функцию Typematic Rate Programming, то в следующих пунктах меню

можете устанавливать параметры автоповтора клавиатуры по умолчанию. Опции: Enabled/Disabled (Разрешено/Запрещено).

Пункт меню: Typematic Rate Delay (задержка автоповтора). Назначение: этот параметр реализуется только при включенном параметре Typematic Rate Programming. Параметр Typematic Rate Delay определяет время, в течение которого нужно удерживать клавишу нажатой, пока не включится автоматическая имитация многократного нажатия этой клавиши. Опции: параметр задержки задается в миллисекундах. Значение по умолчанию составляет 500 мс. Не устанавливайте значение параметра слишком малым во избежание ненужных повторов.

Пункт меню: Typematic Rate (Chars/Sec — частота автоповтора). Назначение: при включенном параметре Typematic Rate Programming можно задавать частоту, с которой происходит автоматическое повторение нажатия клавиши.

Пункт меню: Above 1 MB Memory Test (тест памяти свыше 1 Мбайт). Назначение: если эта функция отключена, то после включения ПК программа самопроверки POST проверяет ОЗУ только в пределах до 1 Мбайт. Остальная часть ОЗУ не проверяется.

Пункт меню: Memory Test Tick Sound. Назначение: с помощью этой функции можно включить или выключить «тиканье», слышимое при проверке памяти после включения компьютера.

Пункт меню: Memory Parity Error Check (тест памяти по четности). Назначение: если системная плата поддерживает эту функцию, то с ее помощью можно включать или выключать проверку по четности. Процедура проверки сводится к определению контрольной суммы для чипов или модулей RAM (ОЗУ). Опции: Enabled/Disabled (Разрешено/Запрещено). Для защиты данных и контроля системы эта функция должна быть включена.

Пункт меню: Hard Disk Type 47 Data Area (область хранения данных НЖМД типа 47). Назначение: в случае, если в ПК установлен НЖМД типа 47, а функция ROM-BIOS-Shadow выключена, параметры жесткого диска типа 47 хранятся либо в области 0:300, либо в области ОЗУ для DOS, которая в результате этого уменьшается на 1 Кбайт и составит уже 639 Кбайт. Опции: 0:300 или DOS 1 Кбайт.

Пункт меню: Wait for <F1> if any Error (ждать нажатия <F1> в случае ошибки). Назначение: если эта функция включена, то AMI-BIOS требует нажатия клавиши <F1> при обнаружении ошибки в процессе самотестирования.

Пункт меню: System Boot Up Num. Lock. Назначение: если включить опцию On, то при запуске ПК включается и Num Lock.

Пункт меню: Weitek Processor. Назначение: с помощью этой функции определяется, встроен (Present) или нет (Absent) сопроцессор Weitek.

Пункт меню: Internal Cache Memory (внутренняя кэш-память). Назначение: с помощью этой функции можно включить или выключить внутреннюю кэш-память процессора 80486. Опции: Enabled/Disabled (Разрешено/Запрещено). Установка Disabled приводит к резкому снижению производительности системы.

Пункт меню: External Cache Memory (внешняя кэш-память). Назначение: если на системной плате имеется внешняя кэш-память, то с помощью этой функции ее можно включать или выключать. Опции: Enabled/Disabled (Разрешено/Запрещено). См. пункт Internal Cache Memory.

Пункт меню: Fast Gate A20 Option. Назначение: с помощью этой функции можно использовать ускоренный, совместимый с PS/2 метод доступа к Extended Memory. Для OS/2 и Других программ эта функция повышает скорость работы. Опции: Enabled/Disabled (Разрешено/Запрещено).

Пункт меню: Turbo Switch Function. Назначение: с помощью этой функции можно включать или отключать действие кнопки Turbo.

Пункт меню: Password Checking Option (условия проверки пароля). Назначение: с помощью этой функции можно включать запрос пароля. При этом имеются различные степени зашиты. Так, опция Always требует постоянного ввода пароля при начальной загрузке, а опция Setup — только для доступа к программе настройки. Опции: Disabled (запрещено для доступа к программе настройки Setup); Always (всегда). Функция Password не имеет в BIOS надежной защиты. Ее можно выключить путем отключения батареи или изменить с помощью различных программ.

Пункт меню Video ROM Shadow & Adapter ROM Shadow. Назначение: благодаря включению функций Shadow-ROM вы копируете содержимое медленных блоков ПЗУ в более быстрое ОЗУ. Соответствующие подпрограммы BIOS затем выполняются из ОЗУ. Конечно, для использования этой функции необходимо иметь представление об адресе и объеме перегружаемого блока ROM. Опции Enabled/Disabled (Разрешено/Запрещено). Различные SCSI-контроллеры заполняют части области ROM и ни в коем случае не могут быть скопированы.

Необходимо запомнить, что в некоторых случаях (например, при замене графической платы на VGA или EGA) должна быть переставлена перемычка на джемпере (при этом нужно руководство-

ваться справочником, в котором описывается соответствующая системная плата).

Оптимизация стартовых файлов CONFIG.SYS и AUTOEXEC.BAT, относящихся к ОС, в основном состоит из организации памяти и установки необходимых драйверов. Начиная с DOS 6.x более важным из них стал CONFIG.SYS. Тем более, что почти все команды AUTOEXEC.BAT могут быть интегрированы в CONFIG.SYS.

Контрольные вопросы

1. Перечислите основные компоненты ПЭВМ.
2. Что находится внутри системного блока?
3. Что такое системная плата и для чего она нужна?
4. Что такое системная шина и для чего она нужна?
5. Перечислите типы дополнительных интегральных микросхем, которые входят в состав ПЭВМ.
6. Какие виды устройств ОП вы знаете?
7. Для чего нужна внешняя память?
8. Из чего состоит интерфейсная часть МП?
9. Перечислите основные классы регистров МП 80x86.
10. Для чего нужен стек?
11. Поясните принцип действия системы прерываний.
12. Что такое система команд компьютера?
13. Опишите функции памяти и функции процессора.
14. Назовите основные части процессора. Каково их назначение?
15. Что понимается под структурой компьютера? Какой уровень детализации описания компьютера может она обеспечить?
16. Что собой представляет шина компьютера? Каковы функции общей шины (магистрали)?
17. Какую функцию выполняют контроллеры?
18. Как характер решаемых задач связан с архитектурой компьютера?
19. Что такое центральный процессор?
20. Какие основные компоненты содержат в себе современные микропроцессоры?
21. Как конструктивно выполнены современные микропроцессоры?
22. Что собой представляет модуль памяти типа SIMM? Какие другие типы модулей памяти Вы знаете?
23. Каково назначение кэш-памяти? Каким образом она реализуется?
24. Что такое специальная память? Характеризуйте ее основные виды.
25. Что такое BIOS и какова ее роль?
26. Какие характеристики компьютера стандартизируются для реализации принципа открытой архитектуры?
27. Что такое аппаратный интерфейс?

28 Каково назначение контроллеров и адаптеров? В чем заключается разница между контроллером и адаптером?
29 Что такое порты устройств? Охарактеризуйте основные виды портов
30 Перечислите функции процессора
31 Перечислите и прокомментируйте основные параметры процессора
32 Какие типы систем команд процессора вы знаете?
33 Зачем нужны регистры?
34 Зачем нужны прерывания?
35 Перечислите настройки BIOS

Заключение

Вычислительная техника не сразу достигла необходимого уровня В ее развитии отмечают предысторию и четыре поколения ЭВМ Предыстория начинается в глубокой древности с различных приспособлений для счета (абак, счеты), а первая счетная машина появилась лишь в 1642 г Ее изобрел французский математик Паскаль Построенная на основе зубчатых колес, она могла суммировать десятичные числа Все четыре арифметических действия выполняла машина, созданная в 1673 г немецким математиком Лейбницем Она стала прототипом арифмометров, использовавшихся с 1820 г до 60-х гг XX в Впервые идея программно-управляемой счетной машины, имеющей арифметическое устройство, устройства управления, ввода и печати (хотя и использующей десятичную систему счисления), была выдвинута в 1822 г английским математиком Бэббиджем Проект опережал технические возможности своего времени и не был реализован Лишь в 40-х гг XX в удалось создать программируемую счетную машину, причем на основе электромеханических реле, которые могут пребывать в одном из устойчивых состояний, «включено» и «выключено» Это технически проще, чем пытаться реализовать десять различных состояний, опирающихся на обработку информации на основе десятичной системы счисления С каждым новым поколением ЭВМ увеличиваются быстродействие и надежность их работы при уменьшении стоимости и размеров, совершенствовались устройства ввода и вывода информации В соответствии с трактовкой компьютера как технической модели информационной функции человека-устройства ввода приближаются к естественному для человека восприятию информации (зрительному, звуковому) и, следовательно, операции по вводу в компьютер становятся все более удобными для человека

ЭВМ характеризуются множеством показателей Это — система команд (общее количество выполняемых операций), быстродействие центрального процессора, емкости оперативной и внешней памятей, количество и номенклатура одновременно подсоединяемых

периферийных устройств, потребляемая электроэнергия и др. На всех этапах истории развития ЭВМ создатели стремились все больше повысить их производительность, а пользователи мечтали (и мечтают) получить в свое распоряжение еще более мощные и функционально более богатые вычислительные средства. Появились разного типа вычислительные системы, разнообразные многопроцессорные архитектуры, в архитектуре стало применяться множество различных функциональных элементов и узлов (например, таких, как кэш-память) и т. д. Если же рассмотреть историю развития информационных технологий, то с появлением ЭВМ, параллельно с развитием последних, также успешно развивались существующие и появлялись новые ИТ. С появлением микропроцессоров в истории вычислительной техники началась эпоха микроЭВМ, а с развитием микроэлектронной технологии, по мере возрастания степени интеграции и расширения функциональных возможностей микроэлектронных схем, стали появляться новые микропроцессоры и на их базе более совершенные ВМ. В конечном счете этот период ознаменовался широчайшим распространением во всех сферах человеческой деятельности персональных ЭВМ (ПЭВМ) и развитием технологии автоформализации знаний.

Увеличение емкости системы памяти и повышение общей производительности ПК привели к более широкому распространению баз данных в различных автоматизированных системах управления. Это и ряд других факторов привели к тому, что стали более бережно относиться и ценить различные информационные ресурсы. Появилась необходимость предоставить доступ к этим ресурсам многим пользователям. Возникли локальные вычислительные сети (ЛВС), которые дали возможность в некоторой степени решить эту задачу, а также повысить эффективность использования дорогостоящих аппаратных средств (например, принтеров, дисков и др.).

Расширялись функциональные возможности новых ПК и одновременно появлялись новые задачи, требующие еще более мощные накопители и процессоры и т. д. Объединение технологии сбора, хранения, передачи и обработки информации на компьютерах с техникой связи привели к идее создания более мощных и широких глобальных вычислительным сетей. Разрабатывались новые ИТ, ставящие свои специфические требования перед требуемой вычислительной техникой. Этот лавинообразный циклический процесс прогресса продолжается непрерывно. Так, ЭВМ, объединенные в сеть, разделяются на основные (абонентские, клиентские) и вспомогательные (серверы). Последние служат для преобразования и пе-

редачи информации от одной ЭВМ к другой и к коммуникационным ЭВМ (Host-ЭВМ) по каналам связи.

В качестве ЭВМ клиента может быть использована любая ЭВМ (клиент-приложение, посылающее запрос к серверу).

К сетевому серверу по качеству и мощности предъявляются повышенные требования. Эта ЭВМ выполняет функции по обслуживанию клиента, распределению ресурсов системы, а также поддерживает выполнение функций сетевой ОС.

К серверу баз данных предъявляются немного иные требования. Он обеспечивает обработку запросов в многопользовательских системах, является средством решения сетевых задач и т. д.

Коммуникационная ЭВМ устанавливается в узлах сети и решает коммуникационные задачи. Поэтому перед этими ЭВМ ставятся другие требования.

В общем виде и очень приближенно наблюдаемые в настоящее время тенденции развития вычислительной техники можно обозначить следующим образом:

- в результате дальнейшего развития микроэлектронных технологий все большее увеличение степени интеграции микроэлектронных схем, повышение вычислительной мощности и расширение функциональных возможностей микропроцессоров (плотность размещения транзисторов более 3 млн на 1 $см^2$, тактовая частота уже превосходит величину 3000 МГц);
- разработка новых архитектурных решений в области однопроцессорных и многопроцессорных вычислительных систем;
- создание различных функционально расширенных специализированных больших и сверхбольших интегральных схем — СБИС. Например, корпорация Intel предложила программируемую логическую интегральную схему (ПЛИС) с уникальной особенностью — возможностью проектирования и, если надо, моделирования и модификации цифровой схемы (со стороны разработчика, на своем рабочем месте, за несколько часов) специализированного СБИС, заменяющего 50—80 стандартных логических схем;
- разработка принципиально новых элементов памяти и различных функциональных схем элементной базы ЭВМ;
- расширение набора функций, выполняемых в одном компьютере (обработка алфавитно-цифровой информации, воспроизведение и запись звука, создание специальных эффектов, прием телепередач, запись кадров и их обработка, воспроизведение аналоговых и цифровых сигналов, различные режимы работы в компьютерных сетях — вот неполный перечень вы-

полняемых функций) Все это требует существенного расширения номенклатуры и повышения мощности базовых блоков и узлов компьютера,

- на основе отмеченных и большого количествадрvгих разработок, создание широкой номенклатуры более производительных, многофункциональных и отличающихся своими высокими техническими и эксплуатационными характеристиками устройств и блоков ЭВМ, а также автономных средств вычислительной и организационной техники (например, компания Seagate Technology создала жесткие диски семейства с интерфейсом UltraDMA емкостью 80 Гбайт и со скоростью вращения шпинделя 7200 об /мин)

Естественно, что этот перечень можно продолжать очень долго, так как данная область науки и техники развивается интенсивно и многосторонне

Разрабатываются новые офисные и мультимедийные ПК, высокопроизводительные серверы, профессиональные рабочие станции, суперкомпьютеры, графические рабочие станции, презентационные ноутбуки и много других средств

Создаются средства защиты информации, такие, как программ но-аппаратный комплекс, обеспечивающий шифрование данных с гарантированной стойкостью и аутентификацию с проверкой целостности передаваемой информации, различные системы защиты информации, устройства криптографической защиты данных и др

С появлением качественно новых вычислительных средств, средств оргтехники и управления проводится большая работа по совершенствованию существующих и разработке совершенно новых ИТ В качестве примера можно отметить персональные информационные системы, связанную с ними GPS-технологию (Global Positioning System — Система Глобального Позиционирования) Эта система обеспечивает информацией, приспособленной для некоторого индивидуума и доставляющейся непосредственно ему с помощью портативных персональных информационных устройств (PID — Personal Information Device) В качестве PID могут использоваться персональные цифровые ассистенты — карманные или портативные ПК Они могут как переноситься, так и перемещаться в автомобиле, снабжены средствами беспроводного сетевого соединения и сетевыми портами Каждый портативный ПК оснащен GPS-картой GPS-технология основана на спутниковых системах, генерирующих и посылающих абонентам специальные сообщения, позволяющие с высокой точностью определять их географическое местонахонаждение GPS-карта принимает и интерпретирует спут-

никовые сообщения (снабжена специальным программным обеспечением) Таких примеров можно показать очень много

Таким образом, можно утверждать, что развитие компьютерной и связанной с неи другой техники (например, сетевой техники и техники связи), а также различных информационных технологий происходит непрерывно, они тесно взаимосвязаны и все время взаимно стимулируют процессы развития

Список литературы

1. Computing & Multimedia. Словарь. М.: Внешсигма, 1996.

2. Dictionary of Computing (Data Communication, Hardware and Software Basics, Digital Electronic) Ed. by Frank J. Galland, John Wiley & Sons, Datology Press Ltd, Windsor, England, 1983.

3. *Айден К., Колесниченко О.* и др. Аппаратные средства PC. 2-е изд., перераб. и доп. СПб.: BHV—Санкт-Петербург, 1998.

4. *Айден К., Фибельман Х., Драмер М.* Аппаратные средства IBM PC. СПб.: BHV—Санкт-Петербург, 1996.

5. *Алферова З. В.* Теория алгоритмов. М.: Статистика, 1973.

6. *Аскеров Т. М.* ЭВМ и программное обеспечение. Часть I. Технические средства ЭВМ: Учеб. пособие / Под общей ред. К. И. Курбакова. М.: Изд-во Рос. экон. акад. им. Г. В. Плеханова, 1999.

7. *Аскеров Т. М., Кашканов А. Ю.* История ЭВМ и ВТ: Учеб. пособие / Под общей ред. К. И. Курбакова. М.: Изд-во Рос. экон. акад. им. Г. В. Плеханова, 2004.

8. *Ахметов К.* Курс молодого бойца. М.: Комьпютер-пресс, 1996.

9. *Боглаев Ю. П.* Вычислительная математика и программирование. М.: Высшая школа, 1990.

10. *Болски М. И.* Язык программирования Си. Справочник: Пер. с англ. М.: Радио и связь, 1988.

11. *Бордовский Г. А.* Информатика в понятиях и терминах. М.: Просвещение, 1991.

12. *Борзенко А.* и др. Мультимедиа для всех. М.: Компьютер-Пресс, 1996.

13. *Ван Тассел Д.* Стиль, разработка, эффективность, отладка и испытание программ. М.: Мир, 1981.

14. *Голицына О. Л., Максимов Н. В., Попов И. И.* Базы данных: Учеб пособие. М.: ФОРУМ: ИНФРА-М, 2003.

15. *Голицына О. Л., Попов И. И.* Основы алгоритмизации и программирования: Учеб. пособие. М.: ФОРУМ: ИНФРА-М, 2002.

16. *Гусева А. И.* Технология межсетевых взаимодействий. NetWare — Unix —Windows — Internet. М.: Диалог-МИФИ, 1997.

17. *Дютер Р. С., Полунов Ю. Л.* От абака до компьютера. М.: Знание, 1975. (Жизнь замечательных идей).

18. *Ефремова О., Шафрин Ю.* Практикум по компьютерной технологии. М.: ABF, 1997

19. *Зотов В. В.* и др. Терминологический словарь по автоматике, информатике и вычислительной технике. М.: Высшая школа, 1989

20. *Каган Б. М.* Электронные вычислительные машины и системы. М.: Энергоиздат, 1991.

21. *Казаков С. И.* Основы сетевых технологий. 1998.

22. *Каймин В. А.* Информатика: Учебник. М.: ИНФРА-М, 2000.

23. *Кирмайер М.* Мультимедиа. С.-П, 1994.

24. *Курбаков К. И.* Информационный ресурс и национальная система баз данных и баз знаний высшей школы России // Проблемы информатизации. 1993. № 3—4. С. 73—89.

25. Основы информатики (учебное пособие для абитуриентов экономических вузов) / К. И. Курбатов, Т. Л. Партыка, И. И. Попов, В. П. Романов. М.: Экзамен, 2004.

26. *Максимов Н. В., Попов И. И.* Компьютерные сети: Учеб. пособие. М.: ФОРУМ: ИНФРА-М, 2003.

27. *Малиновский Б. Н.* История вычислительной техники в лицах. Киев: Фирма КИТ; ПТОО А.С.К., 1995.

28. *Мельников В.* Зашита информации в компьютерных системах. М.: Финансы и статистика; Электроинформ, 1997.

29. Методы, системы, стандарты // Мир связи и информации. Connect. 1996. № 12. С. 54—55.

30. *Нортон П.* Программно-аппаратная организация IBM PC. М., 1991.

31. *Нортон П., Сандлер К., Батпей Т.* Персональный компьютер изнутри. М.: Бином, 1995.

32. *Олифер В. Г., Олифер Н. А.* Компьютерные сети. Принципы, технологии, протоколы. СПб.: Издательство Питер, 2000.

33. *Партыка Т. Л., Попов И. И.* Информационная безопасность: Учеб. пособие. М.: ФОРУМ: ИНФРА-М, 2002.

34. *Партыка Т. Л., Попов И. И.* Операционные системы, среды и оболочки. М.: ФОРУМ: ИНФРА-М, 2003.

35. *Петроченков А. А.* Компьютер и периферия. М., 1995.

36. *Попов И. И.* Автоматизированные информационные системы (по областям применения): Учеб. пособие / Под общей ред. К. И. Курбакова М.: Изд-во Рос. экон. акад., 1999.

37. *Попов И. И.* Информационные ресурсы и системы: реализация, моделирование, управление. М.: ТПК Альянс, 1996.

38. *Попов И. И.* Информатика и информационные системы (программа курса). М.: РГГУ, 1998.

39. *Попов И. И.* Моделирование и оптимизация автоматизированных документальных информационных систем (учебное пособие). М.: РГГУ, 1996.

40. *Попов И. И., Храмцов П. Б., Максимов Н. В.* Введение в сетевые информационные ресурсы и технологии (учебное пособие). М.: РГГУ, 2001.

41. *Попов И. И., Храмцов П. Б.* Мировые информационные ресурсы и сети (методы доступа к ним). Учебник / Под ред. К. И. Курбакова. М.: Изд-во Рос. экон. акад., 1998.

42. *Скотт Мюллер.* Модернизация и ремонт персональных компьютеров. М.: Бином, 1997.

43. *Смирнов Ю. П.* История вычислительной техники: Становление и развитие: Учеб. пособие. Изд-во Чуваш. ун-та, 1994.

44. *Стерлинг, Дональд Дж.* Волоконная оптика. Техническое руководство. М.: ЛОРИ, 1998.

45. Толковый словарь по вычислительным системам / Под ред. В. Иллингуотера и др. М.: Машиностроение, 1990.

46. Федеральный закон «Об информации, информатизации и защите информации». Принят Государственной думой 25 января 1995 г.

47. *Фигурнов В. Э.* IBM PC для пользователя. М.: ИНФРА-М, 1995.

48. *Фролов А. В. Фролов Г. В.* Локальные сети персональных компьютеров. Монтаж сети, установка программного обеспечения. Т. 7. М.: Диалог-Мифи, 1994.

49. *Фролов А. В., Орлов Г. В.* Модемы и факс-модемы. М.: Диалог-МИФИ, 1995.

50. *Фролов А. В., Фролов Г. В.* Программирование модемов. М.: Диалог-МИФИ, 1993.

51. *Якубайтис Э. А.* Информационные сети и системы. Справочная книга. М.: Финансы и статистика, 1996.

Глоссарий терминов и сокращений (русский язык)

Абак (греч. abax — стол, счетная доска) — счетный прибор у древних греков и римлян в виде доски, разделенной на полосы. При счете камешки или палочки передвигались по полосам, которые определяли разряды чисел. Утратил свое значение с изобретением счетов.

Абстрактная вычислительная машина — теоретическое построение, с помощью которого дается математическое определение алгоритма. Известны машины Тьюринга, Поста, Колмогорова и др.

Автоматизированное рабочее место (АРМ, рабочая станция) — место оператора, оборудованное средствами, необходимыми для выполнения определенных функций. В системах обработки данных и учреждениях обычно АРМ — это дисплей с клавиатурой, но может использоваться также и принтер, внешние ЗУ и др.

Автономный резким — режим работы нескольких систем независимо друг от друга, хотя физически они соединены между собой Часто называют режимом «off-line» (вне системы). Противоположен диалоговому режиму, при котором системы взаимодействуют между собой.

Ада — универсальный язык программирования процедурного типа, достаточно мощный, предназначенный для разработки разнообразных систем управления, в том числе встроенных (бортовых). Разработан по заказу Министерства обороны США в 1980 г. фирмой Honeywell и ее французским филиалом Cii-Bull. Назван в честь Ады Лавлейс, дочери английского поэта Дж. Байрона, участвовавшей в разработке программ для первой программно-управляемой вычислительной машины — аналитической машины Бэббиджа.

Адаптер *(лат. adaptare* — прилаживать, короче говоря, — «приспособление») — устройство сопряжения центрального процессора и периферийных устройств компьютера; кроме этого, иногда осуществляет функции управления периферийным устройством Обычно выполнен в виде микросхемы и помещен на системную плату, может быть представлен отдельной платой. Иногда называется *картой* или *контроллером.*

Адаптер графический — устройство, управляющее дисплеем и обеспечивающее вывод графических изображений. Определяет разрешающую способность дисплея (количество точек на единицу площади экрана), количество цветов. Обычно включает в себя видеопамять и

средства преобразования данных, находящихся в видеопамяти, в видеосигнал. Известны пять основных типов адаптеров: (1) MGA (Monochrome Graphics Adapter) — монохромный графический адаптер, иногда называемый — Hercules Graphics Adapter; (2) CGA (Color Graphics Adapter) — цветной графический адаптер; (3) EGA (Enhanced Graphics Adapter) — улучшенный графический адаптер, (4) VGA (Video Graphics Array) —видеографическая матрица; (5) SVGA (Super Video Graphics Array) — видеографическая матрица высокого класса.

Адаптер локальной сети (Адаптер сетевой, Networking Adapter) — адаптер для подключения компьютера к локальной сети компьютеров. Например, для подключения персонального компьютера к сети Ethernet используется адаптер NE-2000

*Адрес ETHERNET (Ethernet **Adress)*** — система описания компьютера и порта передачи данных в локальной сети Ethernet.

Адрес — номер конкретного байта оперативной памяти компьютера.

Аккумулятор — (1) устройство, вырабатывающее электричество путем преобразования химической энергии в электрическую. Имеется возможность многократной перезарядки. Используются в настольных компьютерах, как вспомогательное энергопитание, в компьютерах переносного типа, как основное, кроме этого — в устройствах бесперебойного питания; (2) ячейка памяти, используемая для хранения результатов вычисления; обычно так называют один из регистров в арифметико-логическом устройстве процессора.

Аксессуар (фр. accesoire — принадлежность) — элемент компьютера или программной среды, который может быть использован только вместе со всей системой, но приобрести и установить его можно отдельно. Например, к мультимедийным аксессуарам компьютера относятся: компакт-диски, звуковые адаптеры и колонки и пр.

Активное устройство — физическое или логическое устройство, с которым работает система в данный момент времени. Активным может быть также и некоторая программа, файл или база данных. Это означает, что в данный момент они готовы для ввода/вывода данных. Например, приглашение в MS-DOS — `b:\>` означает, что активным в данный момент является дисковод B.

Алгебра логики — раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Алгоритм — понятное и точное предписание (указание) исполнителю совершить определенную последовательность действий, направленных на достижение указанной цели или решение поставленной задачи (приводящую от исходных данных к искомому результату).

Алфавит — фиксированный для данного языка набор основных символов, т. е. «букв алфавита», из которых должен состоять любой текст на этом языке. Никакие другие символы в тексте не допускаются.

Аналоговая вычислительная машина — вычислительная машина, которая оперирует данными, представленными в аналоговом виде. Аналоговые вычислительные машины практически всегда жестко специализированы. Отличаются от цифровых большей скоростью выполнения операций и простотой программирования Предполагается, что аналоговые вычислительные машины получат свое дальнейшее развитие при создании нейрокомпьютера.

Аналого-цифровой преобразователь — устройство, преобразующее аналоговый сигнал в цифровой и обратно. Например, для передачи данных по цифровой телефонной сети с помощью модема, между модемом и цифровым телефонным каналом ставится аналого-цифровой адаптер.

Аппаратура передачи данных (АПД) — аппаратура, предназначенная для обеспечения возможности вхождения оконечного оборудования данных (ООД) в канал связи. АПД обеспечивает интерфейс ООД с сетью передачи данных. Модем является аппаратурой передачи данных.

Арифметико-логическое устройство (АЛУ) — часть процессора, которая производит выполнение операций, предусмотренных данным компьютером.

Арифметические операции — четыре действия арифметики (+, -, *, /) и операция получения остатка от деления (%) образуют группу арифметических операций. Их выполнение не имеет каких-либо особенностей, кроме как преобразование типов переменных при их несовпадении. Если в одном выражении встречаются переменные разных типов, как правило, осуществляется преобразование (приведение) типов.

Архитектура двойной независимой шины (DIB, Dual independent Bus) — архитектура построения процессора, при которой данные передаются по двум шинам независимо друг от друга, одновременно и параллельно. Наличие двух независимых шин дает возможность процессору получать доступ к данным, передающимся по любой из шин одновременно и параллельно, в отличие от последовательного механизма, характерного для систем с одной шиной.

Архитектура открытая — архитектура, разработанная фирмой IBM для персональных компьютеров Основные признаки: наличие общей информационной шины, к которой возможно подключение различных дополнительных устройств через разъемы расширения; модульное построение компьютера; совместимость всех новых устройств и программных средств с предыдущими версиями по принципу

«сверху-вниз», т. е последующие разработки должны поддерживать более ранние.

Архитектура фон Неймана — архитектура компьютера, имеющего одно арифметико-логическое устройство, через которое проходит поток данных, и одно устройство управления, через которое проходит поток команд.

Архитектура ЭВМ — общее описание структуры и функции ЭВМ на уровне, достаточном для понимания принципов работы и системы команд ЭВМ. Архитектура не включает в себя описание деталей технического и физического устройства компьютера. Основные компоненты архитектуры ЭВМ: процессор, внутренняя (основная) память, внешняя память, устройства ввода, устройства вывода.

Асинхронная передача данных — способ передачи и метод извлечения данных из непрерывного потока сообщений, при которых передающая сторона в каждое данное вводит стартовый и стоповый биты, указывающие, где данное начинается и где кончается.

Аудиоадаптер (Sound Blaster, звуковая плата) — специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

Базовая система ввода/вывода (BIOS, англ. — Basic Input/Output System) — программы, предназначенные для выполнения следующих функций: тестирование основных устройств компьютера; распознавание типов устройств, установленных в компьютере; вызов блока начальной загрузки операционной системы; обслуживание системных прерываний. В большинстве компьютеров BIOS записывается изготовителем компьютера в постоянное запоминающее устройство и пользователь не имеет средств изменять ее. В настоящее время выпускаются компьютеры, у которых BIOS записывают во флэш-память, и у пользователя появляется возможность изменять BIOS по мере необходимости. Некоторые источники считают BIOS частью операционной системы.

Байт — машинное слово минимальной размерности, адресуемое в процессе обработки данных. Размерность байта (8 бит) принята не только для представления данных в большинстве компьютеров, но и в качестве стандарта для хранения данных на внешних носителях, для передачи данных по каналам связи, для представления текстовой информации. Кроме того, байт, размерность всех форм представления данных, устанавливается кратной байту. При этом машинное слово считается разбитым на байты, которые нумеруются начиная с младших разрядов Самые распространенные формы представления данных, использующие одно машинное слово — целое число со знаком и без него. Наиболее простая — целое положи-

тельное число без знака. В нем каждый разряд машинного слова имеет вес, в 2 раза больший, чем вес соседнего правого, т. е. 1, 2, 4, 8, 16 и т. д. или последовательные степени 2. Тогда значение числа в машинном слове равно сумме произведений значений разрядов на их веса. Одним байтом кодируется символ в ASCII, EBCDIC. Для измерения объема памяти применяются также такие единицы, как Кбайт, Мбайт, Гбайт и Тбайт

Библиотека стандартных подпрограмм — совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм.

Бит (англ. Binary dIgiT — двоичная единица) — единица измерения количества информации, равная количеству информации, содержащемуся в опыте, имеющем два равновероятных исхода. Это наименьшая единица информации в цифровом компьютере, принимающая значения «О» или «1».

Блок-схема — (1) графическое представление алгоритма, повышающее наглядность алгоритма. Составление блок-схем особенно полезно начинающим программистам; (2) графическое представление состава технических средств, или структуры системы.

Бод (baud), бит/с (bps) — единица измерения скорости передачи данных.

Булева алгебра — раздел математической логики, изучающий высказывания и операции над ними Частный случай алгебры логики. Под высказываниями понимается любое утверждение, которое бывает либо истинным, либо ложным. Над высказываниями возможны операции: И (конъюнкция &, л); ИЛИ (дизъюнкция, v); «если..., то» (импликация, →); двусторонняя импликация (эквивалентность, ~); НЕ (отрицание, ¬). Введено понятие функций, которые могут задаваться таблицами (таблицы истинности). Логические операции подчиняются законам: коммутативности, ассоциативности, поглощения, дистрибутивности, противоречия и исключенного третьего.

Буфер — (1) дополнительная память для временного хранения данных. Буфер предназначен для компенсации более низкой скорости работы выходного устройства, по сравнению со скоростями работы процессора и оперативной памяти. Буфер многих лазерных принтеров, например, составляет от 1 до нескольких Мбайт; (2) часть оперативной памяти, используемая системой для временного хранения данных при выполнении операций копирования и переноса.

***Быстродействие** накопителя* — скорость чтения/записи данных в накопителе. Определяется двумя параметрами: средним временем доступа и скоростью передачи данных.

Быстродействие процессора — скорость выполнения операций процессором. Так как скорость выполнения отдельных операций у процессора разная, то за скорость работы всего процессора принимают либо скорость выполнения команд «регистр—регистр», либо скорость выполнения команд над числами с плавающей запятой. Последняя имеет специальное название — флопс (Flops — floating point operations per second). Обобщенным показателем скорости процессора является тактовая частота и тип процессора. Например, при тактовой частоте 66 МГц у процессора 486-DX скорость 54 млн команд/с; у Pentium при той же частоте — 112 млн команд/с.

БЭСМ — быстродействующая электронно-счетная машина семейства машин общего назначения. Разработка БЭСМ начата С. А. Лебедевым в 1949 г. в Киеве и закончена в 1952 г. в Москве в Институте точной механики и вычислительной техники АН СССР. Первая модель БЭСМ создана в 1953 г., имела память 2048 ячеек и скорость 8 тыс. операций в секунду. Серийный выпуск начался в 1954 г. с БЭСМ-1. В 1967 г. выпущена самая мощная (1 млн операций в секунду) БЭСМ-6.

Ввод речевой — процесс ввода данных с голоса пользователя. Для обеспечения речевого ввода необходим компьютер, оснащенный микрофоном, специальной платой для превращения звуковых колебаний в цифровые коды, базами данных (словарями), в которых собраны распознаваемые слова, и программами, которые ставят произнесенное слово в соответствие слову в словаре.

Ввод — считывание информации с внешнего устройства в память компьютера.

Векторная графика — способ представления изображения как совокупности графических элементов (графических примитивов — отрезков, дуг и пр.), описанных любым способом, в том числе графическими командами. Векторная графика хранится в метафайлах, которые чаще всего представляются, как файлы в двоичном коде, но могут иметь вид ASCII-текста. В связи с техническим принципом представления данных на экране дисплея в виде точек любая графика в конце концов — растровая.

Величина — (1) элемент данных, определенный либо своим именем, либо значением, либо и тем и другим; (2) часть памяти, которая задается именем (идентификатором) и значением, которое хранится в памяти.

Величина аналоговая — величина, у которой значения изменяются непрерывно и ее конкретное значение зависит только от точности прибора, производящего измерение. Например, температура воздуха.

Величина дискретная — величина, значения которой изменяются скачкообразно. Например, величина, характеризующая наличие или отсутствие тока в электрической цепи, является дискретной и может принимать значения «да» или «нет» («О» или «1»).

Вещественное число — тип данных, содержащий числа, записанные с десятичной точкой и (или) с десятичным порядком.

Видеоадаптер — электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Содержит видеопамять, регистры ввода-вывода и модуль BIOS. Посылает в дисплей сигналы управления яркостью лучей и сигналы развертки изображения.

Видеопамять — дополнительная память для обеспечения качественного изображения на дисплее. Является частью видеоадаптера, имеет объем до нескольких десятков мегабайт. В видеопамяти формируются изображения одного или нескольких экранов, которые затем подаются на дисплей. В некоторых компьютерах видеопамять выделяется из оперативной памяти.

***Виртуальная** память* — отличается от обычной ОП тем, что какие-то ее редко используемые фрагменты могут находиться на диске и подгружаться в реальную ОП по мере необходимости. Такая организация памяти позволяет снять ограничение, накладываемое объемом физической памяти, установленной на ЭВМ. Для реализации ВП используют, например, динамическую переадресацию. Сегментом в терминах ВП называется область памяти, состоящая из страниц. Вначале по номеру в таблице сегментов отыскивается сегмент. Таблица сегментов содержит начальный адрес таблицы страниц. Вторая часть адреса используется для обращения в эту таблицу, и по ней находится физический адрес данной страницы. Результаты поиска по таблицам запоминаются в быстродействующем ассоциативном ЗУ, называемом TLB. Наиболее часто употребляемые адреса откладываются в TLB и поэтому 98—99 % обращений к памяти идут без просмотра таблиц.

Внешние устройства — устройства ввода и вывода информации. Поскольку, как правило, они работают значительно медленнее остальных, управляющее устройство должно приостанавливать программу для завершения операции ввода-вывода с соответствующим устройством.

Волоконно-оптический кабель — кабель, передающий данные с помощью света, что увеличивает скорость и качество передачи. Используется в компьютерных сетях. В простейшем случае световод представляет собой волоконный (гибкий) диэлектрик, выполненный на основе кварцевого стекла и окруженный оболочкой с показателем преломления, меньшим, чем у сердцевины.

Восьмеричная система счисления — позиционная система счисления с основанием 8. Для записи чисел используются цифры 0, 1, 2, 3, 4, 5, 6, 7. Например, 123 в восьмеричной системе равно числу $1 \cdot 8^2 + 2 \cdot 8^1 + 3 \cdot 8^0 = 64 + 16 + 3 = 83$ в десятичной системе.

Второе поколение компьютерной техники — машины, созданные в 1955—1965 гг. Элементная база — дискретные транзисторные логические элементы. Оперативная память на магнитных сердечниках Высокопроизводительные устройства работы с магнитными лентами, магнитные барабаны и диски. Быстродействие — до сотен тысяч операций в секунду, емкость памяти — до нескольких десятков тысяч слов. Языки высокого уровня, широкий набор библиотечных программ, мониторные системы, управляющие режимом трансляции и исполнения программ.

Вывод — результаты работы программы, выдаваемые компьютером пользователю, другому компьютеру или во внешнюю память.

Выделенные каналы связи — каналы связи, закрепленные за сетью связи или специально созданные для нее. В отличие от коммутируемых каналов являются двух- и четырехпроводными. Иначе называются *некоммутируемыми* каналами связи.

Высказывание — понятие математической логики, определяемое как повествовательное предложение, которое может быть истинным или ложным, но не может быть истинным и ложным одновременно Над высказываниями возможно производить логические операции. Из простых высказываний строятся сложные.

Вычислительная сеть — комплекс компьютеров, вспомогательного оборудования, каналов связи и специального программного обеспечения для передачи данных между элементами сети. В зависимости от задач, типа оборудования и линий связи вычислительные сети разделяются на локальные, корпоративные, территориальные и глобальные. Сети создаются для более полного использования ресурсов или их перераспределения, для быстрой и автоматической связи с передачей больших объемов данных. Ресурсы бывают как вычислительные и технические (например, задача решается на нескольких компьютерах, или несколькими компьютерами используется один принтер), так и информационные (например, абоненты сети могут пользоваться одной базой данных).

Генератор тактовой частоты — устройство для выработки последовательности импульсов через равные отрезки времени. Время между двумя последовательными импульсами называется *тактом*. Некоторые команды процессора выполняются за несколько тактов. Импульсы, проходя через все элементы компьютера, заставляют их работать в едином такте — синхронно. Частота генерации тактовых импульсов определяет быстродействие компьютера.

Гибкий магнитный диск — диск из гибкой пластмассы в защитной пластмассовой упаковке, в которой прорезаны отверстия для подхода магнитных головок ввода/вывода. Диск покрыт магнитным составом. Часто называется флоппи-диском (floppy — свободно висящий) или дискетой Широко распространены диски диаметром 5,25 и 3,5 дюймов. Предназначен для длительного хранения небольших массивов данных, используется для хранения резервных копий и переноса данных с одного компьютера на другой. Для чтения/записи требуются соответствующие дисководы. Данные записываются по концентрическим окружностям, называемым треками или дорожками. На дискетах обычно ставится следующая маркировка: DS/SD — Double Sided/Single Density — двусторонние/единичной плотности; DS/DD (2S/2D, 2DD) - Double Sided/Double Density - двусторонние/двойной плотности; DS/HD (2HD) — Double Sided/ High Density — двусторонние/высокой плотности. В настоящее время выпускаются дискеты 3,5 дюйма под названием «Go anywhere» («Вездеход»), предназначенные для работы в условиях повышенной запыленности и влажности. Кроме того, разработана 3,5-дюймовая дискета емкостью 120 Мбайт.

Гигабайт (Гбайт) — единица измерения количества данных или объема памяти, $2^{30} = 1\ 073\ 741\ 824$ байта Иногда считают, что 1 Гбайт = $= 10^9 = 1\ 000\ 000\ 000$ байт Расхождение составляет около 7 %.

Главная (внутренняя, оперативная) память компьютера — упорядоченная последовательность байтов или машинных слов (ячеек памяти), проще говоря, — массив. Номер байта или слова памяти, через который оно доступно как из команд компьютера, так и во всех других случаях, называется адресом. Если в команде непосредственно содержится адрес памяти, то такой доступ этому слову памяти называется прямой адресацией.

Глобальная компьютерная сеть — совокупность отдельных компьютеров и локальных сетей, расположенных в разных странах, соединенных различными каналами связи и работающих в разных программных средах. Данная совокупность имеет согласованные протоколы взаимодействия.

Графика — наиболее общий способ визуального представления данных в компьютере, в котором объединяются текстовые данные и графические образы. Способы или форматы представления самого графического изображения на машинных носителях бывают двух типов: растровая и векторная графика.

Графический пользовательский интерфейс (Graphical User Interface — GUI) — программы, обеспечивающие пользователю работу с графическими образами. Первые разработки выполнены в компании Xerox Затем реализованы в операционных системах Apple Macintosh и Microsoft Windows. Графический пользовательский ин-

терфейс позволяет разработчикам прикладных систем работать с файлами, окнами и пр. в графическом режиме, осуществлять метафору «рабочего стола».

Графопостроитель — устройство для вывода из компьютера информации в виде графиков и чертежей на неподвижную или вращающуюся на барабане бумагу.

Датчик — устройство, обеспечивающее регистрацию какой-либо физической величины, преобразование ее в сигналы (обычно электрические) и передачу этих сигналов для обработки в систему управления. Например, в принтере стоит датчик конца листа бумаги. Если лист кончается, принтер перестает печатать и звуковым сигналом сообщает об этом.

Двоичная система счисления — позиционная система счисления с основанием 2. Для записи чисел используются двоичные цифры 0 и 1. Например, 101101 в двоичной системе равно числу: $1 \cdot 2^5 + 0 \cdot 2^4 + 1\ 2^3 + 1\ 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 45$ в десятичной системе. Является основной в вычислительной технике, так как приборы, имеющие два устойчивых состояния, проще, чем приборы с любым другим числом состояний. Используются производные системы счисления (степени 2) — восьмеричная и шестнадцатеричная.

Двойная точность (Double) — числовое данное (с фиксированной или плавающей точкой), размещенное в двух машинных словах, требует наличия операций специальной арифметики.

Двойное слово — машинное слово двойной длины (двойное слово) — используется для увеличения диапазона представления целых чисел. Двойные слова обрабатываются либо отдельными командами процессора, либо программно (эмуляция).

Дефрагментация диска — процесс размещения файлов на смежных секторах диска. Файлы при записи на диск разбиваются на фрагменты (кластеры), которые могут располагаться в разных частях диска. Это связано с работой файловой системы ОС. Разбитые на куски файлы затрудняют доступ, увеличивают время обращения к файлам. Считается, что следует выполнять процедуру дефрагментации диска раз в 1—2 недели. Соответствующие утилиты имеются в любой операционной системе. В MS-DOS и Windows это программа `defrag.exe`.

Джойстик (англ. joy stick — веселая палочка) — устройство ввода данных в компьютер. Представляет собой рычаг, выполненный в виде ручки, от поворота которой изменяется положение курсора на дисплее, и кнопки, которая фиксирует положение курсора. Часто применяется в компьютерных играх.

Диалоговый режим — режим непосредственного взаимодействия между человеком и компьютером, компьютерами в сети или между компь-

ютером и периферийным устройством, при котором связь между взаимодействующими системами не прерывается. Часто называется *интерактивным* режимом или режимом «*on-line*».

Дигитайзер (или ***диджитайзер, англ.*** *digitizer —* ***оцифровыватель)*** — устройство для ввода графических данных в компьютер, которое может основываться на различных технических принципах Как правило, при использовании любого из них контур изображения обводится специальным пером.

Динамическая оперативная память (англ. Dynamic Random Access Memory (DRAM) — динамическая память с произвольным доступом) — тип полупроводниковой оперативной памяти. Каждый двоичный разряд (бит) хранится в схеме, состоящей из транзистора и конденсатора. Если конденсатор заряжен, то это соответствует 1, разряженный конденсатор соответствует 0. Транзистор управляет доступом к конденсатору. Микросхема ДОП содержит, как правило, 2^{20} = 1 048 576 бит, и из них набирается оперативная память. В 1993 г. разработана микросхема емкостью 256 Мбит. Память называется динамической потому, что конденсаторы не могут долго хранить заряд и их необходимо каждые несколько миллисекунд подзаряжать.

Диск — носитель данных в форме круглой пластины, на которую осуществляется запись разными способами. Часто под диском подразумеваются несколько дисков, объединенных в пакет. Устройство, которое записывает (читает) данные на/с диск/а, называется *накопителем данных.* Диски различаются по способу записи/чтения данных, возможности их замены, плотности записи. По способу записи/чтения диски делятся на магнитные, лазерные (оптические), магнитооптические. Магнитные диски, в свою очередь, делятся на гибкие и жесткие. Лазерные диски предназначены в основном только для чтения данных. Магнитооптические диски могут использоваться как для чтения, так и для записи данных. Диски бывают съемные и постоянные. Съемные магнитные диски называются гибкими магнитными, устройство для чтения/записи называется накопителем на гибком магнитном диске. Несъемные магнитные диски называются жесткими магнитными дисками, устройство для чтения/записи называется накопителем на жестком магнитном диске или винчестером. Запись/чтение данных на жесткий магнитный диск производится на большой скорости (до 7200 оборотов в минуту) Объем данных на диске до 100 Гбайт.

Дисковод — устройство, управляющее вращением магнитного диска, чтением и записью данных на нем.

Дисплей (монитор) — устройство визуального отображения информации (в виде текста, таблицы, рисунка, чертежа и др.) на экране электронно-лучевого прибора.

Дисплей на ЭЛТ — дисплей, на экране которого изображение создается с помощью электронно-лучевой трубки (ЭЛТ). В дисплее на ЭЛТ изображение создается бомбардировкой флуоресцирующего экрана электронным лучом Луч движется по экрану слева направо и сверху вниз, за один проход формируются все горизонтальные строки Важными показателями являются: строчная частота дисплея — число строк, формируемое дисплеем за 1 с; частота обновления кадров (кадровая частота). Так как изображение создается одним лучом, то чем выше разрешение, тем больше надо создать строк, тем больше замедляется частота обновления кадров. Человек замечает мерцание экрана при понижении частоты смены кадров до 60 Гц. Рекомендуемая частота — 75 Гц. В настоящее время выпускают дисплеи с маркировкой N1 (Noninterlaced), означающей частоту обновления кадров более 60 Гц. Дисплеи на ЭЛТ характеризуются размерами экрана по диагонали в дюймах (чем больше, тем лучше) и расстоянием между точками растра (шаг точки) в мм (чем меньше, тем лучше); для дисплея с диагональю 14" нормальный шаг точки — 0,28 мм. Управление дисплеем на ЭЛТ бывает аналоговым и цифровым. В настоящее время наиболее распространены дисплеи с цифровым управлением. Дисплеи на ЭЛТ должны быть безопасными с точки зрения радиоизлучения, такие дисплеи маркируются LR (Low Radiation — низкое излучение)

Дисплейная панель — дисплей, построенный на плазменной или жидкокристаллической технологии, в отличие от дисплея на ЭЛТ. Основные достоинства — отсутствие жесткого излучения, низкое энергопотребление, плоская конструкция.

Дисциплина обслуживания FIFO (First In First Out) — в порядке поступления: «первым пришел — первым обслуживается». Все заявки на обслуживание поступают в конец очереди. Первыми обслуживаются заявки, находящиеся в начале очереди.

Дисциплина обслуживания LIFO — в порядке, обратном порядку поступления: «последним пришел — первым обслуживается» (Last In First Out). Является основой для построения стековой памяти.

Дополнительный код — беззнаковая форма представления чисел со знаком. В двоичной системе счисления дополнение каждой цифры выглядит как инвертирование двоичного разряда, т. е. замена 0 на I и наоборот. Если же знак числа представляется старшим разрядом машинного слова, то получается простой способ представления отрицательного числа: взять абсолютное значение числа в двоичной системе; инвертировать все разряды, включая знаковый; добавить 1.

Дорожка — концентрическая окружность на магнитной поверхности диска, где располагается информация. Дорожки нумеруются с 0-й (дорожка с самым большим радиусом).

Доступ — возможность чтения/записи данных в любых типах памяти. Различают методы прямого (произвольного) и последовательного доступа, которые тесно связаны с устройствами. Прямой доступ означает, что чтение/запись конкретных данных возможна без чтения/записи других данных. Последовательный доступ предполагает просмотр многих, если не всех, данных, для того чтобы прочитать/записать необходимые данные. На дисковых устройствах может быть организован как прямой, так и последовательный доступ к данным, поэтому дисковод называется устройством прямого доступа. Магнитную ленту в худшем случае надо перемотать всю, чтобы найти конкретное данное, поэтому магнитофон называется устройством последовательного доступа. Таким образом, методы доступа отличаются временем доступа: прямой — быстрый, последовательный — медленный.

Драйвер (Driver) — резидентный программный модуль, осуществляющий управление внешним устройством и связь с операционной системой и прикладными программами. Драйвер может входить в состав библиотек операционной системы. При подключении к компьютеру нового устройства необходимо иметь драйвер, который обеспечит работу этого устройства. Разработка драйвера, если он не поставляется с устройством, возможна программистом.

Дуплексное соединение — логическое или физическое соединение двух точек сети, между которыми может осуществляться передача данных одновременно в обоих направлениях.

Единая система ЭВМ (ЕС ЭВМ) — семейство компьютеров, объединенное единой идеологией, предназначенное охватить практически все виды деятельности вычислительными услугами, кроме управления технологическими процессами. Разработку и производство осуществляли с начала 80-х гг. группы институтов и заводов СССР и стран — членов СЭВ. Программно совместимы с компьютерами серий IBM-360 и IBM-370. Выпускались машины От очень мощных (ЕС1060) до персональных (ЕС1 845).

Жесткий магнитный диск — диск для долговременного хранения данных на компьютерах. В отличие от гибкого магнитного диска, который является съемным, жесткий магнитный диск составляет единое целое с дисководом. Поэтому жесткий магнитный диск и накопитель на жестком магнитном диске (НЖМД) являются для пользователя совпадающими понятиями. Как правило, в накопителе имеется несколько дисков, образующих пакет. Сам диск сделан из алюминиевого сплава и магнитного покрытия. Диски вместе с высокоскоростным двигателем и тонкопленочными головками чтения/записи помещаются в герметический корпус. Существуют переносные жесткие магнитные диски, которые подключаются к параллельным портам компьютера.

Запоминающее устройство (ЗУ) — устройство для записи, хранения и выдачи данных. Различают устройства: долговременного и оперативного хранения данных, они же энергонезависимые и энергозависимые; только для чтения данных (постоянное запоминающее устройство, компакт-диски) и как для чтения, так и для записи В зависимости от физических принципов хранения данных различают. магнитные, магнитооптические, оптические и полупроводниковые (схемные) устройства В некоторых случаях целесообразно разделить само устройство чтения/записи (накопитель) и носитель, на котором происходит хранение, например, накопитель на гибких магнитных дисках и гибкий диск, оптический накопитель и компакт-диск. Невозможно разделить накопитель и носитель в постоянном запоминающем устройстве, оперативной памяти и пр. Основными техническими характеристиками ЗУ является их емкость и быстродействие. ЗУ часто называется памятью.

Защита памяти — осуществляется путем блокировки доступа к памяти других процессов, а также блокировки доступа к памяти ядра. Один из способов — вся память делится на страницы, и у каждой есть замок — 4-битовый признак, который можно установить только привилегированной командой. В процессоре есть 4-битовый регистр — ключ, который также можно установить только привилегированной командой При обращении происходит сравнение замка и ключа С появлением многозадачности появилась проблема распределения памяти. При работе реальной программы обращения к ОП имеют тенденцию к локализации. Память можно разделить на используемую и неиспользуемую. Чтобы отследить использование области памяти, всю ОП можно разбить на страницы фиксированного размера (4 Кбайт) и с каждой страницей связать бит, который устанавливать в «1» при обращении к данной странице.

Защищенный режим (protected mode) — режим работы процессора Intel 386, при котором он выполняет множество проверок корректности обращений к памяти, вызовов функций, доступа к портам ввода-вывода и т. д. Такая защищенность позволяет операционной системе обрабатывать ошибочные операции. Для того чтобы иметь возможность использовать все адресное пространство и преимущества виртуальной памяти 386 процессора, приложение должно работать в защищенном режиме.

Интегральная схема — реализация электронной схемы, выполняющей некоторую функцию, в виде единого полупроводникового кристалла, в котором изготовлены все компоненты, необходимые для осуществления этой функции. Включает совокупность транзисторов, диодов, конденсаторов, резисторов и др. связанных между собой микропроводниками. Интегральная схема изготавливается по специальной технологии, обеспечивающей предельно малые размеры и

высокую надежность. По количеству элементов интегральные схемы условно делят на малые (МИС) — с количеством элементов в кристалле до 10^2 (100), средние (СИС) — до 10^3 (1000), большие (БИС) — до 10^4 (10000), сверхбольшие (СБИС) — до 10^6 (1 000 000), ультрабольшие (УБИС) - до 10^9 (1 000 000 000) и гигабольшие (ГБИС) — более 10^9 элементов в кристалле Выпускаются интегральные схемы с различными функциями, микропроцессоры — это интегральные схемы Для представления о размерах интегральной схемы и ее наполненности элементами можно рассмотреть микропроцессор Pentium: количество электронных компонентов — 3,1 млн, площадь кристалла — 292 мм^2 (это квадрат со стороной около 1,7 см), высота — несколько миллиметров. Довольно часто используется другое название интегральной схемы — *чип,* от *англ. chip* — тонкий кусочек.

Интерфейс (англ. inter — между и face — лицо) — (1) взаимодействие между элементами системы или системами; (2) совокупность средств, стандартов, сигналов, обеспечивающая обмен данными между устройствами; (3) взаимодействие между человеком и компьютером. Среди множества вариантов интерфейса человек—компьютер есть два принципиально отличных вида: «вспоминай-и-набирай» (язык команд, которые сначала надо вспомнить, потом набрать и выполнить); «смотри-и-выбирай» (язык меню и пиктограмм, в котором следует выбрать необходимое, после чего произойдет соответствующее действие).

Информация, единицы измерения — элементарной единицей информации является бит — это один двоичный разряд. 8 бит образует 1 байт. Информация в памяти ЭВМ измеряется в следующих единицах: килобайт (Кбайт), который равен 1000 байтам (1 Кбайт = 1024 байта (2^{10})); мегабайт (Мбайт), который равен 1000 килобайтам (1 Мбайт =1024 Кбайт); гигабайт (Гбайт) равен 1000 мегабайт (1 Гбайт =1024 Мбайт).

Информация — сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые воспринимают информационные системы (живые организмы, управляющие машины и др.) в процессе жизнедеятельности и работы. Применительно к обработке данных на компьютерах — произвольная последовательность символов, несущих смысловую нагрузку.

Исполняемый модуль — модуль, содержащий готовую к выполнению программу; может быть двух видов: *точный образ памяти* программы с привязкой к абсолютным адресам (в MS-DOS — формат файла * .COM); *перемещаемый* исполняемый формат.

Источник бесперебойного питания (ИБП) — система, которая обеспечивает защиту электронных приборов, в том числе компьютеров и сетей, от бросков напряжения, перекоса фаз или внезапного прекра-

щения подачи энергии. Обычно такие сбои приводят к порче или потере данных. Существует много видов ИБП — от простых до самонастраивающихся. Как правило, ИБП состоит из электронных схем и батарей аккумуляторов, которые при нарушениях в питании переходят на автономное питание, обеспечивающее сохранение данных в системе Выбор ИБП зависит от величины ожидаемых потерь при некачественном энергопитании.

Исходный код программы — код, написанный на языке программирования. Может включать модули на ЯВУ и модули с подпрограммами на языке ассемблера.

Итерационный цикл — вид цикла, для которого число повторений операторов тела цикла заранее неизвестно. На каждом шаге вычислений происходят последовательное приближение и проверка условия достижения искомого результата. Выход из цикла осуществляется в случае выполнения заданного условия.

Канал связи — технические устройства и физическая среда, обеспечивающие передачу данных. Каналы связи разделяются на аналоговые и цифровые, на телефонные, телеграфные, радиочастотные, телевизионные, инфракрасные и оптические. Кроме этого, каналы связи бывают выделенные и коммутируемые. Дуплексный канал — канал, по которому передача данных происходит в оба направления одновременно. Симплексный канал — канал, по которому передача данных в каждый момент времени происходит только в одном направлении. Основной характеристикой канала является его пропускная способность.

Картридж (англ. cartrige — патрон, кассета) — сменяемая часть устройства. Обычно это кассета, в которой хранится красящая лента для принтеров, тонер для лазерных принтеров или множительных аппаратов, чернила для струйных принтеров и пр. Картридж полностью готов к работе, для этого его достаточно поставить на место.

Каталог файлов — логическое разбиение дисковой памяти (для операционных систем UNIX, MS DOS, Windows и ряда других) на части, в которых могут храниться файлы и другие каталоги. Совокупность каталогов создает дерево каталогов с корневым каталогом. Все каталоги, кроме корневого, называются подкаталогами, но так как свойства у каталога и подкаталога одинаковы, то применяют общее название — каталог. В каждом каталоге могут быть другие каталоги и файлы. В одном каталоге не может быть непосредственных подкаталогов и файлов с одинаковыми именами. Чтобы сделать доступным какой-либо файл, необходимо указать последовательно все промежуточные каталоги, начиная с корневого, и разделить их символом «\»; последним указывается имя требуемого файла.

Килобайт (Кбайт) — единица измерения количества данных или объема памяти, равная 2^{10} = 1024 байтов. Иногда считают, что 1 Кбайт =

$= 10^3 = 1000$ байтов. Расхождение составляет 2,4 %. Правильнее говорить «Кбайт», потому что «кило» обычно означает 1000.

Клавиатура — устройство, предназначенное для ручного ввода данных в компьютер. Клавиатуры различаются количеством клавиш. Стандартным для IBM-подобных компьютеров является клавиатура со 101 клавишей, где выделены блоки: функциональных клавиш; букв, цифр и вспомогательных символов; клавиш управления курсором; цифровой клавиатуры (дублируется для удобства ввода). При каждом нажатии клавиши в память посылается не код символа, который нарисован на клавише, а код клавиши, который затем программным путем связывается с символом. Такой подход позволяет гибко менять набираемые символы.

Кластер (англ. cluster — группа) — единица хранения данных на гибких и жестких дисках. Кластер содержит несколько рядом стоящих секторов.

Клиент (Client) — программно-технический комплекс, обеспечивающий интерфейс с пользователем (другой активной стороной) при отправлении и получении запросов от сервера.

Клиент-сервер архитектура (Client-Server) — распределенная обработка запросов в сети, реализуемая на двух взаимодействующих программно-технических комплексах (клиент и сервер)

Коаксиальный кабель (лат со — совместно и *axis* — ось) — кабель, состоящий из двух соосных проводников, между которыми расположен изолятор Используется в каналах связи компьютерных сетей. Применяется для передачи сигналов с несущей частотой до $3 \cdot 10^{10}$ Гц. Коаксиальный кабель, предназначенный для работы на частотах ниже 105 Гц, называется экранированным проводом.

Код ASCII (англ. American Standard Code for Information Interchange — Американский стандартный код для обмена информацией) — стандарт кодирования символов латинского алфавита, цифр и вспомогательных символов или действий в виде однобайтового двоичного кода (1 байт = 8 бит). Первоначально стандарт определял только 128 символов, используя 7 битов (от 0 до 127). Использование всех восьми битов позволяет кодировать еще 128 символов. В этом случае говорят о расширенном ASCII-коде. Дополнительные символы могут быть любыми, им отводятся коды от 128 до 255. Русские символы кодируются именно в этой части ASCII-кода. Например, служебное действие «ввод» (клавиша <Enter>) имеет код — 13, символ «1» имеет код 49, символ «W» — 87, символ «w» — 119, русские символы «Б» и «б» — соответственно 129 и 161 (при альтернативной кодировке). Другие обозначения — IA-5, ANSI X.34, ISO-7 (код ISO-7 отличается 10-ю кодовыми комбинациями, зарезервированными для национальных применений).

Код EBCDIC (EBCDIC code) — внутренний 8-битовый код хранения символьной информации в больших машинах *(mainframes)* фирмы IBM и ряда других — Extended Binary Coded Decimal Interchange Code).

Код Unicode — стандарт для представления символов с использованием 16-разрядных кодов (2 байта) Допускает 65 536 символов. С этим стандартом работает Windows NT и Windows 95 Стандарт должен в перспективе заменить ASCII, так как удобнее пользоваться одним кодом для разных языков, чем менять перекодировочные таблицы в ASCII-коде. Организован консорциум пользователей Unicode, в который вошли практически все известные фирмы-разработчики программного обеспечения: IBM, Microsoft, Borland и др.

Код Бодо (Baudot code) — международный телеграфный код IA-1 (International Aphabet 1), предшественник IA-2 (M-2, MKKTT-2, CCITT-2).

Код MKKTT-2 (CCITT-2 code) — телеграфный код, предложенный Международным консультативным комитетом по телефонии и телеграфии (MKKTT), бинарный 5-разрядный, трехрегистровый; он же IA-2 (International Alphabet 2).

Код Холлерита (Hollerith code) — код, используемый для представления информации на 80-колонных перфокартах, 12-разрядный, избыточный.

Кодек (англ. Codec — COmpress — DECompress — сжимать — восстанавливать) — аппаратно-программный комплекс, обеспечивающий работу персонального компьютера с видеоинформацией. Кодек позволяет достичь качества сигнала VHS или лучше за счет использования аппаратных и программных методов сжатия данных.

Кодирование (Coding) — установление согласованного (узаконенного) соответствия между набором символов и сигналами или битовыми комбинациями, представляющими каждый символ для целей передачи, хранения или обработки данных.

Кодовая таблица (Code Page) — таблица, устанавливающая стандартизованное соответствие графических символов и бинарных кодов, определяемое применением (алфавит, программы, устройства ЭВМ).

Кольцо — способ соединения компьютеров в сеть, когда данные в сети передаются последовательно от одной станции к другой. Как правило, данные передаются только в одну сторону, поэтому, чтобы передать сообщение рядом стоящему, но находящемуся против движения данных компьютеру, нужно пройти все компьютеры в сети. Преимущество — простота управления, недостаток — возможность отказа всей сети при сбое в канале между двумя узлами.

Команда — описание элементарной операции, которую должен выполнить компьютер. Обычно содержит код выполняемой операции,

указания по определению операндов (или их адресов), указания по размещению получаемого результата. Последовательность команд образует программу.

Командный язык модема (Modem ***AT-command*** *(Hayes AT command))* — элемент командного языка, управляющего работой Hayes-совместимого модема.

Коммутация — (1) процесс соединения или переключения вычислительных систем, в том числе компьютеров; (2) различают коммутацию пакетов сообщений, под которой понимают объединение некоторых данных и их передачу по каналам связи.

Коммутируемые каналы связи — каналы связи общего назначения, которые используются конкретной сетью только на момент связи. В территориальных и глобальных компьютерных сетях, как правило, используются телефонные каналы общего назначения, которые по вызову подключаются (коммутируются) к данной сети. Коммутируемые каналы являются низкоскоростными в отличие от выделенных каналов.

Компакт-диск — диск для постоянного хранения данных, представляющий собой круг из пластика или алюминиевого сплава, покрытый защитной прозрачной пленкой. Запись производится по одной спиралевидной, очень длинной дорожке настолько плотно, что на 5-дюймовый диск помещается до 700 Мбайт данных. Осуществляется запись в стационарных условиях на специальных устройствах, и затем для массового потребления штампуются компакт-диски только для чтения (CD-ROM — Compact Disk Read Only Memory). Чтение производится маломощным (следовательно, значительно более дешевым) лазером по тому же принципу: диск вращается с достаточно большой скоростью, лазерный луч фокусируется на дорожке, и читающее устройство ловит отраженный луч, который падает на фотодиод. Если луч попадает на отражающую поверхность диска, интенсивность отраженного луча одна, если на рассеивающую поверхность — другая; этим различаются нули и единицы, с помощью которых записаны данные.

Компьютер — программируемое электронное устройство, способное обрабатывать данные и производить вычисления, а также выполнять другие задачи манипулирования символами. Основу компьютеров образует аппаратура *(hardware),* построенная, в основном, с использованием электронных и электромеханических элементов и устройств. Принцип действия компьютеров состоит в выполнении программ *(software)* — заранее заданных, четко определенных последовательностей арифметических, логических и других операций.

Конечный ***пользователь*** *(End User)* — пользователь, на обслуживание которого ориентирована система (информационно-поисковая, операционная и пр.).

Контроллер (англ. control — управлять) — устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования Контроллер выполняет интерпретацию команд процессора для отдельных устройств

Косвенная адресация — случай, когда машинное слово содержит адрес другого машинного слова. Тогда доступ к данным во втором машинном слове через первое называется косвенной адресацией. Команды косвенной адресации имеются в любом компьютере и являются основой любого регулярного процесса обработки данных. Действительно, содержимое первого машинного слова можно формировать программно, работая с различными (например, последовательно расположенными) словами памяти.

Курсор — светящийся участок на экране дисплея, указывающий позицию, на которой будет отображаться следующий вводимый с клавиатуры знак.

Кэш-память — сверхоперативная память, обращение к которой намного быстрее, чем к оперативной, и в которой хранятся наиболее часто используемые участки последней При обращении к памяти сначала нужные данные ищутся в кэш-памяти. При отсутствии производится обращение к оперативной памяти, в результате общее время доступа к памяти сокращается

Логические операции — над значениями условных выражений можно выполнить логические операции И (&, AND), ИЛИ (|, OR) и НЕ (!, ¬, NOT), которые Объединяют по правилам логики несколько условий в одно. Благодаря тому, что любая логическая операция может быть представлена с помощью трех основных логических операций, набора элементов И, ИЛИ и НЕ в принципе достаточно для построения любого устройства процессора компьютера, а также для описания любых алгоритмов.

Логический элемент (вентиль) — часть электронной логической схемы, выполняющая элементарную логическую функцию.

Логическое высказывание — любое предложение, в отношении которого можно однозначно сказать, истинно оно или ложно

Логическое устройство — устройство, созданное программным способом, но функционально работающее как физическое Одному логическому устройству может соответствовать несколько физических и наоборот Например, один физический жесткий диск может быть поделен на несколько логических дисков.

Лэптоп (наколенник) — laptop — портативный компьютер, по своим размерам близкий к портфелю По быстродействию и памяти примерно соответствует настольным персональным компьютерам.

Магнитооптический накопитель — накопитель для работы с магнитооптическими дисками. Магнитооптический диск (МО-диск) изготавливается из алюминиевого сплава и заключен в пластиковую оболочку Технология записи данных: лазерный луч нагревает точку на диске, а электромагнит изменяет магнитную ориентацию этой точки в зависимости от того, что необходимо записать — «О» или «1» Считывание производится лазерным лучом меньшей (чем при записи) мощности, который, отражаясь от этой точки, меняет свою полярность. МО-диски (и соответственно дисководы) выпускаются двух размеров: 3,5 дюймов, (емкость 500 Мбайт); 5,25 дюймов (2,3 Гбайт). Время доступа к данным составляет около 50 мс. Магнитооптические накопители выпускаются двух типов· перезаписываемые и типа WORM (Write-Once, Read-Many — однократная запись, многократное считывание)

Манипулятор *(лат. manus — рука)* — устройство, позволяющее управлять состоянием компьютера, в том числе и вводить данные с помощью рук. К манипуляторам относятся джойстик, мышь, трекбол, сенсорная панель, перо, трекпойнт, J-клавиша.

Маршрутизатор — электронное устройство, иногда с программным блоком, определяющее оптимальный путь (маршрут) пакета сообщений в компьютерных сетях.

Массив дисков RAID (англ. Redundant Arrays of Independent Disks — избыточный массив независимых дисков). Набор жестких дисководов, конструктивно объединенных в один блок с общим интеллектуальным контроллером. Как правило, используется в серверах для обеспечения надежности за счет дублирования данных. Существует восемь уровней (от 0 до 7) реализаций RAID, которые отличаются друг от друга степенью избыточности, методом доступа и пр. Стандарт седьмого уровня отличается собственной операционной системой и высокой производительностью.

Масштабируемость — свойство системы или ее отдельных частей, характеризующее возможность системы приспосабливаться к уменьшению или увеличению ее отдельных параметров. Например, операционные системы Windows имеют масштабируемый пользовательский интерфейс, который обеспечивает одинаковый внешний вид при использовании дисплеев разных размеров

Математический сопроцессор — интегральная схема, дополнительная к главному (центральному процессору), которая выполняет команды, работающие с числами, представленными в форме с плавающей точкой (запятой). За счет использования сопроцессора скорость работы ПК увеличивается в 4—20 раз. Этот выигрыш получается не только при решении вычислительных задач, но и при работе с графикой. При работе с текстами сопроцессор не используется. Для микропроцессора i8086 выпускался сопроцессор i8087, для i80286 —

i80287. С микропроцессором i80386 могут использоваться сопроцессоры i80287 и i80387. Микропроцессоры i486 выпущены со встроенным сопроцессором, однако в клоне i486SX сопроцессор заблокирован.

Матричный принтер — принтер, у которого печатающий узел представляет собой металлическую пластину с отверстиями (матрицу), в которых свободно двигаются штырьки (иголочки). Штырьки, управляемые магнитом, бьют по красящей ленте (такой же, как у пишущей машинки), и на бумаге точками создается символ. Матричный принтер обеспечивает приемлемую скорость печати и качество. Основной недостаток — значительный шум при печати. Иногда говорят: игольчатая печать, печать ударного типа (impact printer).

Машина Поста (автор — математик Э. Пост, США) — абстрактная конструкция, предназначенная для уточнения понятия алгоритма (если для решения задачи можно построить машину Поста, то она алгоритмически разрешима). Машиной называется потому, что при описании используются понятия — память, команда и пр. Машина Поста и машина Тьюринга эквивалентны по своим возможностям. Разработаны практически в одно и то же время (в 1936 г.) независимо друг от друга. Машина Поста состоит из неограниченной длины ленты, разделенной на ячейки, которые последовательно пронумерованы целыми числами, как положительными, так и отрицательными. В каждой ячейке ленты стоит либо признак того, что в ячейке записана метка, либо ячейка пустая. Состояние ленты — это данные о том, какие ячейки заняты, а какие пусты. Кроме ленты, имеется головка чтения/записи, которая: умеет двигаться вперед, назад и стоять на месте; умеет читать содержимое, стирать и записывать метку; управляется программой, в которую могут входить в любой комбинации и любом количестве шесть команд: Вправо; Влево; Поставить метку; Стереть метку; Передача управления на один номер команды в программе, если в текущей ячейке есть метка; если метки нет, то Передача управления на другой номер команды; Прекращения работы. Состояние машины — это состояние ленты и положение головки чтения/записи. Машина Поста, несмотря на внешнюю простоту, может производить различные вычисления, для чего надо задать начальное состояние машины и программу, которая эти вычисления сделает.

Машина Тьюринга (автор — математик А. Тьюринг, Англия) — абстрактная конструкция, предназначенная для определения понятия алгоритма (если для решения задачи можно построить машину Тьюринга, то она алгоритмически разрешима). Машиной называется потому, что при построении используются некоторые понятия реальных машин — память, команда и пр. Машина Тьюринга состоит из неограниченной в обе стороны ленты, разделенной на ячейки, кото-

рые последовательно пронумерованы целыми числами, как положительными, так и отрицательными. В каждой ячейке ленты может стоять любой символ из заданного алфавита, в котором выделен «пустой» символ — признак того, что ячейка пустая. Машина имеет конечное множество внутренних состояний, начальное (с него начинается работа машины) и конечное состояние, попав в которое, машина прекращает работу. Кроме ленты, имеется головка чтения/записи, которая, во-первых, умеет двигаться вперед, назад и стоять на месте; во-вторых, умеет читать содержимое, стирать и записывать символы из данного алфавита; в-третьих, управляется программой. Программа представляет собой таблицу, в которой в каждой клетке записана команда. Каждая клетка определяется двумя параметрами — символом алфавита и состоянием машины. Команда представляет собой указание, куда передвинуть головку чтения/записи из текущего состояния, какой символ записать в текущую ячейку и в какое состояние перейдет машина.

Машинное слово — упорядоченное множество двоичных разрядов, используемое для хранения команд программы и обрабатываемых данных. Каждый разряд, называемый битом, — это двоичное число, принимающее значения только 0 или 1. Разряды в слове обычно нумеруются справа налево начиная с 0. Количество разрядов в слове называется размерностью машинного слова или его разрядностью.

Машинный язык — совокупность машинных команд компьютера, отличающаяся количеством адресов в команде, назначением информации, задаваемой в адресах, набором операций, которые может выполнить машина, и др.

МДП-структура — структура «металл—диэлектрик—полупроводник», используемая при создании электронных приборов, в том числе микропроцессоров, памяти для компьютеров. Представляет собой упорядоченную совокупность очень тонких (менее 1 мкм) слоев металла и диэлектрика, нанесенных на полупроводниковую пластину. Если в качестве диэлектрика используются оксиды (оксид алюминия, диоксид кремния), то образуется МОП-структура («металл—оксид—полупроводник»). Метод создания приборов на основе таких структур называется МДП-технологией или МОП-технологией.

Мегабайт (Мбайт) — единица измерения количества данных или объема памяти, равная $2^{20} = 1\ 048\ 576$ байт. Иногда считают, что 1 Мбайт = $10^6 = 1\ 000\ 000$ байт. Расхождение составляет более 4,8 %.

Медленная связь — подключение через модем (быстродействие от 9600 до 28 800 бит/с).

Микрокомпьютер — компьютер, в котором в качестве управляющего и арифметического устройства используется микропроцессор.

Микрометр (мкм) — 10^{6} м, 1000 нанометров (нм).

Микросекунда (мс) — 10^{-6} с, 1000 наносекунд (нс).

Микропроцессор — устройство, осуществляющее обработку данных и управляющее этим процессом, выполненное в виде одной или нескольких больших (сверхбольших) интегральных схем. Микропроцессоры встраиваются в устройства управления и входят основной частью в компьютер. Например, в автомобиле марки «БМВ» установлено 54 интегральные схемы, которые управляют тормозами с антиблокировкой и воздушными подушками безопасности, в доме средней американской семьи используется около 50 интегральных схем, управляющих бытовыми приборами. В состав микропроцессора входят: арифметико-логическое устройство, выполняющее арифметические и логические операции; блок управления и синхронизации; блок ввода/вывода; регистры и пр Самыми известными микропроцессорами для персональных компьютеров являются микропроцессоры фирмы Intel. Один из учредителей и президент фирмы Intel Гордон Мур (Gordon Moor) в 1965 г. сформулировал правило (получившее его имя): число транзисторов (количество элементов), которое может содержать кристалл микропроцессора, удваивается каждые полтора года.

Многозадачность — режим одновременного решения нескольких задач на компьютере. Под задачей в данном случае понимается часть работы, выполняемой процессором.

Модем (Modem) — устройство преобразования цифровой информации в аналоговую и обратно посредством модуляции/демодуляции несущей частоты для передачи данных по телефонным линиям. Дискретные (двоичные) данные из компьютера попадают в модем, где кодируются соответствующим образом (модулируются) и передаются в линию связи. На другом конце линии они попадают в другой модем, где преобразуются (демодулируются) в двоичные сигналы и поступают в принимающий компьютер.

МОП-структура — структура материала, из которого изготовляются транзисторы, конденсаторы и др. электронные приборы. Сокращение от «металл—оксид—полупроводник».

Мультимедиа — собирательное понятие для различных компьютерных технологий, при которых используется несколько информационных сред, таких, как графика, текст, видео, фотография, движущиеся образы (анимация), звуковые эффекты, высококачественное звуковое сопровождение. Мультимедиа-компьютер — это компьютер, снабженный аппаратными и программными средствами, реализующими технологию мультимедиа.

Мышь (mouse) — манипулятор, позволяющий выбирать данные на дисплее, вводить графические данные. Включает в себя шар и две или

три кнопки. Шар заставляет курсор перемещаться по экрану, а кнопки играют роль клавиш <Enter> (ввод) и <Esc> (выход). Удобство применения мыши заключается в непосредственном доступе к каждой точке экрана. Название устройство получило за свое сходство с настоящей мышью, за счет «хвоста» — провода, соединяющего устройство с процессорным блоком.

Мэйнфрейм (*англ. main* — главный, *frame* — сооружение, буксир, тягач) — большой, очень мощный компьютер общего назначения, используемый для работы в качестве суперсерверов в мощных сетях и объемных научных расчетах. Мэйнфреймы занимают промежуточное место между персональными и суперкомпьютерами.

МЭСМ— малая электронная счетная машина. Первая в СССР и континентальной Европе вычислительная машина, выполняющая регулярные расчеты Разработана в Киеве, в Институте электротехники, под руководством и при непосредственном участии С. А. Лебедева в 1951 г. МЭСМ имела скорость 50 операций в секунду, содержала 6000 электронных ламп и занимала несколько комнат. Первоначально МЭСМ задумывалась как макет БЭСМ.

Набор импульсный/тональный (Dialing Pulse/Tone) — операции установления соединения по коммутируемым телефонным каналам, выполняемые модемом.

Накопитель — устройство для записи/чтения данных на/с определенный носитель. Накопители относятся к внешним запоминающим устройствам. Различают накопители на дисках, лентах, картах. Различают также накопители: со съемными носителями (в этом случае носитель данных можно поменять — например, гибкие магнитные диски, магнитные ленты, реже — пакет жестких МД); с постоянными носителями (в этом случае носитель встроен в накопитель и его нельзя сменить, например, жесткий магнитный диск). Существуют переносные накопители, которые можно свободно и быстро менять, например, специальные съемные накопители на жестком магнитном диске (НЖМД). Переносные накопители могут подключаться к порту LPT. Каждому накопителю должен соответствовать адаптер, который иногда является частью накопителя, а иногда самостоятельным устройством, например, в компьютерах PS/2 адаптеры накопителя на гибких магнитных дисках и НЖМД встроены в системную плату.

Накопитель на гибком магнитном диске (НГМД) — устройство для записи/чтения данных на гибкий пластиковый диск, покрытый магнитным слоем. Устройство состоит из двух двигателей, один из которых вращает диск (360 об/мин), а другой (шаговый) передвигает головки чтения/записи по концентрическим окружностям (трекам). Диски свободно вставляются в устройство. НГМД часто называют дисководом. Различают два типа НГМД по размеру рабочего дис-

ка — 5,25" (пятидюймовые) и 3,5" (трехдюймовые). Количество записываемых данных зависит как от накопителя, так и самого диска. На 5-дюймовых записывают данные объемом от 180 Кбайт до 1,2 Мбайт. На 3-дюймовых: 720 Кбайт — 1,44 Мбайт. На ПК обычно устанавливают один 5-дюймовый и один 3-дюймовый дисководы. В последнее время от 5-дюймовых дисководов отказываются.

Накопитель па жестком магнитном диске (НЖМД)— устройство для записи/чтения данных на жестком диске (иногда называют «винчестером»). НЖМД впервые использован в персональном компьютере в 1983 г. фирмой IBM. Наиболее массовое запоминающее устройство большой емкости, в котором носителями информации являются круглые алюминиевые пластины — *платтеры*, обе поверхности которых покрыты слоем магнитного материала. Используется для постоянного хранения больших объемов информации. На каждый диск приходится две (на каждую сторону) головки чтения/записи. Имеется два двигателя, один из которых вращает диски (3600—7200 об/мин), второй перемещает головки по трекам. Двигатели, головки и диски помещены в металлический герметичный корпус. За счет герметичности и постоянства дисков достигается высокая плотность записи и, следовательно, возможность хранить большие объемы данных (до 100 Гбайт). Есть несколько версий происхождения названия «винчестер». Приведем две из них: от названия города Winchester в Англии, в котором филиалом фирмы IBM разработан данный тип накопителя; от маркировки первого жесткого диска, сходной с калибром знаменитой винтовки Winchester (30/30). Здесь тоже две легенды: (1) диск имел емкость 30 Мбайт, время доступа 30 миллисекунд; (2) накопитель состоял из двух дисков по 30 Мбайт каждый.

Накопитель на компакт-диске (CD-ROM) — накопители также называют оптическими. Технологию CD-ROM изобрел в 1965 г. Джеймс Рассел (James Russell). Устройство позволяет только считывать данные с компакт-диска, именно поэтому на английском языке он называется CD-ROM (Compact Disk Read Only Memory — Память только для чтения на компактном диске). Содержит двигатель для вращения диска, генератор лазерного луча и преобразователь отраженного лазерного луча в электрические сигналы, соответствующие «О» и «1». На компакт-диск предварительно в стационарных условиях записываются данные в виде микроскопических меток, отражающих или рассеивающих лазерные лучи и расположенных последовательно на одной спиралевидной дорожке. Процесс чтения происходит следующим образом: на дисковод устанавливается носитель (компакт-диск), этот диск начинает вращаться относительно лазерной головки; луч лазера попадает на диск и отражается с разной интенсивностью в зависимости от того, попал ли на отражающую

или рассеивающую поверхность; отраженный луч попадает на фотодиод, с помощью которого импульсы света превращаются в нули и единицы. Накопитель CD-ROM имеет высокую скорость передачи данных За единицу скорости такого накопителя принимают 150 Кбайт/с (1-х), в настоящее время выпускаются 50-скоростные (150 х 50 = 7,5 Мбайт/с) накопители (50-х).

Нанометр (нм) — 10^{-9} м = 0,001 микрометра (мкм).

Наносекунда ***(нс)*** — 10^{-9} с = 0,001 микросекунды (мс).

Нанотехнология — технология изготовления интегральных схем для процессоров в компьютерах, основанная на работе на уровне молекул и атомов. Базируется на величинах, соответствующих нанометрам и наносекундам. Например, выражение «технология (процесс) 130 нм (или 0,13 мкм)» означает, что размеры структурного элементов микросхемы не превосходят 130 нм.

Непозиционная система счисления — система счисления, при которой для обозначения чисел вводятся определенные знаки, количественное значение которых всегда одинаково и не зависит от месторасположения. В позиционной системе тоже вводятся знаки — цифры, но их количество ограничено основанием системы. Непозиционные системы используются редко, так как плохо приспособлены для вычислений. Характерный пример — римская система счисления. Имеются символы латинского алфавита со следующими значениями: **I** - 1, **V** - 5, **X** - 10, **L** - 50, **C** - 100, **D** - 500, **M** - 1000. Для записи чисел используется следующий алгоритм: каждый меньший знак, поставленный слева от большего, вычитается из него, а каждый меньший знак, поставленный справа от большего, прибавляется к нему. Например, число 124 в десятичной системе представляется как CXXIV в римской.

Ноутбук (блокнот) — портативный компьютер, по своим размерам близкий к книге крупного формата, помещается в портфель-дипломат. Обычно комплектуется модемом и снабжается приводом CD-ROM.

Оперативная память — предназначена для хранения программ и данных, которыми они манипулируют. Физически выполнена в виде некоторого числа микросхем. Логически ОП можно представить как линейную совокупность ячеек, каждая из которых имеет свой номер, называемый адресом. Время записи и чтения из ОП в современных машинах занимает доли микросекунды, а для других устройств это время в 10—1000 раз больше.

Организация ввода-вывода — в современных ЭВМ осуществлена с использованием *прерываний.* Это связано с тем, что УВВ работают намного медленнее, чем процессор и оперативная память. Поэтому управляющее устройство должно приостанавливать выполнение

программы и ждать завершения операции ввода-вывода с внешним устройством. При выводе все результаты выполненной программы должны быть выведены на ВУ, после чего процессор переходит к ожиданию сигналов от ВУ При вводе, например, с клавиатуры получение значений нажатых клавиш осуществляется при поступлении прерывания от клавиатуры

Основание системы счисления — количество различных цифр, используемых для изображения чисел в данной системе счисления.

Открытых систем взаимодействие (OSI) — совокупность требований ISO для установления взаимодействия открытых систем (ВОС) в сетях (Open system Interconnection reference model).

Палмтоп (наладонник) — *palmtop* — самый маленький современный персональный компьютер. Умещается на ладони, магнитные диски в нем заменяет энергонезависимая электронная память.

Пейджинг — механизм виртуальной памяти, при котором страницы вытесняются на диск или «подкачиваются» с диска.

Первое поколение компьютерной техники — машины, созданные на рубеже 50-х гг. В схемах использовались электронные лампы. Набор команд небольшой, схема арифметико-логического устройства и устройства управления простая, программное обеспечение практически отсутствовало. Быстродействие — 10—20 тыс. операций в секунду.

Персональный ***компьютер*** — микрокомпьютер универсального назначения, рассчитанный на одного пользователя и управляемый одним человеком.

Поколения компьютеров — условная, нестрогая классификация вычислительных систем по степени развития аппаратных и программных средств, а также способов общения с ними.

Пользователь (User) — физическое или юридическое лицо, непосредственно применяющее информационный ресурс, систему, технологию для решения задач.

Порты устройств — электронные схемы, содержащие один или несколько регистров ввода-вывода и позволяющие подключать периферийные устройства компьютера к внешним шинам микропроцессора. Последовательный порт обменивается данными с процессором побайтно, а с внешними устройствами — побитно. Параллельный порт получает и посылает данные побайтно.

Постоянная память (ПЗУ) — энергонезависимое запоминающее устройство, изготовленное в виде микросхемы. Используется для хранения данных, не требующих изменения. Содержание памяти специальным образом «зашивается» в ПЗУ при изготовлении. В ПЗУ находятся программа управления работой самого процессора, программы управления дисплеем, клавиатурой, принтером, внешней

памятью, программы запуска и остановки компьютера, тестирования устройств.

Прерывания — специфические сигналы, посылаемые процессору устройством или программой, когда требуется его немедленное вмешательство. В этом случае он останавливает всякую другую деятельность и вызывает программу *обработчик прерывания* По окончании ее работы он продолжает прерванную работу с того места. где она остановилась. Прерывания бывают двух типов — *аппаратные* (генерируются схемами ПК в ответ на какое-либо действие), например, при нажатии клавиши на клавиатуре генерируется прерывание (иногда аппаратные прерывания генерируются устройством в случае некорректной работы программы, например деление на 0); *программные* — генерируются программой для вызова различных подпрограмм из ОЗУ и ПЗУ.

Принтер — печатающее устройство, преобразует закодированную информацию, выходящую из процессора, в форму, удобную для чтения на бумаге.

Принципы фон-Неймана — включают в себя: *Принцип программного управления.* Программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности. *Принцип адресности.* Основная память состоит из перенумерованных ячеек; процессору времени доступна любая ячейка. *Принцип однородности памяти.* Программы и данные хранятся *в* одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

Протокол коммуникации — согласованный набор конкретных правил обмена информацией между разными устройствами передачи данных.

Процессор — центральное устройство компьютера. Назначение процессора: управлять работой ЭВМ по заданной программе; выполнять операции обработки информации. Возможности компьютера как универсального исполнителя по работе с информацией определяются системой команд процессора. Эта система команд представляет собой язык машинных команд (ЯМК). Отдельная команда определяет отдельную операцию (действие) компьютера. В ЯМК существуют команды, по которым выполняются арифметические и логические операции, операции управления последовательностью выполнения команд, операции данных из одних устройств памяти в другие и пр. В состав процессора входят: устройство управления (УУ), арифметико-логические устройство (АЛУ), регистры процессорной памяти.

Реальный режим (real mode) — режим работы процессора Intel 80386, совместимый с процессором Intel 8086. В реальном режиме невозможны доступ к *виртуальному адресному пространству* процессора 386 или такие возможности, как *замещение страниц по требованию.*

Регистр команд — регистр УУ для хранения кода команды на период времени, необходимый для ее выполнения.

Регистр — специальная запоминающая ячейка, выполняющая функции кратковременного хранения числа или команды и выполнения над ними некоторых операций. Отличается от ячейки памяти тем, что может не только хранить двоичный код, но и преобразовывать его.

Сверхоперативная память — очень быстрое ЗУ малого объема. Используется для компенсации разницы в скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Свопинг — алгоритм реализации виртуальной памяти. Его можно разбить на три части: управление пространством на устройстве выгрузки, выгрузка процессов из основной памяти и подкачка процессов в основную память. В качестве устройства выгрузки используют раздел на устройстве типа жесткого МД (swap-partition) или дисковый файл (swap-file или page-file) на таком устройстве.

Сектор — каждая дорожка, размещенная на диске, делится на секторы (блоки). Каждый сектор имеет размер 512 байт (для MS DOS).

Сервер (server) — сетевой компьютер, на котором находятся доступные клиентам *ресурсы.* Ресурсами сервера могут быть файлы, принтеры или приложения-серверы (такие, как многопользовательские базы данных).

Сервер файловый (File Server) — выделенная машина с установленным программным обеспечением, поддерживающим общие информационные ресурсы в локальной сети.

Сеть информационно-вычислительная (Distibuted computation network) — совокупность вычислительных средств, терминалов и каналов связи, предназначенная для совместного обращения совокупности пользователей к распределенным информационным ресурсам и вычислительным мощностям.

Сеть компьютерная — совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети. По степени географического распространения сети делятся на локальные, городские, корпоративные, глобальные и др. *Локальная сеть (ЛВС)* — связывает ряд компьютеров в зоне, ограниченной пределами одной комнаты, здания или предприятия. *Глобальная сеть (ГВС)* — соединяет компьютеры, удаленные географически на

большие расстояния друг от друга. Отличается от локальной сети более протяженными коммуникациями (спутниковыми, кабельными и др.). *Городская сеть* — обслуживает информационные потребности большого города.

Сеть локальная (Local Area Network(LAN)) — оборудование и программное обеспечение, предназначенные для комплексирования малых и средних ЭВМ для совместного использования локальных ресурсов.

Сеть передачи данных (Data Transfer Network) — комплексы средств связи и управляющих компьютеров, обеспечивающие передачу данных для различных приложений.

Символьное данное (Character) — тип данных, предназначенный для ввода и отображения алфавитной, цифровой и спецсимвольной информации; типу соответствуют определенные операции над переменными и функции (строчные).

Система команд — совокупность операций, выполняемых некоторым компьютером.

Система счисления — совокупность правил наименования и изображения чисел с помощью набора *символов,* называемых *цифрами.* Системы счисления делятся на *позиционные* и *непозиционные.* Пример непозиционной системы счисления — римская, к позиционным системам счисления относится двоичная, десятичная, восьмеричная, шестнадцатеричная. Здесь любое число записывается последовательностью цифр соответствующего алфавита, причем значение каждой цифры зависит от места (позиции), которое она занимает в этой последовательности. В непозиционных системах счисления не представляются дробные и отрицательные числа.

Сканер — устройство для ввода в компьютер документов — текстов, чертежей, графиков, рисунков, фотографий. Создает оцифрованное изображение документа и помещает его в память компьютера.

Стандартное машинное слово — машинное слово, размерность которого совпадает с разрядностью процессора. Большинство команд процессора использует для обработки данных стандартное машинное слово.

Стек — среда для размещения данных для возврата из подпрограмм, а также их аргументов и автоматических данных. Все это может потребовать достаточно большого размера стека. Как правило, программист может определять размер стека в программе.

Страничная организация памяти — организация, при которой адресное пространство памяти разбивается на малые участки-*страницы.* Используется для управления памятью в системах, работающих в защищенном режиме. Как правило, такая организация памяти подразумевает *пейджинг.*

Стример — устройство для резервного копирования больших объемов информации. В качестве носителя применяются кассеты с магнитной лентой емкостью 1—2 Гбайта и более.

Сумматор — электронная логическая схема, выполняющая суммирование двоичных чисел.

Суперкомпьютер — очень мощный компьютер с производительностью свыше 100 мегафлопов (1 мегафлоп, (Мфлоп, Mflop) — миллион операций с плавающей точкой в секунду). Представляет собой многопроцессорный и (или) многомашинный комплекс, работающий на общую память и общее поле внешних устройств. Архитектура основана на принципах параллелизма и конвейеризации вычислений.

Схема алгоритма (блок-схема) — графическое представление алгоритма в виде последовательности блоков, соединенных стрелками

Счетчик команд (СчАК, РАК) — регистр УУ, содержимое которого соответствует адресу очередной выполняемой команды; служит для автоматической выборки команд программы из последовательных ячеек памяти.

Таблица истинности — табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

Терминал (Terminal) — терминальное устройство — сочетание устройств ввода и вывода данных в ЭВМ.

Тип данных — форма представления данных, которая характеризуется: способом организации данных в памяти; множеством допустимых значений; набором операций. Иначе говоря, тип данных — это схема определенного вида переменных, заложенная в транслятор. Сама переменная — это не что иное, как область памяти программы, в которой размещены данные в соответствующей форме представления, т. е. определенного типа. Поэтому любая переменная в языке имеет раз и навсегда заданный тип. Область памяти всегда ассоциируется в трансляторе с именем переменной. Тип данных INTEGER, (`int`, FIXED и пр.) задает свойства *целой* переменной: она может принимать только целые значения в определенном диапазоне, зависящем от разрядности процессора, например, -32 767...+32 767. Все арифметические операции над целыми числами не выходят за рамки этого представления, т. е. дают также целый результат. Тип данных REAL (double, FLOAT и пр.) задает свойства переменной с *плавающей точкой.* Число с плавающей точкой (или вещественное, действительное число) может принимать значения десятичной дроби (например, 287,3) и имеет ограничения на количество значащих цифр (точность представле-

ния). Все операции над переменными данного типа не выходят за рамки этой формы представления. Если же в операции присутствуют переменные обоих типов, то первый тип преобразуется во второй (целое число в вещественное)

Топология компьютерной сети — логический и физический способы соединения компьютеров, кабелей и других компонентов, в целом составляющих сеть. Топология характеризует свойства сетей, не зависящие от их размеров. При этом не учитывается производительность и принцип работы этих объектов, их типы, длины каналов, хотя при проектировании эти факторы очень важны. Наиболее распространенные виды топологий: линейная, кольцевая, древовидная, звездообразная, ячеистая, полносвязная.

Трекболл — устройство управления курсором. Небольшая коробка с шариком, встроенным в верхнюю часть ее корпуса. Пользователь рукой вращает шарик и перемещает соответственно курсор.

***Третье** поколение компьютерной техники* — семейства программно совместимых машин с развитыми операционными системами. Обеспечивают мультипрограммирование. Быстродействие внутри семейства — от нескольких десятков тысяч до миллионов операций в секунду. Емкость оперативной памяти — до нескольких сотен тысяч слов. Элементная база — интегральные схемы.

Триггер — электронная схема, применяемая в регистрах компьютера для запоминания одного бита информации. Имеет два устойчивых состояния, которые соответствуют двоичным «1» и «О».

Устройство управления (УУ) — часть процессора, выполняющая функции управления устройствами компьютера.

*Файл **(File)*** — именованный организованный набор данных определенного типа и назначения, находящийся под управлением операционной системы. Это однородная по своему составу и назначению совокупность информации, хранящаяся на носителе информации и имеющая имя. Правила образования имен файлов и объединения файлов в файловые системы зависят от конкретной операционной системы. Например, в операционной системе MS-DOS 6.0 имя файла состоит из двух частей: собственно имени и расширения имени (т. е. типа файла). Собственно имя файла состоит из не более чем восьми символов, исключая знаки арифметических операций, пробелов, отношений, пунктуации. В качестве имен файлов запрещены имена, являющиеся в MS-DOS именами устройств, например con, `lpt1`, `lpt2`. Расширение имени может состоять не более чем из трех символов, в том числе может отсутствовать. Если расширение есть, то от основного имени оно отделяется точкой, например `pict.bmp`, `lett.txt`, `doc.doc`. По имени файла можно судить о его назначении, так как для расширений установили некоторые соглашения, фиксирующие для ОС тип обработки файлов.

Расширение com или exe имеют файлы программ, предназначенные для исполнения по вызову пользователя; doc — файлы с документами, подготовленные в текстовом редакторе Microsoft World; bak — резервные копии; bas — файлы с текстами программ на языке Бейсик.

Файл ASCII (ASCII-file) — файл, содержащий символьную информацию в коде Latin-1 и символьную разметку.

Файл бинарный (Binary File) — файл, содержащий произвольную двоичную информацию (текст с бинарной разметкой, программа, графика, архивный файл).

Файл графический (Image file) — бинарный файл, содержащий данные, обычно полученные с помощью растрового сканера и соответствующие двумерному изображению объекта.

Файл табличный (Table of data file) — файл, содержащий организованные данные, соответствующие строкам и столбцам некоторой таблицы и являющийся объектом или продуктом табличного процессора или СУБД.

Файл текстовый (Text file)— файл, содержащий символьную информацию в одном из соответствующих кодов, и коды, управляющие режимом отображения символов на печать и экранные устройства.

Файловая система (File management system) — динамически поддерживаемая информационная структура на устройствах прямого доступа (диски) и обеспечивающая функцию управления данными ОС путем указания связи «имя—адрес».

Фиксированная точка (fixed) — простейший тип числовых данных, когда число размещено в машинном слове, и диапазон значений зависит только от разрядности слова.

Флоппи-диск (дискета) — съемный гибкий магнитный диск.

Флопс (Floating Point Operations per Second — FLOPS) — мера быстродействия в операциях с плавающей точкой за секунду.

Хост машина (Host computer) — главная ЭВМ (в сети или автономно), поддерживающая информационные и вычислительные ресурсы и предоставляющая их удаленным пользователям.

Цилиндр — совокупность дорожек с одним и тем же номером, расположенных на разных поверхностях диска (для флоппи-диска под цилиндром подразумевается две дорожки). Цилиндр — пространство, доступное для записи-считывания при фиксированном положении блока головок дисковода.

Четвертое поколение компьютерной техники — современное поколение машин, разработанных после 1970 г. Эти компьютеры проектировались в расчете на эффективное использование высокоуровневых языков и упрощение процесса программирования для конечного пользователя. Элементная база — интегральные схемы. Емкость

ОЗУ — десятки Мбайт. Машины этого поколения представляют собой персональные компьютеры либо многопроцессорные и (или) многомашинные комплексы, работающие на общую память и общее поле внешних устройств. Быстродействие — до нескольких десятков — сотен миллионов операций в секунду.

Числа с плавающей точкой (Float) — числовое данное, размещенное в машинном слове в форме мантиссы и порядка, что позволяет представлять широкий диапазон значений; предполагает наличие встроенной или эмулируемой арифметики (операции с плавающей точкой). Для использования чисел с дробной частью, а также для расширения диапазона используемых числовых данных вводится форма представления вещественных чисел или чисел с плавающей точкой: $X = m \times 10^p$, например $25{,}4 = 0{,}254 \times 10^2$, где $0{,}1 < m < 1$ — значащая часть числа, приведенная к интервалу $0{,}1...1$, называемая мантиссой, а p — целое число, называемое порядком. Аналогично, если взять основание степени — 2, то получим: $X = m \times 2^p$, где $0{,}5 < m < 1$ — мантисса, а p — двоичный порядок.

Чувствительный экран — позволяет осуществлять общение с компьютером путем прикосновения пальцем к определенному месту экрана монитора.

Шина (bus) — устройство, способное управлять, по крайней мере, еще одним устройством. К шине подключаются платы адаптеров. С точки зрения подсистемы Plug & Play шиной является всякое устройство, способное обеспечивать ресурсы.

Штриховой код (бар-код) — серия широких и узких линий, в которых зашифрован номер торгового изделия. Имеет большое распространение в организации компьютерного обслуживания торговых предприятий.

Язык ассемблера — система обозначений, используемая для представления в удобочитаемой форме программ, записанных в машинном коде. Перевод программы с языка ассемблера на машинный язык осуществляется специальной программой, которая называется *ассемблером* и является, по сути, простейшим транслятором.

Глоссарий терминов (английский язык)

10BaseT *(Twisted-Pair Ethernet)* — Ethernet на витой паре. Кабель выполнен на неэкранированной витой паре UTP 3—5 категории, топология — звезда, в центре которой находится хаб (Hub). Преимущества по сравнению с Шиной: к каждому узлу подходит только один гибкий кабель; повреждение одного лучевого кабеля приводит к отказу соединения только одного узла; несанкционированное «прослушивание» пакетов в сети затруднено.

ACPI (Advanced Configuration and Power Interface) — современный интерфейс конфигурирования и управления энергопотреблением — стандарт, разработанный фирмами Intel, Microsoft и Toshiba для унификации функций управления энергопотреблением компьютера. Является ключевым элементом Operating System Directed Power Management (OSPM — непосредственное управление энергопотреблением операционной системой).

ADC (Analog Digital Convertor) — аналого-цифровой преобразователь (АЦП). Предназначен для преобразования аналогового сигнала в цифровой код, т. е. каждому значению напряжения входного аналогового сигнала соответствует определенное значение выходного цифрового кода.

Additional ROM ***BIOS*** — дополнительный ROM BIOS.

API (Applications Programmer's Interface) — интерфейс прикладного программирования — спецификация набора функций, которую должны выдержать разработчики программного обеспечения для совместимости своих программ с соответствующими операционными системами.

Arcnet — сеть магистральной или иерархической топологии, скорость — 2,5 Мбит/с, максимальное число узлов — 255, максимальная длина — 6600 м.

ARLL (Advanced RLL) — усовершенствованный RLL (метод записи на диск)

ASPI (Advanced SCSI Programming Interface) — усовершенствованный программируемый SCSI-интерфейс.

AT (Advanced Technology) — улучшенная технология.

ATX (AT extension) — расширение формата AT — конструктив корпуса персонального компьютера. Начал массово использоваться после появления процессоров Pentium II, так как системные платы для этого процессора выпускаются только в формате ATX (за очень ред-

ким исключением). Основные различия: плата крепится только на винтах, что повышает надежность и жесткость ее крепления; разъемы всех портов ввода/вывода (принтерный, COM1, COM2 и т. д.) жестко крепятся на системной плате в ее правом верхнем углу, причем могут устанавливаться друг над другом; процессор (процессоры) и его «обвязка» устанавливаются справа от интерфейсных разъемов, что позволяет устанавливать полноразмерные платы расширения; блок питания запускается/выключается от специального сигнала, который может быть выработан как кнопкой включения компьютера, так и BIOS через соответствующую схему на системной плате. Это позволяет программно выключать блок питания.

APA (All Points Adressable) — возможность адресации любого адреса.

Bank Switching — переключение банков (в EMS-памяти).

Banks — банки (памяти).

BEDO DRAM (Burst **EDO DRAM)** - EDO DRAM с групповым способом чтения.

BIOS (Basic Input Output System) — базовая система ввода-вывода.

BIOS (Plug & Play BIOS) — базовая система ввода-вывода персонального компьютера. BIOS обеспечивает интерфейс самого низкого уровня с такими устройствами, как системные часы, жесткий диск и монитор. Plug & Play BIOS дополняет функции BIOS рядом процедур, поддерживающих некоторые действия подсистемы Plug & Play, например перечисление устройств.

BIOS Setup (установка BIOS) — программа-утилита конфигурации системы.

BIU (**B**us Interface Unit) — интерфейсный блок.

bpi (bit per inch) — количество бит на дюйм.

BR (Boot Record) — загрузочная запись.

Buffered Write Through — буферизированная сквозная запись (в кэш-памяти).

BPB (BIOS Parameter Block) — блок параметров BIOS (в загрузочной записи — Boot-сектор).

Cache Level 1 (L1) — кэш первого уровня (внутренняя память процессора).

Cache Level 2 (L2) — кэш второго уровня (внешняя память, на системной плате).

Cache Memory — кэш-память (сверхоперативная память).

Cache-line — строка кэша.

Call Far — дальний вызов (в ассемблере).

CAS (Column Address Strobe) — сигнал выборки столбца адреса в DRAM.

Cache Memory (кэш-память) — память, необходимая для того, чтобы центральный процессор меньше простаивал из-за низкого быстродействия основной памяти, расположена между процессором и основной памятью. Объем и быстродействие кэш-памяти являются определяющими параметрами быстродействия системной платы и/или центрального процессора для подавляющего большинства задач, решаемых на компьютере.

CCS (Common Command Set) — набор общих команд (в SCSI-2).

CD Plus — музыкальные мультимедиа диски, содержащие две сессии — аудио (воспройзводится также на любом стандартном аудиоплейере) и CD-ROM.

CD-DA — *Digital Audio* — классический аудиодиск. Поддерживается практически всеми приводами.

CD-I — CD Interactive — видеозапись со звуковым сопровождением для воспроизведения на видеоплейере со стандартным телевизором. Некоторыми приводами не поддерживается.

CD-R (Recordable) — CD-WORM — Write Once Read Many times — CD-WO — Write Once — устройства, записывающие данные на CD. Технология записи на золотое напыление отличается от массовой (штамповки), что теоретически не влияет на считывание, однако некоторые приводы CD-ROM не читают диски с многократными сеансами записи.

CD-ROM (Compact Disk Read Only Memory) — носитель и устройство для считывания компакт-дисков (CD). Диск диаметром 5 дюймов емкостью 640—700 Мбайт имеет одну спиральную дорожку. Время доступа относительно велико (у лучших моделей — 80 нс), чувствителен к вибрациям при работе. Интерфейсы: SCSI, IDE (E-IDE, IDE ATAPI). Исполнение — внутреннее и внешнее (SCSI, LPT-порт). Могут отличаться как по поддержке различных форматов, так и по следующим возможностям: Multisession CD-ROM — позволяют считывать данные, записанные за несколько сеансов на записывающем CD-ROM (не более 9). В противном случае читаются данные только первой сессии. XA-Ready CD-ROM позволяют читать XA-диски, но не имеют собственного ADPCM-декодера. Caddy-Type — приводы, у которых CD укладывается в специальную защитную кассету (картридж), аналогичную чехлу дискеты. Сохраняет диски от внешних повреждений, но при частой смене диска желательно иметь несколько кассет. CD-changer (juke box) — устройства, в которые можно одновременно установить несколько CD, текущая работа возможна только с одним, смена текущего диска — автоматическая (1—5 с).

CD-ROM XA (extended Architecture) — расширенная архитектура, совместима с ISO 9660 и High Sierra дисками. При создании оригина-

ла используется Interleaving — чередование сегментов данных, аудио- и видеоинформации. Аудиосигнал сжимается по методу ADPCM (Adaptive Differential Pulse Code Modulation).

CD-RW (CD Rewritable) — перезаписываемые диски — оптические диски, допускающие многократную запись информации. Как правило, возможно выполнить до 1000 циклов записи на один диск.

Chipset — набор микросхем (обычно для системной платы).

CHS Standard (Cylinder, Head, Sector Standard) — формат физического адреса на магнитных накопителях.

CISC (Complex Instruction Set Computer) архитектура — компьютер со сложным набором команд (большинство современных компьютеров, вплоть до Pentium Pro).

CMOS RAM (Complimentary Metal Oxide Semiconductor Memory) — КМОП-память.

Controller — контроллер, специализированный процессор управления обменом с внешними устройствами.

Conventional Memory — стандартная (базовая) память.

CP-437 — стандарт IBM для интерпретации 2-й половины (128—256) кода ASCII, таблица предназначена для греческого алфавита.

CP-850 — стандарт IBM для интерпретации 2-й половины (128—256) кода ASCII, таблица предназначена для восточно-европейских алфавитов.

CP-852 — стандарт IBM для интерпретации 2-й половины (128—256) кода ASCII, таблица предназначена для греческого алфавита.

CP-862 — стандарт IBM для интерпретации 2-й половины (128—256) кода ASCII, таблица предназначена для иврита.

CP-S66 — стандарт IBM для интерпретации 2-й половины (128—256) кода ASCII, таблица предназначена для русской кириллицы.

CPL (Current Privilege Level) — текущий уровень привилегий.

CPS (Characters Per Second) — знаков в секунду (скорость принтера).

CPU (Central Processing Unit) — центральный процессор.

CR (Carriage Return) — возврат каретки (управляющий символ).

CRC (Cyclic Redundancy Check) — циклический избыточный контроль.

CS (Code Segment) — код сегмента (содержит начальный адрес сегмента памяти).

CYL (CYLinders) — цилиндры.

DA (Data Area) — массив (область) данных.

DAT (Digital Audio Tape) — цифровая аудиолента.

Data Sharing — разделение данных.

DD (Double Density) — двойная плотность записи (на дискетах).

DDR SDRAM (Double Data Rate SDRAM) — SDRAM с удвоенной скоростью обмена данными — вид памяти. Как видно из названия, про-

пускная способность DDR SDRAM в два раза выше обычной. Этот вид памяти также иногда называется SDRAM II.

Dedicated — выделенный режим сервера.

Device — устройство

DIMM(Dual In-line Memory Module) — двухсторонний модуль памяти — конструктив модуля памяти, ставший с 1997 г. фактическим стандартом для компьютеров. Имеет по 84 вывода с каждой стороны. Собственно память, размещенная на модуле, может быть как FPM или EDO так и SDRAM. Память в DIMM имеет разрядность 64 (с четностью 72) бита и может использоваться поодиночке, а не парами, как обычные SIMM.

Direct-mapped cache — кэш прямого отображения.

DMA (Direct Memory Access) — прямой доступ к памяти — режим обмена данными между памятью и устройством ввода/вывода, управляемый специальным устройством (контроллером DMA) и выполняемый без участия центрального процессора. Использование этого режима значительно ускоряет пересылку данных, так как исключает пересылки данных в процессор и обратно.

Dot Positions — точечные позиции (на поверхности магнитного диска).

Double-Sided — двусторонние (о дискетах).

DPL (Descriptor Privilege Level) — уровень привилегий дескриптора.

DPMI (DOS-интерфейс защищенного режима) — старый способ, благодаря использованию которого могли работать 32-разрядные программы защищенного режима.

Drag-and-Drop («захватить-и-перетащить», «перетащить-и-отпустить», «буксировать» и пр.) — элемент технологии интерфейсов WIMPD, состоящий из следующих действий — «захват» экранного объекта (ярлык, имя файла и пр.) с помощью указателя «мыши», «буксировка» к месту назначения на экране при нажатой клавише, «сбрасывание» объекта при отпускании клавиши.

DRAM (Dinamic Random Access Memory) — динамическая память прямого доступа — память, схемотехнически выполненная в виде двумерной матрицы (строки х столбцы) конденсаторов. Достаточно дешева, но требует постоянной регенерации *(refresh)* заряда на конденсаторах. Регенерация выполняется как «пустое» чтение памяти. Этот процесс отнимает значительное время, так как в этот период никакое устройство не может получить доступ к памяти, кроме контроллера регенерации.

DVD (Digital Versatile Disk) — цифровой универсальный диск — современный стандарт хранения информации на оптическом (лазерном) диске. Отличается от обычного CD-ROM увеличенной почти в 30 раз емкостью (до 17 Гбайт). Возможны следующие варианты изготовления DVD дисков: односторонний однослойный с емкостью

4,7 Гбайт; односторонний двухслойный с емкостью 8,5 Гбайт; двухсторонний однослойный с емкостью 9,4 Гбайт; двухсторонний двухслойный с емкостью 17 Гбайт.

DVD-1 — условное название первого поколения приводов для DVD дисков. Имеют скорость чтения обычных CD-ROM дисков не выше 8-х и, кроме этого, не могут читать диски CD-R и CD-RW.

DVD-2 — условное название второго поколения приводов для DVD дисков Имеют скорость чтения обычных CD-ROM дисков до 24-х и, кроме этого, могут читать диски CD-R и CD-RW

DVD-Audio — стандарт на аудиодиски — за счет увеличенной емкости диска увеличена частота дискретизации и разрядность. Кроме этого, звук может быть записан объемным (трехмерным).

DVD-ROM — диск, доступный только для чтения; может считываться только на приводе DVD. DVD-Video предназначен для записи видеофильмов и может воспроизводиться как в приводах DVD в компьютерах, так и в DVD плейерах

DVI(Digital Video Interactive) — система аппаратного сжатия движущихся видеоизображений с коэффициентом сжатия до 160:1 и записи звукового сопровождения по методу сжатия ADPCM

ECC (Error Control Correction) — режим контроля функционирования памяти с восстановлением ошибок. Применяется в некоторых системных платах с соответствующими наборами микросхем для особо критичных к надежности функционирования компьютеров. В основном это серверы и мощные графические станции с большими объемами памяти. В этом режиме возможно исправление только одиночных ошибок, т. е. ошибку в одном бите из 64, которые считывает процессор.

ECP (Enhanced Capability Port) — порт с расширенными возможностями, отличается от стандартного принтерного порта с интерфейсом Centronics тем, что передаваемая информация разделяется на команды и данные с поддержкой режима DMA и кодирования по методу RLE (Run-Length Encoding — кодирование повторяющихся последовательностей данных).

ED (Extra High Density) — сверхвысокая плотность записи (на дискетах).

EDO DRAM (Extended Data Output DRAM) — DRAM с уменьшенным временем доступа к данным Extended Data Output DRAM при работе на запись не дает никаких преимуществ по сравнению с FPM DRAM, но требует существенно меньше времени при чтении за счет того, что данные удерживаются на выходе микросхемы EDO DRAM дольше, чем в FPM DRAM. EDO (как и FPM) не дает никаких преимуществ при произвольной выборке данных из памяти, но такая ситуация возникает крайне редко. На микросхемах EDO

DRAM указывается время доступа в наносекундах к данным в случайном порядке.

EDRAM (**E**nhanced**DRAM)** — усовершенствованная DRAM.

EEPROM(**E**lectrically Erasable Programmable **ROM)** — электрически стираемое программируемое ПЗУ.

EIDE-контроллер(Enhanced **IDE)** — улучшенный EIDE-контроллер.

EISA (Extended Industry Standard Architecture) — устройство *шины,* которое позволяет использовать 32-разрядные адаптеры и допускает некоторое автоматическое распознавание и конфигурирование устройств.

EMM(Expanded Memory Manager) — менеджер дополнительной памяти.

EMS (Expanded Memory Specification) — спецификация дополнительной памяти.

EPP (Enhanced Parallel Port) — расширенный параллельный порт — двунаправленный вариант принтерного порта с максимальной скоростью приема/передачи данных до 2 Мбайт/с. Стала возможной адресация нескольких устройств, ввод 8-разрядных данных. Для буферизации данных используется память с FIFO организацией объемом в 16 байт.

EPROM(Erasable Programmable **ROM)** — вид программируемого ПЗУ, стираемого ультрафиолетовым излучением.

ESDI-контроллер (Enhanced Small Device Interface) — улучшенный интерфейс малых устройств.

Ethernet — сеть с общей шиной, скорость — 10 Мбит/с, максимальная длина сегмента — 500 м, максимальная длина сети — 2500 м, максимум 100 компьютеров на сегмент.

EP (Even Parity) — четный бит контроля четности.

EPA (Environment Protection Agency) — агентство защиты окружающей среды.

ECP (Extended Capabilities Port) — порт с расширенными возможностями для лазерных принтеров.

FAT (**F**ile Allocation Table) — таблица размещения файлов.

FDD (Floppy **Disc** Drive) — накопитель на гибких магнитных дисках.

FDDI(Fiber Distributed Data Interface) — стандартизированная спецификация ANSI X3T9.5 для сетевой архитектуры высокоскоростной передачи данных в основном по оптоволоконным линиям. Скорость передачи — 100 Мбит/с. Топология — либо кольцо (двойное), либо гибридная (включение звездообразных или древовидных подсетей в главную сеть через концентратор). Метод доступа — маркерный с возможностью одновременного циркулирования множества кадров в кольце. Максимальное количество станций — 1000, расстояние между станциями до 2 км при многомодовом и до 45 км при одномодовом кабеле, затухание сигнала между станциями 11 дБ, длина

кольца до 100 км (может увеличиваться за счет применения дополнительных повторителей). В некоторых случаях вторичное кольцо используется для удвоения пропускной способности. Реализация сети в настоящее время стоит достаточно дорого

Feature Connector ~ разъем для подключения дочерних плат.

FH (Full Height) — один из параметров форм-фактора.

FIFO (First Input First Output) «первым вошел — первым вышел» — способ организации памяти, при котором записанные данные сдвигаются вперед при поступлении новых данных. Как правило, память с такой организацией используется в качестве буферной при приеме/передаче данных.

Flash Memory — флеш-память, хранящая информацию BIOS при выключенном ПК.

Flops (floating point operation per second) — быстродействие в количестве операций с плавающей точкой, выполняемых в секунду. Производные единицы — Mflops (10^6 оп/с), Gflops (10^9 оп/с), Tflops (10^{12} оп/с). Употребляются также транслитерации: флопс, Мфлопс, Гфлопс, Тфлопс.

Form-factor — стандартизированные габаритные размеры (горизонтальные и вертикальные) винчестеров и других устройств.

FPM (Fast Page Mode) — быстрый страничный метод (вид памяти DRAM).

FPU (Floating Point Unit) — устройство для арифметических операций с плавающей точкой.

Frame Grabbing — оцифровка и сохранение отдельного кадра.

Fully associative cache — полностью ассоциативная кэш-память.

GDT (Global Descriptor Table) — глобальная дескрипторная таблица.

Hayes — система команд для обмена данными между модемами, названная по имени фирмы, ее разработавшей.

HD (High Density) — высокая плотность записи (на дискетах).

HD-CD (High Density CD) — CD высокой плотности.

HDD (Hard Disc Drive) — накопитель на жестких магнитных дисках (винчестер).

Heads (HD) — магнитные головки в накопителях на магнитных дисках.

High Sierra Format (*HSF*, или HSG — High Sierra Group) — фактический стандарт на доступ к данным из среды DOS, UNIX и других ОС. Начальная дорожка содержит информацию об организации диска — VTOC (Volume Table Of Content).

Hub — хаб является обязательным (кроме двухточечной сети) соединительным элементом сети на витой паре и средством расширения топологических, функциональных и скоростных возможностей для любых сред передачи. Простейшие хабы являются многопортовыми

повторителями. Хабы могут иметь набор разъемов BNC, RJ-45, AUI, обеспечивая выбор кабеля для передачи от источника к приемнику. К порту хаба можно подключить как отдельный узел, так и другой хаб. Хабы с набором разнотипных портов позволяют объединять сегменты сетей с различными кабельными системами. Существуют более сложные и дорогие варианты Switched HUB, Stackable HUB.

iCOMP (Intel Comparative Microprocessor Performance) — тест для вычисления производительности микропроцессоров.

ICW (Initialization Command Words) — слова команд инициализации (ICW1—ICW4) в контроллере прерываний.

IDE-контроллер (Integrated Drive Electronics) — интегрированная электроника накопителя.

IEEE (Institution of Electrical and Electronical Engineers) — общество инженеров по электротехнике и радиоэлектронике (США).

IFS (Installable File System) — инсталлируемая файловая система (в Windows 95).

Impact — ударный (принтер).

Interleaving — интерливинг (чередование, перекрытие).

IP (Instruction Pointer) — указатель команды (регистр в МП, хранящий адрес выполняемой команды).

IPL2 (Initial Program Loading 2) — программа начальной загрузки (инициализации).

IrDA (Infrared Data Association) — инфракрасный порт приема-передачи данных (для принтеров Hewlett Packard).

ISA (Industry Standard Architecture) — стандарт системной шины.

ISO 9660 — первый стандарт (1988 г.) для хранения данных на CD-ROM, файловая система аналогична MS-DOS, имена файлов по схеме: 8 символов — имя, 3 символа — расширение имени, глубина вложенности каталогов до 8.

JPEG (Joint Photographic Expert Group) — метод сжатия неподвижных изображений, основанный на одновременной обработке информации матрицы пикселей (например, 8 x 8) в пространстве Y-U-V с приоритетом сохранения яркостной информации. Изображение восстанавливается с некоторыми небольшими потерями качества. Степень сжатия зависит от характера изображения и размера матрицы квантования. Этот формат стал фактическим стандартом в Internet для обмена изображениями, в основном фотографиями.

LAN (Local Area Network) — локальная сеть.

Latin-1 — международный стандарт (ISO-8859-1) для интерпретации 2-й половины (128—256) кода ASCII, таблица предназначена для латиницы.

Latin-8 — международный стандарт (ISO-8859-8) для интерпретации 2-й половины (128—256) кода ASCII, таблица предназначена для иврита

Latin-C — международный стандарт (ISO-8859) для интерпретации 2-й половины (128—256) кода ASCII, таблица предназначена для кириллицы.

LBA (Logical Block Address) — логический адресный блок.

LCD (Liqid Crystal Display) — жидкокристаллический дисплей.

LDT (Logical Drive Table) — таблица логического диска (при разбиении HDD).

LED-принтеры (Light Emitting Diode) — светодиодный принтер.

LF (Line Feed) — перевод строки (управляющий символ).

LIFO (Last Input First Output) — «последним вошел — первым вышел» (дисциплина обслуживания стековой памяти).

LQ (Letter Quality) — машинописное качество (печати).

MBR (Master Boot Record) — главная загрузочная запись (при разбиении дисков на разделы)

MDRAM (Multibank DRAM) — мультибанковая DRAM.

Memory Refresh — регенерация памяти.

MIDI (Musical Instruments Digital Interface) — стандарт на язык и аппаратуру представления звуков различных инструментов. Команды MIDI сообщают аппаратуре, у какого инструмента, на какой октаве и какая нота должна звучать. Поэтому запись мелодии в MIDI-командах очень компактна. Существует много разновидностей этого стандарта — General MIDI, Roland MT-20 и т. д.

MIPS (Millions Instruction Per Second) — быстродействие в миллионах операций в секунду.

M-JPEG (Motion JPEG) — метод сжатия для обработки движущихся изображений. Используется в ряде устройств среднего (по стоимости) уровня для ввода в компьютер видеоинформации.

Movie Grabbing — оцифровка и сохранение «живого» видео.

MPEG (Motion Picture Expert Group) — организация-разработчик стандартов на типы кодирования видео- и аудиосигналов.

MPEG-1 — также стандарты ISO/IES 11172 — тип кодирования видеоизображения и/или звука, позволяющий при потоке данных на уровне 1,5 Мбит/с (170 Кбайт/с) передавать изображение с качеством бытового кассетного видеомагнитофона стандарта VHS (Video Home System) со стереофоническим звуковым сопровождением. Исходное изображение — 352 x 240 пикселей, 30 кадров в секунду. В стандарт также входит программная реализация кодера и декодера на языке С. Низкая скорость потока данных позволяет использовать в качестве носителя видеоинформации обыкновенный четы-

рех- и более скоростной CD-ROM. Диски в MPEG-1 формате обычно обозначаются как Video CD.

MPEG-2 — ISO/IEC 13818 — стандарт на кодирование для высококачественной передачи и хранения изображений в вещательном формате (720 x 480 пикселей), аудиоинформации и данных при потоке 28 Мбит/с (3,5 Мбайт/с). Стандарт предусматривает одновременную передачу множества TV-каналов с возможностью шифрования для ограничения доступа к информации. Допускается многоканальная передача аудиоданных (два канала аудиопотока MPEG-2 эквивалентны потоку MPEG-1). Этот формат пока не имеет массового применения, но с появлением DVD накопителей CD-ROM началось расширение сферы его использования.

MPR-II — стандарт безопасности мониторов, разработанный Национальной лабораторией измерения и тестирования Швеции в 1987 г. Стал активно поддерживаться производителями мониторов с 1990 г. Этим стандартом устанавливается максимальный уровень напряженности электрического поля 2,5 В/м на расстоянии 50 см от монитора.

MS DOS — однопользовательская операционная система, обслуживающая семейство IBM PC-совместимых ПЭВМ.

Multiword DMA Mode — метод многословного прямого доступа к памяти.

NIC (Network Interface Card) — сетевая интерфейсная карта (для сетевых лазерных принтеров),

NLQ (Near Letter Quality) — почти машинописное качество печати (хуже, чем машинописное).

Non dedicated — невыделенный режим сервера.

NorthBridge — северный мост — принятый среди изготовителей chipset (наборов микросхем системной платы) термин, обозначающий системный контроллер, в который входит контроллер системной шины, шин AGP и PCI, памяти и кэш-памяти (для наборов под обычный Pentium). Как правило, это одна микросхема и именно по ней называется весь набор.

OCR (Optical Character Recognition) — оптическое распознавание символов.

OCW (Operation Control Words) — команды управления рабочим режимом (OCW1—OCW3) в контроллере прерываний.

OEM (Original Equipment Manufacturer) — изготовитель оригинального оборудования.

OP (Odd Parity) — нечетный бит контроля четности.

OTROM (One Time Programmable ROM) — однократно программируемая ROM.

Page Frame — окно (страница) при организации EMS-памяти.

***Paging Mode* — страничный режим.**

Partition Table — таблица разделов (MBR, HDD).

PCI bus — разработанная Intel шина, которая предназначена для поддержки высокоскоростного 32-разрядного обмена данными между устройствами, памятью и процессором Подсистема Plug & Play полностью поддерживает PCI.

PCL (Printer Control Language) — язык управления принтером.

PCMCIA (Personal Computer Memory Card International Association) — международная ассоциация карт памяти для персональных компьютеров.

PD/CD — комбинированный накопитель, записывающий диски на специальный носитель (и их же считывающий) по методу изменения фазы вещества. Пока значительного распространения не имеет и вряд ли будет иметь. Устройство также может читать и обычные CD.

Peer to peer — одноранговая сеть.

PhotoCD — стандарт, разработанный фирмой Kodak для хранения высококачественных изображений.

PIO Mode (Processor Input-Output Mode) — метод процессорного ввода-вывода в FATA-интерфейсе.

Pipeline Burst — конвейерная пакетная (тип кэш-памяти).

Pits — впадины, углубления (на компакт-диске).

Pixel (Picture Element) — пиксель (точка экрана монитора в графическом режиме).

Plug and Play, PnP («включай и работай») — технология современных BIOS, стандарт для устройств, позволяющий работать с ними практически сразу после вставки в слот расширения (применяется в Windows 95-XP).

POP — ассемблерная команда считывания из стека.

POST (Power-On Self Test) — автотест при включении питания ПК.

Primary — первичный (раздел в HDD).

PRML (Portal Response Maximum Likelihood) — максимальная вероятность приоритетной реакции (в дисках для фильтрации пиков аналоговых сигналов при детектировании сигналов от магнитных головок).

Programmable ROM — программируемая ROM.

Protected Mode — защищенный режим (в МП).

Proximity — головки близкого расположения.

PSRAM — псевдостатическая память.

PUSH — ассемблерная команда записи в стек.

QIC (Quarter Inch Cartridge) — картридж для МЛ шириной в 0,25".

RAID (Redundant Array of Independent Discs) — система надежного хранения больших массивов информации с использованием минимальной избыточности (в серверах).

RAM (Random Access Memory) — оперативное запоминающее устройство (с произвольным доступом).

RAS (Row Address Strobe) — строб адреса строки в микросхеме DRAM или же сигнал в памяти

RDRAM (Rambus DRAM) — одна из разновидностей DRAM.

Read-write *head* — головка считывания-записи.

Real Mode — реальный режим (в МП).

Remote Boot ROM — удаленный загрузчик.

Repeater — репитер (устройство для удлинения сети).

Resolution — разрешающая способность.

Resource Sharing — разделение ресурсов.

Ret Far — дальний возврат.

RTC (Real Time Clock) — часы реального времени (один из блоков информации, хранимый в CMOS RAM).

RISC (Reduced Instruction Set Computer) архитектура — компьютер с сокращенным набором команд.

RLE алгоритм (Run Length Encoding) — алгоритм кодирования переменной длины.

RLL (Run Length Limited) — ограниченная длина неперемагничиваемых участков дорожки (в накопителях на жестком диске).

ROM (Read Only Memory) — постоянное запоминающее устройство (только для чтения).

RPL (Requested Privilege Level) — заявленный (запрошенный) уровень привилегий (в виртуальной памяти, одно из полей селектора сегмента).

RS-232 порт (RS-232-C serial port) — стандарт (протокол, формат) соединения ЭВМ с последовательными внешними устройствами.

RSM (Resume) — команда возобновления работы МП.

SCAM (SCSI Configuration AutoMatically) — автоматизированная настройка SCSI.

SCSI (Small Computer System Interface) — интерфейс малых компьютерных систем.

SD (Single Density) — одинарная плотность записи (на дискетах).

SDRAM (Synchronous DRAM) — синхронная DRAM. Вид памяти, который имеет преимущества только при последовательной выборке данных из памяти. Но при последовательной выборке (или потоке — burst) чтение/запись выполняются в 2 раза быстрее, чем для EDO DRAM. На микросхемах SDRAM указывается время доступа к данным в наносекундах при последовательной выборке. Реально же цифры на корпусах микросхем синхронной памяти фактически сообщают максимальную тактовую частоту системной шины, на которой данная память может работать. SDRAM выпускается сейчас

только в 168-выводных 64-разрядных модулях DIMM. В отличие от обычных модулей SIMM эти могут устанавливаться на системной плате поодиночке. В соответствии со стандартом JEDEC на модуле DIMM должна быть установлена специальная микросхема SPD-устройства. Некоторые современные системные платы, например фирмы Intel на наборе (chipset) 440LX, не запускаются, если установлена плата памяти без SPD. Микросхемы SDRAM так же широко используются в качестве локальной памяти видеокарт.

Semi-contact — полуконтактные магнитные головки.

Set-associativecache — наборно-ассоциативная кэш-память.

Setup — установка, программа установки, программа CMOS Setup.

SFFC (Small Form Factor Committee) — комитет по малым форм-факторам.

SFNM (Special Fully Nested Mode) — специальный полный вложенный режим (в контроллере прерываний).

SGRAM — синхронная графическая память — разновидность обычной синхронной памяти, применяемая в качестве локальной памяти на видеокартах. Отличается наличием регистра страницы, который позволяет выполнять запись по нескольким адресам одновременно, что дает возможность быстрого заполнения областей экрана или очистки их же.

SIMD (Single Instruction Multiple Data) — одна команда с множеством данных; (1) один из классов компьютеров по Флинну; (2) основной принцип построения MMX-команд для микропроцесоров. Эти команды в качестве операндов используют регистры сопроцессора, предназначенные для хранения операндов в 80-разрядной сетке.

SIMM (Single In-line Memory Module) — односторонний модуль памяти. Имеет 72 вывода с каждой стороны, но пары выводов с одной и другой стороны замкнуты между собой, поэтому они считаются односторонними. Собственно память, размещенная на модуле, может быть как FPM, так и EDO. Память в SIMM имеет разрядность 32 (с четностью — 36) бита и может использоваться в компьютерах с процессорами Pentium только парами.

SMM (System Management Mode) — режим управления системой (в i4086 и Pentium).

SMP (Symmetric Multi-Processing) — симметричная многопроцессорная обработка — метод, позволяющий более чем одному процессору распределять между собой вычислительную нагрузку. Intel Pentium и Pentium II поддерживают такой режим только для двух процессоров, Pentium Pro — для четырех.

SMRAM (System Management RAM) — память для управляющей системы.

SMT (Surface Mounting Technology) — технология поверхностного монтажа.

Socket Services — гнездо обслуживания.

SouthBridge — южный мост — обозначение периферийного контроллера в chipset (наборе микросхем), включающего обычно контроллер EIDE, клавиатуры, моста PCI-to-PCI, последовательных/параллельных портов, шины USB и других подобных устройств.

SP (Stack Pointer) — указатель стека (один из регистров-указателей в МП).

SPD (Serial Presence Detect) — устройство определения присутствия с последовательным доступом, изготавливается на специальной микросхеме (как правило, это электрически перепрограммируемая память), содержащей информацию о типе устройства и его основных характеристиках. Объем такой памяти — 512 байт.

SPP (Standard Parallel Port) — стандартный параллельный порт — классический принтерный интерфейс, называемый, как правило, Centronics, по имени давно ликвидированной фирмы, разработавшей этот интерфейс. Интерфейс позволяет передавать данные по байту со скоростью до 80 кБайт/с. При необходимости приема данных можно использовать четыре линии сигнала от принтера (обрыв бумаги, буфер принтера полон и т. д.).

SPT (Sectors Per Track) — количество секторов на дорожку.

SRAM (Static RAM) — статическая RAM.

stack — стековая память.

***Stackable** Hub* — наращиваемый хаб, имеет специальные средства соединения нескольких хабов в стек, выступающий в роли единого целого. При этом обычно интеллектуальность одного хаба делает интеллектуальным весь стек. Расстояние между хабами в стеке может быть коротким (локальный стек) и длинным, до сотен метров (распределенный стек, более гибкий элемент для оптимизации кабельной системы).

SVGA (Super VGA) — развитие стандарта VGA.

swap — обмен сегментов.

Switched Hub — коммутирующий хаб — дальнейшее развитие технологии Ethernet, повышающее производительность работы сети. В этом случае управление доступом к среде практически переносится с узлов в центральное коммутирующее устройство, обеспечивающее установление виртуальных выделенных каналов между парами портов — источниками и получателями пакетов. От узлов-передатчиков коммутирующий хаб почти всегда готов принять пакет либо в свой буфер, либо практически без задержки передать его в порт назначения (коммутация с таким хабом двух компьютеров, обменивающиеся «на лету» — On-the-fly Switching). Используя обмен данными между собой через коммутирующий хаб, компьютеры

не будут загружать общий трафик. Такие хабы также применяются для соединения между собой сетей Ethernet и Fast Ethernet.

Synchronous Burst DRAM — синхронная пакетная (тип кэш-памяти).

Tag — тег (ярлык, признак).

Target Buffer — буфер кэширования блоков.

TCO 92, 95 — стандарты, утвержденные Федерацией профсоюзов Швеции (The Swedish Confederation of Professional Employees). По сравнению с MPR-II устанавливают более жесткие нормы на излучение от мониторов. Максимально допустимый уровень напряженности электрического поля установлен в 1 В/м на расстоянии в 30 см. Это в несколько раз более строгие требования, чем в MPR-II. Стандарт TCO 95 предъявляет такие же требования по излучению, но обязывает также изготавливать монитор из материалов, подлежащих вторичной переработке и не наносящих вред окружающей среде Еще более жесткие требования по излучению введены в новом стандарте TCO 99, в котором по сравнению с TCO 95 ужесточены следующие требования: минимально допустимая частота кадров не менее 85 Гц; уменьшен вдвое уровень потребления электроэнергии в режиме Standby; время восстановления из режима Standby в рабочий не более 3 с. Стандарт впервые предъявляет жесткие требования к качеству самого изображения — должна быть минимальной расфокусировка изображения по углам экрана по отношению к центру, оговаривается уровень отражения света от экрана (блики) и т. п.

TI (Table Indicator) — индикатор таблицы в виртуальной памяти как селектора сегмента.

Token Ring — сеть кольцевой топологии, максимальная скорость 4 Мбит/с, максимальное число узлов 260.

TouchPad — панель, чувствительная к касанию — специальная панель, размером приблизительно 6 × 6 см, заменяющая мышь. Панель отслеживает как перемещение пальца, так и нажатие им (щелчок). Применяется в мобильных компьютерах и встраивается в некоторые модели клавиатур.

TPI (Track Per Inch) — количество дорожек на дюйм.

Track — дорожка (на магнитном диске).

TrackBall — шар с отслеживаемым перемещением — специальное устройство в виде шара и двух или трех кнопок, служащих для замены мыши. Вращение шара пальцем эквивалентно перемещению мыши.

Transceiver — устройство для подключения ПК к толстому кабелю Ethernet.

UMA (Unified Memory Architecture) — унифицированная архитектура памяти, используется в недорогих видеокартах, размещенных, как правило, на системных платах. Принцип действия основан на использовании видеокартой обычной памяти компьютера как памяти

экрана и соответственно удешевления компьютера. При этом выделенная для видеокарты память «не видна» операционной системе типа DOS и доступ к ней возможен только через BIOS видеокарты, поэтому такие видеокарты значительно медленнее обычных PCI карт. Под операционными системами типа Windows разница существенно меньше, но она все равно есть.

UMB(Upper Memory Block) — блок верхней памяти.

UNIX — операционная система универсальных ЭВМ коллективного пользования.

UPS (Uninterruptible Power Supply) — устройство, обеспечивающее бесперебойное питание ПК.

UV-EPROM (Ultra Violet Erasable PROM) — стираемая ультрафиолетом перепрограммируемая память

VFAT (Virtual File Allocation Table) — виртуальная таблица размещения файлов (в Windows 95).

VGA (Video Graphics Array) — видеографический массив (название стандарта).

Video CD — высококачественная цифровая видеозапись в формате MPEG, может воспроизводиться на компьютере с программным или аппаратным MPEG-декодером.

VideoRAM — RAM видеоадаптера (двухвходовая).

VLIW(Very Long Instruction Word) — сверхдлинные команды процессора.

VTOC (Volume Table of Contents) — таблицы содержания тома (системная часть CD ROM).

Wavetable-память — ROM для хранения образцов звуковых сигналов (в звуковых картах).

WIMPD(Windows, Menu, Pointng Device — окна, меню, указывающее устройство) — аббревиатура, обозначающая графические интерфейсы (как перечень основных «действующих лиц» в подобном интерфейсе).

WORM (Write Once Read Many) — память с одноразовой записью и многоразовым чтением (на компакт-дисках).

WRAM(Window RAM) — двухвходовая оконная RAM (используется в видеосистемах).

Write Back — обратная запись; термин применяется при описании устройств кэш-памяти. Если установлен режим обратной записи, то в случае изменения данных, находящихся в кэш-памяти, процессор меняет их только в кэше, но не в основной памяти. Только при замене одной области данных в кэше на другую процессор сохраняет данные из кэш-памяти в основную память. Реально это означает, что данные кэшируются на запись и на чтение, и именно такой режим применяется сейчас в подавляющем большинстве компьютеров.

Write Combining — объединенная запись — термин применяется при описании устройств кэш-памяти и означает накопление записываемой информации в кэш-памяти с последующим «выстреливанием» готового пакета данных на шину Этот режим позволяет ускорить запись информации, например в память видеокарты.

Write Through — сквозная запись; термин применяется при описании устройств кэш-памяти. Если установлен режим сквозной записи, то в случае изменения данных, находящихся в кэш-памяти, процессор одновременно меняет их как в кэше, так и в основной памяти. Реально это означает, что данные кэшируются только по чтению, поэтому этот способ кэширования применяется в редких случаях при сознательной необходимости.

WT MusicSynthesizer — синтезаторы с табличным синтезом (Wave Table), хранящие в своей постоянной памяти образцы (волновые таблицы — цифровые последовательности выборок) сигналов настоящих «живых» инструментов для нескольких нот диапазона по каждому инструменту. Как правило, минимальный объем таблицы — 1 Мбайт. На многих звуковых картах возможна установка дополнительной памяти для загружаемых таблиц, которые могут быть созданы самим пользователем. Качество волнового синтеза высокое, но при более высокой цене.

XIP-механизм (eXecute In Place) — исполнять на месте (в PC Card).

XMS (extended Memory Specification) — спецификация расширенной памяти.

Zone bit Recording — зонно-секторная запись (на жестких дисках).

Хронология информатики и вычислительной техники

История счетных устройств насчитывает много веков. Ниже в хронологическом порядке приводятся некоторые наиболее значимые события этой истории, их даты и имена участников.

/// в. — счеты с передвигающимися костяшками позволили ускорить вычисления.

Около 500 г. н. э. Изобретение счетов (абака) — устройства, состоящего из набора костяшек, нанизанных на стержни.

1614 г. Шотландец Дж Непер изобрел логарифмы. Вскоре после этого Р. Биссакар создал логарифмическую линейку. В счетном устройстве операция умножения производилась путем сложения чисел, расположенных в прилегающих друг к другу сегментах.

1642 г. Французский ученый Б. Паскаль приступил к созданию арифметической машины — механического устройства с шестернями, колесами, зубчатыми рейками и т п. Она умела «запоминать» числа и выполнять элементарные арифметические операции.

1673 г. Калькулятор Лейбница ускорил выполнение операций умножения и деления (механический калькулятор).

1804 г. Французский инженер Жаккар изобрел перфокарты для управления автоматическим ткацким станком, способным воспроизводить сложнейшие узоры Работа станка программировалась колодой перфокарт, каждая из которых управляла одним ходом челнока.

1822 г. Разностная машина английского ученого Ч. Бэббиджа предназначалась для расчетов математических таблиц.

1834 г. Ч. Бэббидж составил проект «аналитической» машины, в которую входили: устройства ввода и вывода информации, запоминающее устройство для хранения чисел, устройство, способное выполнять арифметические операции, и устройство, управляющее последовательностью действий машины.

1876 г. Английский инженер А Белл изобрел телефон.

1890 г. Американский инженер Г. Холлерит создал статистический табулятор, в котором информация, нанесенная на перфокарты, расшифровывалась электрическим током. Табулятор использовался для обработки результатов переписи населения в США.

1892 г. Американский инженер У. Барроуз выпустил первый коммерческий сумматор.

1897 г. Английский физик Дж Томсон сконструировал электронно-лучевую трубку.

1901 г. Итальянский физик Г. Маркони установил радиосвязь между Европой и Америкой

1904—1906 гг. Сконструированы электронные диод и триод.

1930 г. Профессор Массачусетского технологического института (МТИ) В. Буш построил дифференциальный анализатор, с появлением которого связывают начало современной компьютерной эры. Это была первая машина, способная решать сложные дифференциальные уравнения, которые позволяли предсказывать поведение таких движущихся объектов, как самолет, или действие силовых полей, например гравитационного поля.

1936 г. Английский математик А. Тьюринг и независимо от него Э Пост выдвинули и разработали концепцию абстрактной вычислительной машины. Они доказали принципиальную возможность решения автоматами любой проблемы при условии возможности ее алгоритмизации.

1937 г. Дж. Стибиц построил двоичный сумматор (электромеханическая схема, выполняющая операцию двоичного сложения).

1938 г. Немецкий инженер К. Цузе построил первый чисто механический компьютер.

1938 г. Американский математик и инженер К Шеннон показал возможность применения аппарата математической логики для синтеза и анализа релейно-контактных переключательных схем

1939 г. Американец болгарского происхождения профессор физики Дж. Атанасофф создал прототип вычислительной машины на базе двоичных элементов.

1941 г. К. Цузе сконструировал первый универсальный компьютер на электромеханических элементах. Он работал с двоичными числами и использовал представление чисел с плавающей запятой.

1943 г. Построенная на электронных вакуумных лампах специализированная машина, работающая по принципу «машины Тьюринга» (она имела название «Колосс») принялась за работу, пытаясь расшифровать секретные немецкие коды.

1944 г. Под руководством американского математика Г. Айкена создана автоматическая вычислительная машина «Марк-1» с программным управлением, в которой использовались электромеханические реле, программы обработки данных записывались на перфорационной ленте, а данные вводились в машину в виде десятичных чисел, закодированных на перфорационных картах.

1945 г. Дж. фон Нейман в отчете «Предварительный доклад о машине Эдвак» сформулировал основные принципы работы и компоненты современных компьютеров.

1946 г. Американцы Дж. Эккерт и Дж. Моучли сконструировали первый электронный цифровой компьютер «Эниак» (Electronic Numerical Integrator and Computer). Машина имела 20 тысяч электронных ламп и 1,5 тысячи реле. Она работала в тысячу раз быстрее, чем «Марк-1», выполняя за одну секунду 300 умножении или 5000 сложений.

1948 г. В американской фирме Bell Laboratories физики У. Шокли, У. Браттейн и Дж. Бардин создали транзистор. За это достижение им была присуждена Нобелевская премия.

1948 г. Н. Винер опубликовал книгу «Кибернетика, или управление и связь в животном и машине», оказавшую влияние на все последующие исследования в области искусственного интеллекта.

1949 г. В Англии под руководством Мориса Уилкса построен первый в мире компьютер с хранимой в памяти программой ЭДСАК (EDSAC).

1951 г. Вступил в строй первый коммерческий компьютер «Лео».

1951 г. В Киеве построен первый в континентальной Европе компьютер МЭСМ (малая электронная счетная машина), имеющий 600 электронных ламп. Создатель — С. А. Лебедев.

1951—1955 гг. Благодаря деятельности российских ученых С. А. Лебедева, М. В. Келдыша, М. А. Лаврентьева, И. С. Брука, М. А. Карцева, Б. И. Рамеева, В. С. Антонова, А. Н Невского, Б. И. Буркова и руководимых ими коллективов Советский Союз вырвался в число лидеров вычислительной техники, что позволило в короткие сроки решить важные научно-технические задачи овладения ядерной энергией и исследования Космоса.

1952 г. Под руководством С. А. Лебедева в Москве построен компьютер БЭСМ-1 (большая электронная счетная машина), в то время самая производительная машина в Европе и одна из лучших в мире.

1953 г. Дж. Форрестер реализовал оперативную память на магнитных сердечниках (core memory), которая существенно удешевила компьютеры и увеличила их быстродействие. Память на магнитных сердечниках широко использовалась до начала 70-х гг. На смену ей пришла память на полупроводниковых элементах.

1955—1959 гг. Российские ученые А. А. Ляпунов, С С. Камынин, Э. 3. Любимский, А. П. Ершов, Л. Н. Королев, В. М. Курочкин, М. Р. Шура-Бура и др. создали «программирующие программы» — прообразы трансляторов. В.В. Мартынюк создал систему символьного кодирования (ССК) — средство ускорения разработки и отладки программ.

1955—1959 гг. Заложен фундамент теории программирования (А. А. Ляпунов, Ю. И. Янов, А. А. Марков, Л. А. Калужин) и численных методов (В. М. Глушков, А. А. Самарский, А. Н. Тихонов). Моделиру-

ются схемы механизма мышления и процессов генетики, алгоритмы диагностики медицинских заболеваний (А. А. Ляпунов, Б. В. Гнеденко, Н. М. Амосов, А. Г. Ивахненко, В. А. Ковалевский и др.).

1957 г. Первое сообщение о языке Фортран (Дж. Бэкус).

1958 г. Дж. Килби из фирмы Texas Instruments создал первую интегральную схему.

1957 г. Американской фирмой NCR создан первый компьютер на транзисторах.

1959 г. Под руководством С. А. Лебедева создана машина БЭСМ-2 производительностью 10 тыс. опер./с. С ее применением связаны расчеты запусков космических ракет и первых в мире искусственных спутников Земли.

1959 г. Создана машина М-20, главный конструктор С. А. Лебедев. Для своего времени одна из самых быстродействующих в мире (20 тыс опер./с). На этой машине было решено большинство теоретических и прикладных задач, связанных с развитием самых передовых областей науки и техники того времени. На основе М-20 была создана уникальная многопроцессорная М-40 — самая быстродействующая ЭВМ того времени в мире (40 тыс. опер./с). На смену М-20 пришли полупроводниковые БЭСМ-4 и М-220 (200 тыс. опер./с).

1959 г. Первое сообщение о языке Алгол, который надолго стал стандартом в области языков программирования (Algol-60).

1961 г. Фирма IBM Deutschland реализовала подключение компьютера к телефонной линии с помощью модема.

1964 г. Начат выпуск семейства машин третьего поколения — IBM/360.

1965 г. Дж. Кемени и Т. Курц в Дортмундском колледже (США) разработали язык программирования Бейсик.

7965 *г.* С. Псйпсрт разработал язык LOGO — компьютерный язык для детей.

1967 г. Под руководством С. А. Лебедева организован крупносерийный выпуск шедевра отечественной вычислительной техники — миллионника БЭСМ-6, — самой быстродействующей машины в мире. За ним последовал «Эльбрус» — ЭВМ нового типа, производительностью 10 млн опер./с.

1968 г. Основана фирма Intel, впоследствии ставшая признанным лидером в области производства микропроцессоров и других компьютерных интегральных схем

1970 г. Швейцарец Н Вирт разработал язык Паскаль

1971 г. Э. Хофф разработал микропроцессор Intel-4004, состоящий из 2250 транзисторов, размещенных в кристалле размером не больше шляпки гвоздя.

1971 г. Французский ученый А. Колмари разработал язык логического программирования Пролог (PROgramming in LOGic).

1972 г. Д. Ритчи из Bell Laboratories разработал язык Си.

1973 г. К. Томпсон и Д. Ритчи создали операционную систему UNIX.

1973 г. Фирма IBM (International Business Machines) сконструировала первый жесткий диск «винчестер»

***1974** г.* Фирма Intel разработала первый универсальный 8-разрядный микропроцессор 8080 с 4500 транзисторами.

1974 г. Э. Роберте, молодой офицер ВВС США, инженер-электронщик, построил на базе процессора 8080 микрокомпьютер Альтаир, имевший огромный коммерческий успех, продававшийся по почте и широко использовавшийся для домашнего применения.

1975 г. Молодой программист П. Аллен и студент Гарвардского университета Б. Гейтс реализовали для Альтаира язык Бейсик. Впоследствии они основали фирму Майкрософт (Microsoft), являющуюся сегоднякрупнейшимпроизводителемпрограммногообеспечения.

1975 г. Фирма IBM начала продажу лазерных принтеров.

***1976** г.* Студенты С. Возняк и С. Джобе, устроив мастерскую в гараже, реализовали компьютер Apple-1, положив начало корпорации Apple.

***1978** г.* Фирма Intel выпустила микропроцессор 8086.

***1979** г.* Фирма Intel выпустила микропроцессор 8088. Корпорация IBM приобрела крупную партию этих процессоров для вновь образованного подразделения по разработке и производству персональных компьютеров (IBM PC, IBM PC XT, IBM PC AT).

***1979** г.* Фирма Software Arts разработала первый пакет деловых программ VisiCalc (Visible Calculator) для персональных компьютеров.

1980 г. Корпорация Control Data выпустила суперкомпьютер Cyber (Сайбер) 205.

1980 г. Японские компании Sharp, Sanyo, Panasonic, Casio и американская фирма Tandy выпустили первый карманный компьютер, обладающий всеми основными свойствами больших компьютеров.

1981 г. Фирма IBM выпустила первый персональный компьютер IBM PC на базе микропроцессора 8088.

***1982** г.* Фирма Intel выпустила микропроцессор 80286, содержащий 134000 транзисторов и способный выполнять любые программы, написанные для его предшественников. С тех пор такая программная совместимость остается отличительным признаком семейства микропроцессоров Intel.

***1982** г.* М. Капор представил систему Lotus 1-2-3, которая победила в конкурентной борьбе Visicalc.

1983 г. Корпорация Apple Computers построила персональный компьютер Lisa (Mitch Kapor) — первый офисный компьютер, управляемый манипулятором мышь.

***1983** г.* Гибкие диски получили распространение в качестве стандартных носителей информации.

1983 г. Фирмой Borland выпущен в продажу компилятор Turbo Pascal, разработанный А. Хейльсбергом.

1984 г. Создан первый компьютер типа Laptop (наколенный), в котором системный блок объединен с дисплеем и клавиатурой в единый блок.

1984 г. Фирмы Sony и Phillips разработали стандарт записи компакт-дисков CD-ROM.

1984 г. Корпорация Apple Computer выпустила компьютер Macintosh на 32-разрядном процессоре Motorola 68000 — первую модель знаменитого впоследствии семейства Macintosh с удобной для пользователя операционной системой, развитыми графическими возможностями, намного превосходящими в то время те, которыми обладали стандартные IBM-совместимые ПК с MS-DOS. Эти компьютеры быстро приобрели миллионы поклонников и стали вычислительной платформой для целых отраслей, таких, например, как издательское дело и образование.

1984 г. Появилась некоммерческая компьютерная сеть FIDO. Ее создатели — Т. Дженнингс и Дж. Мэдил. В 1995 г. в мире насчитывалось около 20 тыс. узлов этой сети, объединяющих 3 млн человек.

1985 г. Фирма Intel выпустила микропроцессор 80386, насчитывающий 275 000 транзисторов. Этот 32-разрядный многозадачный процессор обеспечивал возможность одновременного выполнения нескольких программ.

1985 г. Б. Страуструп из Bell Laboratories опубликовал описание созданного им объектно-ориентированного языка C++.

1989 г. Американская фирма Poquet Computers Corporation представила новый компьютер класса Subnotebook — Pocket PC.

1989 г. Т. Бернерс-Ли предложил язык гипертекстовой разметки HTML (HyperText Markup Language) в качестве одного из компонентов технологии разработки распределенной гипертекстовой системы World Wide Web.

1989 г. Фирма Intel выпустила микропроцессор Intel 486 DX. Поколение процессоров i486 ознаменовало переход от работы на компьютере через командную строку к режиму «укажи и щелкни». Intel 486 стал первым микропроцессором со встроенным математическим сопроцессором, который существенно ускорил обработку данных, выполняя сложные математические действия вместо центрального процессора. Количество транзисторов — 1,2 млн. Корпорация Microsoft выпустила графическую оболочку MS Windows 3.0.

1990 г. Выпуск и ввод в эксплуатацию векторно-конвейерной супер-ЭВМ «Эльбрус 3.1». Разработчики — Г, Г. Рябов, А. А. Соколов,

А Ю Бяков Производительность в однопроцессорном варианте — 400 Мфлопс

1991 г. Финский студент Л Торвальдс распространил среди пользователей Интернет первый прототип своей операционной системы Linux (на основе ОС Unix) Заинтересованные в этой работе программисты стали поддерживать Linux, добавляя драйверы устройств, разрабатывая разные продвинутые приложения и др Атмосфера работы энтузиастов над полезным проектом, а также свободное распространение и использование исходных текстов стали основой феномена Linux В настоящее время Linux — очень мощная система, к тому же, некоммерческая

1992 г. **В** этом году начался бурный рост популярности Internet и World Wide Web в связи с появлением web-браузера Mosaic, разработанного в Национальном центре по приложениям для суперкомпьютеров в Университете штата Иллинойс Разработчики — Э Бина и М Андриссен

1993 г. Фирма Intel выпустила микропроцессор Pentium, который дал возможность компьютерам работать с атрибутами «реального мира» — звук, голосовая и письменная речь, фотоизображения

1994 г. Начало выпуска фирмой Power Mac серии фирмы Apple Computers — Power PC

1994 г. Компания Netscape Communication выпустила браузер Netscape Navigator

1995 г. Фирма Microsoft выпустила операционную систему Windows 95

1995 г. Фирма Microsoft выпустила браузер Internet Explorer Началась война браузеров, в которой пока побеждает Internet Explorer

1995 г. Фирма Intel выпустила микропроцессор Pentium Pro, насчитывающий 5,5 млн транзисторов Процессор разрабатывался как мощное средство наращивания быстродействия 32-разрядных приложений для серверов и рабочих станций, систем автоматизированного проектирования, программных пакетов, используемых в машиностроении и научной работе Все процессоры Pentium Pro оснащены второй микросхемой кэш-памяти, еще больше увеличивающей быстродействие

1997г. Фирма Intel выпустила микропроцессор Pentium II, насчитывающий 7,5 млн транзисторов Процессор Pentium II использует технологию Intel MMX, обеспечивающую эффективную обработку аудио, визуальных и графических данных Кристалл и микросхема высокоскоростной кэш-памяти помещены в корпус с односторонним контактом, который устанавливается на системной плате с помощью одностороннего разъема (Slot 1) — в отличие от прежних процессоров, имевших множество контактов (Socket 7) Процессор дает пользователям возможность вводить в компьютер и обрабатывать

цифровые фотоизображения, создавать и редактировать тексты, музыкальные произведения, сценки для домашнего кино, передавать видеоизображения по обычным телефонным линиям

1997 г. Компания Sun Microsystems приняла стандарт объектно-ориентированного языка программирования Java (произносится «джава»), созданного для реализации принципа «написано однажды — работает везде» В применении к интернету Java — технология создания «апплетов» — небольших программ, которые загружаются на компьютер пользователя вместе со страницей сайта и позволяют «оживлять» эту страницу Апплеты могут обеспечивать странице дополнительную функциональность, например реализовывать мультипликационные иллюстрации

1998 г. Выпуск в свет операционной системы Windows 98

1999 г. Появление 64-разрядного микропроцессора Mersed

2000 г. Появление 64-разрядных микропроцессоров Itanium и AMD

2000 г. Выпуск в свет операционной системы Windows 2000

Основная таблица ASCII-символов и их кодировки

Таблица соответствия значений клавиш терминала в десятичном (основание 10), шестнадцатеричном (основание 16), восьмеричном (основание 8), а также в коде ASCII (Американский стандартный код для обмена информацией) Последовательности клавиш, включающие клавишу <Control> (<Ctr>l) вводятся одновременным нажатием клавиши <Control> и указанной клавиши Эти последовательности основаны на тех, которые описаны для большинства стандартных терминалов, и могут быть описаны иначе на других терминалах

Таблица П4. **ASCII-символы и их кодировки**

Десятичный код	Двоичный код	Восьмеричный код	Шестнадцатеричный код	Символы кода ASCII	Клавиши терминала IBM
0	0000 0000	00	00	NUI	<Ctrl+@>
1	00000001	01	01	SOH	<Ctrl+A>
2	0000 0010	02	02	STX	<Ctrl+ B>
3	00000011	03	03	ETX	<Ctrl O
4	0000 0100	04	04	EOT	<Ctrl D>
5	0000 0101	05	05	ENQ	<Ctrl E>
6	0000 0110	06	06	ACK	<Ctrl+F>
7	00000111	07	07	BEL	<Ctrl+G>
8	0000 1000	10	08	BS	<Ctrl+H>
9	0000 1001	11	09	HT	<Ctrl+I>
10	0000 1010	12	0A	LF	<Ctrl+J>
11	00001011	13	0B	VT	<Ctrl+K>
12	0000 1100	14	0C	FF	<Ctrl+L>
13	0000 1101	15	0D	CR	<Ctrl+M>
14	0000 1110	16	0E	SO	<Ctrl+N>
15	00001111	17	0F	SI	<Ctrl+O>

Продолжение табл П4

Десятичный код	Двоичный код	Восьмеричный код	Шестнадцатеричный код	Символы кода ASCII	Клавиши терминала IBM
16	0001 0000	20	10	DLE	<Ctrl+P>
17	0001 0001	21	11	DC1	<Ctrl+Q>
18	0001 0010	22	12	DC2	<Ctrl+R>
19	0001 0011	23	13	DC3	<Ctrl+S>
20	0001 0100	24	14	DC4	<Ctri+T>
21	0001 0101	25	15	NAK	<Ctrl+U>
22	0001 0110	26	16	SYN	<Ctrl+V>
23	0001 0111	27	17	ETB	<Ctrl+W>
24	0001 1000	30	18	CAN	<Ctrl+X>
25	0001 1001	31	19	EM	<Ctrl+Y>
26	0001 1010	32	1A	SUB	<Ctrl+Z>
27	0001 1011	33	1B	ESC	<Esc>
28	0001 1100	34	1C	FS	<Ctrl+\>
29	0001 1101	35	1D	GS	<Ctrl+ >
30	0001 1110	36	1E	RS	<Ctrl+=>
31	0001 1111	37	1F	US	<Ctrl+ >
32	0010 0000	40	20	SP	<Space> (Пробел)
33	00100001	41	21	!	! (Восклицательный знак)
34	00100010	42	22	«	» (Кавычки)
35	00100011	43	23	*	* (Знак числа)
36	0010 0100	44	24	$	$ (Знак «доллар»)
37	00100101	45	25	%	% (Процент)
38	00100110	46	26	&	& (Амперсанд)
39	00100111	47	27		(Апостроф или диакритический знак)
40	0010 1000	50	28	(	(Открывающие (круглые) скобки
41	0010 1001	51	29	)	) Закрывающие (круглые) скобки
,2	0010 1010	52	2A	*	* (Звездочка)

Продолжение табл. П4

Десятичный код	Двоичный код	Восьмеричный код	Шестнадцатеричный код	Символы кода ASCII	Клавиши терминала IBM
43	0010 1011	53	2B	+	+ (Плюс)
44	0010 1100	54	2C	,	, (Запятая)
45	00101101	55	2D	-	- (Дефис, тире или минус)
46	0010 1110	56	2E		(Точка)
47	00101111	57	2F	/	/ (Косая черта)
48	0011 0000	60	30	0	0 (Ноль)
49	0011 0001	61	31	1	1
50	0011 0010	62	32	2	2
51	00110011	63	33	3	3
52	0011 0100	64	34	4	4
53	0011 0101	65	35	5	5
54	0011 0110	66	36	6	6
55	0011 0111	67	37	7	7
56	0011 1000	70	38	8	8
57	0011 1001	71	39	9	9
58	0011 1010	72	3A	:	: (Двоеточие)
59	0011 1011	73	3B	;	; (Точка с запятой)
60	0011 1100	74	3C	<	< (Меньше, чем)
61	0011 1101	75	3D	=	= (Равняется)
62	0011 1110	76	3E	>	> (Больше, чем)
63	00111111	77	3F	?	? (Вопросительный знак)
64	0100 0000	100	40	@	@ (Коммерческий знак at)
65	0100 0001	101	41	A	A (Латиница, прописные)
66	0100 0010	102	42	B	B
67	0100 0011	103	43	C	C
68	01000100	104	44	D	D
69	01000101	105	45	E	E
70	01000110	106	46	F	F
71	01000111	107	47	G	G

Продолжение табл П4

Десятичный код	Двоичный код	Восьмерич-ный код	Шестнадца-теричный код	Символы кода ASCII	Клавиши терминала IBM
72	0100 1000	110	48	H	H
73	0100 1001	111	49	I	I
74	0100 1010	112	4A	J	J
75	0100 1011	113	4B	K	K
76	0100 1100	114	4C	L	L
77	0100 1101	115	4D	M	M
78	0100 1110	116	4E	N	N
79	0100 1111	117	4F	O	O
80	0101 0000	120	50	P	P
81	0101 0001	121	51	Q	Q
82	0101 0010	122	52	R	R
83	0101 0011	123	53	S	S
84	0101 0100	124	54	T	T
85	0101 0101	125	55	U	U
86	0101 0110	126	56	V	V
87	0101 0111	127	57	W	W
88	0101 1000	130	58	X	X
89	0101 1001	131	59	Y	Y
90	0101 1010	132	5A	Z	Z
91	0101 1011	133	5B	[	[(Открывающая квадратная скобка)
92	0101 1100	134	5C	\	\ (Обратная косая черта)
93	0101 1101	135	5D	]	] (Закрывающая квадратная скобка)
94	0101 1110	136	5E	^	" (Диакритический знак)
95	0101 1111	137	5F	_	_ (Символ подчеркивания)
96	01100000	140	60	`	' (Диакритический знак)
97	0110 0001	141	61	a	a (Латиница, строчные)
98	01100010	142	62	b	b
99	01100011	143	63	c	c
100	01100100	144	64	d	d

Окончание табл П4

Десятичный код	Двоичный код	Восьмерич-ный код	Шестнадца-теричный код	Символы кода ASCII	Клавиши терминала IBM
101	01100101	145	65	e	e
102	01100110	145	66	f	f
103	01100111	147	67	g	g
104	0110 1000	150	68	h	h
105	01101001	151	69	i	i
106	01101010	152	6A	j	j
107	01101011	153	6B	k	k
108	0110 1100	154	6C	l	l
109	01101101	155	6D	m	m
110	01101110	156	6E	n	n
111	01101111	157	6F	o	o
112	0111 0000	160	70	P	P
113	01110001	161	71	q	q
114	0111 0010	162	72	r	r
115	0111 0011	163	73	s	s
116	0111 0100	164	74	t	t
117	0111 0101	165	75	u	u
118	01110110	166	76	v	v
119	0111 0111	167	77	w	w
120	0111 1000	170	78	x	x
121	0111 1001	171	79	y	y
122	0111 1010	172	7A	z	z
123	0111 1011	173	7B	{	{ (Открывающая фигурная скобка)
124	0111 1100	174	7C	|	| (Вертикальный штрих (логическое ИЛИ))
125	0111 1101	175	7D	}	) (Закрывающая фигурная скобка)
126	0111 1110	176	7E	~	~ (Знак «тильда»)
127	0111 1111	177	7F	DEL	<Del> (Удаление)

Описания непечатаемых (неотображаемых) символов ASCII

ACK (подтверждение);
BELL (звонок);
BS (возврат, стирание символа);
CAN (отмена);
CR (возврат каретки);
DC1—DC4 (управление устройством);
DEL (удаление символа);
DLE («авторегистр 1» — управляющий символ при передаче данных);
EM (конец носителя данных);
ENQ (запрос);
EOT (конец передачи);
ESC («авторегистр 2» — начало управляющей последовательности);
ETB (конец блока передачи);
EXT (конец текста);
FF (перевод страницы);
FS,GS,RG,US (символы-разделители файлов, групп, записей и устройств);
HT (горизонтальная табуляция);
LF (перевод строки);
NAK (отрицательное подтверждение);
NUL (нуль — заполнитель пространства);
SI (переключение на стандартный регистр);
SO (переключение на дополнительный регистр);
SOH (начало заголовка);
STX (начало текста);
SUB (замена);
SYN (синхронизация — управляющий символ при передаче данных);
VT (вертикальная табуляция).

Содержание

Максимов Николай Вениаминович
Партыка Татьяна Леонидовна
Попов Игорь Иванович

Архитектура ЭВМ и вычислительных систем

Учебник

Редактор *А. В. Волковицкая*
Корректор *А. В. Алешина*
Компьютерная верстка *И. В. Кондратьевой*
Оформление серии *Р. Остроумова*

Сдано в набор 25.10.2004. Подписано в печать 15.12.2004. Формат 60x90/16 ,
Печать офсетная. Гарнитура «Тайме». Усл. печ. л. 32. Уч.-изд. л. 31,6.
Бумага типографская № 2. Тираж 6000 экз. Заказ № 4504009.

ЛР № 071629 от 20.04.98
Издательский Дом «ФОРУМ»
101831, Москва — Центр, Колпачный пер., д. 9а
Тел./факс: (095) 925-32-07, 925-39-27
E-mail: forum-books@mail.ru

ЛР № 070824 от 21.01.93
Издательский Дом «ИНФРА-М»
127214, Москва, Дмитровское ш , 107
Тел.: (095) 485-70-18, 485-74-00
Факс: (095) 485-53-18
E-mail books@infra-m ru
Http'//www.infra-m.ru

Отпечатано с готовых диапозитивов
на ФГУИПП «Нижполиграф».
603006, Нижний Новгород, ул. Варварская, 32.