

Лекция 16

История создания, состояние и тенденции развития вычислительных систем

В основе автоматической обработки информации, информационных систем находится ЭВМ (Computer) – средство предназначенное для автоматической обработки информации – данных. В зависимости от формы представления информации, вычислительные машины делятся на три класса: АВМ (аналоговые), ЦВМ (цифровые) и ГВМ (гибридные).

Аналоговые вычислительные машины – вычислительные машины непрерывного действия, работают с информацией, представленной в непрерывной (аналоговой) форме, т.е. в виде непрерывного ряда значений какой-либо физической величины (чаще всего электрического напряжения). Они просты и удобны в эксплуатации, программирование для них, как правило, нетрудоемкое, скорость решения изменяется по желанию оператора и может быть сделана сколь угодно большой (больше, чем у ЦВМ), но точность решения задач очень низкая (относительная погрешность 2-5%).

Цифровые вычислительные машины – вычислительные машины дискретного действия, работают с информацией, представленной в дискретной, а точнее, в цифровой форме. ЦВМ воспринимает различные виды информации в форме данных, представленных в цифровой форме, хранит эти данные и пересылает их внутри системы по линиям связи, перерабатывает их с большой скоростью и выдает результаты этих действий как информацию. Наиболее широкое применение находят ЭВМ – электронные вычислительные машины (ЦВМ с электрическим представлением дискретной информации). **ЭВМ – аппаратурно-программный комплекс (hardware&software), предназначенный для автоматического выполнения логико-вычислительной работы (ввода (сбора), обработки, хранения и вывода (передачи) информации) (т.е. основанный на модели вычислителя).**

Функционирование ЭВМ определяется:

- ее технической конструкцией (hardware – аппаратурной частью);
- алгоритмом обработки информации (решения задач), который обязательно записывается на языке, доступном машине (программа);
- системными программами (software – программные средства).

Гибридные вычислительные машины – вычислительные машины комбинированного действия, работают с информацией, представленной и в цифровой, и в аналоговой форме. Они совмещают в себе достоинства АВМ и ЦВМ и целесообразны для решения задач управления сложными быстродействующими техническими комплексами.

Вычислительная система (ВС) – комплекс технических средств ЭВМ и программ, обеспечивающих их эксплуатацию; средство обработки информации, функционирование которого основано на имитации работы не одиночных вычислителей, а коллективов вычислителей.

Функциональная структура вычислительной системы – блок-схема, отражающая состав основных функциональных устройств (блоков), связи и взаимодействия между ними.

Архитектура вычислительной системы – концепция взаимосвязи и функционирования его аппаратурных (hardware) и программных (software) компонентов.

Понятие архитектуры – есть одна из основных интегральных характеристик ЭВМ.

Архитектура ЭВМ включает:

- внутреннюю структуру ЭВМ,
- систему команд ЭВМ и ее автокод,
- объемы памяти всех уровней и взаимодействие между ними,
- организацию общения центрального процессора с периферией,

- совокупность свойств и характеристик, определяющих информационные связи между центральными процессорами, периферийными процессорами, оперативными ЗУ, внешними ЗУ и периферией,
- взаимодействие и структуру потоков команд и потоков данных.

Описание вычислительной системы может ограничиваться описанием ее структуры, но может быть продолжено до архитектурного описания.

Совокупность таких характеристик ЭВМ, как элементная (но не элементарная!!!) база, архитектура, режим работы и область применения составляют понятие «поколение» ЭВМ, успешно применяемое при рассмотрении истории развития вычислительной техники. Поколения ЭВМ – есть достаточно условное деление, так как эволюция архитектуры и структуры ЭВМ происходит непрерывно и в рамках каждого поколения можно найти вычислительные системы, носящие черты, как предыдущего, так и следующего поколений.

Структура и принципы работы ЭВМ 1-го поколения

В качестве элементной базы машин первого поколения (50-е годы) использовались электронные лампы. Машины были громоздкими, потребляли много электроэнергии.

Режим работы – монопрограммный однопользовательский. Программист присутствовал при отладке и счете своей задачи, контролировал ее ход по световой индикации на пульте управления ЭВМ и мог по ходу вычислений переключать работу в однократный ручной режим для контроля или внесения в программу изменений. Этот режим был очень удобен для программиста, так как машина всегда оставалась в его распоряжении, однако имел чрезвычайно низкий коэффициент использования оборудования.

Способ обработки информации – последовательный. **Алгоритм управления вычислительным процессом** – неизменный (закладывался в аппаратуру при конструировании машины и неизменный впоследствии). **Структурная схема** ЭВМ 1-го поколения представлена на рисунке:

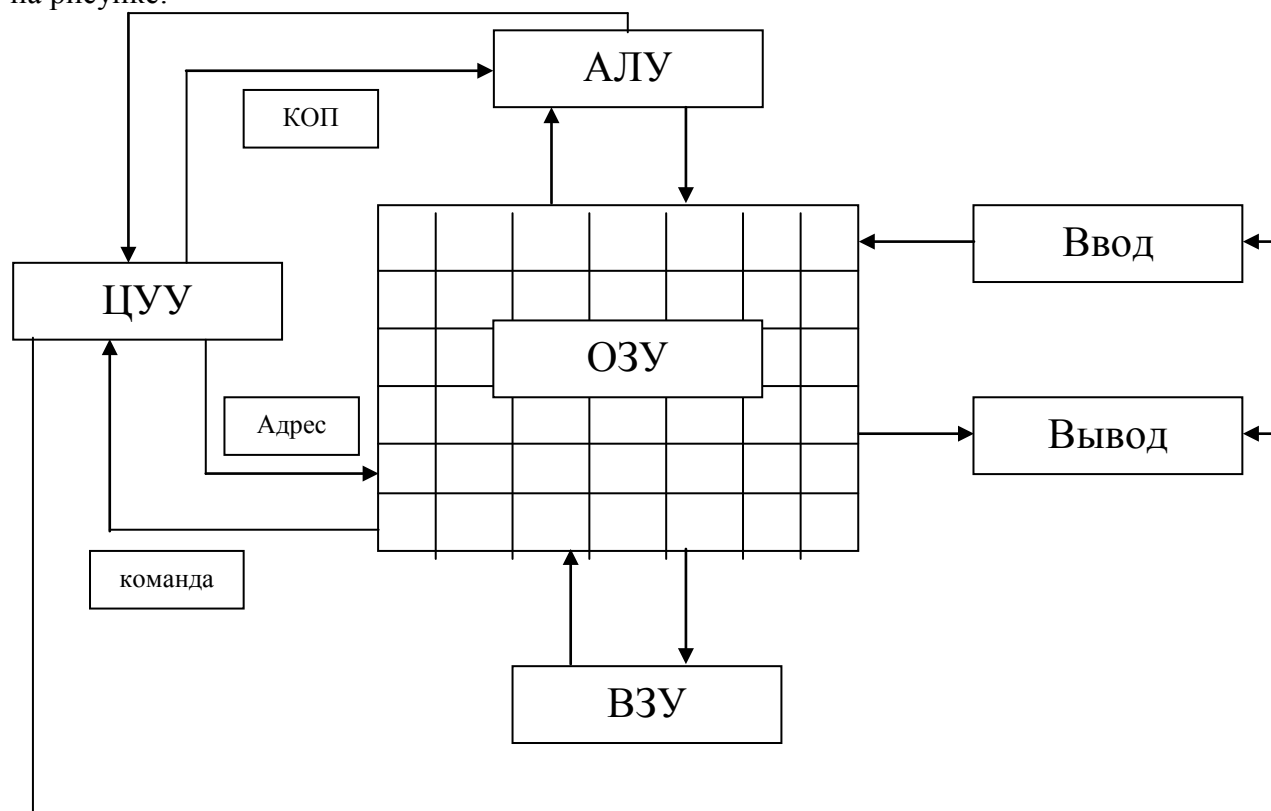


Рис.1.5. Структурная схема ЭВМ 1-го поколения

ОЗУ – оперативное запоминающее устройство, разделенное на отдельные ячейки, называемые адресом и пронумерованные от 0 до N. В каждой ячейке содержится одно слово. Машинное слово равным образом представляет и команду, и число. Команда – инструкция для выполнения одной операции в ЭВМ, записанная в числовой форме. Из команд состоит программа работы ЭВМ. Количество разрядов и в алгебраическом числе, и в командном коде делается одинаковым. Команды заносятся в ОЗУ в порядке номеров ячеек. Если слово «берется» из ОЗУ, то оно как бы считывается, а оригинал остается в ячейке. Когда слово «посылается» в ОЗУ, то оно записывается поверх старого значения (старое значение затирается).

Если слово — команда, то его структура (для первых ЭВМ) выглядит так:

КОП A1 A2 A3,

где КОП – цифровой код операции, выполняющейся в данной команде(можно представить себе как некоторый условный номер в общем списке системы команд);

A1, A2 – адреса операндов, над которыми производится действие, заданное КОП;

A3 – адрес ячейки ОЗУ, куда посылается результат.

Если слово — алгебраическое число, то его структура выглядела так:

знак числа модуль числа

ЦУУ – центральное устройство управления. В него поступают команды. ЦУУ обрабатывает код очередной команды на специальном блоке, называемом дешифратором, и превращает КОП в серию управляющих импульсов, предназначенных для тех устройств, которые участвуют в данной операции. Например, в случае арифметической команды, управляющие импульсы будут направлены в АЛУ. Затем расшифровывается адресная часть команды, и управляющие импульсы посылаются в ОЗУ для выборки слов по соответствующим адресам.

В основе работы ЦУУ лежит «машинный такт». Каждая машинная команда делится на ряд элементарных унифицированных составных частей, называемых тактами. В зависимости от сложности команды она может быть реализована за разное число тактов. Собственно запоминающее устройство называют оперативным потому, что скорость выборки из него данных или посылка в него результата происходят в темпе работы АЛУ, т.е. в соответствии с режимом выполнения машиной одной операции за один рабочий такт.

При выполнении команды ЭВМ проделывает определенные стандартные действия:

- согласно содержимому счетчика адреса команд, считывается очередная команда программы (ее код обычно заносится на хранение в специальный регистр УУ. Который носит название регистра команд);
- счетчик команд автоматически изменяется так, чтобы в нем содержался адрес следующей команды (в простейшем случае для этой цели достаточно к текущему значению счетчика прибавить некоторую константу, определяющуюся длиной команды);
- считанная в регистр команд операция расшифровывается, из оперативной памяти извлекаются необходимые данные и над ними выполняются требуемые действия;
- полученный результат сохраняется в оперативной памяти.

Полный такт работы ЭВМ представлен на следующем рисунке:

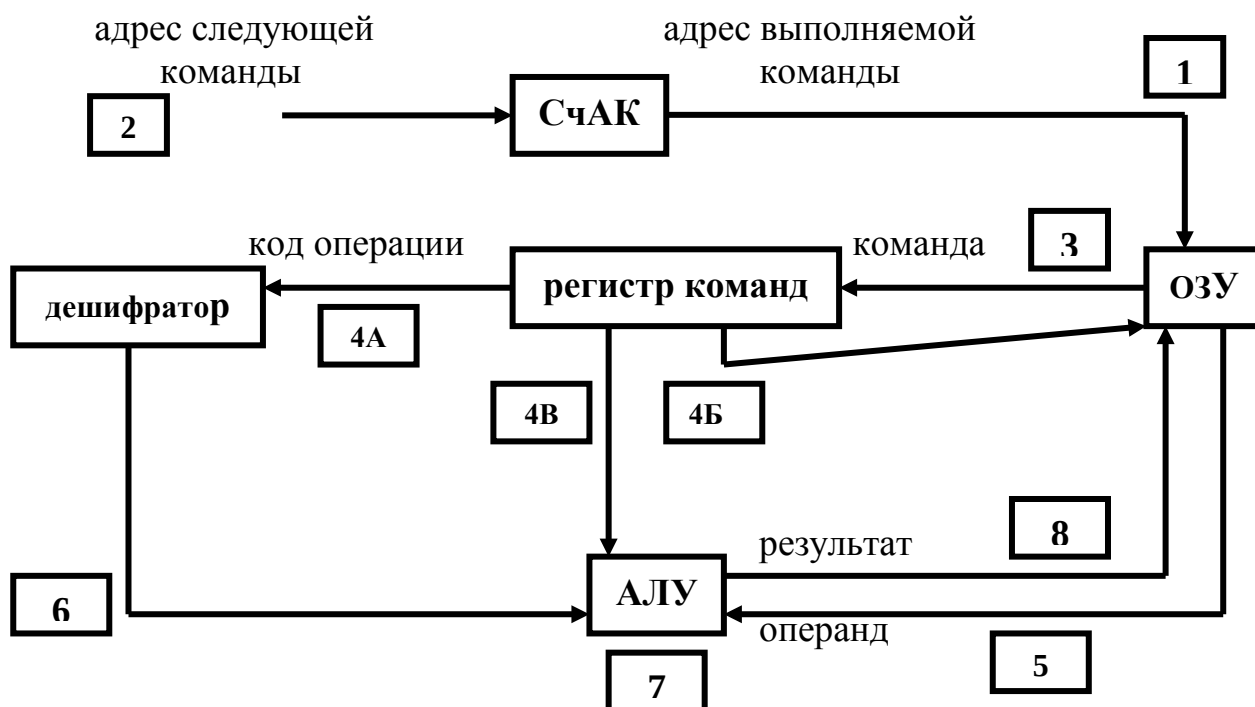


Рис.1.6. Полный такт работы ЭВМ

где:

- *СчАК* – счетчик адреса команд – регистр, содержащий адрес выполняемой в данный момент команды; во время каждого полного такта из ОЗУ выбирается слово с адресом, равным содержимому СчАК, а в СчАК добавляется 1.
- *регистр команд* – регистр, предназначенный для хранения команды, выполняемой в данный момент и выбранной из ОЗУ;
- *АЛУ* – арифметико-логическое устройство;
- *ОЗУ* – оперативно-запоминающее устройство;
- *Дешифратор* – устройство (принадлежит ЦУУ), выполняющее расшифровку команды. КОП из регистра команд попадает в дешифратор, превращается там в управляющие импульсы и посылается на те устройства, которые должны выполнять расшифрованную операцию. По адресной части команды вырабатываются сигналы управления для ОЗУ, обеспечивающие чтение операндов по адресам А1 и А2, посылку их в АЛУ и запись результатов по адресу А3.
- **N** – номер последовательности выполнения действия.

АЛУ¹ – арифметико-логическое устройство, выполняющее, как минимум, арифметические и логические операции. Для выполнения машинной операции АЛУ должно быть настроено на ее выполнение. Операнды в АЛУ посылаются из ОЗУ, туда же записывается и результат операции.

ВЗУ – внешнее запоминающее устройство.

Область применения первых ЭВМ: научные расчеты.

Технология программирования: в машинных кодах, автокод (или ассемблер – упрощенное кодирование команд, предполагающее мнемонический КОП и нумерацию ячеек не по физическим адресам, а с опорой на буквенный базис: а+1, а+2 и т.д.).

ЭВМ поставлялись в вычислительные центры заводами-изготовителями в «голом» виде, без какого-либо вспомогательного программного обеспечения. Вопрос о программных средствах (software) решался просто: все операции, которые нужно было выполнить компьютеру для решения задачи, программисту приходилось полностью описывать в своей

¹ В дальнейшем, ЦУУ и АЛУ, объединенные вместе, составили понятие «процессор» ЭВМ

программе. Не было никакого системного программного обеспечения, кроме библиотек математических и служебных подпрограмм, которые программист мог использовать для того, чтобы не писать каждый раз коды, вычисляющие, например, значение какой-либо математической функции. Все задачи организации вычислительного процесса решались каждым программистом с пульта управления, который представлял собой устройство, состоящее из кнопок, переключателей и индикаторов.

Машинами первого поколения были Стрела, Бэсм-2, М-2, М-3, Урал-1, Урал-2.

Заложенные в первых ЭВМ архитектурно-функциональные принципы на многие годы определили подходы к построению ЭВМ. Эти принципы известны также как принципы фон Неймана, потому что были изложены венгерским математиком Дж.фон Нейманом и его коллегами Г.Голдстейном и А.Берксом (в 1946г., в отчете «Предварительное обсуждение логического конструирования вычислительного устройства»).

Содержание принципов следующее:

Программное управление работой ЭВМ – программы состоят из отдельных шагов – команд; команда осуществляет единичный акт преобразования информации; последовательность команд, необходимая для реализации алгоритма, является программой; все разновидности команд, используемые в конкретной ЭВМ, в совокупности являются языком машины или системой команд машины.

Принцип условного перехода – возможность перехода в процессе вычислений на тот или иной участок программы в зависимости от промежуточных, получаемых в ходе вычислений результатов (обычно в зависимости от знака результата после завершения арифметической операции или от результатов выполнения логической операции). Реализация принципа условного перехода позволяет легко осуществить в программе циклы с автоматическим выходом из них, итерационные процессы и т.д. Благодаря этому, число команд в программе получается во много раз меньше, чем число команд при исполнении данной программы за счет многократного вхождения в работу участков программы.

Принцип хранимой программы – команды представляются в числовой форме и хранятся в том же ОЗУ, что и исходные данные. Только команды для исполнения выбираются из ОЗУ в ЦУУ, а числа (операнды) – в АЛУ. Но для машины и команда, и число являются машинным словом, и если команду направить в АУ в качестве операнда, то над ней можно произвести арифметические операции, изменив ее. Это открывает возможности преобразования программ в ходе их выполнения. Кроме того, этот принцип обеспечивает одинаковое время выборки команд и операндов из ОЗУ для выполнения, позволяет быстро менять программы или части их, вводить не прямые системы адресации, видоизменять программы по определенным правилам.

Принцип использования двоичной системы счисления для представления информации в ЭВМ – существенно упростил конструкцию ЭВМ. Он расширил номенклатуру физических приборов и явлений, которые можно использовать при работе ЭВМ, так как цифры 0 и 1, имеющиеся в этой системе, могут изображаться положением любой двухстабильной системы (триод – в открытом или закрытом состоянии триггера, намагниченным или ненамагниченным участком ферромагнитной поверхности, отсутствием или наличием электрического импульса и т.д.). Количество информации определяется единицей (бит) в двоичной системе счисления, а к логическим схемам, построенным по двоичной системе счисления, может быть применен хорошо разработанный математический аппарат булевой алгебры. В двоичной системе счисления легко производить операцию умножения последовательными сложениями и сдвигом множимого в зависимости от наличия единиц в разрядах множителя. Один из недостатков двоичной системы – ее громоздкость.

Исключительное использование двоичной системы связано с рядом ее особенностей:

- Относительная простота технической реализации с помощью устройств, имеющих два устойчивых состояния (например, наличие или отсутствие заряда в конденсаторе). Для r -ичной системы понадобились бы устройства с r устойчивыми состояниями, более сложные, а значит, менее надежные.
- Относительная простота реализации двоичной арифметики.
- Простота реализации логических операций, то есть операций над значениями «истина» и «ложь» (булева алгебра). Более сложная логика (многозначная) сложна для реализации в автоматических вычислительных устройствах.
- Арифметические операции можно выразить в терминах булевой логики, а булевы логические операции сравнительно легко реализуются цифровыми электронными устройствами.

Принцип иерархичности запоминающих устройств (ЗУ). С самого начала развития ЭВМ существовало несоответствие между быстродействием АЛУ и ОЗУ. Выполнение ОЗУ на тех же элементах, что и АЛУ, удавалось это противоречие частично ликвидировать, но такое ОЗУ получалось слишком дорогим, значительно увеличивало количество радиоламп в ЭВМ, снижая в целом ее надежность. Иерархическое построение ЗУ позволяет иметь быстродействующее ЗУ – ОЗУ – сравнительно небольшой емкости только для команд и операндов, участвующих в счете в данный момент и в ближайшее время. Более низкий уровень – более емкое, более дешевое ЗУ, но с большим временем обращения к нему при записи или считывании. Такими ЗУ самого низкого уровня были в первых ЭВМ магнитные ленты. Иерархичность ЗУ является важным компромиссом между емкостью и быстрым доступом к данным, обеспечивающим требования быстродействия, большой емкости памяти, относительной дешевизны и надежности.

Структура и принципы работы ЭВМ 2-го поколения

Элементная база: ЭВМ второго поколения (60-е гг.) строились на полупроводниковых приборах (транзисторах). Это повысило надежность ЭВМ, уменьшило их габариты и энергопотребление, снизило стоимость.

В АЛУ было встроено некоторое количество сверхбыстрых регистров, что также способствовало повышению скорости работы. Кроме того, появились команды, в которых в качестве адреса фигурировал регистр, а не ячейка ОЗУ.

Была расширена номенклатура ВЗУ на магнитных носителях. Появились магнитные ленты (МЛ), магнитные барабаны (МБ), магнитные диски (МД), магнитные карты.

Расширилась и номенклатура устройств ввода-вывода. Стали использоваться перфокарты, перфоленты, АЦПУ (алфавитно-цифровая печать) на широкую бумажную ленту.

Способ обработки информации – последовательный, но так как появились прерывания, то стало возможным реализовать последовательно-параллельный **алгоритм управления вычислительным процессом** (совмещение ввода/вывода с другими операциями).

Переход к машинам второго поколения обозначил прогресс и в области автоматизации программирования и организации вычислительных работ. Появились алгоритмические языки высокого уровня (АЛГОЛ-60, ФОРТРАН-2, КОБОЛ) и трансляторы с них, которые, помимо библиотек математических и служебных подпрограмм, составили новый тип системного программного обеспечения. Стало происходить различение программ на исходный модуль и объектный модуль (программа в машинных кодах после трансляции). Выполнение каждой программы стало включать большое количество вспомогательных работ: загрузка нужного транслятора, запуск транслятора и получение результирующей программы в машинных кодах, связывание программы с библиотечными подпрограммами, загрузка программы в оперативную

память, запуск программы, вывод результатов на периферийное устройство. Для организации эффективного совместного использования трансляторов, библиотечных программ и загрузчиков в штат вычислительных центров были введены должности операторов, профессионально выполняющих работу по организации вычислительного процесса для всех пользователей этого центра.

Стали шире применяться библиотеки объектных модулей, чтобы избежать лишней работы по программированию исходных модулей, транслированию и отладке; появились диспетчеры, как простейшие версии ОС; сервисные программы (редактирование, отладка).

Область применения ЭВМ расширилась. Кроме научных расчетов они стали использоваться для решения экономических задач. Собственно расширение сферы применения и обеспечило расширение номенклатуры устройств ввода-вывода и ВЗУ.

Режим работы – закрытый. Программиста как бы «отлучили» от машины. Выполнение программ было организовано через посредника (оператора).

Переход от открытого к закрытому режиму использования ЭВМ как бы стимулировал развитие двух направлений программирования: *прикладное программирование*, связанное с разработкой программ для решения конкретных научно-технических задач, и *системное программирование*, связанное с разработкой специальных программ, автоматизирующих процесс написания и отладки прикладных программ и обеспечивающих эффективное использование ЭВМ при их исполнении. **Совокупность таких вспомогательных программ называли математическим или программным обеспечением ЭВМ. Прикладные программы являются конечной целью программирования (производства). Системное программирование играет как бы роль средства производства.**

Для повышения эффективности работы компьютера (так как скорость работы оператора не может состязаться со скоростью работы процессора, и он часто простаивал) были разработаны системы пакетной обработки, которые автоматизировали всю последовательность действий операторов по организации вычислительного процесса. Фактически системы пакетной обработки явились прообразом современных операционных систем, они стали первыми системными программами, предназначенными не для обработки данных, а для управления вычислительным процессом. В ходе реализации систем пакетной обработки был разработан формализованный язык управления заданиями, с помощью которого программист сообщал системе и оператору, какие действия и в какой последовательности он хочет выполнить на вычислительной машине.

Оператор составлял пакет заданий (как правило, набор перфокарт, который иногда переносился на магнитную ленту или магнитный диск), которые в дальнейшем без его участия последовательно запускались на выполнение управляющей программой – *монитором* (находившейся на перфокартах или, в более поздних версиях, на магнитной ленте или магнитном диске).

Это способствовало повышению эффективности работы вычислительных центров, однако имело и свои недостатки, вызвав недовольство программистов.

Типичными машинами этого поколения были Мир, Наيري, Минск-2, Минск-22, Минск-32, М-220, Бэсм-6.

Структура и принципы работы ЭВМ 3-го поколения

Элементная база: ЭВМ 3-го поколения (70-е гг.) выполнены на полупроводниковых интегральных схемах с малой и средней степенью интеграции (сотни – тысячи транзисторов в одном корпусе).

Осуществлен переход к стандартным структурам данных и команд: байт (6, в дальнейшем 8 бит), полуслово (16 бит, 2 байта), слово (32 бита, 4 байта), двойное слово (64 бита, 8 байтов).

К этому времени можно констатировать существенное изменение в распределении функций между аппаратными и программными средствами компьютера: большую часть действий по организации вычислительного процесса взяла на себя операционная система.

Операционные системы (ОС) становились неотъемлемыми элементами компьютеров, играя роль «продолжения» аппаратуры. Все управление ВС автоматизировано. Верхний уровень автоматизации осуществляет комплекс программ, объединенных в операционную систему. Пользователь общается с ВС через ОС, которая синхронизирует работу аппаратной части ВС через систему прерываний и таймер. Более того, были реализованы практически все основные механизмы, присущие современным операционным системам: мультипрограммирование, мультипроцессирование, поддержка многотерминального многопользовательского режима, виртуальная память, разграничение доступа и сетевая работа. В эти годы начинается расцвет системного программирования, которое превращается в отрасль индустрии.

Режимы работы ВС 3-го поколения стали разнообразными. Самым простым режимом остался монопольный, в котором работали ЭВМ 1-го поколения. Использовался также пакетный режим, используемый при работе машин 2-го поколения.

Революционным событием данного этапа явилась промышленная реализация мультипрограммирования. Мультипрограммирование – способ организации вычислительного процесса, при котором в памяти компьютера находилось одновременно несколько программ, попеременно выполняющихся на одном процессоре.

Мультипрограммирование было реализовано в двух вариантах – в системах пакетной обработки и разделения времени. В мультипрограммном пакетном режиме процессор не простаивал, пока одна программа выполняла операцию ввода-вывода (как это происходило при последовательном выполнении программ в системах ранней пакетной обработки), а переключался на другую готовую к выполнению программу. Наличие в ВС периферийных и специальных процессоров, расслоение ОЗУ на отдельные независимо работающие блоки, развитая система прерываний и широкие возможности ОС смогли обеспечить одновременную параллельную обработку нескольких программ: одной – на центральном процессоре, остальных – на периферийном процессоре.

В мультипрограммных системах пакетной обработки пользователь по-прежнему был лишен возможности интерактивно взаимодействовать со своими программами. Системы разделения времени, рассчитанные на многотерминальные системы, вернули пользователю ощущение непосредственного контакта с компьютером. Создавалась иллюзия единоличного владения машиной за счет периодического выделения каждой программе своей доли процессорного времени.

Суть режима разделения времени состоит в следующем. Каждой программе, находящейся в оперативной памяти и готовой к исполнению, выделяется для исполнения фиксированный, задаваемый в соответствии с приоритетом пользователя интервал времени (интервал мультиплексирования). Если программа не выполнена до конца за этот интервал, ее исполнение принудительно прерывается, и программа переводится в конец очереди. Из начала очереди извлекается следующая программа, которая исполняется в течение соответствующего интервала мультиплексирования, затем поступает в конец очереди и т.д. в соответствии с циклическим алгоритмом. Если интервал мультиплексирования достаточно мал (приблизительно 200 мс), а средняя длина очереди готовых к исполнению программ невелика (приблизительно 10), то очередной квант времени выделяется программе каждые 2 с. В этих условиях ни один из пользователей практически не ощущает задержек, так как они сравнимы со временем реакции человека.

Одной из разновидностей режима разделения времени является *фоновый режим*, когда программа с более низким приоритетом работает на фоне программы с более высоким приоритетом.

Режим разделения времени был направлен на то, чтобы каждый из пользователей мог работать с ВС в диалоговом режиме с хорошей «реактивностью» (малым временем ответной реакции ВС). Для этого ВС снабжалась большим числом дисплейных терминалов. Этот режим использовался, например, при отладке задач, когда каждому пользователю на время работы

отводился отдельный дисплейный терминал. Перед предоставлением ВС пользователю, все необходимые на данном этапе данные задачи вводились в ОЗУ и инициировалась работа программы пользователя, но время, отводимое пользователю, лимитировано. Через определенный интервал времени от таймера выдавалось прерывание, и система переходила к решению другой задачи. Обработку прерываний и обслуживание запросов пользователей выполняла ОС.

Машины 3-го поколения работали и в режиме реального времени. Режим реального времени – это такой режим организации вычислительного процесса, при котором ЭВМ управляет некоторым внешним процессом, обрабатывая данные и информацию, непосредственно поступающую от объекта управления. Этот режим встречается, в основном, в динамических системах управления, наблюдения, диагностики и позволял быстро получать результат (хотя и дорогой ценой). Всякая система реального времени на самом деле работает с некоторым запаздыванием, величина интервала времени запаздывания принимается такой, чтобы основной процесс обслуживаемой системы не успел существенно изменить значений характеризующих его параметров, поэтому правильнее назвать такой режим режимом псевдореального времени.

Реализация мультипрограммирования потребовала внесения очень важных изменений в аппаратуру компьютера, непосредственно направленных на поддержку нового способа организации вычислительного процесса. При разделении ресурсов компьютера между программами необходимо обеспечить быстрое переключение процессора с одной программы на другую, а также надежно защитить коды и данные одной программы от непреднамеренной или преднамеренной порчи другой программой. В процессорах появился привилегированный и пользовательский режимы работы, специальные регистры для быстрого переключения одной программы на другую, средства защиты областей памяти, а также развитая система прерываний.

В привилегированном режиме, предназначенном для работы программных модулей операционной системы, процессор мог выполнять все команды, в том числе и те из них, которые позволяли осуществлять распределение и защиту ресурсов компьютера. Программам, работающим в пользовательском режиме, некоторые команды процессора недоступны. Таким образом, только ОС могла управлять аппаратными средствами и исполнять роль монитора и арбитра для пользовательских программ.

Прерывание – ситуация, возникающая в результате воздействия некоторого независимого события, приводящего к временному прекращению выполнения последовательности команд одной программы с целью выполнения последовательности команд другой программы. Механизм взаимодействия основной программы и подпрограммы прерывания не отличается от организации обращений к библиотечной программе и выхода из нее. Только там инициатива исходит от основной программы, а в случае прерываний – от сигнала прерывания.

Сигналы прерывания поступают на регистры прерывания, которые периодически, через короткий интервал времени, опрашиваются. Внутренние сигналы прерывания поступают от ОС, ЦУУ при окончании программы, при нехватке данных, при сбоях и т.д. Внешние прерывания поступают от периферийных процессоров, от периферийных устройств, закончивших свою работу, или готовых к работе, от таймера.

Система прерываний включает аппаратные и программные (входят в ОС) блоки. Ее назначение состоит в следующем: если по ходу работы ВС возникает необходимость выполнить срочную работу, не входящую в выполняемую в данный момент программу, то выполнение программы приостанавливается, включается подпрограмма требуемой работы и после ее выполнения вновь восстанавливается работа прерванной программы. Для этого при прерывании основной программы состояние всех регистров ЦУУ и использовавшихся

регистров АЛУ запоминается, а также запоминается адрес следующей по порядку команды основной программы, управление передается подпрограмме ОС, обрабатывающей прерывание. В конце этой подпрограммы помещены команды, восстанавливающие состояние ВС перед прерыванием, и последней командой управление вновь передается основной программе.

Введены специальные периферийные процессоры (каналы обмена, процессоры ввода-вывода, контроллеры внешнего устройства или просто контроллеры (К)) для управления ВЗУ и периферией, создан единый ресурс (селекторный и мультиплексный каналы, общая шина и т.д.), через который осуществляется взаимодействие между центральным процессором и другими устройствами (см. рис.).

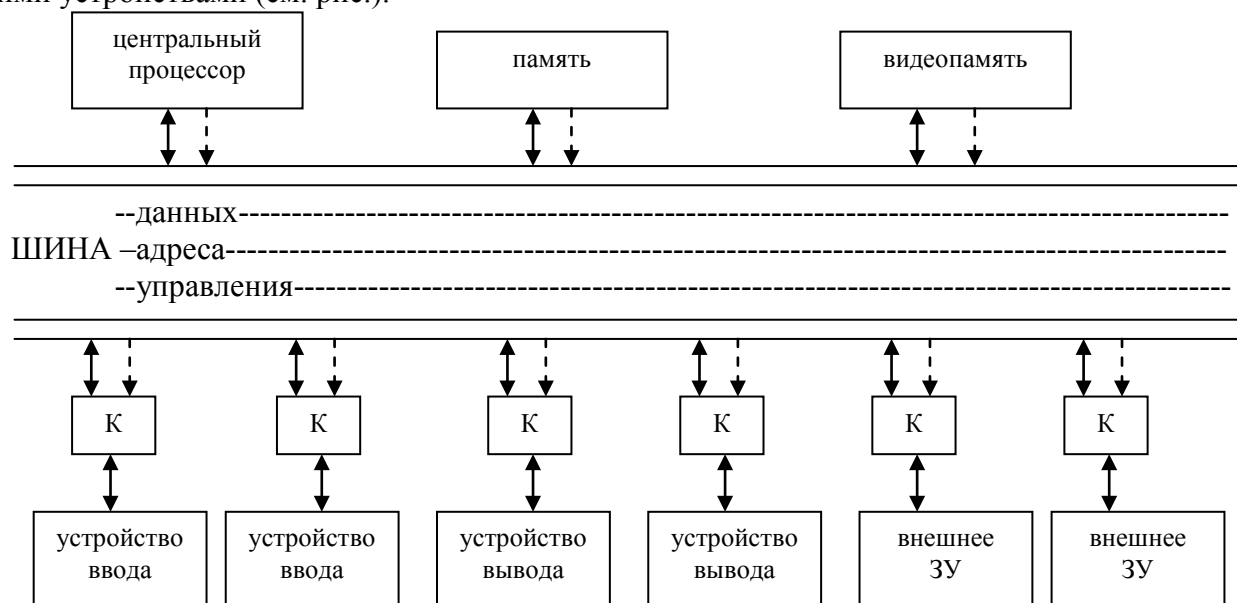


Рис. Шинная структура ЭВМ

Система прерываний позволяла синхронизировать работу различных устройств компьютера, работающих параллельно и синхронно (каналы ввода-вывода, диски, принтеры и пр.). Аппаратная поддержка операционных систем с тех пор стала неотъемлемым свойством практически любых ВС.

В ЭВМ 3-го поколения была установлена минимальная единица информации для обмена с ОЗУ – 1 байт. Ввели новые единицы измерения информации – машинное слово (4 байта), двойное слово, полуслово. Адрес (номер ячейки ОЗУ) стал относиться к одному байту. Адреса слов меняются уже не через 1; их приращение зависит от длины машинного слова.

В машинах третьего поколения появилось еще несколько особенностей: разная длина команд в зависимости от способа адресации данных, наличие специальной сверхоперативной регистровой памяти, вычисление эффективного адреса ОЗУ как суммы нескольких регистров и пр.

Память ЭВМ первых двух поколений была организована просто: она состояла из отдельных ячеек, содержимое каждой из которых считывалось как единое целое. Номер ячейки и получил название адреса. При необходимости обработки последовательно расположенных ячеек (массивов) адрес модифицировался значением специальных индексных ячеек. Меняя содержание индексной ячейки, можно было получить доступ к различным элементам массива.

В ЭВМ 3-го поколения *усилена иерархия памяти*: ОЗУ делится на блоки с независимыми системами управления, могущие работать одновременно; в процессоре появляются элементы ограниченной сверхбыстродействующей памяти на электронных регистрах; ячеечная структура ОЗУ дополняется более крупным структурным объединением – страницей, сегментом.

Значительно расширена номенклатура и число периферийных устройств внешней памяти, в том числе в качестве основного устройства внешней памяти вводятся накопители на магнитных дисках (НМД).

Несколько моделей одной архитектуры, отличающихся производительностью, но программно совместимых «снизу вверх», с совместимыми для всех моделей периферийными и внешними запоминающими устройствами, объединены в одно семейство (ряд).

В этот период стали активно создаваться и использоваться базы данных (БД). С 50-х до 70-х годов (в мировом масштабе) компании превратились в автоматизированные предприятия. Но самое главное, можно сказать, что, пришло понимание того, что надо делать, чтобы повысить эффективность работы с информацией: информация всегда доступна, оперативна, если она организована в БД.

Наиболее известными представителями машин данного поколения были машины серии IBM 360/370, ЕС ЭВМ.

СТРУКТУРА И ПРИНЦИПЫ РАБОТЫ ЭВМ 4-ГО И ПОСЛЕДУЮЩИХ ПОКОЛЕНИЙ В КОНТЕКСТЕ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ВС

Э.Дейкстра сказал, что «Для появления джентельмена, по убеждению англичан, нужны три благородных поколения; это справедливо и для вычислительных машин». Эволюция совершенствования средств вычислительной техники связана с постоянной борьбой за увеличение производительности (наращивание памяти, повышение надежности, улучшение технико-экономических показателей). Анализ возможностей совершенствования ЭВМ позволил выделить несколько факторов (предпосылок) совершенствования их архитектуры. Было замечено, что путь эволюционного совершенствования средств ВТ на основе принципов *последовательной обработки информации, фиксированности структуры* (невозможность адаптации ЭВМ к области применения) и *конструктивной неоднородности* не дает кардинальных улучшений их технических характеристик.

80-е годы можно назвать периодом проблем и противоречий. К этому времени вычислительная система стала более сложной. Она состояла уже из многих функциональных устройств, включая центральный процессор, вспомогательные (периферийные) процессоры, иерархическую систему запоминающих устройств, разветвленную систему устройств ввода-вывода и отображения информации, разветвленную систему управления, включая программную часть. Совершенно понятно, что общая производительность такой системы зависит от производительности и сбалансированности всех составляющих частей, а также от четкой целевой направленности на решение задач определенного класса. Специализированная ВС будет стоить во много раз дешевле, чем универсальная (общего пользования).

Конструктивный принцип неоднородности в машинах Неймана реализован на нескольких уровнях:

- структура ЭВМ в целом нерегулярна, а состав гетерогенный (неоднородный): каждое из 5 устройств (УУ и АЛУ, объединяемые в центральный процессор, память, внешнее запоминающее устройство, устройства ввода и вывода) имеет свое функциональное назначение и свою логическую организацию, основывается на специфических принципах, обладает своими особенностями технической реализации;
- в пределах всей конфигурации ЭВМ однородность достигается лишь фрагментарно (например, лишь оперативная память допускает однородную реализацию).

Построение неоднородных ЭВМ находится в резком противоречии с тенденцией развития микро- и нанoeлектроники.

Поэтому в качестве факторов совершенствования архитектуры ЭВМ стали рассматриваться отход от принципов *последовательной обработки информации, фиксированности структуры и неоднородности ЭВМ*.

В качестве одного из первых рассматривается отход от принципов последовательной обработки информации. Задача повышения производительности ВС всегда была основной для разработчиков систем. Существуют следующие способы повышения производительности ЭВМ при обработке информации:

1. совершенствование и разработка алгоритмов решения задач;
2. создание эффективного ПО и оптимизация программы;
3. повышение быстродействия и улучшение физико-технических свойств элементов и внутримашинных информационных каналов;
4. улучшение алгоритмов выполнения машинных операций и соответствующая модификация структуры процессора;
5. модернизация алгоритма управления вычислительными процессами и канонической структуры ЭВМ.

1-й и 2-ой способы основываются на фундаментальных достижениях математики, на тщательном анализе исходной задачи, выборе методов и алгоритмов ее решения, на скрупулезном программировании, в максимальной степени учитывающем архитектурные возможности машины. Для ЭВМ широкого назначения не дают ощутимого эффекта. Необходимым, но далеко не достаточным, является быстродействие центрального процессора, т.е. количество операций, выполняемых им за единицу времени (за секунду). Производительность ВС зависит, естественно, от загрузки процессора вычислениями, что в свою очередь, зависит от наличия для этого данных в памяти. Производительность ВС зависит и от возможностей и эффективности работы операционной системы и входящих в нее трансляторов и сервисно-отладочных программ. Поэтому при рассмотрении структурных схем ВС необходимо учитывать, насколько эффективно их функционирование поддерживается программным обеспечением.

3-ий способ – использование новых физических явлений, материалов, совершенствование технологии производства БИС. Для современной кремниевой технологии изготовления БИС достигнут практически теоретический предел показателей, который ограничивает плотность упаковки и быстродействие БИС.

В ЭВМ первых поколений производительность ВС увеличивалась в основном за счет увеличения быстродействия элементной базы. Но постепенно требования к увеличению времени срабатывания электронных схем, превысили возможности физической реализации логических элементов на несколько порядков. Время срабатывания электронных схем на кремниевой основе, даже в суперЭВМ составляло от 350 пикосекунд ($1\text{пс} = 10^{-12}\text{с}$) до 5 наносекунд ($1\text{нс} = 10^{-9}\text{с}$). Переход с кремниевой основы, применяемой в качестве полупроводника, на арсенид галлия или использование криогенной техники не могли обеспечить выполнение новых требований.

Кроме того, так как скорость прохождения сигнала не превосходит скорости света, то удаление электронных схем в устройстве хотя бы на 1 метр, приводит к запаздыванию сигнала², поэтому необходимо было уменьшать размеры машины, располагать ближе друг к другу логические элементы. Но миниатюризация центрального процессора, как необходимое условие

² Для примера рассмотрим задачу. Пусть расстояние между процессором и памятью равно 30см. Спрашивается, может ли такая ЭВМ работать со скоростью 600млн. оп/сек? Нет, не может, т.к. информация распространяется со скоростью $300\,000\text{км/сек} = 300\,000\,000\text{м/сек}$. При выполнении одной операции сигнал проходит $30+30=60\text{см/сек}$, для заданной скорости работы ЭВМ он должен проходить в секунду $600\,000\,000 \times 60\text{см/сек} = 36\,000\,000\,000\text{см/сек} = 360\,000\,000\text{м/сек} = 360\,000\text{км/сек}$, что невозможно.

поднятия тактовой частоты ЭВМ, выдвигало сразу же проблему отвода тепловой энергии, выделяемой в ходе работы логических элементов.

4-ый способ – поиск форм представления чисел и алгоритмов, убыстряющих реализацию машинных операций. Основные логические операции (типа сравнения) дальнейшему ускорению не поддаются (значительно возрастает техническая сложность в реализации процессора).

5-ый способ – модернизация алгоритма управления: вычислительного процесса, т.е. канонической функциональной структуры ЭВМ. Конвейерные вычисления – наиболее яркая новация.

Направлениями устранения ограничений на пути развития производительности ВС становятся конвейеризация и векторизация вычислений, имевшие довольно глубокие аналоги с организацией производства, и параллелизм в обработке информации (многомашинные и многопроцессорные системы). При этом предполагается как параллельное выполнение вычислительной системой нескольких программ, так и распараллеливание во времени выполнения одной программы.

Конвейер - возможность независимого выполнения (как на настоящем конвейере) последовательности действий машинного такта. В случае классического конвейера это есть такие действия, как: выборка команды, дешифрация команды, подготовка операндов, выполнение команды, запись результатов. На первом шаге производится выборка первой команды. На втором шаге первая команда подается на дешифрацию, и производится выборка второй команды. На третьем шаге для первой команды проводится подготовка операндов, для второй – дешифрация, для третьей – выборка и т.д. Через пять таких шагов с конвейера выходит первая выполненная команда. Каждая следующая команда выходит на каждом следующем шаге, а не через пять очередных шагов, как это было бы в рамках традиционной архитектуры. Иногда метод конвейеризации называют «опережающей выборкой». Ведь по окончании команды чаще всего оказывается, что следующая команда уже выбрана из ОЗУ, дешифрована и подготовлена к исполнению. В случае нарушений естественного порядка выполнения команд в программе (например, при безусловном переходе) опережающая выборка оказывается напрасной и конвейер очищается. Следующая за переходом команда выполняется дольше, так как, чтобы конвейер «работал на полную мощность», необходимо его предварительно заполнить. Таким образом, в конвейерной машине время выполнения программы может зависеть не только от составляющих ее команд, но и от их взаимного расположения.

Классический конвейер имеет пять названных ступеней. Суперконвейерная архитектура – число ступеней, большее 5.

В задачах, где удастся создать длинные векторы операндов, целесообразно введение в систему команд ВС команд для быстрой выборки из ОЗУ и формирования векторов. ВС такого типа называются **векторными**.

ступени конвейера	«выход» команды					
				1-ой	2-ой	
5. запись результатов					1	2
4. выполнение команды				1	2	3
3. подготовка операндов			1	2	3	4
2. дешифрация		1	2	3	4	5
1. выборка команды	1	2	3	4	5	6

Рис. Схема классического конвейера

Одним из направлений устранения ограничений на пути развития производительности ВС становится отныне дальнейшее развитие иерархической многоуровневой системы ЗУ с обменом данными в соседних уровнях ЗУ, не зависящим от работы ЗУ других уровней.

Таковыми новыми ВЗУ явились *накопители на магнитных дисках типа «Винчестер»*, с несъемным фиксированным пакетом дисков, помещаемым в изолированный от внешней среды объем, затем электронные диски.

Стали активно использоваться и так называемые *постоянные (пассивные) ЗУ* (ПЗУ), или память типа ROM (Read Only Memory), в которых данные, будучи раз записаны, хранятся надежно и долго. Обращение к этим данным (их считывание) может быть осуществлено с тем же темпом или даже быстрее, чем к данным, хранящимся в ОЗУ, но зато нельзя однажды записанные в ячейки данные, заменять другими.

Большое значение приобрели *кэш-память* и *регистровая память*. Кэш-память используется для помещения в нее наиболее часто употребляемых частей программы и данных на очередном этапе счета или для хранения общих изменяемых данных при распараллеливании решения одной задачи на многопроцессорных комплексах. Она более быстродействующая, чем ОЗУ, но с меньшим объемом. Регистровая память является наиболее быстродействующей.

Иногда регистровую память применяют совместно со *стековой (магазинной) памятью*. В ней устанавливается очередность – «последний вошел – первый вышел» - LIFO (last in first out) или «первым вошел – первым вышел» – FIFO (first in first out). Удобство обращения к стеку в режиме LIFO заключается в том, что запись в стек и считывание из него производятся через ячейку, называемую вершиной (указателем) стека. Данные, занесенные через стек, не меняют своих физических адресов, а «плавающим» вверх – вниз является адрес в указателе вершины стека. Стековая память используется для различных целей: хранения промежуточных результатов и значений параметров при обращении к процедуре и организации ее работы, при получении результатов и выходе из процедуры, для запоминания и восстановления состояния машины при прерывании программы и выходе из прерывания, при программировании вычислений арифметических выражений³.

При программировании больших задач испытывается недостаток в ресурсах памяти верхнего уровня. Естественным решением является предоставление пользователю всего объема доступной памяти как памяти одного верхнего уровня. Этот объем памяти может быть структурирован постранично, а внутри страницы – по ячейкам, как в ОЗУ. Таким образом, для пользователя объем прямоадресуемой оперативной памяти значительно больший, чем физический объем памяти ОЗУ. Этот объем памяти носит название *виртуальной памяти*. Для программиста виртуальная память эквивалентна ОЗУ, а в ЭВМ она распределена по нескольким иерархиям ЗУ, и задача ОС состоит в том, чтобы вовремя обеспечить опережающую подкачку нужных страниц в верхние уровни.

Следует упомянуть и о двух современных взаимно конкурирующих направлениях в построении системы команд ЭВМ: компьютер с полным набором команд CISC (Complex Instruction Set Computer) и с ограниченным набором – RISC (Reduced Instruction Set Computer).

³ Существует польская инверсная форма записи арифметических выражений, при которой без скобок, согласно последовательности действий записываются операнды и следом – знак операции. Если записывается один операнд и знак операции, то в качестве второго операнда берется результат предыдущей операции; если записывается только знак операции, то в качестве операндов берутся два предыдущих результата. Если результат не используется в следующей операции, то он запоминается. Например, пусть переменной X надо присвоить значение дроби, числитель которой равен $(a+b)*(c+d)*f$, а знаменатель равен k. В польской записи это выглядит так: $ab+cd+*f*k/$. Если занести эту запись в стек справа налево, то получится кусок программы, для выполнения которого достаточно знать только адрес вершины стека. Последней операцией будет запись в вершину стека числа X.

Разделение возникло из-за того, что основную часть времени компьютеру приходится выполнять небольшую часть из своего набора команд, остальные же используются эпизодически. Таким образом, если существенно ограничить набор операций до наиболее простых и коротких, зато тщательно оптимизировать их, получится достаточно эффективная и быстродействующая RISC-машина. Правда за скорость придется платить необходимостью программной реализации «отброшенных» команд, но для решения ряда задач быстродействие существенно важнее проблем программирования.

Основной набор команд слабо изменился в ходе бурной эволюции ЭВМ. в то же время способы указания адреса расположения информации в памяти претерпели значительное изменение.

Команды могут быть одно-, двух- и трехадресные в зависимости от числа участвующих в них операндов. Трехадресная команда легко расшифровывалась и была удобна в использовании, но с ростом объемов ОЗУ ее длина становилась непомерно большой. Поэтому появились двухадресные команды, исключавшие адрес записи результата. В таких случаях результат операции оставался в специальном регистре (сумматоре) и был пригоден для использования в следующих вычислениях. В ряде случаев результат записывался вместо одного из операндов. В системе команд ЭВМ делаются двухадресные команды с адресами регистра-регистра или регистра-ОЗУ. Эти команды вычисляются быстрее, чем команды ОЗУ-ОЗУ. Дальнейшее упрощение привело к появлению одноадресных команд.

Типовая структура двухадресной команды – КОП A1 A2,
а одноадресной команды – КОП A1.

Является неверным представление, что для решения задачи одноадресных команд потребуется в три раза больше, чем трехадресных. Например, для сложения двух чисел (с адресами ОЗУ A1 и A2) и записи их результата по адресу A3 требуется выполнить три команды: извлечь содержимое ячейки A1 в сумматор; сложить сумматор с числом из A2; записать результат из сумматора в A3. Но уже для вычисления выражения $A5 = (A1 + A2) * A3 / A4$ потребуется три трехадресных команды и всего пять одноадресных.

Заметим, что при использовании стека возможна реализация и безадресной (нуль-адресной) машины.

Начиная с четвертого поколения доминирует байтовая организация памяти, и байт является минимально адресуемой единицей информации.

Разнообразие ВС, классифицированное по взаимодействию ЦП, ОЗУ и ПП, можно свести к следующим структурам (для простоты ограничимся рассмотрением только структурных связей в архитектуре ВС):

- *Однопроцессорная (с одним ЦП и периферийным процессором или каналом), со связью через ЦП. ЦП с использующей его ОС осуществляет всю координацию работы системы, устанавливая при необходимости прямые связи с периферийными устройствами через каналы или ПП. Такая структура характерна для большинства ЭВМ средней производительности 3-го поколения (IBM 360/370, ЕС ЭВМ и др.);*
- *С одним мощным ЦП и обслуживающими его несколькими периферийными процессорами. Для сверхпроизводительных систем нецелесообразно часто прерывать работу центрального процессора и переводить его в режим работы ОС, чтобы прореагировать на любое изменение, обусловленное сигналом прерывания. Концентрическая структура с ЦП, который взаимодействует с ОЗУ и рядом вспомогательных процессоров, подготавливающих для него работу и выводящих от него результаты, позволяет полнее использовать время ЦП, упростить ОС, рассредоточить ее действия между ЦП и одним из ПП. ЦП в таких ВС, как правило,*

бывает достаточно сложным, состоящим из множества отдельных функциональных устройств.

- *Многопроцессорная (с несколькими ЦП и ПП и секционированием ОЗУ).* Ведь повышать производительность ВС можно не только путем увеличения быстродействия ЦП и освобождения его от выполнения подпрограмм ОС, но и путем увеличения числа ЦП в системе с общим ЗУ. При этом система может работать как на решение одной общей задачи по распараллеленному на несколько ЦП алгоритму, так и решать на каждом из них свою задачу.
- *С магистральной шиной.* Концепция магистрально-шинной структуры состоит в том, что все части ВС (ЦП, ОЗУ, ВЗУ, периферия) могут быть подсоединены к одной общей многопроводной шине, захватывая ее на короткий период машинного такта для обмена информацией. Чтобы не возникали конфликтные ситуации между устройствами при захвате шины, имеется устройство, допускающее захват шины только одним обменом в данный момент времени. Эта структура нашла широкое применение в мини- и микро-ЭВМ, в связях объектов с ЭВМ в автоматизированных системах научных исследований. Недостатком структуры считалось возрастание конфликтных столкновений при увеличении количества подключенных к шине приборов и при интенсификации обменов между ними. Это ограничение было снято развитием многошинных структур.
- *Сетевая.* Идея вычислительных сетей состоит в том, что территориально разнесенные ЭВМ образуют узлы, соединенные линиями связи. Между узлами происходит передача сообщений по единым и строго регламентированным правилам – протоколам. В узлах сообщения проходят некоторую обработку: проверку правильности передачи, определение, адресовалось ли данное сообщение именно этому узлу (если да, то оно передается находящейся в узле основной ЭВМ, в противном случае – направляется в следующий узел, лежащий на пути к адресу назначения). В зависимости от длины линий связи между узлами сети могут быть глобальными (межконтинентальными), государственными, региональными (для обслуживания района или группы родственных предприятий), локальными с длинами линий связи порядка 1-2км. При стягивании сети с уменьшением длин линий связи до порядка метров, но с сохранением правил передачи информации между узлами и переработки ее в узлах могут быть образованы мультипроцессорные ВС сетевой архитектуры.
- *Функционально-перестраиваемая многопроцессорная;*
- *Масс-процессорная (ВС с числом процессоров порядка сотен и тысяч).*

Разнообразие ВС, классифицированное по взаимодействию потока команд и потока данных (операндов), можно свести к следующим структурам:

ОКОД - один поток команд – один поток данных (одна команда – одни данные). В ЭВМ классической архитектуры ведется последовательная обработка данных. Команды поступают одна за другой (за исключением точек ветвления программы), и для них из ОЗУ или регистров также последовательно поступают операнды. Одной команде соответствует один необходимый для нее набор операндов (как правило, два для бинарных операций).

ОКМД – одна команда – много данных. Этот тип охватывает такие структуры, в которых одной командой обрабатывается набор из множества данных. На этой основе строятся специальные процессоры для обработки изображений и др.

МКОД – много команд – один поток данных. Примером может служить конвейер, т.е. последовательная обработка одного потока данных многими обрабатывающими ступенями конвейера.

МКМД – много команд – много данных. Этот тип предусматривает наиболее полное и независимое распараллеливание процесса.

Данная классификация отражает основные взаимодействия потоков данных и команд, но все же не охватывает всех возможных взаимодействий, встречающихся на практике.

Как мы отмечали, в этот период происходит смещение интереса в сторону поиска и апробации путей повышения производительности ВС за счет архитектурных решений. В индустрии больших компьютеров интерес смещается из сферы самих компьютеров в сферу процессов проектирования и построения приложений, т.е. разработка приложений переходит из области искусства в область инженерной дисциплины (автоматизированную разработку программных средств).

Сравнительная характеристика показателей ЭВМ разных поколений представлена в таблице 1.1.

Классификацию ЭВМ можно проводить на основе разных классификационных признаков.

По назначению ЭВМ можно разделить на: универсальные (общего назначения), проблемно-ориентированные и специализированные.

Универсальные ЭВМ предназначены для решения самых различных инженерно-технических задач (экономических, математических, информационных и др.), отличающихся сложностью алгоритмов и большим объемом обрабатываемых данных. Их основными чертами являются: высокая производительность, большая емкость оперативной памяти, разнообразие форм обрабатываемых данных (двоичных, десятичных, символьных, при большом диапазоне их изменения и точности их представления), обширная номенклатура выполняемых операций (арифметических, логических, специальных), развитая организация системы ввода-вывода, обеспечивающая подключение разнообразных видов внешних устройств.

Проблемно-ориентированные ЭВМ служат для решения более узкого круга задач, как правило, с управлением технологическими объектами; регистрацией, накоплением и обработкой относительно небольших объемов данных; выполнением расчетов по относительно несложным алгоритмам. Они обладают ограниченными по сравнению с универсальными ЭВМ аппаратными и программными ресурсами.

Специализированные ЭВМ (например, программируемые микропроцессоры специального назначения, адаптеры и контроллеры, выполняющие логические функции управления отдельными несложными техническими устройствами, агрегатами и процессами, устройства согласования и сопряжения работы узлов вычислительных систем) используются для решения узкого круга задач или реализации строго определенной группы функций. Это позволяет четко специализировать их структуру, существенно снизить сложность и стоимость при сохранении высокой производительности и надежности их работы.

Таблица 1.1. Сравнительная характеристика показателей ЭВМ разных поколений.

показатели	поколения ЭВМ					
	1	2	3	4		5
год появления	1949-1954	1958-1960	1965-1966	А) 1976-1979	Б) С 1983г.	
элементная база процессора	электронные лампы	транзисторы	ИС	большие ИС (БИС)	сверхбольшие ИС (СБИС)	+оптоэлектроника+криоэлектроника
элементная база ОЗУ	электронно-лучевые трубки	ферритовые сердечники	ферритовые сердечники	БИС	СБИС	СБИС
возможность пользования	задача, пассивный режим	набор задач, пассивный режим	набор задач, активный режим			

алгоритм управления, структура	последовательный алгоритм, фиксированная структура	последовательный алгоритм, фиксированная структура	последовательно-параллельный алгоритм, ручное изменение структуры	решение сложной задачи	суперсложная система задач	
производство	индивидуальное	мелкосерийное	серийное			
ПО	машинные языки	+ ассемблер, трансляторы, диспетчеры	+ процедурные языки высокого уровня (ЯВУ), система языков ОС	+ новые процедурные ЯВУ	+ непроцедурные ЯВУ	+ новые непроцедурные ЯВУ
средства обмена	УВВ	УВВ, каналы связи	УВВ, каналы связи, оптические устройства			
производительность опер/с	10^4	10^6	10^7	10^8	10^9 + многопроцессорность	10^{12} + многопроцессорность
объем ОЗУ (байт) V	10^2	10^3	10^4	10^5	10^7	10^8
безотказность (часы) g	1..10	10^2	10^3	$\geq 10^4$	$\geq 10^5$	
средства связи пользователя с ЭВМ	пульт управления и перфокарты	перфокарты и перфоленты	алфавитно-цифровой терминал	монохромный графический дисплей, клавиатура	цветной графический дисплей, клавиатура, мышь и др.	+ устройство голосовой связи с ЭВМ
цена опер/с σ	10	10^0	10^{-1}	$\leq 10^{-2}$	$\leq 10^{-3}$	

По размерам и функциональным возможностям ЭВМ можно разделить на: сверхбольшие (суперЭВМ), большие, малые, сверхмалые (микроЭВМ).

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

Первая большая ЭВМ ЭНИАК (1946г.) имела массу более 50 тонн, быстродействие – несколько сотен операций в секунду, оперативную память емкостью 20 чисел и занимала огромный зал площадью около 100 кв.м. Большие ЭВМ часто называют *мэйнфреймами* (Mainframe). Основные направления эффективного применения мэйнфреймов – решение научно-технических задач, работа в вычислительных системах с пакетной обработкой информации, работа с большими базами данных, управление вычислительными сетями и их ресурсами. Последнее направление – использование мэйнфреймов в качестве больших серверов вычислительных сетей – часто отмечается специалистами среди наиболее актуальных. Родоначальником современных больших ЭВМ, по стандартам которого в последние несколько десятилетий развивались ЭВМ этого класса в большинстве

стран мира, является фирма IBM. Ее модели IBM 360 и IBM 370, их архитектура и программное обеспечение взяты были за основу при создании отечественной системы больших ЭВМ – ЕС ЭВМ.

Избыточность ресурсов больших ЭВМ для ряда приложений, а также прогресс в области электронной элементной базы привели к появлению *малых ЭВМ*. Малые ЭВМ (мини-ЭВМ) – надежные, недорогие и удобные в эксплуатации компьютеры, обладающие несколько более низкими по сравнению с мэйнфреймами возможностями. Наиболее мощные из них – супермини-ЭВМ, возникновение которых явилось результатом успехов в области элементной базы и архитектурных решений. Этот класс компьютеров по архитектуре, размерам и стоимости относится к классу малых ЭВМ, а по производительности – сравним с большими ЭВМ.

Изобретение микропроцессора (1969г.) привело к появлению в 70-х гг. еще одного класса ЭВМ – *микроЭВМ*. Именно наличие микропроцессора (МП)⁴ послужило первоначально определяющим признаком микроЭВМ. Сейчас микропроцессоры используются во всех без исключения классах ЭВМ. МикроЭВМ обычно классифицируют следующим образом:

- *Многопользовательские микроЭВМ* – мощные микроЭВМ, оборудованные несколькими видеотерминалами и функционирующие в режиме разделения времени, что позволяет эффективно работать на них сразу нескольким пользователям.
- *Персональные ЭВМ (ПК)* – однопользовательские микроЭВМ, удовлетворяющие требованиям общедоступности и универсальности применения. Для этого ПК должен иметь следующие характеристики: малую стоимость, находящуюся в пределах доступности для индивидуального покупателя; автономность эксплуатации без специальных требований к условиям окружающей среды; гибкость архитектуры, обеспечивающую адаптивность у разнообразным применениям в сфере управления, науки, образования, в быту; «дружественность» интерфейса ОС и прочего программного обеспечения, обуславливающую возможность работы с ней пользователя без специальной профессиональной подготовки; высокую надежность работы. Среди ПК следует отметить компьютеры фирм: IBM, Compaq Computer, Apple (Macintosh), Hewlett Packard, Dell, DEC (Digital Equipment Corporation) и др. Самыми распространенными сегодня являются, пожалуй, IBM PC с микропроцессорами Pentium и Pentium Pro.
- *Рабочие станции* – однопользовательские мощные ЭВМ, специализированные для выполнения определенного вида работ (графических, инженерных, издательских и др.).
- *Серверы* – многопользовательские мощные микроЭВМ в вычислительных сетях, выделенные для обработки запросов от всех станций сети.

Следует отметить, что данная классификация весьма условна, так как мощный современный ПК, оснащенный проблемно-ориентированным программным и аппаратным обеспечением, может использоваться и как полноправная рабочая станция, и как многопользовательская микроЭВМ, и как хороший сервер, по своим характеристикам почти не уступающий малым ЭВМ.

Однако производительность даже больших ЭВМ оказалась недостаточной для ряда задач: например, прогнозирования метеообстановки, управления сложными оборонными комплексами, моделирования экологических систем и др. Это явилось предпосылкой для разработки и создания *суперЭВМ*, самых мощных вычислительных систем, интенсивно развивающихся в настоящее время. К суперЭВМ относятся мощные многопроцессорные вычислительные машины с быстродействием сотни миллионов – десятки миллиардов операций в секунду.

⁴ МП представляет собой функционально законченное устройство обработки информации, управляемое хранимой в памяти программой и конструктивно выполненное в виде интегральной схемы.

Создать такую высокопроизводительную ЭВМ по современной технологии на одном микропроцессоре не представляется возможным ввиду ограничения, обусловленного конечным значением скорости распространения электромагнитных волн, потому что время распространения сигнала на расстояние несколько миллиметров (линейный размер стороны МП) при быстродействии 100 млрд.оп/с становится соизмеримым с временем выполнения одной операции. Поэтому суперЭВМ создаются в виде высокопараллельных *многопроцессорных вычислительных систем* (МПВС), имеющих несколько разновидностей:

Магистральные (конвейерные) МПВС, в которых процессоры одновременно выполняют разные операции над последовательным потоком обрабатываемых данных - тип МКОД или MISD (Multiple Instruction Single Data).

Векторные МПВС, в которых все процессоры одновременно выполняют одну команду над различными данными – тип ОКМД или SIMD (Single Instruction Multiple Data).

Матричные МПВС, в которых МП одновременно выполняют разные операции над несколькими последовательными потоками обрабатываемых данных – МКМД или MIMD (Multiple Instruction Multiple Data).

В суперЭВМ используются все три варианта архитектуры МПВС.

Главной тенденцией развития вычислительной техники в настоящее время является дальнейшее расширение сфер применения ЭВМ и, как следствие, переход от отдельных машин к их системам – вычислительным системам и комплексам разнообразных конфигураций с широким диапазоном функциональных возможностей и характеристик. Наиболее перспективные, создаваемые на основе персональных ЭВМ, территориально распределенные многомашинные вычислительные системы – *вычислительные сети* – ориентируются не столько на вычислительную обработку информации, сколько на коммуникационные информационные услуги: электронную почту, системы телеконференций и информационно-справочные системы.

Объединение глобальных, региональных и локальных сетей позволяет создавать многосетевые иерархии для обработки огромных информационных массивов и доступа к неограниченным информационным ресурсам.

Специалисты считают, что в начале XXI века в цивилизованных странах произойдет смена основной информационной среды (удельный объем информации, получаемой по компьютерным сетям, превысит объем информации, получаемой по традиционным каналам – радио, телевидение, печать). Широкое внедрение средств мультимедиа, в первую очередь аудио- и видеосредств ввода и вывода информации, позволит общаться с компьютером на естественном языке.

Принцип централизованной обработки данных не отвечает высоким требованиям к надежности процесса обработки. Появление малых ЭВМ, микроЭВМ и ПК потребовало нового подхода к организации обработки данных, а именно: перехода от использования отдельных ЭВМ в системах централизованной обработки данных к распределенной их обработке.

Распределенная обработка данных – обработка данных, выполняемая на независимых, но связанных между собой компьютерах, представляющих распределенную систему.

Реализацию распределенной обработки данных можно проводить на многомашинных вычислительных комплексах (локальных – группе компьютеров, установленных в одном помещении, объединенных с помощью средств сопряжения, дистанционных – группе компьютеров, установленных на значительном расстоянии от центральной ЭВМ и использующих для передачи данных телефонные каналы связи). Но для этой цели служат и компьютерные (вычислительные) сети. Основное отличие сети от многомашинного вычислительного комплекса – это размерность (сеть может состоять из десятков и даже сотен ЭВМ, расположенных друг от друга от нескольких метров до десятков, сотен и даже тысяч метров), разделение функций между ЭВМ, необходимость решения задачи маршрутизации

сообщений (сообщение может быть передано по различным маршрутам в зависимости от состояния каналов связи, соединяющих ЭВМ друг с другом).

Абонент сети — объект, потребляющий или генерирующий информацию в сети. Абонентами сети могут быть отдельные ЭВМ, комплексы ЭВМ, терминалы, промышленные роботы, станки с числовым программным управлением и т.д. Любой абонент сети подключается к *станции* (аппаратуре, выполняющей функции, связанные с передачей и приемом информации).

Для организации взаимодействия абонентов в сети необходима физическая передающая среда — линии связи или пространство, в котором распространяются электрические сигналы, и аппаратура передачи данных. Физическая передающая среда сети представлена тремя типами кабелей: *витая пара* (самый простой ее вариант — телефонный провод, популярна из-за дешевизны, но имеет низкую скорость передачи и плохую помехозащищенность); *коаксиальный кабель* (толстый и тонкий); *оптоволоконный кабель* — идеальная передающая среда.

Обработка данных в компьютерных сетях распределена между двумя объектами: *клиентом* и *сервером*.

Сервер — компьютер, подключенный к сети и обеспечивающий ее пользователей определенными услугами. Сервер выполняет запрос, поступивший от клиента. Серверы могут осуществлять хранение данных, управление базами данных, удаленную обработку заданий, печать заданий и ряд других функций. Сервер — источник ресурсов сети. Одним из самых распространенных типов является файловый сервер, хранящий данные пользователей сети и обеспечивающий им доступ к этим данным. Это компьютер с оперативной памятью и жесткими дисками большой емкости, а также дополнительными накопителями на магнитной ленте (стримерами).

Клиент — это задача, рабочая станция или пользователь компьютерной сети. Клиент обрабатывает полученные данные и представляет результаты обработки в виде, удобном для пользователя. В принципе обработка данных может быть выполнена и на сервере.

Рабочая станция — персональный компьютер, подключенный к сети, через который пользователь получает доступ к ее ресурсам. Рабочая станция функционирует как в сетевом, так и в локальном режиме. Она оснащена собственной ОС и обеспечивает пользователя всеми необходимыми инструментами для решения прикладных задач.

Для подобных систем приняты термины — системы клиент-сервер или архитектура клиент-сервер. Использоваться такая архитектура может как в одноранговых локальных вычислительных сетях, так и в сети с выделенным сервером.

В *одноранговой сети* нет единого центра управления взаимодействием рабочих станций и нет единого устройства для хранения данных. Сетевая ОС распределена по всем рабочим станциям. Каждая станция сети может выполнять функции как клиента, так и сервера. Пользователю сети доступны все устройства, подключенные к другим станциям (диски, принтеры). Достоинством одноранговых сетей является низкая стоимость и высокая надежность. Недостатком — зависимость эффективности работы от количества станций, сложность управления сетью, сложность обеспечения защиты информации, трудности обновления и изменения программного обеспечения станций.

В *сети с выделенным сервером* один из компьютеров выполняет функции хранения данных, предназначенных для использования всеми рабочими станциями и ряд сервисных функций. Такой компьютер называют сервером сети. На нем устанавливается сетевая ОС, к нему подключаются все разделяемые внешние устройства (жесткие диски, принтеры, модемы). Взаимодействие между рабочими станциями в сети, как правило, осуществляется через сервер. Достоинства этого типа сети — надежная защита информации, высокое быстродействие, отсутствие ограничений на количество рабочих станций, простота управления по сравнению с одноранговыми сетями. Недостатки — высокая стоимость из-за выделения одного компьютера

под сервер, зависимость быстродействия и надежности сети от сервера, меньшая гибкость по сравнению с одноранговой сетью.

Для способа обращения к передающей среде и методов управления сетью безразлично, как расположены абонентские ЭВМ.

Топология ЛВС – это усредненная геометрическая схема соединений узлов сети (т.е. устройств, непосредственно подключенных к передающей среде сети). Топологии вычислительных сетей могут быть самыми различными, но для локальных вычислительных сетей типичными являются всего три: кольцевая, шинная, звездообразная.

Кольцевая топология предусматривает соединение узлов сети замкнутой кривой. Информация передается от узла к узлу по кольцу. Каждый промежуточный узел между передатчиком и приемником ретранслирует посланное сообщение. Принимающий узел распознает и получает только адресованные ему сообщения.

Шинная топология – одна из самых простых. Данные от передающего узла сети распространяются по шине в обе стороны. Промежуточные узлы не ретранслируют поступающих сообщений.

Звездообразная топология базируется на концепции центрального узла, к которому подключаются периферийные узлы. Каждый периферийный узел имеет свою отдельную линию связи с центральным узлом. Вся информация передается через центральный узел, который ретранслирует, переключает и маршрутизирует информационные потоки в сети. Работоспособность сети целиком зависит в данном случае от центрального узла.

Следует отметить, что передающая среда является общим ресурсом для всех узлов сети. Чтобы получить возможность доступа к этому ресурсу из узла сети, необходимы специальные механизмы – методы доступа к передающей среде (т.е. совокупности правил, по которым узлы получают доступ к ресурсу).

При детерминированном методе доступа передающая среда распределяется между узлами с помощью специального механизма управления, гарантирующего передачу данных узла в течение некоторого, достаточно малого интервала времени. Наиболее распространенными детерминированными методами доступа являются метод опроса и метод передачи права.

При недетерминированном методе доступа предусматривается конкуренция всех узлов сети за право передачи. Наиболее распространенным недетерминированным методом доступа является CSMA/CD - множественный метод доступа с контролем несущей частоты и обнаружением коллизий.

В случае необходимости расширения ЛВС (например, при исчерпании существующих технических возможностей сети), происходит объединение нескольких ЛВС.

Самый простой вариант объединения ЛВС – *мост* – это объединение двух сетей, использующих одинаковые методы передачи данных. Мост – это, как правило, отдельная ЭВМ со специальным ПО и дополнительной аппаратурой. Хотя иногда мост может располагаться на одной из ЭВМ сети и совмещать функцию моста с функцией абонентской ЭВМ. Мост может соединять сети разных топологий, но работающие под управлением однотипных сетевых ОС.

Сеть сложной конфигурации, представляющая собой соединение нескольких сетей, нуждается в специальном устройстве, задача которого – отправить сообщение адресату в нужную сеть. Называется такое устройство *маршрутизатором* (или роутером). Он обеспечивает балансировку нагрузки в сети, перенаправляя потоки сообщений по свободным каналам связи. Маршрутизатор может выбрать наилучший путь для передачи сообщения абоненту сети, фильтрует информацию, проходящую через него, направляя в одну из сетей только ту информацию, которая ей адресована.

Для объединения ЛВС совершенно различных типов, работающих по существенно отличающимся друг от друга протоколам, предусмотрены специальные устройства – *шлюзы*. С

помощью шлюзов можно подключить локальную сеть к глобальной или же к главному компьютеру.

Мосты, маршрутизаторы и даже шлюзы конструктивно выполняются в виде плат, устанавливаемых в компьютерах.

Internet («между сетей») – глобальная компьютерная сеть, соединяющая отдельные сети. Важной ее особенностью является равноправие всех подключенных компьютеров и отсутствие иерархии. Каждый подключенный к сети компьютер имеет свой адрес, по которому его может найти абонент из любой точки сети. К адресам станций предъявляются специальные требования: он должен нести некоторую информацию о своем владельце и иметь формат, позволяющий вести его обработку автоматически. С этой целью для каждого компьютера устанавливаются два адреса: цифровой IP-адрес и доменный адрес. Цифровой адрес удобен для обработки на компьютере, доменный – для восприятия пользователем.

Цифровой адрес имеет длину 32 бита. Для удобства он разделяется на 4 блока по 8 бит, которые можно записать в десятичном коде. Два блока определяют адрес сети, один – адрес подсети, один адрес компьютера в подсети. Например, в десятичном адресе 196.23.9.200: адрес сети – 196.23, адрес подсети – 9, адрес компьютера – 200.

Чтобы абонентам Internet было достаточно просто связаться друг с другом, все пространство ее адресов разделяется на области – домены. В системе адресов Internet приняты домены, представленные географическими регионами и имеющие имя, состоящее из букв (Беларусь – by, Россия – ru, США – us, Франция – fr и т.д.), домены, разделенные по тематическим признакам, имеющие трехбуквенное сокращенное название (учебные заведения - edu, правительственные учреждения - gov, коммерческие организации - com). Доменный адрес, определяет область, представляющую ряд хост-компьютеров (т.е. компьютеров, самостоятельно подключенных к Internet). В отличие от цифрового адреса он читается в обратном порядке: вначале идет имя компьютера, затем имя сети, в которой он находится. Компьютерное имя включает, как минимум два уровня доменов, отделяющихся друг от друга точкой. Слева от домена верхнего уровня располагаются другие имена – поддомены для общего домена. Например, в имени tutor.sptu.edu: tutor – поддомен sptu, который является поддоменом edu.

Для пользователей Internet адресами могут быть просто их регистрационные имена на компьютере, подключенном к сети. За именем следует знак @. Все это слева присоединяется к имени компьютера. Например, victor@tutor.sptu.edu.

Среди популярных служб Internet можно назвать:

- *Электронную почту*, основным преимуществом которой является независимость от времени (письмо приходит сразу же после его отправления и хранится в почтовом ящике до получения адресатом).
- *World-Wide-Web* (всемирная информационная сеть- паутина), предоставляющая клиентам возможность взаимодействовать с другими приложениями Internet с помощью графического интерфейса, используя при этом технологию гипертекста (текста, содержащего в себе связи с другими текстами, графической, видео- или звуковой информацией). Для создания таких текстов используется специальный язык HTML. HTML-документы представляют собой текстовые файлы, в которые встроены специальные команды. Связь между гипертекстовыми документами осуществляется с помощью ключевых слов. Найдя такое слово, пользователь может перейти в другой документ, который также будет иметь гипертекстовые ссылки.
- *Телеконференции Usenet*, любой абонент которой может направить свое сообщение по интересующей его теме либо по какому-то конкретному адресу электронной почты, либо в распоряжение всех участников телеконференции.
- *Передача файлов с помощью протокола FTP* – для обмена файлами и целыми программами. Протокол обеспечивает способ перемещения файлов между двумя

компьютерами (с различными ОС и независимо от работающего оборудования) и позволяет абоненту сети Internet получить в свое распоряжение множество файлов. Пользователь получает доступ к различным файлам и программам, хранящимся на компьютерах, подключенных к сети.